Information Management

IBM

Draft Document for Review January 13, 2011 8:13 pm

SG24-7925-00

IBM Optim Performance Manager for DB2 for Linux, UNIX, and Windows

Installation, configuration, and integration with ITCAM and, Optim Query Tuner

Detailed functions and features usage guide

Performance tuning scenarios

Whei-Jen Chen
Ute Baumbach
Robert Borovsky
Sonali Kenge
Marcia Miskimen
Ling Xu

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM Optim Performance Manager for DB2 for Linux, UNIX, and Windows

December 2010

SG24-7925-00

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

First Edition (December 2010)

This edition applies to Optim Performance Manager Extended Insight Edition Version 4.1.0.1.

This document created or updated on January 13, 2011.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

Preface

Optim™ Performance Manager Extended Edition, a follow-on to DB2® Performance Expert, is one of the key product of the IBM® Optim Solution. Optim Performance Manager Extended Edition provides a comprehensive, proactive performance management approach. It helps organizations resolve emergent database problems before they impact the business.

This book describes the architecture and the components of Optim Performance Manager Extended Edition. We provide information for planning the deployment of Optim Performance Manager and detail steps for successful installation, activation, and configuration of Optim Performance Manager and the Extended Insight client.

Optim Performance Manager delivers new paradigm in terms of how it is used to monitor and manage database and database application performance issues. We describe individual product dashboards and reports and discuss, with various scenarios, how they can be used to identify, diagnose, prevent, and solve database performance problems.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Whei-Jen Chen is a Project Leader at the International Technical Support Organization, San Jose Center. She has extensive experience in application development, database design and modeling, and DB2 system administration. Whei-Jen is an IBM Certified Solutions Expert in Database Administration and Application Development, and an IBM Certified IT Specialist.

Ute Baumbach has been a software developer at the IBM laboratory in Germany for 20 years, where she works in various software development projects and roles. Most of her projects were based on DB2. For seven years, she has worked as a member of the DB2 Performance Expert development team, now the Optim Performance Manager development team. In addition, Ute supports customers in setting up Optim Performance Manager and utilizing it to its best advantage in DB2 data server environments. She is an IBM Certified Database Administrator and a Certified Application Developer for DB2 for Linux®, UNIX®, and Windows®.

Robert Borovsky is an IT specialist at IBM Canada. For the last 10 years he has worked as a Technical Sales specialist covering IBM Information Management portfolio of products. His expertise is in the areas of database technology, data warehousing, database performance tuning and high availability. He holds an engineering degree from the Faculty of Electrical Engineering of the Technical University in Kosice, Slovakia. He is located in Markham, Ontario in Canada.

Sonali Kenge is the Quality Assurance lead for IBM's pureQuery and Optim Performance Manager Extended Insight. Based at IBM Silicon Valley Lab, San Jose, she has over 12 years of experiences with DB2 and Informix® products and has held various roles in development, quality assurance, and product management. Her areas of expertise include application development life cycle and software testing. She has a Bachelor of Physics and a Bachelor of Computer Science from Mumbai University in India.

Marcia Miskimen is an Advisory Software Engineer with the IBM Software Group and works from Minnesota. She has been with IBM for 21 years, most recently working on testing of Optim Performance Manager integration with Tivoli® software. She has worked with DB2 Performance Expert, and now Optim Performance Manager, since Version 1, and this is her fifth Redbooks® publications. Over her many years in IT industry she has done COBOL/IMS development, mainframe system support, consulting, writing, pre-sales technical support, and testing, to name a few. She graduated from The Ohio State University with a bachelor's degree in business administration.

l**Ling Xu** is an IBM software engineer with the BIM Software Group in China. She has been working on performance test, analysis, and tuning for Java software and DB2 for over three years. Before joining IBM, Ling was a Java developer for three years. Now she is leading a team working on IBM Optim Performance Manager and IBM Optim Query Tuner with a focus on performance tuning. Ling and her team also provide assistance in customer enablement for Optim Performance Manager.

Acknowledgement

A Special Thanks to **Simon Harris** his extensive review of this book. Simon is a Performance/QA Architect for Data Studio Administration Tools, and is located in United Kingdom.

Thanks to the following people for their contributions to this project:

Rafael Coss
Jim Stephens
Larry McWilliams
Gary Walters
Vikram Khatri
Susan Althoff
Sven Miller
IBM Software Group, USA

Steve Rees
IBM Toronto Laboratory, Canada

Jun Wong
IBM Software Group, China

Steve Klopfer
IBM Sales and Distribution, USA

Emma Jacobs
International Technical Support Organization, San Jose Center

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Optim Performance Manager overview

This chapter introduces Optim Performance Manager (OPM) product. We talk about the evolution of the product, formerly known as DB2 Performance Expert. We highlight features, components as well as packaging of the product. Architectural diagrams are provided to better illustrate functionality of individual Optim Performance Manager components.

1.1 About Optim Performance Manager

Optim Performance Manager is a performance analysis and tuning tool for managing DB2 systems by using a web interface. Optim Performance Manager helps organizations resolve emergent database problems before they impact the business.

Optim Performance Manager is designed for those individuals responsible for the overall health and availability of their DB2 for Linux, UNIX, and Windows data servers, typically a database administrator (DBA) or an application owner. It watches over DB2 data servers by gathering performance metrics and determining if any key performance indicators are exceeding acceptable thresholds. By constantly monitoring the system, Optim Performance Manager can help detect potential performance problems before users are affected and service level agreements (SLA) are breached.

Optim Performance Manager utilizes a repository of historical performance metrics for problem prevention, trend analysis, customizable reporting and growth planning.

1.1.1 History

Optim Performance Manager was formerly known as DB2 Performance Expert for Multiplatforms product. It was first released in 2002. This tool complimented DB2 database performance monitoring product for z/OS® platform, IBM Tivoli Omegamon XE for DB2 Performance Expert on z/OS.

DB2 Performance Expert for Multiplatforms Version 1 solution consisted of Performance Expert (PE) server and PE client components. PE server and its repository database were collocated with the monitored DB2 database. PE client was a Java™ GUI client installed on the user's workstation, which displayed data collected by PE server.

DB2 Performance Expert for Multiplatform Version 2 was released in 2005. It introduced a new architecture of the product. PE server component was now designed to be installed on its own server. It would then remotely attach to monitored DB2 databases and collect database performance metrics. This architecture allowed a single PE server to manage multiple DB2 database servers. This version also delivered Performance Warehouse functionality, which aggregated collected database performance data into a separate set of tables in PE server's repository database, allowing for trend analysis and capacity planning of monitored database servers.

DB2 Performance Expert for Multiplatforms Version 3 was released in 2008. This version expanded the domain of the product, from just database server monitoring, to also include performance data of DB2 applications connected to DB2 servers using JDBC connectivity. This end-to-end database monitoring capability was delivered in the DB2 Performance Expert Extended Insight feature, which was an add-on feature of DB2 Performance Expert product.

Optim Performance Manager for DB2 for Linux, UNIX, and Windows 4.1 released in 2010, is a major step forward in the database monitoring capabilities that were previously provided in DB2 Performance Expert for Multiplatforms. It introduced a new web-based interface, which significantly simplifies the deployment of the product. Legacy PE client component is still available with this version, to allow for smoother migration of existing DB2 Performance Expert users.

Optim Performance Manager 4.1.0.1 became available in October 2010. It delivered new features and enhancements to Optim Performance Manager 4.1. It is available either as a fix pack for an existing copy of the 4.1 version of the product, or as a full installation. The content of the book is based on the version of the tool at the time of the writing of the book.

1.1.2 Features

Optim Performance Manager provides a comprehensive and proactive performance management solution for database applications. This solution brings new paradigm for database performance management, which can be characterized as a top-down database performance management.

Top-down performance management approach begins at the top level component, which is the database application. From this level, Optim Performance Manager can drill down to database level for related database performance metrics.

Optim Performance Manager complements top-down approach with the bottom-up database performance approach. This is a reactive performance management approach, which is a traditional way of database performance management. It begins at the bottom level component, which is the database server level.

Features and functionality of the product can be grouped into three categories:

- Identifying and diagnosing performance problems
- Preventing performance problems
- Solving performance problems

Identifying and diagnosing performance problems

Optim Performance Manager offers the ability to monitor and analyze multiple DB2 instances (including single partition, multi partition, and pureScale™ databases) running a wide variety of workloads, from a single control point. Predefined, customizable monitoring templates for online transaction processing (OLTP), business intelligence, SAP, and enterprise content management database systems allow users to rapidly and adequately deploy performance monitoring.

Optim Performance Manager extends database monitoring across the database client, the application server, and the network, giving DBAs immediate insight into where database workloads, transactions, and SQL requests are spending their time. It can monitor database end-to-end response time for Java,.Net, and DB2 Call Level Interface (CLI) database applications and provides predefined application views for WebSphere® Application Server, SAP, Cognos®, InfoSphere™ DataStage®, and InfoSphere Warehouse's SQL Warehousing Tool tasks. This feature allows users to:

- Improve availability of mission critical database applications.

  Detect negative trends sooner including: eroding response times for database APIs; network congestion between the application and the database; or other key factors to sustaining defined service level agreements.

- Manage applications to meet defined service level agreements.

  DBAs can see the response time for single SQL requests or complete database transactions from the applications point of view. This holistic view of database application response times can help to create realistic performance indicators that more directly relate to the user's experience.

- Reduce the time needed to isolate performance issues

  Graphical displays allow to see where workloads, transactions, and SQL requests originate in the application, and how the time spent is spread across the database client, the application server, the database server, and the network between them. This brings transparency to the process of isolating performance issues in complex environments, which include application and database servers, spread across multiple physical or virtual servers.

Optim Performance Manager delivers a browser based dashboard approach to help users quickly identify potential problems. The new browser based interface includes performance overview displays with associated health indicators to quickly detect problems in the overall environment and within a specific database. Intelligent diagnostic dashboards provide relevant metrics that focus on a particular problem area and provide the details needed to determine the root cause including:

- Locking conflicts, deadlocks, and timeouts

- Long running or untuned SQL statements
- Sorting or prefetching issues
- Buffer pool, cache, and heap sizes
- Operating system memory or CPU shortages
- Partition skews or overloaded partitions
- Log performance issues

Optim Performance Manager can alert administrators by e-mail or page when events occur to find and fix problems before they affect the business. It can also generate SNMP traps, which can be configured to be sent to enterprise SNMP managers. Users can customize alerts and configuration options, such as type, severity, frequency, and back-outs as required.

Prevent performance problems

Workload management administration and management tooling is now available in Optim Performance Manager for customers that also own the DB2 workload management (WLM) feature. This new tooling provides the ability to define workloads, assign business priorities, and enable concurrency controls and aging to give resource priority to important queries that must meet service level objectives. After you setup WLM environment, you can use Optim Performance Manager to analyze live monitoring data to track resource consumption, system capacity, response times, performance issues, as well as historical data for trend analysis and growth planning.

Integration of Optim Performance Manager and DB2 Workload Manager feature delivers a proactive approach to database performance management. It allows for defining more predictable database server execution environment, by assigning resources (CPU, I/O, memory) according to the priorities of various workloads.

Optim Performance Manager provides predefined report templates that you can use to generate reports for trend detection, proactive monitoring, baselines, and more. Report templates include: disk space consumption, system configuration for the database and the database manager, dynamic SQL statements, WLM, and connections.

Solve performance problems

Integration of Optim Performance Manager with the IBM Optim family of products supports a user's efforts to quickly resolve problems.

Integration with Optim Query Tuner passes problematic SQL statements directly into a query tuning session so the query can be analyzed, tuned, and redeployed into production.

Integration with Optim pureQuery Runtime enables instant isolation of the SQL statement application source code and facilitates collaboration between the DBA and the developer to resolve the problem. Optim pureQuery Runtime also allows for optimizing the SQL without altering the application.

Integration with Tivoli Composite Application Manager (ITCAM) provides a consolidated view of the business transactions across the enterprise while providing the deep database diagnostics found in Optim Performance Manager to support efforts to resolve response time issues quickly. ITCAM provides a common model and consistent view of the performance of an application. From the transaction view within ITCAM, you can launch Optim Performance Manager in context. Additional integration facilitates the ability to view operating system statistics from Optim Performance Manager. This functionality is provided by Optim Performance Manager plug-in for Tivoli Enterprise Portal (TEP).

1.1.3 Components

In this section we describe components of Optim Performance Manager product. We highlight most important characteristics of individual OPM components and how they contribute to OPM's ability to identify, diagnose, prevent, and solve database performance problems.

Optim Performance Manager

Optim Performance Manager provides system overview displays for quicker problem identification and detailed diagnostic displays for detailed problem analysis.

System overview displays with color-coded user interfaces are designed to quickly draw attention to problematic areas within the database. Once a problem has been identified, specific diagnostic dashboards provide focus on that area and present relevant details to provide a well-rounded analysis of the problem. Once the problem has been identified, integration to additional Optim products helps speed the resolution of the problem.

The Health Summary dashboard displays an overview of the performance data for your databases. You can use the Health Summary dashboard, for example, to identify which of your databases have critical issues that require your attention.

The Alerts dashboard displays a list of issues that require attention. Flexible alert notifications can be defined by type, severity, and database. You can also define alert notification parameters such as email addresses, SNMP trap generation, and alert frequency. You can view alerts by alert severity, host and port, or by custom groups.

Inflight dashboards provide recent and historical information about specific databases. Each dashboard provides information about a database that relates to a particular category of potential performance issues: Overview, Buffer Pool and I/O, Locking, Logging, Memory, Active SQL, System, Utilities, and Workload. For example, the Active SQL dashboard provides performance data for currently running queries. You can use the Active SQL dashboard to identify, tune, or terminate SQL queries that degrade the performance of your database.

Reporting feature described earlier in this chapter is also included in OPM.

Optim Performance Manager Extended Insight

Optim Performance Manager Extended Insight (OPM EI) complements OPM technology by monitoring database response time as seen by the application. This integrated monitoring system watches over both the database system and the end-user response times to help organizations achieve their SLA goals.

Extended Insight Dashboard displays end-to-end data about the entire database application system, which includes clients, application servers, data servers, and the network. Monitoring begins when you initiate a transaction, continues as that transaction is processed by each component, such as the client, network, and data server, and ends when the application finishes processing and produces the results.

OPM EI contains a client component, which is collocated with the database application. It intercepts DB2 database traffic from the application and sends performance data about this traffic to Optim Performance Manager. OPM EI supports Java, WebSphere Application Server, DB2 CLI, and .Net applications. It can also be used to monitor DB2 databases on z/OS systems. This functionality requires the use of OPM as well as OMEGAMON® XE for DB2 on z/OS V5.1 product.

> **Note:** OPM EI collects and displays additional monitoring information for applications running in WebSphere Application Server. Application servers such as Weblogic or Tomcat, are treated by OPM EI as generic java applications.

With OPM EI you can proactively, quickly, and intuitively identify:

- *Who* has response time performance issues or causes them to others by identifying the specific set of transactions that make up the problem workload.

  For example, Which end-user ID or applications issues those transactions?

- *When* did the response time performance occur by identifying the problem periods.

For example monitor minutes, hours, days, weeks, or even years)

- *What* specific activities were involved in the response time performance problem by identifying the complete list of involved problem SQL statements.

  For example, check the top N SQL statements by end to end response time or data server time.

- *Why* the response time problem occurred by identifying the exact problem layer that slows down the response time.

  For example, detect whether the transactions are slowed down in the database, network, driver, application, or application server.)

OPM plug-in for Tivoli Enterprise Portal

Optim Performance Manager plug-in for Tivoli Enterprise Portal facilitates integration between OPM EI and IBM Tivoli Composite Application Manager for Application Diagnostics and ITCAM for Transactions in a Tivoli Enterprise Portal Console for end-to-end transaction monitoring. It allows OPM EI to be configured such that it sends database transaction information directly to ITCAM. This data is then being surfaced in the TEP console. This allows operators of the TEP console to be notified when database transactions are not performing well. It also allows them to launch into OPM EI and OPM dashboards for further diagnostic and analysis.

OPM plug-in for TEP also delivers extended operating system performance data by launch-in-context capabilities. For instance, when you launched OPM or OPM EI dashboards from TEP console, you can open the operating system monitoring details for a selected application client by launch-in-context into TEP workspace.

Workload manager configuration tool

Workload manager configuration tool provides workload management administration and management tooling for DB2 customers who have deployed DB2 Workload Management (WLM) feature. It enables proactive approach to database performance management by allocating resources to database workloads up front. This prevents workloads from monopolizing database resources, and creating database performance issues.

You can use this tool to:

- Define workloads
- Enable aging
- Assign business priorities to workloads to ensure service level agreements
- Monitor service classes, workloads, and work classes
- Obtain sophisticated usage statistics and definitions

- View real-time data and historical data
- Understand workload at a point in time
- Understand workload patterns and long-term development issues
- Derive effective service classes and thresholds
- Perform workload analysis for workload profiling and accounting
- Adjust the WLM configuration of a DB2 database continually so that your workloads meet their performance objectives

Data Studio Health Monitor

Data Studio Health Monitor is a web-based health and availability monitor for DB2 for Linux, UNIX and Windows databases. It replaces Data Studio Administration Console product and is now integrated into Optim Performance Manager.

Data Studio Health Monitor offers health alerts and basic monitoring capabilities including:

- Summary views across all monitored databases
- Monitoring for essential characteristics like database server status, storage utilization, recovery pending, backup pending, etc.
- HADR state monitoring
- Drill downs into alert details
- Views into active applications and utilities

Performance Expert Client

Performance Expert Client is the original client user interface of the DB2 Performance Expert product and is still available in the Optim Performance Manager product. Even though the majority of its functionality is now available in the new web browser interface of Optim Performance Manager, you can use Performance Expert client to perform a smoother migration from DB2 Performance Expert product to Optim Performance Manager.

You can also perform the following tasks by using DB2 Performance Expert Client:

- In addition to the features of the Workload Manager tool, you can look at Workload Manager data to monitor and report Workload Manager setup and activities over time.
- In addition to the features of the reporting feature, you can do long term performance analysis through Performance Warehouse.

- In addition to the features of the Optim Performance Manager plug-in for TEP, you can monitor and analyze operating system performance after installing the optional Common Information Model (CIM) server component.
- Real time database monitoring
- More detailed monitoring of partitioned DB2 databases

1.1.4 Packaging

Optim Performance Manager product is available in two editions:

- Optim Performance Manager

  This is the base version of the product that contains a useful set of base capabilities, including new Web-based user-interface, graphical dashboards, reporting capabilities, and Workload Manager tooling. It is licensed according to the type of monitored DB2 database:

  – IBM Optim Performance Manager V4.1.0.1 for DB2 for Linux, UNIX, and Windows, Enterprise Edition

  – IBM Optim Performance Manager V4.1.0.1 for DB2 for Linux, UNIX, and Windows, Workgroup Edition

  – IBM Optim Performance Manager V4.1.0.1 for DB2 for Linux, UNIX, and Windows, Content Manager Edition

- Optim Performance Manager Extended Edition

  Contains capabilities of the Optim Performance Manager product and extends them with Extended Insight and Tivoli integration.

Table 1-1 shows both Optim Performance Manager editions and lists their respective capabilities. Included in the table is also DB2 Performance Optimization Feature bundle (only available with DB2 Enterprise Server Edition) which contains Optim Performance Manager Extended Edition, as well as the activation of the DB2 Workload Manager feature. We also include DB2 9.7 Advanced Enterprise Edition, that contains Optim Performance Manager, as well as DB2 Workload Manager feature.

Table 1-1 Optim Performance Manager editions

Feature	Optim Performance Manager	Optim Performance Manager Extended Insight	DB2 Performance Optimization Feature	DB2 9.7 Advanced Enterprise Server Edition
Alerts and Notifications	✓	✓	✓	✓

Feature	Optim Performance Manager	Optim Performance Manager Extended Insight	DB2 Performance Optimization Feature	DB2 9.7 Advanced Enterprise Server Edition
Overview Health Summary	✓	✓	✓	✓
Diagnostic Dashboards	✓	✓	✓	✓
Reporting	✓	✓	✓	✓
Data Studio Health Monitor	✓	✓	✓	✓
Workload Manager Tool	✓	✓	✓	✓
Extended Insight		✓	✓	
Tivoli ITCAM integration		✓	✓	
DB2 Workload Manager Feature			✓	✓

Note: Optim Performance Manager Extended Edition is also included in following product offerings:

- InfoSphere Warehouse 9.7 Enterprise
- IBM Smart Analytics System

1.2 Architecture

Optim Performance Manager can be deployed in two configuration options:

- Standalone - It only captures monitoring information from DB2 data server.
- Extended Insight - It offers monitoring of DB2 database, as well as database applications.

The following sections document architectural diagrams for both deployment options.

1.2.1 Optim Performance Manager architecture

Figure 1-1shows the architecture of the base Optim Performance Manager product.

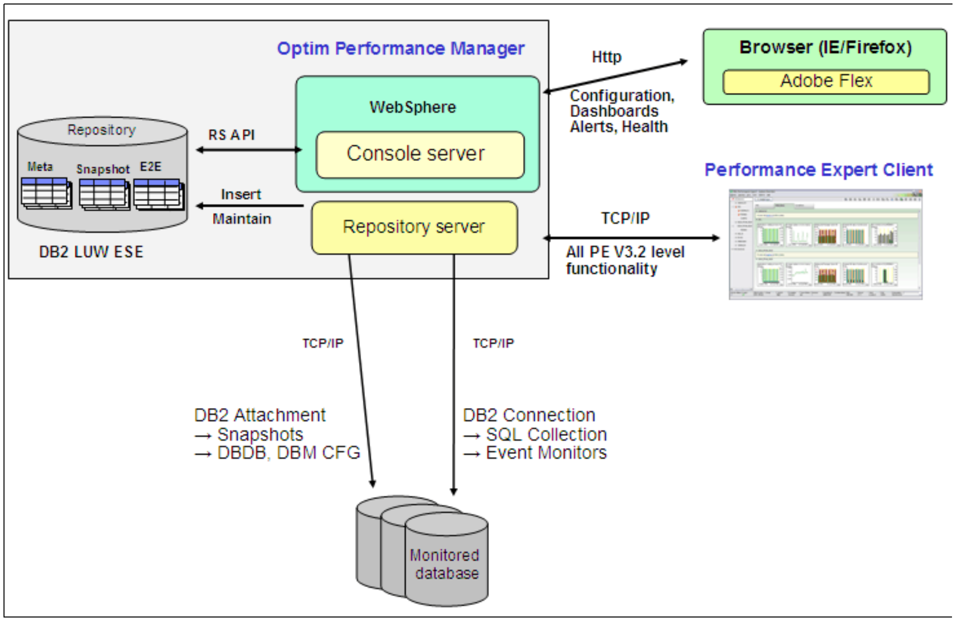


Figure 1-1 Optim Performance Manager architecture

Key components of Optim Performance Manager are:

- Repository Server

  It establishes connection to monitored DB2 database and mainly uses database snapshot commands and DB2 event monitors to collect database performance data. It stores this collected data in its repository database.

- Console Server

  It runs as an application in WebSphere Application Server environment and connects to Optim Performance Manager repository database. It also allows Optim Performance Manager users to use a Web interface to retrieve this data and configure the monitoring behavior of Optim Performance Manager.

- Repository database

  It is a DB2 Enterprise V9.5 database, which is included in the OPM product packaging. It stores database performance data collected by repository server from the monitored DB2 databases and database application data collected by OPM Extended Insight client.

1.2.2 Optim Performance Manager Extended Insight architecture

Figure 1-2 illustrates the basic architecture of the Optim Performance Manager Extended Insight.

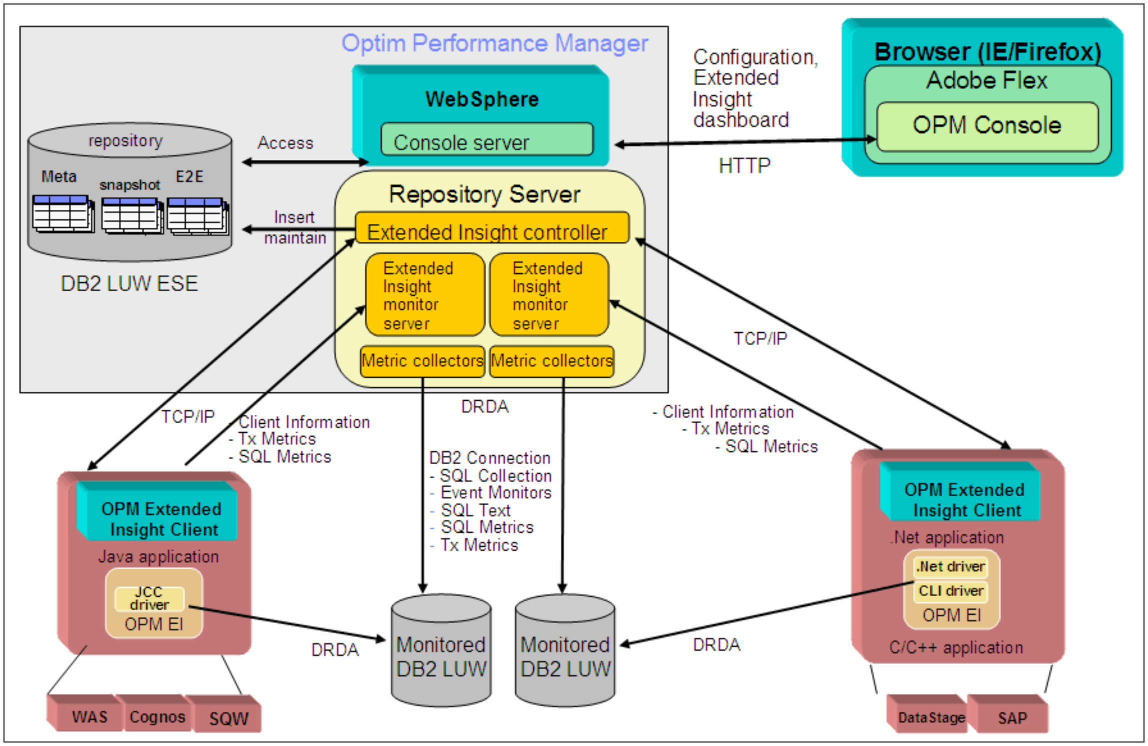


Figure 1-2 Optim Performance Manager Extended Insight architecture

Optim Performance Manager Extended Insight consists of following components:

- Optim Performance Manager Extended Insight client

  It is collocated with the database application. It hooks into JDBC or CLI drivers, intercepts database traffic for the monitored DB2 database and collects response time data about transactions and SQL statements. This data is then periodically forwarded to the Extended Insight monitoring server, which stores it in the repository database.

- Optim Performance Manager Extended Insight controller

  It is embedded in the repository server of the Optim Performance Manager. The Extended Insight controller is a global controller that listens on a port for Extended Insight Clients accessing the controller. It also knows about all available Extended Insight monitoring servers. When an application that you monitor with Extended Insight client starts and connects to the monitored database, the Extended Insight client accesses the controller and asks for the Extended Insight monitor server port which is listening for the Extended Insight data from the monitored database. From that point on, the Extended Insight client sends the collected data to the Extended Insight monitor server for the monitored database over the communicated monitor server port.

You specify the port number of the controller when you activate Extended Insight on Optim Performance Manager and when you configure Extended Insight clients. On both systems the port number is saved in the `pdq.properties` file.

- Optim Performance Manager Extended Insight monitoring server

  It is embedded in the repository server of the Optim Performance Manager. There is one Extended Insight monitoring server available per monitored database for which Extended Insight monitoring is configured. Each monitoring server is listening on a dedicated port for response time data about transactions and SQL statements from Extended Insight clients. Extended Insight clients first access the Extended Insight controller to obtain the port number of the responsible Extended Insight monitoring server. After that Extended Insight clients send the collected response time data periodically to the Extended Insight monitoring server which receives the data and stores the data in the repository database.

  By default the port number of each Extended Insight monitoring server is determined dynamically. If you prefer fixed port numbers, you can specify them when you configure Extended Insight monitoring from Optim Performance Manager web console.

- Optim Performance Manager Extended Insight metric collectors

  It is embedded in the repository server of the Optim Performance Manager. There is one set of metric collectors available per monitored database for which Extended Insight monitoring is configured. The metric collectors collect additional information about the transactions and SQL statements directly from the monitored database, combine the collected data with the data which Extended Insight monitoring server receives from Extended Insight clients and store the data in the repository database. The metric collectors start unit of work or package cache event monitors (DB2 9.7 or above) or use the dynamic SQL snapshots (DB2 9.5 or lower) to collect additional information about the transactions and SQL statements. The additional information consists of time distributions for transaction and SQL statement executions on the database and complete statement text. By combining this data with data received from Extended Insight client you get an end-to-end response time distribution of transactions and SQL statements.

2

Planning

This chapter contains important information for planning your deployment of Optim Performance Manager. We explore the component architecture in more detail and address some common questions about Optim Performance Manager:

- How can I best prepare for installation?
- What kind of system resources are required for the Optim Performance Manager server?
- How can Optim Performance Manager comply with my company's security requirements?
- What kind of footprint does Optim Performance Manager have on my production database?

2.1 Topology

In this section, we look at the Optim Performance Manager architecture and topology. Since this chapter is about planning, this section is intended to explore the components of Optim Performance Manager and what you need to think about as you proceed with your deployment tasks. The information in this section is a follow-on to the component architecture introduced in 1.2, "Architecture" on page 11.

The Optim Performance Manager Information Center has good descriptions of architecture and topology diagrams, see

http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.installconfig.doc/architectures.html

When you think about a performance monitor in the simplest sense, you have a monitor, the monitored object, performance data, a place to store the data and a way to look at the data. Optim Performance Manager has these components, which we explore next.

2.1.1 Optim Performance Manager Server

As described in 1.2.1, "Optim Performance Manager architecture" on page 12, there are three main components of what is called the "Optim Performance Manager server":

- Repository server
- Repository database
- Console server

Generally, when you see the term "Optim Performance Manager", "Optim Performance Manager server" or "OPM Server", it refers to the whole set of components, and if there is a reason to refer to a particular component, it will be called out separately.

> **Important:** The repository server, console server, and repository database must all reside on the same machine.

Repository server

The repository server component is sometimes referred to as the "back end" or "back end server". It is the component that collects, stores, aggregates, and deletes the performance data according to your configured settings.

The repository server could be considered like any application in your shop. It runs as a Java program, under ownership of the Optim Performance Manager DB2 instance user, and stores its data to the repository database.

When you start the repository server a Java process is started on the Optim Performance Manager machine. The Java process consists of a set of threads per monitored database responsible for collecting, storing, aggregating, and deleting performance data in the repository database. Appendix B, “Optim Performance Manager footprint” on page 471 describes the threads in detail.

To collect snapshot data from a monitored database, the repository server calls a stored procedure. The stored procedure periodically attaches to the DB2 instance of the monitored database, retrieves snapshot data using DB2 APIs, and detaches. Stored procedures are running in DB2 `db2fmp` processes. Therefore, a number of `db2fmp` processes are started on behalf of the repository server for collecting snapshot data from the monitored databases and storing it in the repository database. The number of `db2fmp` processes that are started depend on the number of databases the repository server monitors.

Altogether the repository server consists of one Java process, a number of `db2fmp` processes, and some other DB2 processes that are started when the repository server connects to the repository database, for example `db2sysc`.

Repository database

The repository database is the central storage for all performance metrics collected by Optim Performance Manager. As with any database, there are considerations for its placement, size, growth, and security. These considerations are described in several sections in this chapter.

In addition to the planning topics in this chapter, Appendix A, “Managing the repository server and the repository database” on page 437 constains information that can help you understand the layout of the repository database and give you hints to maintain the repository database, for example:

- Tables in the repository database
- Automatic runstats and reorganization
- Backing up the repository database
- Changing database configuration parameters
- Enabling row compression for the repository database

Console server

The Console server is the WebSphere application that presents the performance data to the end user. While Optim Performance Manager ships with an installable WebSphere Application Server, you could also use an existing WebSphere Application Server to host the software, if it meets the pre-requisites. The considerations for deciding how to deploy in WebSphere are described in 2.3.1, “Common Optim Performance Manager installation parameters” on page 26.

2.1.2 Optim Performance Manager Extended Insight

Extended Insight, unlike Optim Performance Manager server, is not a separate running component, but rather a feature with some hooks into other components. Review the architecture discussion in 1.2.2, “Optim Performance Manager Extended Insight architecture” on page 13. With Extended Insight, Optim Performance Manager now monitors not only the database itself, but the application database transactions which use that database. Therefore for planning purposes, deploying Extended Insight requires action at the monitored application client side, and on the Optim Performance Manager server.

- On Optim Performance Manager server, you activate the Extended Insight license and define the port number of the Extended Insight controller. Ensure that this port is open on firewalls which may exist in your shop between the application client machine and Optim Performancde Manager machine. The activation is described in to 3.2.3, “Activating the Optim Performance Manager Extended Insight license” on page 81.
- On Optim Performance Manager server you enable Extended Insight monitoring per monitored database that the client applications use. During enabling, you define configuration properties for the Extended Insight monitor server and the Extended Insight metric collectors per database. For example, you can define a fixed port number of the Extended Insight monitor server instead of letting the Extended Insight monitor server determine a port number dynamically. Defining a fixed port number is recommended if your shop has strict firewall rules between the application client machine and Optim Performancde Manager machine and you have to open ports explicitly. Enabling Extended Insight monitoring is described in 3.3.3, “Configuring the database for monitoring” on page 98 and in more detail in 3.3.5, “Monitoring profile for Extended Insight” on page 119.
- On the application client side, you install Extended Insight Client and configure it. Configuration includes
  - Letting the DB2 .Net, CLI, or JDBC drivers that the application uses to connect to the monitored database know that the Extended Insight is available. The DB2 CLI or JDBC driver your application uses must be at a certain level to be able to communicate with the Extended Insight client in order to provide information about transactions. The CLI and JDBC driver prerequisites are described in 2.2.2, “Optim Performance Manager Extended Insight” on page 23.

 – Specifying the port number of the Extended Insight controller so that the Extended Insight client can establish the communication and knows where to send collected data about transactions and statements.

 Installation and configuration is described in 3.4, “Installing and Configuring Extended Insight Client” on page 129.

The most important concept in understanding the Extended Insight is how the performance data is collected on the application client side and transfered to Optim Performance Manager. When Optim Performance Manager monitors a DB2 database, it retrieves performance data from the database in an active *pull* method, using the standard DB2 functions such as snapshot or event monitors. When monitoring a client application, the data transfer is *pushed* to Optim Performance Manager. Let us describe this pushed data transfer in more detail:

1. Assume that Extended Insight is completly deployed as decribed above.
2. You now start the client application that connects to the monitored database using a DB2 CLI, .Net or JDBC driver. In the following we just call it DB2 driver.
   a. The DB2 driver loads the Extended Insight client which runs in its own thread.
   b. The Extended Insight client accesses the Extended Insight controller of Optim Performance Manager through the controller port and asks for the Extended Insight Monitor server port number for that database. Communication is established.
3. The client application starts a transaction by executing an SQL statement.
   a. The DB2 driver provides information about the transaction and statement such as transaction start time, connection properties, or SQL statement text to the Extended Insight client.
   b. Extended Insight client calculates a hash code for the SQL statement and keeps the provided data in memory.
4. The client application ends a transaction by executing a commit or rollback.
   a. The DB2 driver provides information about the transaction such as transaction end time and the time breakdown. For example, how much time of the transaction was spent in the network or data server.
   b. The Extended Insight client aggregates this information to the information already availalbe in memory. An aggregating example is that it calculates the average response time for all transactions having the same connection properties such as user ID or application name.
5. The client application repeats steps 3.) and 4.)

a. The Extended Insight client further aggregates the provided data in memory.

b. If the client application is a WebSphere application then the Extended Insight client periodically checks the connection pool status and aggregates WebSphere connection pool information in memory as well.

c. After each minute Extended Insight sends the aggregated data to the Extended Insight Monitor server through the monitor server port. Instead of sending the SQL statement text, it just sends calculated hash codes to keep the size of the sent data small.

d. The Extended Insight monitor server reads the data and stores it in the repository database.

e. The Extended Insight metric collectors within the repository server collect additional information about the transactions and statements from the database using the pull method. For each statement hash code the complete statement text is retrieved from the package cache either by using the dynamic SQL snapshot (DB2 9.5 or lower) or by using the package cache event monitor (DB2 9.7 and above).

f. If your monitored database is at DB2 9.7 Fix Pack 1 or higher, the metric collectors collect transaction execution details and statement execution details on the database and combine it with the transaction and statement data received from Extended Insight client. For example, the Extended Insight client provides the information that the average time the transactions spent on the database is three seconds. Using the unit of work event monitor, the metric collectors provide further time breakdown information of the three seconds in time spent for locking, sorting, I/O, and other processing or waits.

6. The client application disconnects from the monitored database.

   The Extended Insight client sends aggregated data a last time and stops processing.

2.1.3 Monitored database

Typically, the monitored database resides on a remote machine. Optim Performance Manager does not require a separate program (agent) to be installed on the remote monitored database server, however, it does create some objects in the monitored database, depending on what monitoring options you choose. These objects are discussed in 2.7, “Objects in the monitored database” on page 57. Addditionally, Appendix B, “Optim Performance Manager footprint” on page 471 describes the created objects and their footprint in detail.

2.1.4 Tivoli monitoring

If you deploy Optim Performance Manager Extended Insight into an environment that also uses Tivoli Composite Application Manager for Transactions, some additional metrics are available through the Tivoli Enterprise Portal. Planning and pre-requisites for deploying into Tivoli environment are discussed in Chapter 10, "Integration with Tivoli monitoring components" on page 323.

2.2 Prerequisites

In this section, we document general prerequisites for installing and running Optim Performance Manager solution. Check the product document for the detailed information about software levels of individual components of Optim Performance Manager at the following website:

http://publib.boulder.ibm.com/infocenter/idm/docv3/index.jsp?topic=/com.ibm.datatools.perfmgmt.installconfig.doc/pm_install_reqs.html

2.2.1 Optim Performance Manager

Following are the hardware and software requirements for the deployment of the Optim Performance Manager product.

Hardware and operating system

We recommend installing Optim Performance Manager in a separate physical or virtual server from the monitored DB2 database. This approach would prevent Optim Performance Manager to share CPU, memory, and disk resources with monitored database, thus allowing it to collect unbiased database performance data.

You can install Optim Performance Manager in the UNIX (AIX®, Solaris), Linux (RedHat, SuSE), and Windows environments. Size of the server in which it is installed would generally depend on:

- The number of monitored DB2 databases
- The number of partitions on your monitored DB2 databases
- The number of DB2 objects on your monitored DB2 databases
- The number of monitoring functions that are activated in Optim Performance Manager
- The volume of workload against monitored database (for instance, number of SQL statements per minute)

DB2 data server for Optim Performance Manager

Optim Performance Manager uses DB2 database as a repository for storing collected performance data as well as its own configuration information. If the server where Optim Performance Manager will be installed already contains a copy of DB2 server product, you can use it. Otherwise, the product ships with a restricted use license of DB2 Enterprise Server Edition Version 9.5.

Optim Performance Manager supports the following data servers for its repository database:

- IBM DB2 Enterprise Server Edition Version 9.1 for Linux, UNIX, and Windows
- IBM DB2 Enterprise Server Edition Version 9.5 for Linux, UNIX, and Windows
- IBM DB2 Enterprise Server Edition Version 9.7 for Linux, UNIX, and Windows

Note: The DB2 instance where the repository server runs may not run in Oracle compatibility mode. For more information on Oracle compatibility mode, please refer to DB2 documentation site:
`http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.apdv.porting.doc/doc/r0052867.html`

Note: Optim Performance Manager requires DB2 Enterprise Server Edition product, because its repository database uses DB2 Enterprise Server Edition features such as table partitioning.

WebSphere Application Server

Optim Performance Manager requires WebSphere Application Server to run the console server component. If the server where Optim Performance Manager is installed already has WebSphere Application Server 7.0.0.3 or later installed, it can use it. Otherwise, it installs a copy of WebSphere Application Server 7.0.0.5.

Monitored DB2 database

The following data servers are supported for the DB2 instances to be monitored by Optim Performance Manager. 64-bit DB2 instances are supported on each of these data servers. 32-bit DB2 instances are supported only on Linux on System x® and Windows.

- IBM DB2 Enterprise Server Edition Version 9.1 for Linux, UNIX, and Windows
- IBM DB2 Enterprise Server Edition Version 9.5 for Linux, UNIX, and Windows
- IBM DB2 Enterprise Server Edition Version 9.7 for Linux, UNIX, and Windows
- IBM DB2 Enterprise Server Edition Version 9.8 for Linux, UNIX, and Windows
- IBM DB2 Workgroup Server Edition Version 9.1 for Linux, UNIX, and Windows

- IBM DB2 Workgroup Server Edition Version 9.5 for Linux, UNIX, and Windows
- IBM DB2 Workgroup Server Edition Version 9.7 for Linux, UNIX, and Windows
- IBM DB2 for z/OS Version 9.1
- IBM DB2 for z/OS Version 10.1

Web browsers

New web interface of Optim Performance Manager is supported in following web browsers:

- Mozilla Firefox Version 3.6 or later with Adobe® Flash Player 10.1.53.64 or later
- Microsoft® Internet Explorer Version 7.0 with Adobe Flash Player 9.0.124 or later
- Microsoft Internet Explorer Version 8.0 with Adobe Flash Player 9.0.124 or later

2.2.2 Optim Performance Manager Extended Insight

Optim Performance Manager Extended Insight enables end-to-end monitoring of DB2 database applications from following generic database client environments:

- CLI applications:

  Require the use of DB2 Data Server Client Packages for Version 9.7 Fix Pack 2 or later.

- .Net applications:

  Require the use of DB2 Data Server Client Packages for Version 9.7 Fix Pack 3.

- WebSphere Applications Server z/OS:

  Use the following versions of WebSphere Application Server for z/OS:

  - IBM WebSphere Application Server Version 6.1.0.31 or later
  - IBM WebSphere Application Server Version 7.0.0.11 or later

- WebSphere Application Server:

  Use the following versions of WebSphere Application Server for Linux, UNIX, and Windows:

  - IBM WebSphere Application Server Version 6.1.0.27 or later with Patch PK98171

- IBM WebSphere Application Server Version 6.1.0.31 or later
- IBM WebSphere Application Server Version 7.0.0.5 or later with Patch PK98171
- IBM WebSphere Application Server Version 7.0.0.11 or later

> **Note:** Application servers such as Weblogic or Tomcat are treated by Optim Performance Manager Extended Insight as generic Java applications.

- JDBC and SQLJ applications

  To have a complete set of extended insight monitoring data, use the following DB2 data server client versions:

  - IBM Data Server Drivers for JDBC and SQLJ Version 9.7 Fix Pack 2 or later for Linux, UNIX, and Windows
  - IBM Data Server Drivers for JDBC and SQLJ Version 9.5 Fix Pack 6 or later for Linux, UNIX, and Windows

> **Note:** When using Optim Performance Manager Extended Insight for end-to-end monitoring, the level of collected and presented performance data differs according to the level of monitored DB2 database. For instance:
>
> - For DB2 Version 9.1, use Fix Pack 6 s to obtain information about data server time.
> - To be alerted for lock wait and lock wait timeout events, DB2 9.7 is required
>
> You must have have DB2 Version 9.7 Fix Pack 1 or later to obtain layers for time that is spent in locking, sort, logging, queue time, and so on. Otherwise, you only have one layer that indicates DB2 data server time.
>
> The following data server client versions are supported, but only a subset of extended insight monitoring data is collected. For example, if a transaction includes static and dynamic executions, you can monitor only dynamic executions.
>
> - IBM Data Server Drivers for JDBC and SQLJ Version 9.7 for Linux, UNIX, and Windows
> - IBM Data Server Drivers for JDBC and SQLJ Version 9.5 Fix Pack 5 for Linux, UNIX, and Windows

If you plan to monitor SAP, Cognos, InfoSphere Warehouse, or Information Server, use the following versions:

- SAP kernel Version 7.0 SR3 or later
- Cognos Version 8.4 Fix Pack 2 or later
- InfoSphere Warehouse Version 9.7 Fix Pack 1
- Information Server Version 8.5

2.2.3 Integration with Tivoli

If you plan to run Optim Performance Manager Extended Insight monitoring from your Tivoli Enterprise Portal (TEP) console, you must have following products installed:

- IBM Tivoli Monitoring (ITM) Version 6.2.2 Fix Pack 1 or later
- IBM Tivoli Composite Application Manager (ITCAM) for Transactions Version 7.2 or later
- ITCAM for Application Diagnostics Version 7.1 or later

If you use J2EE 1.4 application in your environment, use ITCAM for Application Diagnostics Version 7.1 Fix Pack 1 or later.

2.3 Installation options

The Optim Performance Manager Extended Edition product consists of the following components:

- Server component: Optim Performance Manager
- Client components:
  - Optim Performance Manager Extended Insight client
  - Optim Performance Manager Extended Insight plug-in for Tivoli Enterprise Portal
  - Performance Expert client

All components must be installed separately because, typically, these components are installed on separate systems.

The Optim Performance Manager Extended Insight client, Optim Performance Manager Extended Insight plug-in for Tivoli Enterprise Portal, and Performance Expert client installations do not require any license. To install version 4.1.0.1 of these components you have the following options:

- Install the version 4.1.0.1 available from the Optim Performance Manager Extended Edition 4.1.0.1 installation package.
- Install the version 4.1 Fix Pack 1 available from the fix pack download site:

http://www-01.ibm.com/support/docview.wss?rs=434&uid=swg27008647#opm-lib

- If you have the version 4.1 already installed, update it to 4.1.0.1 by installing version 4.1.0.1 or Fix Pack 1 on the top of version 4.1.

Optim Performance Manager requires a license. To install version 4.1.0.1 of Optim Performance Manager you have the following options:

- Direct installation option:

  Install the version 4.1.0.1 available from the Optim Performance Manager Extended Edition 4.1.0.1 installation package. This package includes the license activation kit.

  Use this option for a fresh installation. The installation installs Optim Performance Manager and sets up the DB2 repository database. It also sets up WebSphere Application Server and optionally installs it first if the product is not available yet. After installation, you configure Optim Performance Manager by adding databases and configure them for monitoring using the Optim Performance Manager web console.

- Update option:

  Install version 4.1 available from the Optim Performance Manager Extended Edition 4.1 installation package. This package includes the license activation kit. Afterward, update it to 4.1.0.1 by installing Fix Pack 1 on the top of version 4.1.

  Typically, you use this option if you already have installed and used version 4.1 and want to update it to the newest level.

- Migration option:

  If you have Performance Expert V3.2 installed, you can migrate to Optim Performance Manager version 4.1.0.1 by installing version 4.1.0.1 from the Optim Performance Manager EE 4.1.0.1 installation package and continue use the repository database of Performance Expert for Optim Performance Manager.

You obtain the same set of functional features independent on the installation option you use. The version 4.1.0.1 of Optim Performance Manager offers new installation features. If you are new to Optim Performance Manager, we recommend using the direct installation option instead of using the update option. We describe the new installation features in 2.3.2, “Direct installation option” on page 29.

2.3.1 Common Optim Performance Manager installation parameters

In this section, we describe the parameters that you can set during Optim Performance Manager installation. These parameters are common to the direct installation, update, and migration options and they are important in planning your installation. You can choose between a typical installation and an advanced installation. If you choose typical installation then the default values are taken for some parameters.

DB2 instance selection

Optim Performance Manager requires a DB2 instance to host the repository database. During installation, you can specify which DB2 instance you want to use. If the DB2 instance does not yet exist then the Optim Performance Manager installation creates it.

DB2 user specification

The Optim Performance Manager installer uses the user that you specify to create the repository database in the selected DB2 instance. Later at Optim Performance Manager run time, this user is used by Optim Performance Manager to connect to the repository database to access the collected data. This user must have SYSADM authority on the DB2 instance. Learn more about user privileges in Optim Performance Manager in 2.6, “User authorization” on page 52.

Repository database specification (Advanced installation only)

The repository database is the database of Optim Performance Manager to store the collected performance data. The Optim Performance Manager installer creates this database. You can use the advanced installation mode to specify the following settings for the repository database:

- Database name
- Database location
- Table space location for small SMS table spaces storing control and metadata

Table space type selection (Advanced installation only)

For each database that Optim Performance Manager monitors Optim Performance Manager creates a set of table spaces in the repository database. The table spaces are created when you configure a database for monitoring after Optim Performance Manager is installed. During installation you can specify which type of table spaces (SMS, DMS, or Automatic Storage) Optim Performance Manager should create. These table spaces can grow to multiple GBs in size. Refer to 2.4, “Capacity planning” on page 34 to learn how large these table spaces can get. 2.5, “Storage options” on page 49 gives you more details about these table spaces.

Working directory specification (Advanced installation only)

The Optim Performance Manager repository server uses the directory that you specify to write log and trace files during run time. In addition, this directory contains the property files that the Optim Performance Manager repository server uses.

Performance Expert client group specification

If you want to use Performance Expert client then all users that are part of this operating system group can logon from Performance Expert client to the Optim Performance Manager repository server. The operating system group must be available on the Optim Performance Manager machine. If you do not want to use Performance Expert client then it does not matter which group you specify.

WebSphere Application Server selection

Optim Performance Manager requires a copy of WebSphere Application Server on the Optim Performance Manager machine to serve the Web User Interface (Optim Performance Manager web console). During installation, you specify whether you want to reuse an existing copy of WebSphere Application Server for Optim Performance Manager or you want to let the Optim Performance Manager installer install and set up WebSphere Application Server. The Optim Performance Manager installer list the existing copies of WebSphere Application Server that can be used for Optim Performance Manager. Only WebSphere Application Server copies that are at version 7.0.0.3 or higher and that have been installed as root (Linux, UNIX) are listed

Parameter summary

Table 2-1 summarizes the installation parameters and describes the defaults for the typical installation mode.

Table 2-1 Installation parameter summary

Parameter	Specify always	Specify in advanced mode	Default value for typical mode
DB2 instance	Yes	-	-
DB2 user	Yes	-	-
Repository database name	-	Yes	PERFDB or PERFDB[x] If PERFDB exists then x is replaced with a positive number
Repository database location	-	Yes	Default database path (DFTDBPATH) from database manager configuration

Parameter	Specify always	Specify in advanced mode	Default value for typical mode
Repository database small SMS table spaces location	-	Yes	Within database location directory
Table space type	-	Yes	SMS in version 4.1 DMS in version 4.1.0.1
Working directory			► Windows: `<OPM install dir>\RepositoryServer\instances` ► Linux/UNIX: `<Home directory of DB2 instance owner>/opm/v4`
WebSphere Application Server selection	Yes	-	-

2.3.2 Direct installation option

Use this option for a fresh installation of Optim Performance Manager version 4.1.0.1. This installation installs Optim Performance Manager and sets up the DB2 repository database and installs or sets up WebSphere Application Server.

Optim Performance Manager version 4.1.0.1 introduces new installation features. These new features broaden the environments in that the Optim Performance Manager can be installed, for example environments that use LDAP authentication. Additionally, these new features simplify the configuration and interaction with WebSphere Application Server compared to Optim Performance Manager 4.1

This section describes these features:

- Ability to specify the WebSphere Application Server profile name during installation that is to be used by Optim Performance Manager
- Ability to run the WebSphere Application Server profile that is used by Optim Performance Manager as the DB2 instance owner instead of root
- Ability to set up the authentication for the Optim Performance Manager Web console users through the repository database
- Ability to install Optim Performance Manager in environments that use LDAP authentication for DB2 users

WebSphere Application Server profile specification

You can specify the name of the WebSphere Application Server profile that Optim Performance Manager should use. If you selected to reuse an existing copy of WebSphere Application Server, the installer lists all available profiles that are either owned by the root user (Linux and UNIX), the SYSTEM user (Windows), or the owner of the DB2 instance. You can select one of the listed profile or specify a new one. If you specify a non-existing WebSphere Application Server profile name, the Optim Performance Manager installer creates it. Having the ability to specify the WebSphere Application Server profile name has the following advantages:

- You can use a dedicated WebSphere Application Server profile for Optim Performance Manager and, therefore, avoid interference with other WebSphere Application Server applications.
- You can share one WebSphere Application Server installation between multiple Optim Performance Manager installations on the same machine by specifying different profiles for each Optim Performance Manager installation.

A WebSphere Application Server profile defines the WebSphere Application Server runtime environment. Learn more about profiles:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/cpro_overview.html

Run WebSphere Application Server profile as DB2 instance owner

On Linux and UNIX, if you specify a new WebSphere Application Server profile, the Optim Performance Manager installer creates the profile under the DB2 instance owner of the instance used for Optim Performance Manager. You then can start the profile as the DB2 instance owner and it runs under the instance owner ID. This means that you can use the same user to run both WebSphere Application Server and Optim Performance Manager repository server.

On Windows, WebSphere Application Server is always running under the SYSTEM account.

Optim Performance Manager console authentication through the repository database

By default, only authorized users can logon to the Optim Performance Manager Web console. During installation, the Optim Performance Manager installer sets the authentication method. The default authentication method is through the repository database. That means that users who can connect to the repository database can be authorized to use the Optim Performance Manager Web console. Right after the installation, only the user who was specified as the DB2 user during installation can logon to the Optim Performance Manager Web console.

You can authorize more users to logon to Optim Performance Manager Web console through the Console Security panel. You also can use the Console Security panel to change the Optim Performance Web console security to be no authentication required or to be authenticated through WebSphere Application Server. If the authentication is through the WebSphere Application Server, then authorizing the users to use the Optim Performance Manager Web console is performed through the WebSphere Application Server administrative console.

For more details refer to 2.6, "User authorization" that introduces you to the authorization and privilege concept of Optim Performance Manager.

LDAP authentication for DB2 users

During Optim Performance Manager 4.1.0.1 installation, you can specify a DB2 user for the repository database access and a group for the Performance Expert client logon that are authenticated through LDAP. The prerequisite is that the DB2 instance that you select for Optim Performance Manager must be created and configured to use LDAP-based authentication through the LDAP security plug-in or using transparent LDAP.

Take the following additional considerations if you want to use LDAP authentication:

- On Linux and UNIX, the WebSphere Application Server profile must run under the DB2 instance owner of the DB2 instance used for Optim Performance Manager. This is set up automatically when you specify a new WebSphere Application Server profile.
- On Windows, the user who starts the Optim Performance Manager installation (for example Administrator) and any user who uses the `peconfig` configuration tool must be defined in the LDAP directory.
- If you prefer to authenticate users to the Optim Performance Manager Web console through WebSphere Application Server instead of through the repository database, you must configure WebSphere Application Server to use LDAP through the WebSphere Application Server administrative console. For more information about how to do that, refer to:

  http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_ldap.html

2.3.3 Update option

We recommend using this option only if you already have installed Optim Performance Manager version 4.1. To update Optim Performance Manager version 4.1 to version 4.1.0.1, apply Fix Pack 1 on the top of Optim Performance Manager 4.1 by selecting the same installation path. The fix pack installer does not change the existing WebSphere Application Server setup and the Optim Performance Manager console authentication. This means that none of the new installation features that we described in 2.3.2, “Direct installation option” on page 29 is applied during installation.

The following describes briefly how the Optim Performance Manager 4.1 installer sets up WebSphere Application Server and Optim Performance Manager console authentication:

- The WebSphere Application Server profile that the Optim Performance Manager uses is always `AppServer1`.
- On Linux and UNIX, the WebSphere Application Server profile is always set up to run as root. Therefore, you cannot use one ID to start both WebSphere Application Server and Optim Performance Manager. You must start the WebSphere Application Server as root and start the Optim Performance Manager repository server as the DB2 instance owner.
- The Optim Performance Manager Web console authentication is performed through WebSphere Application Server. After installation, the user that you specified as DB2 user during installation has access to the Optim Performance Manager Web console. You must add additional users who are allowed to logon to the Optim Performance Manager Web console using the WebSphere Application Server administrative console. Refer to the Optim Performance Manager Information Center about how to do that:

  http://publib.boulder.ibm.com/infocenter/idm/docv3/index.jsp?topic=/com.ibm.datatools.perfmgmt.installconfig.doc/pm_configure_user_access_to_opm.html

Tip: If you have already had the Optim Performance Manager 4.1 installed and would like to update to version 4.1.0.1 as well as leverage the new installation features of Optim Performance Manager 4.1.0.1, use this procedure:

- Uninstall Optim Performance Manager 4.1. Do not reconfigure it so you can keep the repository database.
- Install Optim Performance Manager version 4.1.0.1. During installation, specify that you want to use an existing database and provide the name of your repository database.

This method allows you to keep the repository database with the configuration of the monitored databases and collected data. In the meantime, you can take the advantages of the new installation features including running the WebSphere Application Server profile as the DB2 instance owner, using LDAP authentication, and specifying a dedicated WebSphere Application Server profile.

2.3.4 Migration option

You can migrate an existing Performance Expert V3 installation to Optim Performance Manager version 4.1.0.1. Migration means the following:

- The performance database of Performance Expert is used for Optim Performance Manager and updated to the enhanced database schema of Optim Performance Manager.
- On Linux and UNIX, the working directory of Performance Expert Server is used by Optim Performance Manager. On Windows, the default location is used as working directory, see “Parameter summary” on page 28. The important property files are copied to the new location.

The migration is possible only if the same DB2 instance as for Performance Expert Server is used for Optim Performance Manager.

The migration is performed during the installation of Optim Performance Manager. You must select the advanced installation mode to get the option to migrate. The installer lists the DB2 instances that are used for Performance Expert. You can select from the list the DB2 instance to be migrated.

The result of the migration is the following:

- Optim Performance Manager version 4.1.0.1 is installed and it uses the same DB2 instance and database as Performance Expert.

- The monitored databases are configured as they have been configured for the Performance Expert and Optim Performance Manager continues monitoring them.
- You can continue performing configuration changes using `peconfig` or you can use the configuration wizard of Optim Performance Manager Web console for further configuration.
- Performance Expert Server is not uninstalled, but cannot be started anymore. The `pestart` script is renamed to avoid using the script accidentally.

Further considerations include these:

- In Performance Expert, you configure DB2 instances for monitoring whereas in Optim Performance Manager you configure databases for monitoring. In Performance Expert, any monitoring configuration such as collection intervals or retention times applies to all databases of the monitored DB2 instance. After migration, Optim Performance Manager continues to monitor all added databases of the monitored instance using the same configuration settings. If you change the configuration settings for one of the databases of the monitored instance using the configuration wizard of the Optim Performance Manager Web console, the changes are applied to all databases of the monitored instance.
- Performance Expert client must be updated to version 4.1.0.1 as well.
- If you have the Extended Insight Feature activated for Performance Expert, the license is no longer valid for Optim Performance Manager. Run the Extended Insight Activation Kit again to activate the license.

2.4 Capacity planning

In this section, we provide guidelines for a high-level estimation of hard disk, CPU, and memory required for the Optim Performance Manager server to help you plan for your Optim Performance Manager environment. The guideline is for the Optim Performance Manager V4.1 that monitors non-partitioned databases. For capacity planning when the partitioned databases are monitored or for more detailed calculation, contact IBM support.

This section uses Optim Performance Manager monitoring configuration terminology. We discuss monitoring configuration in 3.3, “Configuring Optim Performance Manager” on page 91.

2.4.1 Factors influencing capacity planning of Optim Performance Manager servers

The Opim Performance Manager Server resource (CPU, memory, disk, and network) consumed by Inflight monitoring data depends heavily upon five variables:

- The Inflight monitoring profiles you intend to enable
- The Inflight sampling interval
- The retention time for Inflight data
- The schema of the monitored database server
- The workload being executed on the monitored database server

It is easy to see how the first three variables in the list above will contribute to the Opim Performance Manager Server resource consumption – the more monitoring profiles enabled, the more frequently the data is collected, and the greater the retention time of the monitoring data will all lead to an increase in Opim Performance Manager Server resource consumption to store and process the monitoring data.

For the last two variables, it is perhaps less obvious.

The schema of the monitored database server can have a large impact on disk space consumption if the “I/O and Disk Space” profile is enabled and:

- The “Collect table information” sub-option is selected and there are many tables in the monitored database server, or
- The “Collect table space information” sub-option is selected and there are many table spaces in the monitored database server, or
- The “Collect tablespace information with container information” sub-option is selected and there are many table space containers in the monitored database server.

For example, consider a database which contains 50,000 tables. If the “Collect table information” sub-option is selected for this database, then at each sampling interval, Opim Performance Manager will collect information on each of those 50,000 tables. If the default sampling interval of one minute, and Inflight retention period of 50 hours are used, then the Opim Performance Manager Server will hold 150 million (50,000 * 50 * 60) records to describe these tables. This is likely to consume a lot of Opim Performance Manager Server disk (and other) resource in order to process and store this information.

Similarly, consider a DB2 partitioned database system with 50 partitions, 20 table spaces on each partition, and five containers for each table space on each partition. If the “Collect tablespace information with container information” sub-option is selected, this will result in monitoring information for 5000 table space containers being processed and collected by Opim Performance Manager – during each sampling interval. Consequently, the Opim Performance Manager Server will hold 15 million (5,000 * 50 *60) records to describe the table space containers.

The workload being executed on the monitored database can also impact the resource consumed by the Opim Performance Manager Server. For example, if the “Dynamic SQL” profile is enabled and the monitored database has a large package cache (DB2 database configuration parameter PCKCACHESZ). Consider a database with a package cache of 2GB, which on average holds 15,000 SQL statements. At each sampling interval, Opim Performance Manager will retrieve information about each of the SQL statements in the package cache (including the SQL statement text) and stores this in the repository database. If the default sampling interval of one minute, and Inflight retention period of 50 hours are used, then the Opim Performance Manager Server will store 45 million (15,000 *50 * 60) records to describe the SQL statements.

Another example would be if the “Locking” profile and “Collect lock wait information” sub-option are enabled, and the monitored database server encounters lots of lock wait situations, then collection of this monitoring data will consume Opim Performance Manager Server resource.

As can be seen from these examples, the amount of monitoring data processed and held in the Opim Performance Manager Server can be substantial.

The Opim Performance Manager Server resource consumed by the Extended Insight depends heavily upon the following factors:

- The Extended Insight monitoring profiles enabled.
- The retention times for aggregation levels 1, 2, 3 and 4.
- The number of unique SQL statements in the workload being monitored.
- The transaction rate at the database server.

For the first two items, the more EI monitoring profiles enabled, and the greater the retention period of the aggregated data, then the greater the Opim Performance Manager resource required to store and process the data.

When “Collect statement and transaction metrics on client” is enabled, then the number of unique SQL statements in the workload being monitored will have an impact on Opim Performance Manager Server resource consumed. Information about the SQL statements and transactions issued on the client or Application Server are collected at one minute intervals. All statements issued during this one minute interval are aggregated based on a hash code calculated from the SQL statement text. Therefore, if the workload contains many different SQL statements more data will be aggregated than if the workload contains just a few different statements.

As an example, consider application A which uses literals, and application B which uses parameter markers. Both applications achieve exactly the same end-result, and both have Extended Insight monitoring enabled.

Table 2-2 and Table 2-3 lists the SQL statements run during a one-minute interval by the application A and B respectively.

Table 2-2 Application A issues the following SQL during a one-minute interval

SQL Statement	Number of times executed
INSERT INTO ORDERS (order_num, part, quantity, cost) VALUES (1,'widget A',3,5,60)	1
INSERT INTO ORDERS (order_num, part, quantity, cost) VALUES (2,'widget B',2,12.0)	1
INSERT INTO ORDERS (order_num, part, quantity, cost) VALUES (3,'widget C',7,2.21)	1
INSERT INTO ORDERS (order_num, part, quantity, cost) VALUES (4,'widget A',5,8.60)	1
SELECT order_num, cost FROM ORDERS WHERE order_num=3	1
SELECT order_num, cost FROM ORDERS WHERE order_num=1	1

Table 2-3 Application B issues the following SQL during a one-minute interval

SQL Statement	Number of times executed
INSERT INTO ORDERS (order_num, part, quantity, cost) VALUES (?,?,?,?)	4
SELECT order_num, cost FROM ORDERS WHERE order_num=?	2

For application A, Opim Performance Manager will have six distinct records to process and store. For application B, Opim Performance Manager will have only two records to process and store. Therefore, the Opim Performance Manager resource required to monitor application A will be more than that required to monitor application B.

If we consider a more realistic example where application A issues 10,000 unique SQL statements per minute (because it does not use parameter markers), and each statement is executed just once. In this case, Opim Performance Manager will process 10,000 different statements each minute. If the default level 1 retention period of 24 hours is used, then Opim Performance Manager will requires resource to store and process 14.4 million records during that period.

Now let us consider the same application which issues 10,000 SQL statements per minute, but because it users parameter markers, only 500 of these statements are unique. Therefore, due to aggregation of the statements, Opim Performance Manager will only process 500 different statements each minute. If the default level 1 retention period of 24 hours is used, then Opim Performance Manager will requires resource to store and process 720,000 records during that period.

The primary cause for having many unique SQL statements in your workload would be when literal values are used instead of host variables and parameter markers. Best practice dictates that good coding should use host variables or parameter markers.

When “Collect transaction metrics on data server” EI profile is enabled, the number of transactions (commits plus rollbacks) issued on the monitored database server will influence the resource consumed by Opim Performance Manager. If the Unit of Work event monitor is turned on, it will collect information on each transaction executed on the database server, and Opim Performance Manager will periodically read and process this information. Therefore, the greater the number of transactions occurring on the monitored database server, the greater the Opim Performance Manager resource required to process this data. For example, suppose a database is executing 500 transactions per second. Opim Performance Manager will then have to process and store 30,000 transactions every minute. If the default Extended Insight level 1 retention period of 24 hours is used, then Opim Performance Manager will store 43.2 million records to describe the transactions at 1 minute aggregation intervals.

2.4.2 Hard disk requirement estimation

In order to provide monitoring history, Optim Performance Manager keeps and maintains the collected data in the repository database that can be big if much monitoring data is collected and retained. Apart from the disk space used for DB2 and Optim Performance Manager installation, the monitoring data itself, along with the indexes, is the major disk space consumer of the repository server. The database log files also take some additional space. The installation and log files consume a relatively small and constant space, compared with the repository database. To learn more about the installation consumption, refer the following link:

http://publib.boulder.ibm.com/infocenter/idm/docv3/index.jsp?topic=/com.ibm.datatools.perfmgmt.installconfig.doc/pm_install_reqs.html

The majority of the monitoring data kept in the repository database are the Inflight monitoring data and the Extended Insight monitoring data. Therefore, for Hard Disk Capacity, this section focuses on the sizing of the Inflight monitoring data sizing and the Extended Insight monitoring data.

We provide algorithm and math formulas for estimating disk space consumption.

Disk space required by the Inflight monitoring data

The disk space consumption of Inflight monitoring data depends on many factors including the number of tables and buffer pools in the monitored database. To make the calculation simple and quick, here is an algorithm for high level estimation for the disk space consumed for *each* monitored database.

```
(0.3GB * retention period for Inflight data)/sampling interval
```

where `retention period for Inflight data` is in *hour* and `sampling interval` is in *minute*.

You may set different sampling interval and retention period of Inflight data for the monitored databases. To calculate the total disk space consumption of Inflight monitoring on all monitored databases, apply the above algorithm to each monitored data server and then make a sum. The formula shown in Figure 2-1 on page 39 calculates the overall disk space consumption of Inflight monitoring on *all* databases and the result is in gigabytes.

$$HD_{inf}(GB) = \sum_{i=1}^{ndb} 0.3 * \frac{1}{samp_i} * retain_i$$

Figure 2-1 Overall disk space consumption of Inflight monitoring on all databases

The parameters in the formula are:

- *ndb*: The number of monitored database
- *samp*: The sampling interval for Inflight monitoring
- *retain*: Inflight data Retention Time (hour) for each monitored database

Tips: In the formula shown in Figure 2-1 on page 39, the number of the monitored databases is *ndb*. For each of the monitored database, which is represented as the *i(th)* database (*i* is from 1 to *ndb*), the corresponding sampling interval and retention period for that database is $samp_i$ and $retain_i$. The values can be different from each databases. For each value of i, calculate the value of $(0.3*samp_i)/retain_i$. Then total the results to obtain the overall disk space consumption of Inflight monitoring on all databases.

This "sum" algorithm is also used in other formulas in this section.

Example 1

Suppose one customer wants to monitor five databases and the sampling interval and retention time of Inflight monitoring for each them are as shown in Table 2-4.

Table 2-4 Example: 5 databases to be monitored

Monitored database name	Sampling interval in minute (*samp*) for this monitored database	Retention period in hour (*retain*) for this monitored database	Disk consumption (GB) of Inflight monitoring for this monitored database
DB1	1	50	0.3*50/1 = 15
DB2	5	240	0.3*240/5 = approx 15
DB3	3	168	0.3*168/3 = approx 17
DB4	15	300	0.3*300/15 = 6
DB5	10	300	0.3*300/10= 9

As a result, the overall disk space consumption of Inflight monitoring for these five monitored databases is (15+15+17+6+9 = 62GB).

Disk required by the Extended Insight monitoring data

The size of the Extended Insight client and Extended Insight server monitoring data depends mainly on the following factors:

- Transaction rate (*txt_rate*): This rate refers to the average number of executed transactions of the monitored application per minute during the retention of the aggregation level one.

- Unique SQL statement rate (`uni_sql`): This rate is the average number of unique SQL statements executed per minute during the retention of aggregation level one.
- Retention time (`retain_ei_a1, retain_ei_a2, retain_ei_a3, retain_ei_a4`): Retention time refers to the retention time you set for each of the aggregation levels when the Extended Insight monitoring is configured.

Here we provide the estimation methods for using the default and non-default retention time for the aggregation levels.

Using the default retention time

If you take the default retention level values for aggregation 1, 2, 3, and 4 – one day, one month, three months, and two years, respectively, you can apply the formula shown in Figure 2-2 to calculate the disk space consumption for *each* monitored database. The result of this formula is gigabytes.

$$HD_{ei\_each\_db}(GB) = 7.5 * ei * \frac{uni\_sql}{1000} + 1 * txt * \frac{txt\_rate}{1000}$$

Figure 2-2 disk space consumption of Extended Insight monitoring for each database

The parameters for this formula are:

- `ei`: Whether Extended Insight monitoring is enabled on a monitored database. If yes, ei=1; otherwise, ei=0
- `uni_sql`: The average number of Unique SQL statements per minute executing on a monitored database
- *txt*: Whether "Collect transaction metrics on data server" is selected in monitoring configuration. If yes, txt=1; otherwise, txt=0. It is only applicable when the monitored database is DB2 v97fp1 or above.
- `txt_rate`: Transaction rate per minute on a monitored database

The total disk space consumption of the Extended Insight monitoring for *all* databases is the sum of the disk space required of each monitored database. Figure 2-3 shows the formula. The result of this formula is gigabytes.

$$HD_{ei}(GB) = \sum_{i=1}^{ndb}(7.5 * ei_i * \frac{uni\_sql_i}{1000} + 1 * txt_i * \frac{txt\_rate_i}{1000})$$

Figure 2-3 Overall disk space consumption of Extended Insight monitoring for all databases

The parameters for this formula are:

- `ndb`: The number of monitored database.

- *ei*: Whether Extended Insight monitoring is enabled on a monitored database. If yes, ei=1; otherwise, ei=0
- *uni_sql*: The average number of Unique SQL statements per minute executing on a monitored database
- *txt*: Whether "Collect transaction metrics on data server" is selected in monitoring configuration. If yes, txt=1; otherwise, txt=0. It is only applicable when the monitored database is DB2 v97fp1 or above.
- *txt_rate*: Transaction rate per minute on a monitored database.

Example 2

Suppose one customer wants to monitor five databases and the monitoring profile is as shown in Table 2-5 on page 42.

Table 2-5 Extended Insight monitoring profiles for the example monitored databases

Monitored database name	Whether Extended Insight monitoring is enabled (*ei*)?	The average number of Unique SQL statements per minute (*uni_sql*) executing on a monitored database	Whether "Collect transaction metrics on data server" is selected in monitoring configuration (*txt*)	Transaction rate per minute (*txt_rate*) on a monitored database	Disk space consumption (GB) of Extended Insight monitoring for this monitored database
DB1	1 (YES)	6,000	1 (YES)	4,500	49.5
DB2	1 (YES)	4,000	1 (YES)	10,000	40
DB3	1 (YES)	5,000	0(NO)	3,000	37.5
DB4	0 (NO)	15,000	0 (NO)	5,000	0
DB5	0(NO)	5,000	0(NO)	3,000	0

As a result, the overall disk space consumption for Extended Insight monitoring for these 5 databases is (49.5+40+37.5=127GB).

Using non-default retention time

Considering that you may want to set different retention time for each aggregation level, here are the formulas to calculate the overall disk space consumption of the Extended Insight monitoring for *all* monitored databases by each aggregation level.

Use the formula shown in Figure 2-4 to calculate the overall disk space consumption of Extended Insight monitoring *aggregation level 1* for all monitored databases.

$$HD_{ei\_agg\_1}(GB) = \left(60 * \sum_{i=1}^{ndb} (ei_i * \frac{uni\_sql_i}{1000} * retain\_ei\_a1_i) + 5 * \sum_{i=1}^{ndb} (txt_i * \frac{txt\_rate_i}{1000} * retain\_ei\_a1_i)\right) / 1024$$

Figure 2-4 Overall disk space consumption of Extended Insight monitoring aggregation level 1 for all databases

The parameters in the formula are:

- *ndb*: The number of monitored database.
- *ei*: Whether Extended Insight monitoring is enabled on a monitored database. If yes, `ei`=1; otherwise, `ei`=0.
- *uni_sql*: The average number of Unique SQL statements per minute executing on a monitored database.
- *retain_ei_a1*: The retention time (hour) for aggregation level 1 of Extended Insight monitoring for a database.
- *txt*: Whether “Collect transaction metrics on data server” is selected in monitoring configuration. If yes, txt=1; otherwise, txt=0. It is only applicable when the monitored database is DB2 v97fp1 or above.
- *txt_rate*: Transaction rate per minute on a monitored database.

The formula shown in Figure 2-5 is used for calculating the overall disk space consumption of Extended Insight monitoring *aggregation level 2* for all monitored databases.

$$HD_{ei\_agg\_2}(GB) = \left(4 * \sum_{i=1}^{ndb} (ei_i * \frac{uni\_sql_i}{1000} * retain\_ei\_a2_i) + 0.3 * \sum_{i=1}^{ndb} (txt_i * \frac{txt\_rate_i}{1000} * retain\_ei\_a2_i)\right) / 1024$$

Figure 2-5 Overall disk space consumption of Extended Insight monitoring aggregation level 2 for all databases

The parameters in this formula are:

- *uni_sql*: The average number of Unique SQL statements per minute executing on a monitored database.
- *txt_rate*: Transaction rate per minute on a monitored database for which “Collect transaction metrics on data server” is selected.
- *ndb*: The number of monitored database.
- *ei*: Whether Extended Insight monitoring is enabled on a monitored database. If yes, `ei`=1; otherwise, `ei`=0.

- *retain_ei_a2*: The retention time (hour) for aggregation level 2 of Extended Insight monitoring for a database.

Figure 2-6 on page 44 shows the formula used for calculating the overall disk space consumption of Extended Insight monitoring *aggregation level 3* for all monitored databases.

$$HD_{ei\_agg\_3}(GB) = \left(1 * \sum_{i=1}^{ndb}(ei_i * \frac{uni\_sql_i}{1000} * retain\_ei\_a3_i) + 0.01 * \sum_{i=1}^{ndb}(txt_i * \frac{txt\_rate_i}{1000} * retain\_ei\_a3_i)\right) / 1024$$

Figure 2-6 Overall disk space consumption of Extended Insight monitoring aggregation level 3 for all databases

The parameters for this formula are:

- *uni_sql*: The average number of Unique SQL statements per minute executing on a monitored database.
- *txt_rate*: Transaction rate per minute on a monitored database.
- *ndb*: The number of monitored database.
- *ei*: Whether Extended Insight monitoring is enabled on a monitored database. If yes, ei=1; otherwise, ei=0.
- *retain_ei_a3*: The retention time (hour) for aggregation level 3 of Extended Insight monitoring for a database.

Figure 2-7 shows the formula for calculating the overall disk space consumption of Extended Insight monitoring aggregation level 4 for all monitored databases.

$$HD_{ei\_agg\_4}(GB) = \left(0.05 * \sum_{i=1}^{ndb}(ei_i * \frac{uni\_sql_i}{1000} * retain\_ei\_a4_i) + 0.004 * \sum_{i=1}^{ndb}(txt_i * \frac{txt\_rate_i}{1000} * retain\_ei\_a4_i)\right) / 1024$$

Figure 2-7 Overall disk space consumption of Extended Insight monitoring aggregation level 4 for all databases

The parameters in this formula are:

- *uni_sql*: The average number of Unique SQL statements per minute executing on a monitored database.
- *txt_rate*: Transaction rate per minute on a monitored database.
- *ndb*: The number of monitored database.
- *ei*: Whether Extended Insight monitoring is enabled on a monitored database. If yes, ei=1; otherwise, ei=0.
- *retain_ei_a4*: The retention time (hour) for aggregation level 4 of Extended Insight monitoring for a database.

The total disk space consumption of Extended Insight monitoring including all aggregation levels is the sum of the data size of four aggregation levels. Figure 2-8 shows the formula.

$$\boxed{HD_{ei}(GB) = HD_{ei\_agg\_1}(GB) + HD_{ei\_agg\_2}(GB) + HD_{ei\_agg\_3}(GB) + HD_{ei\_agg\_4}(GB)}$$

Figure 2-8 Overall disk space consumption of Extended Insight monitoring for all databases

To have the overall disk space consumption including both Inflight monitoring and Extended Insight monitoring, use the formula shown in Figure 2-9.

$$\boxed{HD_{total}(GB) = HD_{\inf}(GB) + HD_{ei}(GB)}$$

Figure 2-9 Disk space consumption for both Inflight and Extended Insight monitoring

2.4.3 CPU requirement estimation

The repository server retrieves data from monitored application and data server at regular intervals. Ideally, the overall CPU consumption should show regular peaks. However, the sampling to all monitored system is not synchronized, therefore, it also makes sense if you do not see the regular peaks when you have more than one database monitored. To make sure that the repository server works with acceptable performance, the available CPU should be greater than the peak values. In this document, the overall average of CPU consumption is also provided for a reference.

The overall CPU consumption mainly depends on the number of monitored data servers, number of database objects being monitored, transaction rate on each monitored database, and the number of console users. For any of these factors, the bigger it is, the higher the CPU consumption. The biggest contributors are transaction rate, number of monitored data servers, and number of console users. Table 2-6 shows a summary of required CPU based on these factors.

> **Note:** CPU requirements in Table 2-6 are based on IBM pSeries® P5_64-bit 1.9GHz processors. Appropriate conversion factors should be applied to convert the Opim Performance Manager Server CPU requirements to your intended deployment architecture.

Table 2-6 CPU requirements

Number of monitored data servers	Transaction rate on each monitored data server (K/minute)	Number of console users	Overall CPU when both Extended Insight client and Extended Insight server are disabled on all monitored database	Overall CPU when both Extended Insight client and Extended Insight server are enabled on all monitored database
10	1	5	2 CPU, above 2.0GHZ	3 CPU, above 2.0GHZ
20	2	10	2 CPU, above 2.0GHZ	6 CPU, above 2.0GHZ
30	5	15	6 CPU, above 2.0GHZ	8 CPU, above 2.0GHZ
30	20	15	8 CPU, above 2.0GHZ	12 CPU, above 2.0GHZ
30	40	20	10 CPU, above 2.0GHZ	16 CPU, above 2.0GHZ

2.4.4 Memory requirement estimation

To make sure that the Optim Performance Manager server runs with acceptable performance, adequate memory should be available for Optim Performance Manager console server (WebSphere Application Server), the repository server, and the repository database.

The memory capacity for an Optim Performance Manager system can be derived from the following formula:

```
OPM memory (GB) = Repository Server memory + Console Server memory +
                    Repository DB memory + Operating System memory
```

We suggest to have 2GB RAM available for the operating system memory.

Memory required by the repository server

The memory consumed by the repository server depends on the number of monitored data servers and their transaction rate. The number of monitored applications on each data server also impacts the repository server memory consumption. To make a quick estimate, in this section, the capacity estimation involving Extended Insight is based on only one WebSphere application running for each data server. If you have more than one WebSphere Application severs for one database, contact IBM support for memory requirement estimation.

In general, the memory consumed by the repository server can be calculated by the sum of the following items:

- Basic consumption: approximately 200MB.
- Inflight monitoring for each monitored data server: 8MB.

- Extended Insight monitoring for each monitored database: Increases by 5MB each time the transaction rate increases by 1000 transactions per minute.

Figure 2-10 shows a simple formula to estimate the memory consumption of the Optim Performance Manager repository server:

$$Mem\_rs = 200 + \sum_{i=0}^{ndb} 8 + 5 * EI_i * Tr_i$$

Figure 2-10 Memory consumption of Optim Performance Manager repository server

The parameters used in this formula are:

- *Mem_rs* is the memory consumed by the repository server, in megabytes.
- *ndb* is the total number of monitored data servers.
- *ei* indicates whether the Extended Insight client or server monitoring is enabled on the data server. When either Extended Insight client or server is enabled, ei=1, otherwise, ei=0.
- *Txt* indicates whether "Collect transaction metrics on data server" is selected in monitoring configuration. If yes, txt=1; otherwise, txt=0. It is only applicable when the monitored database is DB2 97 Fix Pack 1or above.
- *Txt_rate* is the transaction rate per minute on a monitored database.

To have an acceptable performance, additional memory is necessary to reduce the Java garbage collection overhead. The minimum requirement should be 1.5GB.

Example 3

This example demonstrates how you can use the formula shown in Figure 2-10 to estimate the memory required by Optim Performance Manager repository server. Table 2-7 shows the information for the five monitored database.

Table 2-7 Estimating the repository server memory requirement

Monitored database name	Whether Extended Insight monitoring is enabled (*ei*)	Whether "Collect transaction metrics on data server" is selected in monitoring configuration (*txt*)	Transaction rate per minute (*txt_rate*) on a monitored database	Memory required by Optim Performance Manager Repository Server (MB)
DB1	1 (YES)	1 (YES)	25,000	133
DB2	1 (YES)	1 (YES)	20,000	108
DB3	1 (YES)	0(NO)	30,000	8

Monitored database name	Whether Extended Insight monitoring is enabled (ei)	Whether “Collect transaction metrics on data server” is selected in monitoring configuration (txt)	Transaction rate per minute (txt_rate) on a monitored database	Memory required by Optim Performance Manager Repository Server (MB)
DB4	0 (NO)	0 (NO)	5,000	8
DB5	0(NO)	0(NO)	3,000	8

As a result, the total memory required by Optim Performance Manager repository server is 200+133+108+8+8+8 = approx 465MB.

Memory required by the Optim Performance Manager console

Memory consumed by the Optim Performance Manager console can simply refer to the free space in WebSphere Application Server Java heap that is dedicated to the Optim Performance Manager console. To allow an acceptable performance for the Optim Performance Manager, extra memory space is necessary to reduce the garbage collection overhead. Generally, the memory required by the Optim Performance Manager console is:

- If the number of concurrent console users is less or equal to 10, then 512M is required;
- If the number of concurrent console users is between 11 and 30, then 768M is required.
- If the number of concurrent console users is between 31 and 50, 1GB is required.

More than 50 concurrent console users are not recommended considering the acceptable performance.

Memory required by the repository database

Considering buffer pool consumption, the approximate memory required by a repository database is:

- If the number of monitored data servers is less or equal to 10, 3GB is required;
- If the number of monitored data servers is between 11 and 20, then 5GB is required;
- If the number of monitored data servers is between 21 and 30, then 7GB is required.

More than 30 data servers monitored by the same Optim Performance Manager instance is not tested in lab. If you need capacity for more than 30 data servers, contact IBM technical support for further analysis.

Example 4

This example is based on the scenario described in "Example 3" on page 47. Suppose that there are five database administrator to access the Optim Performance Manager console concurrently, the memory required will be:

- For the Optim Performance Manager console, 512MB of memory is required.
- For the repository database, 3GB of memory is required
- The total memory consumption of Optim Performance Manager would be (465MB + 512MB + 3GB = approx 4GB)

2.5 Storage options

For each database that Optim Performance Manager monitors, Optim Performance Manager creates two table spaces in the repository database, one for holding the short-term history data and one for holding the long-term history data. The table spaces are created when you configure a database for monitoring after the Optim Performance Manager is installed. During installation, in advanced installation mode, you can specify the type of table spaces (SMS, DMS, or automatic storage) the Optim Performance Manager should create. If you use the typical installation then the table space type defaults to DMS.

2.5.1 Table space type selection

Each table space type has its characteristics. The following DB2 documentation provides a comparison:

`http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.dbobj.doc/doc/c0055446.html`

From a performance point of view, DMS and automatic storage table spaces are faster than SMS table spaces, especially for large tables. Because some tables in the Optim Performance Manager repository database can grow large, we recommend use DMS or automatic storage table spaces, considering performance.

Optim Performance Manager creates table spaces as *large* table space. If you consider the maximum table space size, the DMS and automatic storage table spaces allow higher maximum size than SMS stable spaces when the table spaces are created as large table space. For SMS table spaces the large option is not available. See the table space size comparison in DB2 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.dbobj.doc/doc/c0052381.html

If the way to reclaim storage is important for you then SMS table spaces are the easiest to handle because the table space size shrinks whenever data is deleted from the table space. If the capacity planning results a disk space shortage, take the storage reclaiming behavior into account for your table space type decision. During Optim Performance Manager runtime it might happen that you run out of storage although Optim Performance Manager deletes collected data from the repository database regularly and automatically. Then you must delete data from the repository database manually in order to free disk space. Learn how DMS and Automatic Storage table spaces reclaim storage:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.dbobj.doc/doc/c0055392.html

2.5.2 Table space naming and usage

For each monitored instance, Optim Performance Manager creates the following table spaces to save the collected performance data:

- `SHORTTERM_`*<instance_id>*
- `LONGTERM_`*<instance_id>*

The variable *<instance_id>* is a unique positive number that Optim Performance Manager assigns to each monitored database. The first database that you add for monitoring most likely receives the instance ID 1. For this database the table spaces `SHORTTERM_1` and `LONGTERM_1` are created.

All the collected performance data that can be displayed on the Optim Performance Manager Web console are saved in the `SHORTTERM_<instance_id>` table space. This table space can grow large. 2.4, "Capacity planning" on page 34 discusses the calculation of the required space for this table space. Retention times are set for the collected data and Optim Performance Manager deletes the data automatically when the retention time is reached.

The `LONGTERM_<instance_id>` table space stores the collected and aggregated data for Performance Expert client performance warehouse for long-term trend analysis and reporting. This data is not deleted automatically. Collecting this data is optional. It is collected only if you switch on the Performance Warehouse monitoring profile during the configuration of your monitored database. Since the data is stored in an aggregated format, the size of this table space is typically much smaller than the `SHORTTERM_<instance_id>` table space.

After you configured a database for monitoring, you can check the unique instance ID assigned to the database by Optim Performance Manager to learn which table spaces were created for the database. To check the ID, use the **peconfig -console** command from the RepositoryServer/bin directory of your Optim Performance Manager installation. Within peconfig, use the **list** command that returns information include the instance ID, for example:

```
Instance ID = 1
----------------
   Enabled = Yes
   Status = Inactive
   CIM Object Manager enabled = No
   Monitored Instance Alias = LOCALHOST_50000_PEDEMO
   Node, Host, Port/Service name = NODE7507, LOCALHOST, 50000
   Database, remote alias, local alias, EVM = PEDEMO, PEDEMO, PMDB3902, OFF
```

In this example, the Optim Performance Manager uses instance ID 1 for the configured database. This means that the table spaces for this database are named SHORTTERM_1 and LONGTERM_1.

2.5.3 Table space location

If the DMS or SMS table space type is selected during installation, you specify the table space path during the configuration of a monitored database. Optim Performance Manager creates both table spaces in the specified path. You can either specify an explicit path or choose the default location. If you choose the default location then the table spaces are created under the database directory.

Tip: If you monitor multiple databases then we recommend that you specify an explicit path for each monitored database and ensure that the specified paths reside on multiple disks in order to avoid I/O bottlenecks.

If the Automatic Storage table space type is selected during installation, you specify the storage paths to be used during installation. When configuring a monitored database, Optim Performance Manager creates both table spaces using the automatic storage option. If you need to add storage paths later, you can use the **ALTER DATABASE** command.

2.5.4 Table space DDL

In this section we show sample DDL statements for each table space type that Optim Performance Manager uses to create the table space. This helps you to understand which parameters Optim Performance Manager uses to create the table spaces:

- SMS:

```
CREATE REGULAR TABLESPACE SHORTTERM_1 IN NODEGROUP PENG PAGESIZE 8K MANAGED
BY SYSTEM USING ('SHORTTERM_1 $N')  BUFFERPOOL DATA
```

- DMS:

```
CREATE LARGE TABLESPACE SHORTTERM_1 IN NODEGROUP PENG PAGESIZE 8K MANAGED
BY DATABASE USING ( FILE 'SHORTTERM_1' 5000)  BUFFERPOOL DATA AUTORESIZE
YES
```

- Automatic Storage:

```
CREATE LARGE TABLESPACE SHORTTERM_1 IN NODEGROUP PENG PAGESIZE 8K MANAGED
BY AUTOMATIC STORAGE INITIALSIZE 40M BUFFERPOOL DATA AUTORESIZE YES
```

2.6 User authorization

For user authorization, Optim Performance Manager uses privileges to control the following tasks:

- Global tasks:

  These tasks are checked at Optim Performance Manager and include:

  – Logon to the Optim Performance Manager web console

  – Administrative tasks

    • Changing global notification settings, for example, SMTP server settings

    • Accessing collected monitoring data if the monitored database is not accessible

    • Changing authentication method

    • Granting console privileges to users

  – Configuration tasks

    • Adding databases for monitoring

- Database-specific monitoring tasks:

  These tasks are checked at the monitored database and include:

  – Collecting monitoring data
  – Configuring monitoring
  – Access to collected monitoring data
  – Changing alert settings such as thresholds and notifications

2.6.1 User authorization for global tasks

Optim Performance Manager uses *console privileges* to control who can logon to the Optim Performance Manager web console and perform the administrative tasks. For the configuration tasks, additional privileges are required.

To logon to the web console, the user either needs the *Administrator*, *Operator*, or *Viewer* console privilege. A user with the Administrator privilege can execute the administrative tasks, users with Operator or Viewer privilege cannot. This is the only difference between these three privileges.

Granting console privileges

The DB2 user that you specify during installation automatically receives the Administrator privilege, therefore, this user can logon to the web console right after installation. You can grant the console privileges to more users at any time after installation.

Authentication is controlled by repository database

If you use the direct installation option or migration option to install Optim Performance Manager V4.1.0.1l, the authentication is controlled by the repository database. To grant console privileges, as a user with Administrator privilege, use the *Console Security* panel to grant the Administrator, Operator, or Viewer privilege to DB2 users, DB2 groups, or DB2 roles. The user, group, or role that you specify must already have the CONNECT privilege to the repository database.

By granting the Administrator, Operator, or Viewer privilege to a user, group or role, Optim Performance manager grants execute rights to one of these user defined functions in the repository database:

- `DSWEBSECURITY.CANVIEW`
- `DSWEBSECURITY.CANOPERATE`
- `DSWEBSECURITY.CANADMINISTER`

Alternatively, you can grant the console privileges directly using the DB2 command `GRANT EXECUTE`, for example:

```
GRANT EXECUTE ON FUNCTION DSWEBSECURITY.CANVIEW TO USER <user id>
```

Authentication is controlled by WebSphere Application Server

If you use the update option for Optim Performance Manager V4.1.0.1 installation, the authentication is controlled by WebSphere Application Server because this is the only authentication method that is available in Optim Performance Manager V4.1. Use the WebSphere Administrative Console to grant console privileges to more users. Refer to the Optim Performance Manager Information Center about how to do that:

http://publib.boulder.ibm.com/infocenter/idm/docv3/index.jsp?topic=/com.ibm.datatools.perfmgmt.installconfig.doc/pm_configure_user_access_to_opm.html

If you want to switch to the authentication controlled by the repository database, use the *Console Security* panel.

Privileges required for configuration tasks

For the configuration tasks within the web console you need additionally DB2 privileges on the repository database. If you add a database for monitoring, Optim Performance Manager inserts entries for this database into the repository database tables with the `IBMPDQ` schema. To add a database for monitoring, you must authenticate to the repository database using a user who has SELECT and UPDATE privileges on tables within this schema. The DB2 user that you specify during installation has DBADM authority on the repository database. Either use this one or any other with the appropriate DB2 privileges on the repository database.

For the following tables, the SELECT and UPDATE privileges are required:

- `IBMPDQ.MANAGED_DATABASE`
- `IBMPDQ.MANAGED_DATABASE_PROPS`
- `IBMPDQ.PROFILE`
- `IBMPDQ.PROFILE_PROPS`

The following is an example GRANT statement to grant UPDATE privilege:

```
GRANT UPDATE ON TABLE IBMPDQ.PROFILE TO USER <user id>
```

2.6.2 User authorization for database-specific monitoring tasks

Optim Performance Manager uses database privileges on the monitored database and database manager authorities to control the monitoring tasks. You define the users that execute monitoring tasks during configuration of a monitored database from the Optim Performance Manager web console. The first user you define is the *Optim Performance Manager collection user*. Optim Performance Manger uses this user to collect monitoring data from the monitored database. The monitoring data consists of snapshot and event monitor data. This user must have the following privileges and authorities on the monitored system:

- For DB2 Version 9.7:
  - DBADM
  - SECADM
  - SYSMON or SYSADM
- For DB2 Version 9.5 and earlier:
  - SYSADM

Further, you grant privileges to DB2 users, DB2 groups, or DB2 roles to authorize them to

- Access collected monitoring data.

  Grant the *canMonitor* privilege to accomplish this.

- Change alert settings such as thresholds and notifications.

  Grant the *canManageAlerts* privilege to accomplish this. The `canManageAlerts` privilege includes the `canMonitor` privilege.

The DB2 users, DB2 groups, or DB2 roles you grant a privilege to must already exist and have `CONNECT` privilege on the monitored database.

The Optim Performance Manager collection user automatically receives the `canMonitor` and `canManageAlerts` privilege.

By granting one of these privileges to a user, group, or role, Optim Performance Manager grants `EXECUTE` right to one of these user defined functions that Optim Performance Manager creates in the monitored database:

- `OPM.CAN_MONITOR`
- `OPM.CAN_MANAGE_ALERTS`

The user defined functions do not do any operation, they are in the monitored database to check the `EXECUTE` right.

Alternatively, you can grant the privileges on the monitored database by using the DB2 command `GRANT EXECUTE`, for example:

```
GRANT EXECUTE ON FUNCTION OPM.CAN_MONITOR TO USER <user id>
```

The `canMonitor` privilege is checked when you open a monitoring dashboard in the web console, select a database, and specify user credentials for this database. If the user you specify has `EXECUTE` right on the `CAN_MONITOR` user defined function, the `canMonitor` privilege is confirmed and you are allowed to look at the monitoring data of this database. The `canManageAlerts` privilege is checked on the alert notification and configuration dashboards when you select a database and specify user credentials for this database.

To change the monitoring configuration you must specify the credentials of the Optim Performance Manager collection user. Otherwise, any changes to the configuration cannot be saved.

2.6.3 User authorization examples

In this section, we show two examples that illustrate the flow of privilege checking within Optim Performance Manager.

Example 1: Adding and configuring a database for monitoring

Follow the steps in Figure 2-11 on page 56 to see what authorizations are required to add a database and configure it for monitoring.

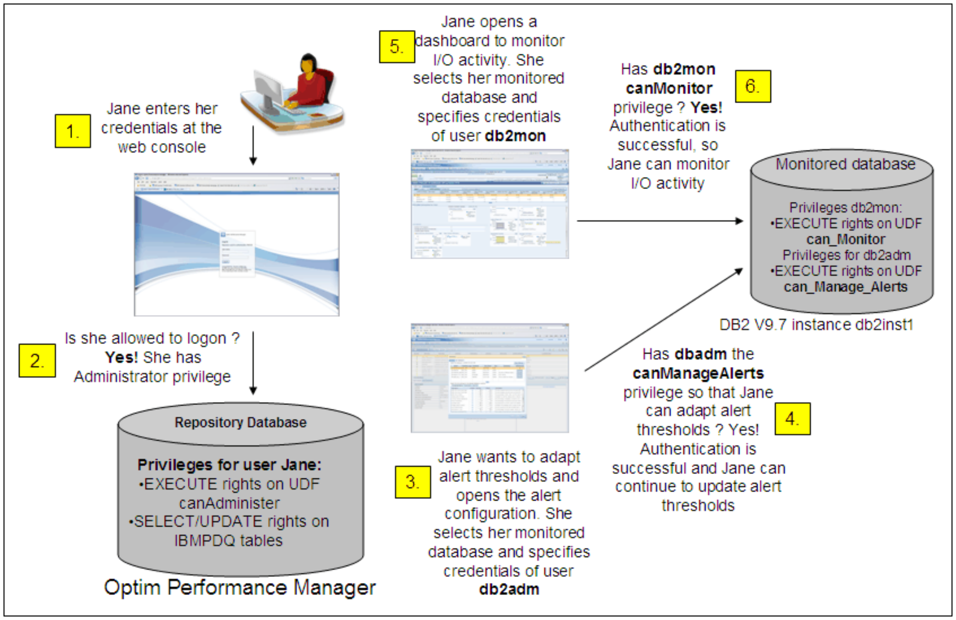


Figure 2-11 User authorization checking during configuration

Example 2: Monitoring using dashboards and alerts

Follow the steps in Figure 2-12 on page 57 to see what authorizations are needed to look at monitoring data of a database and to change alert settings of a monitored database.

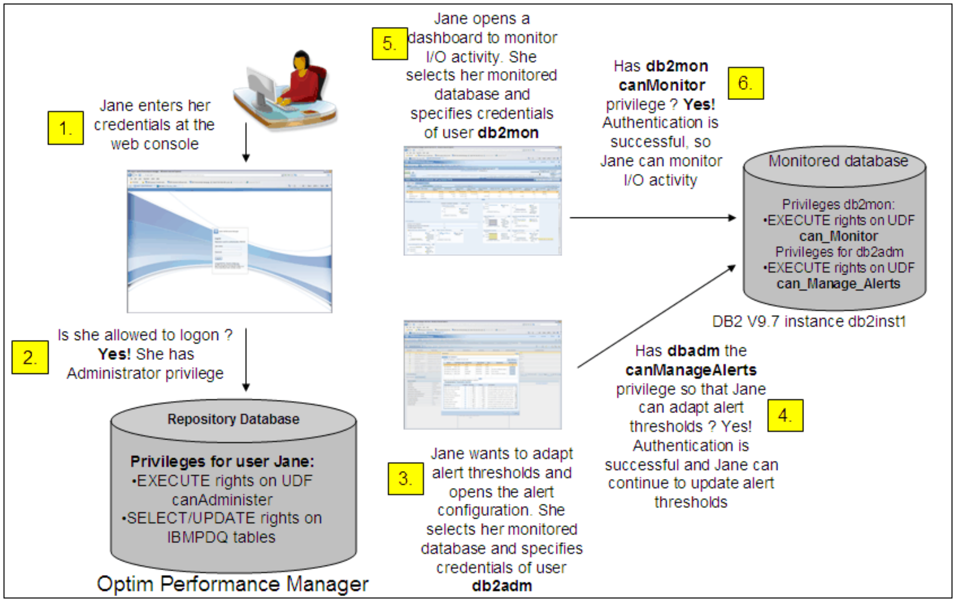


Figure 2-12 User authorization checking during monitoring

2.7 Objects in the monitored database

When monitoring is enabled for a database, Optim Performance Manager creates database objects in the monitored database including event monitors, event monitor tables, user-defined functions (UDFs), and stored procedures.

Depending on the monitoring data that you want to collect, Optim Performance Manager creates the following event monitors after the database is configured and enabled for monitoring:

- For Optim Performance Manager:
  - Lock event monitor (DB2 V9.7 or higher only)
  - Deadlock event monitor
  - Statistic event monitor (DB2 V9.5 or higher only)
- For Optim Performance Manager Extended Insight
  - Package cache event monitor (DB2 V9.7 or higher only)
  - Transaction event monitor (DB2 V9.7 or higher only)

These event monitors write the collected data into tables in the monitored database. Optim Performance Manager maintains these tables by deleting the data after Optim Performance Manager has read and saved it in the repository database. The interval that Optim Performance Manager saves the monitor data is short, for example every minute. By default, the tables are created in the default table space that DB2 chooses. Often it is USERSPACE1. When configuring the database for monitoring using the Optim Performance Manager Web console, you can specify the table space that Optim Performance Manager should use to create the event monitor tables. We recommend that you specify a dedicated table space for the event monitors that Optim Performance Manager creates. Note that for a partitioned system, this table space should exist on all partitions that Optim Performance Manager monitors.

The following event monitors are created only if you use Performance Expert Client and start SQL or DB2 Workload Manager (WLM) activity traces:

- Statement event monitor
- Activity event monitor

These event monitors generate lots of data and, therefore, write the collected data into files on the monitored system instead of into tables. You specify the path for these files during configuration of a monitored database if you enable the performance warehouse monitoring profile. If the monitored database is partitioned, this path must be available on each partition. The instance owner and fenced user must have read and write rights to this path. Optim Performance Manager deletes the files after it reads and stores the contents to the repository database.

The UDFs and stored procedures that Optim Performance Manager creates are used for the following purposes and do not generate overhead on the monitored database

- Control `canMonitor` and `canManageAlerts` privileges.
- Control and read the event monitor files resulting from SQL or WLM activity traces.
- Watchdog for event monitors to ensure that event monitors are dropped on the monitored database in case Optim Performance Manager looses connection to the monitored database.

Installing and configuring Optim Performance Manager

This chapter describes the steps necessary for a successful installation, activation, and configuration of Optim Performance Manager and the Extended Insight client.

3.1 Lab environment

In this section we describe the environment used for this Redbooks publication, shown in Figure 3-1. Other chapters and scenarios may reference these servers. The diagram here is a simplified, logical diagram, to give you an idea of which components are where, and how they interact.

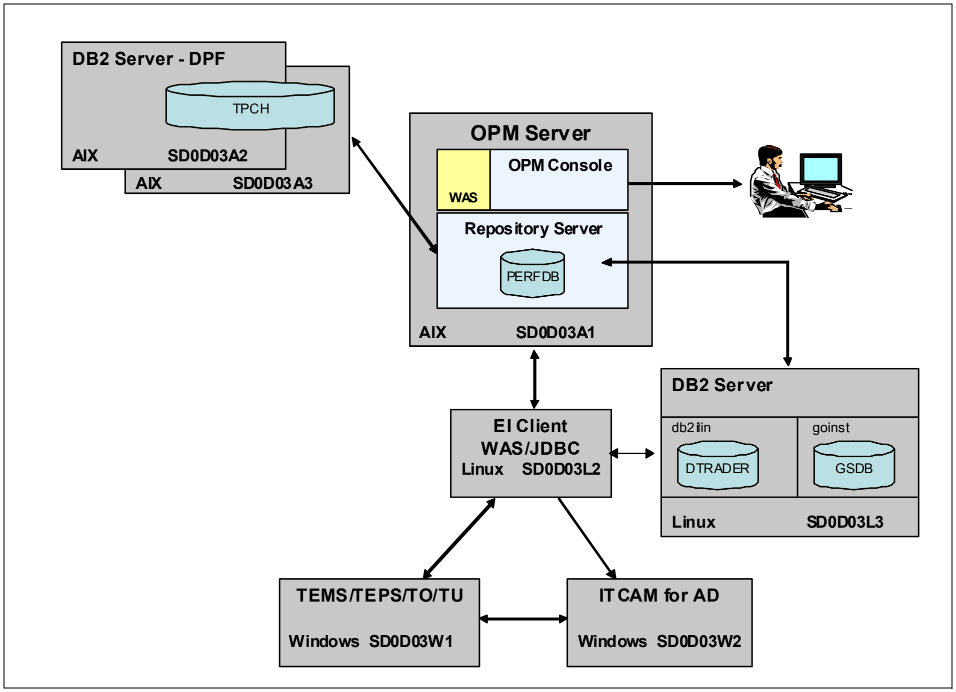


Figure 3-1 Simplified logical view of book lab layout

While this section about our Lab environment describes the complete list of software used in the book, this chapter discusses installation and configuration of the Optim Performance Manager (OPM) server components only.

Optim Performance Manager Server

This machine hosts the Optim Performance Manager server components:

- Optim Performance Manager repository server and its repository database
- Optim Performance Manager console server (WebSphere Application Server) that serves the web interface to the user

The details of this server are:

Machine	SD0D03A1
Operating system	AIX 6.1 TL6
CPU	4
RAM	6GB
Disk	500GB
Software installed	Optim Performance Manager 4.1.0.1 WebSphere Application Server V7.0.0.5 DB2 9.5 fp 6a
Instance	db2inst1
Port	50000
Database	PERFDB

Read about the installation and configuration of Optim Performance Manager Server components starting in section 3.2, “Installing and running Optim Performance Manager” on page 65.

Monitored DB2 server

We demonstrate the Optim Performance Manager monitoring on both single and multi partitioned database.

Single-partition database server

On this Linux server we set up two single-partition DB2 instances with several databases to be monitored by Optim Performance Manager. Installation of this DB2 server is not described in this book.

The system details are:

Machine	SD0D03L3
Operating system	Linux RHEL 5.5
CPU	2
RAM	8GB
Disk	200GB
Software installed	Optim Performance Manager 4.1.0.1 DB2 9.7 fp 2
Instance 1	db2ilin
Port	50001
Database	DTRADER
Instance 2	goinst
Port	50002
Database	GSDB

Muiltipartition database server

Optim Performance Manager has robust capabilities for monitoring mutipartition database instances. The mutipartition database instance used for the scenarios in this book, has four partitions defined across two machines. Machine1 hosts partition 0 and 1, and machine 2 has partitions 2 and 3. All four partitions can be considered data nodes.

Installation of these DB2 servers is not described in this book. The following are the system details:

- Machine #1

Machine	SD0D03A2
Operating system	AIX 6.1 TL6
CPU	4
RAM	6GB
Disk	400GB
Software installed	Optim Performance Manager 4.1.0.1 DB2 9.7 fp2
Instance	db2iaix
Port	50001
Application database	TPCH

- Machine #2

Machine	SD0D03A3
Operating system	AIX 6.1 TL6
CPU	4
RAM	6GB
Disk	400GB
Software installed	Optim Performance Manager 4.1.0.1 DB2 9.7 fp2
Instance	db2iaix
Port	50001
Application database	TPCH

Monitored Extended Insight client

Optim Performace Manager V4.1.0.1 Extended Insight feature now supports monitoring applications that use the DB2 Call Level Interface (CLI). JDBC applications continue to be supported.

You can read about Installing and configuring the Extended Insight (EI) client software in section 3.4, “Installing and Configuring Extended Insight Client” on page 129.

> **Note:** CLI support in Optim Performance Manager is for the DB2 Call Level Interface applications, not “Command Line Interface” which is sometimes also seen abbreviated as “CLI”.

WebSphere Application Server and JDBC applications

Optim Performance Manager Extended Insight feature supports monitoring WebSphere Application Server (WAS) applications that run JDBC transactions. Because WAS can be a complex environment by itself, we put this on its own server to separate it from any other EI client applications.

You can read about Installing and configuring the EI client software in section 3.4, “Installing and Configuring Extended Insight Client” on page 129.

We describe the Tivoli ITCAM components configuration required on this server is in “Enable at ITCAM WebSphere agent side” on page 340.

The detail of this system are as follows:

Machine	SD0D03L2
Operating system	Linux RHEL5.5
CPU	2
RAM	8GB
Disk	200GB
Software installed	WebSphere Application Server 7.0.0.11 DB2 Data Server Driver Package V9.7 fp2 Optim Performance Manager EI client 4.1.0.1 IBM pureQuery runtime V2.25.76 ITCAM Agent for WebSphere 7.1 fp1 ITM OS monitoring agent 6.2.2 fp2
Applications	DayTrader GOCompany

IBM Tivoli monitoring server

For the Tivoli monitoring scenarios, we have a very simple infrastructure defined, with the major IBM Tivoli monitoring components running on a single Windows server. This is not typical for production, but is perfectly suitable for our purposes in this book.

We have Tivoli Enterprise Monitoring Server (TEMS) and Tivoli Enterprise Portal Server (TEPS) running on one server, `SD0D03W1`. Also installed here are the IBM Tivoli Composite Application Manager (ITCAM) for Transactions' Transaction Collector and Transaction Reporter components, and the application support for the ITCAM for Application Diagnostics, and the Optim Performance Manager TEP workspace.

Description of the installation and configuration of the base Tivoli components is not covered in this book. We have installed the Tivoli components using mostly defaults and made very few configuration modifications.

More information about the Tivoli integration is found in Chapter 10, “Integration with Tivoli monitoring components” on page 323.

The system details are:

Machine	SD0D03W1
Operating system	Windows Server 2008 Enterprise SP2 64-bit
CPU	2
RAM	8GB
Disk	100GB
Software installed	ITM 6.2.2 fp2 - TEMS, TEPS ITCAM for Transactions 7.2 fp2 ITCAM for Application Diagnostics application support 7.1 fp1 Optim Performance Manager Extended Insight - Plug-in for Tivoli Enterprise Portal 4.1.0.1 DB2 9.5 fp6a
Instance	DB2
Port	50000
Database	TEPS WAREHOUS

ITCAM for Application Diagnostics Managing Server

The ITCAM for Application Diagnostics provides another user interface (UI) to view deep-dive information about WebSphere transactions. The UI is accessible from both the TEP and from a standalone browser. The Tivoli component that serves up the UI is called the ITCAM for AD Managing Server. We installed this component on a separate Windows server.

Description of the installation and configuration of the ITCAM for AD components is not covered in this book. We have installed the Tivoli components using mostly defaults and made very few configuration modifications.

More information about the Tivoli integration is found in Chapter 10, “Integration with Tivoli monitoring components” on page 323.

The system details are:

Machine	SD0D03W2
Operating system	Windows Server 2008 Enterprise SP2 64-bit
CPU	2
RAM	8GB
Disk	100GB
Software installed	ITCAM for Application Diagnostics Managing Server 7.1 fp1 WebSphere Application Server 7.0.0.0

	DB2 9.5 fp6a
Instance	DB2
Port	50000
Database	OCTIGATE

3.2 Installing and running Optim Performance Manager

Optim Performance Manager (OPM) provides diversified installation options for user. We discuss these options in 2.3, “Installation options” on page 25 to help you decide the option that suit your environment the most.

Different from an installation guide, this section aims to give you an example about how you can set up Optim Performance Manager and does not cover every piece of details. We briefly describe the major steps of fresh installation of Optim Performance Manager V4.1.0.1 with the default option in most of the steps. For more details on installation, refer to the Information Center at:

`http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.installconfig.doc/pm_install_modes.html`

Before installing Optim Performance Manager, read the installation requirement described in the guide. We also discuss the installation prerequisite in 2.2, “Prerequisites” on page 21.

This section covers the installation of the following components:

- Installing Optim Performance Manager repository on AIX
- Activating Optim Performance Manager license on AIX.
- Activating Optim Performance Manager Extended Insight license on AIX
- Installing Performance Expert Client on Windows XP

3.2.1 Installing Optim Performance Manager

This section describes the major steps to install Optim Performance Manager on AIX. Before installing them, you should have DB2 available on the machine because a repository database will be created in DB2 during the installation. You can either create a DB2 instance during the installation or use an existing DB2 instance. The installation demonstrated in this section uses an existing DB2 instance.

A WebSphere Application Server profile is required to run Optim Performance Manager console server. You can set up a WebSphere Application Server profile either before or during Optim Performance Manager installation process. This section only covers setting up a new WebSphere Application Server profile during Optim Performance Manager installation. Refer to the installation guide for the process of other options.

The installation image of Optim Performance Manager for AIX consists of a WebSphere Application Server installer and Optim Performance Manager installer files:

- WebSphere Application Server installer

  WebSphere Application Installer is contained in a folder named “Was”. During Optim Performance Manager installation process, if you choose to set up a new copy of WebSphere Application Server, the installer in this folder would be called automatically. If you choose to use an existing copy of WebSphere Application Server, this folder would not be used.

- Optim Performance Manager installer

  Optim Performance Manager installer contains the following files or folders:

  – `iehs311aix64.jar`
  – `OPM.server.v4.1.0.1.install-on-aix.bin`
  – `DSWeb`
  – `migration`
  – `OPM_sample_install_response_file.rsp`
  – `OPM.server.install.mode.trigger`

To install Optim Performance Manager on AIX in GUI mode, complete the following steps:

1. Launching installation wizard (Figure 3-2):

   Log on as root user and run

   `OPM.server.v4.1.0.1.install-on-aix.bin` to launch the installation wizard. Choose language.

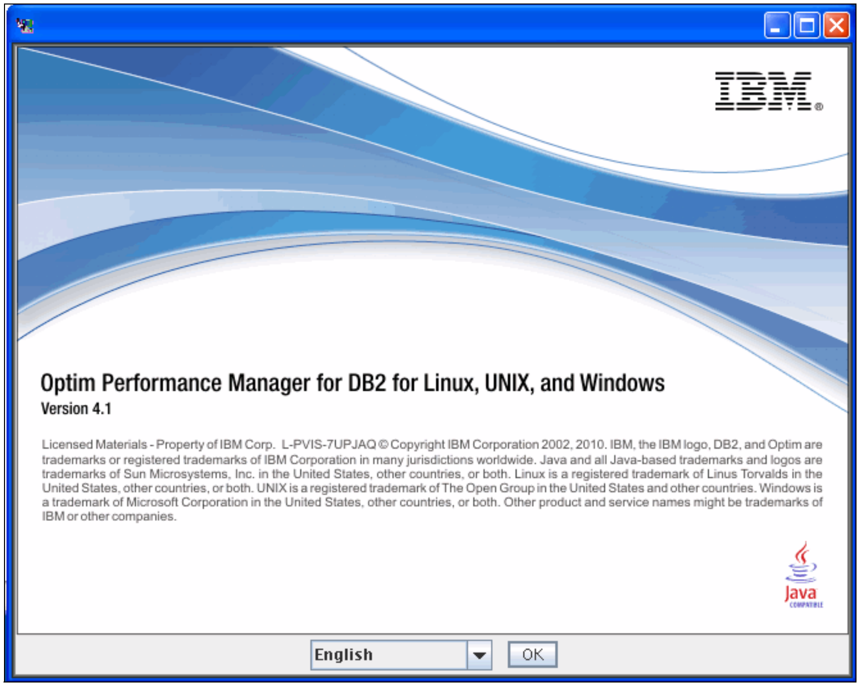


Figure 3-2 Launching installation wizard

2. Introduction:

   The first introduction panel recommends you to quit all programs before continuing the installation. The second panel (Figure 3-3 on page 68) shows the supported minimum level list of AIX. Make sure your Operating System meets the requirement.

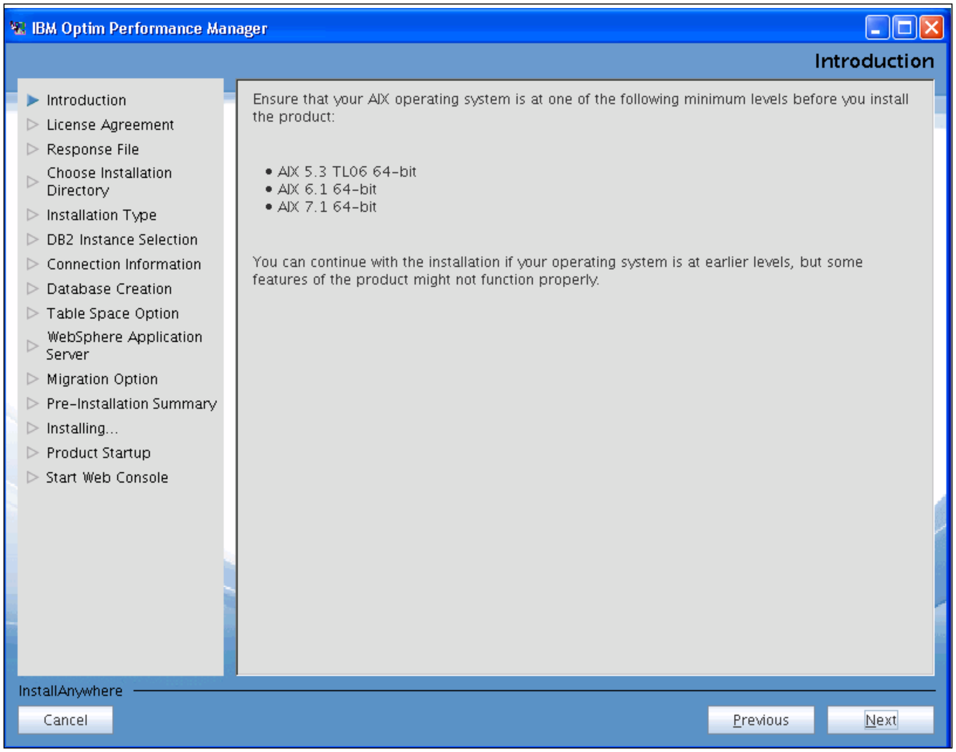


Figure 3-3 Supported AIX level

3. License Agreement (Figure 3-4 on page 69):

   You can choose to install either the Try and Buy edition or a licensed edition. The Try and Buy edition is good for 60 days with the option to activate the license later. The license file, `OPM.EnterpriseEdition.v4.1.0.1.opm_lic`, is in the Optim Performance Manager activation toolkit. You can place the license file anywhere on the machine where you run Optim Peroformance Manager installer, given that it can be read by the installation program. The default locate is in the same directory of Optim Performance Manager installer.

   In this section, we demonstrate installing a licensed edition. We discuss details of the Optim Performance Manager avtivation toolkit and show how to apply license for Try and Buy edition using the activation toolkit in 3.2.2, “Activating the Optim Performance Manager license” on page 80.

   We choose **Install a license edition** and specify the directory of the license file.

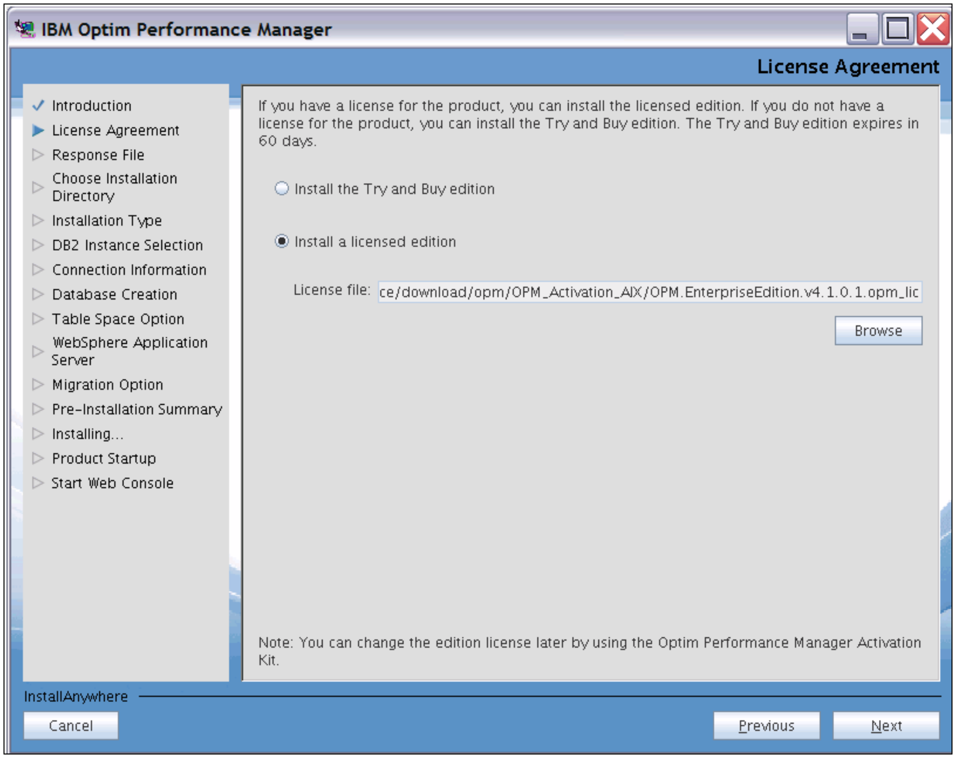


Figure 3-4 Choose an edition

4. License agreement: Read and accept the license agreement.
5. Response File (Figure 3-5 on page 70):

   The installation program of Optim Performance Manager can create a response file for silent installation. The response file contains your input of each step during the installation process. If you choose to create a response file, the file will be created when the installation completes in the directory specified in this step. We choose **Install the production on this machine**.

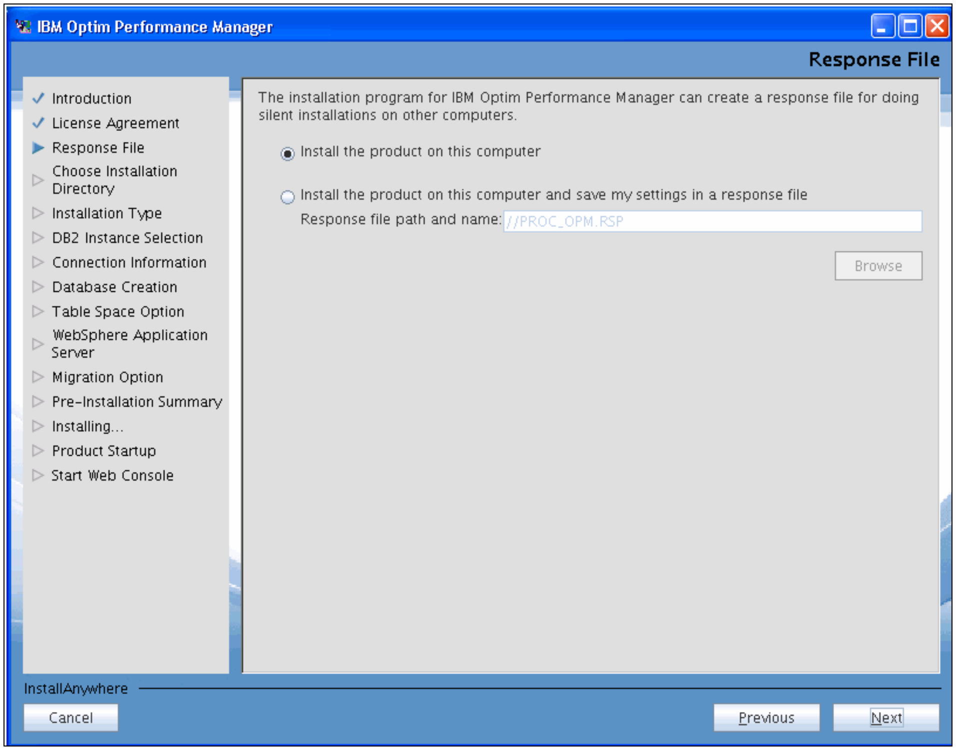


Figure 3-5 Choose whether you want to create a response file

6. Installation directory (Figure 3-6 on page 71):

   When choosing installation directory, you must make sure that there is enough space available for installation. We take the default AIX installation location `/opt/IBM/OPM`.

   Additionally, the installer requires space in the `/tmp` directory during installation. The user (root on UNIX) who installs Optim Performace Manager must have read, write, and execute on `/tmp` or `C:\temp`.

   Make sure that enough free space is available in `/tmp` as described in the System requirements:

   `http://www-01.ibm.com/support/docview.wss?rs=4014&uid=swg27019271`

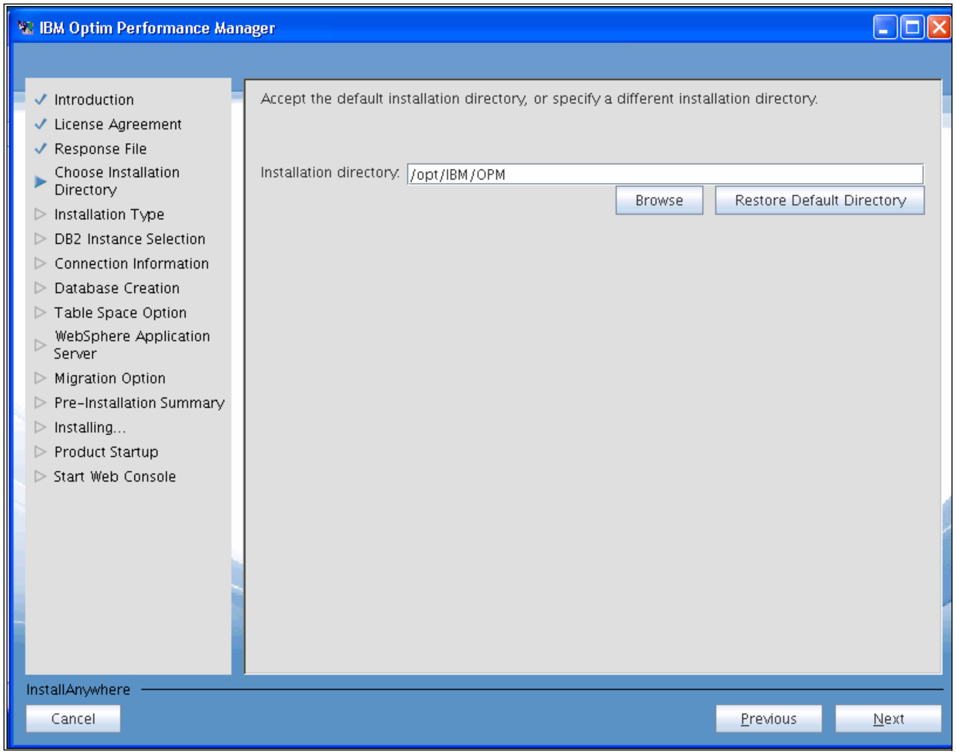


Figure 3-6 Installation directory

7. Installation Type (Figure 3-7 on page 72):

   The options are Typical Installation and Advanced Installation. If you do not want to use the default repository database specifications including name, table space type, and location, choose **Advanced installation** to change them. The table space type to use is one of the planning task described in 2.5, "Storage options" on page 49. You must select the Advance installation to specify the table space type. The default type is DMS. We choose **Typical installation**.

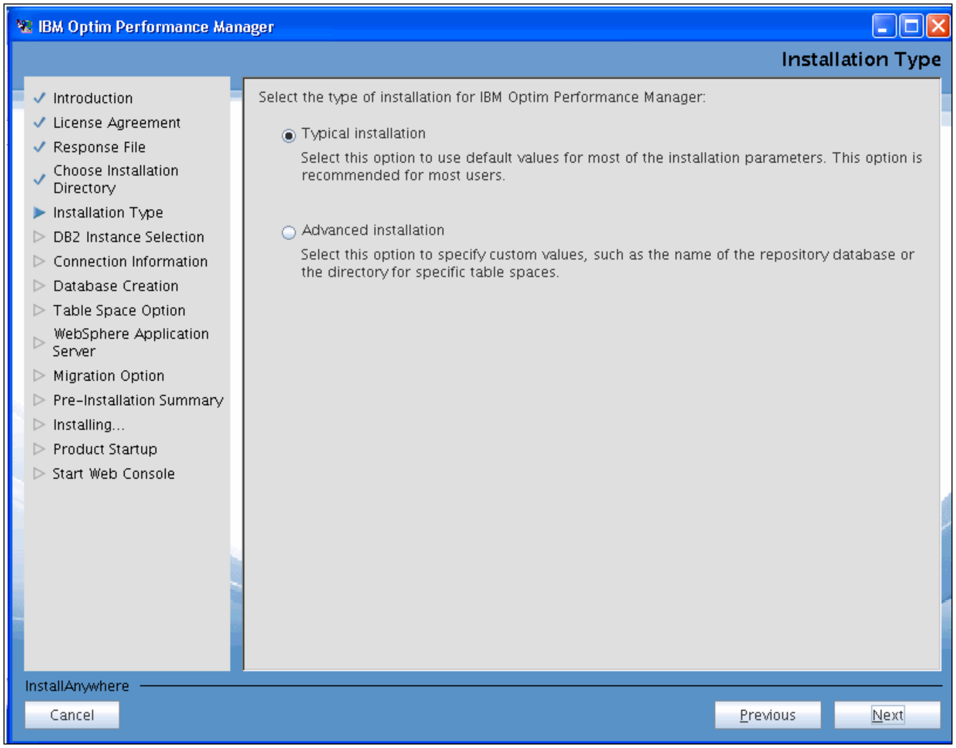


Figure 3-7 Installation Type

8. DB2 Instance Selection (Figure 3-8 on page 73):

   If you have a DB2 instance available for Optim Performance Manager, use the **Select an existing DB2 instance** option to specify the instance. To have the installer create a DB2 instance for you, choose **Create a new DB2 instance** and specify the instance user. In this process, we use an existing DB2 instance.

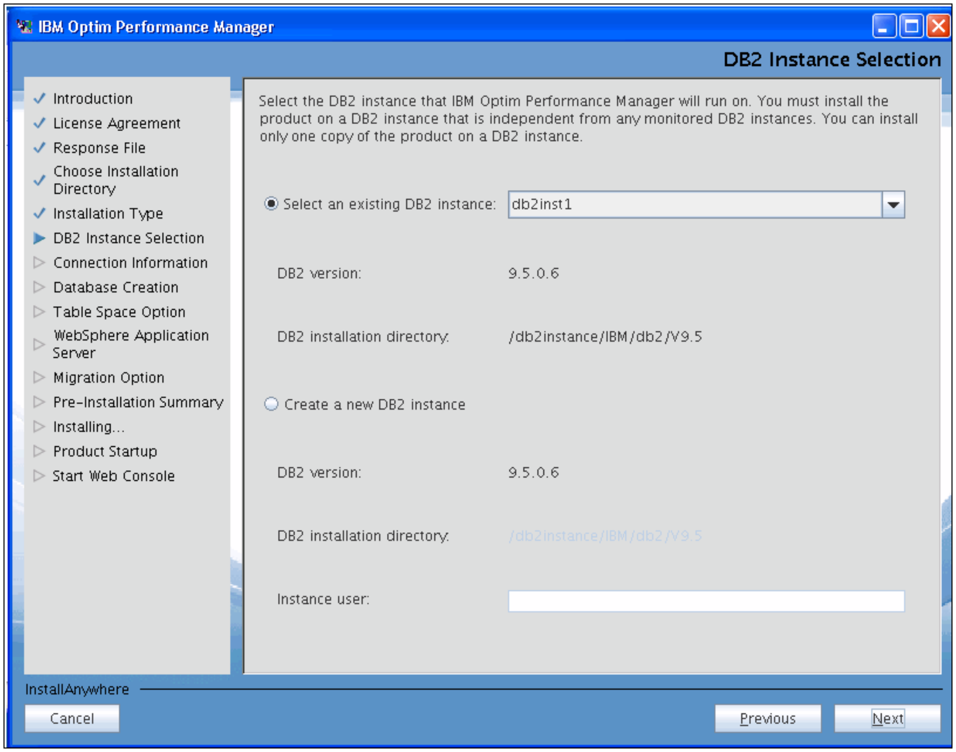


Figure 3-8 Select DB2 instance

9. Connection Information (Figure 3-9 on page 74):

   Input a DB2 instance user name and a password. Optim Performance Manager uses this user name and password to create and access the repository database. This user must have the `SYSADM` authority on the DB2 instance. This user can run the WebSphere Application Server profile on which Optim Performance Manager console serverruns and it can log on to the Optim Performance Manager console.

   Input a group name. The group can be any operating system group. Optim Performance Manager grants the appropriate privileges to this group so the users of this group can log on from Performance Expert Client to the Optim Performance Manager repository server. To learn more about this group, refer to “Performance Expert client group specification” on page 28. The repository database is created on the specified DB2 instance.

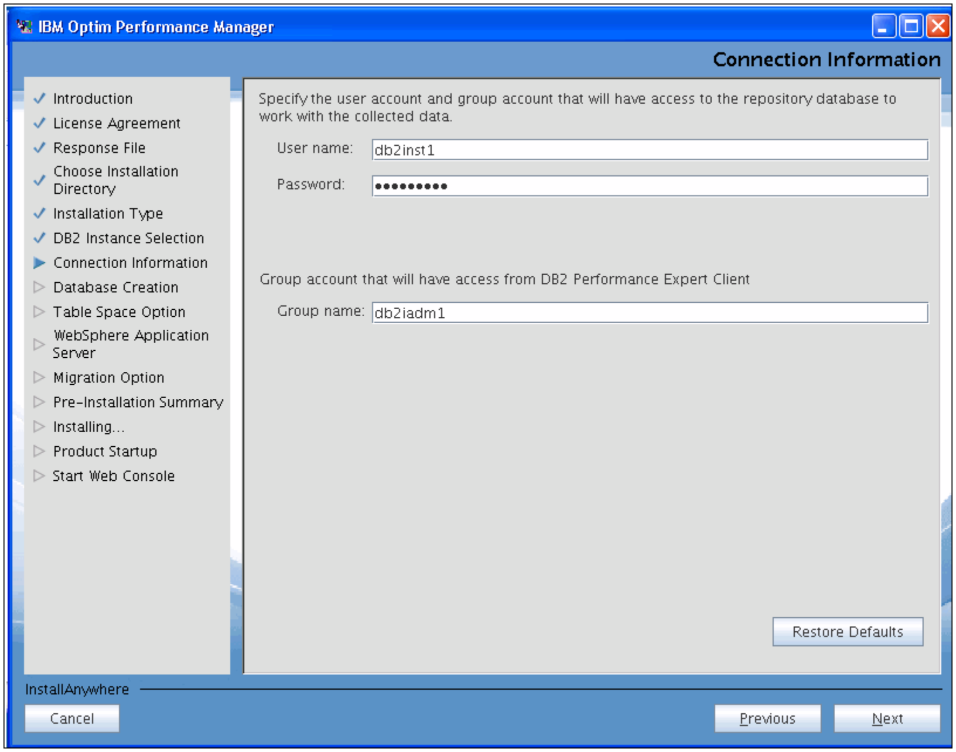


Figure 3-9 DB2 connection information

10. WebSphere Application Server (Figure 3-10 on page 75):

WebSphere Application Server is required for running Optim Performance Manager console server. Optim Performance Manager installer detects available WebSphere Application Server copies on the machine and list them in the *Use an existing copy of WebSphere Application Server* field for you to select the one to be used.

If you elect to create a new copy by choosing **Install a new copy of WebSphere Application Server**, which we recommend, Optim Performance Manager installer installs the WebSphere Application Server included in the Optim Performance Manager installation image.

To show how the Optim Performance Manager installer installs WebSphere Application Server, we choose **Install a new copy of WebSphere Application Server**” and input the installation directory. Refer to the disk space requirement described in the system requirements to make sure that there is enough space available in the specified directory:

http://www-01.ibm.com/support/docview.wss?rs=4014&uid=swg27019271

On AIX, by default, the new copy of WebSphere Application Server is set up in `/opt/IBM/`.

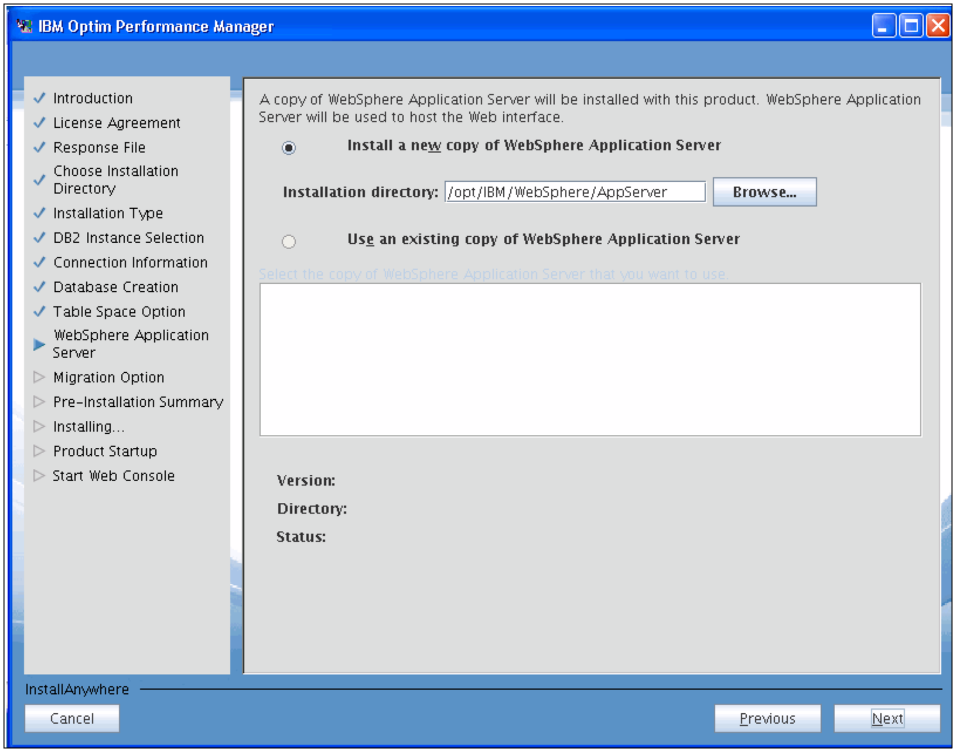


Figure 3-10 WebSphere Application Server

11.WebSphere Application Server profile (Figure 3-11 on page 76):

A WebSphere Application Server profile defines the runtime environment. The profile includes all of the files that the server processes in the runtime environment and that you can change. To learn more about WebSphere Application Server profile, refer to the information center:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/cpro_overview.html`

A WebSphere Application Server profile is required for running Optim Performance Manager console server. If you chose an existing WebSphere Application Server in last step and there are profiles on the server, in this step, you can either create a new profile or use an existing one. Optim Performance Manager installer detects available profiles on the specified WebSphere Application Server and list under the *Use an existing profile for WebSphere Application Server* field.

If you chose to create a new copy of WebSphere Application Server in last step, you can only choose to create a new profile and specify a profile name. Since we chose to create a new WebSphere Application Server in last step, we choose **Create a new profile for WebSphere Application server** and take the default profile name.

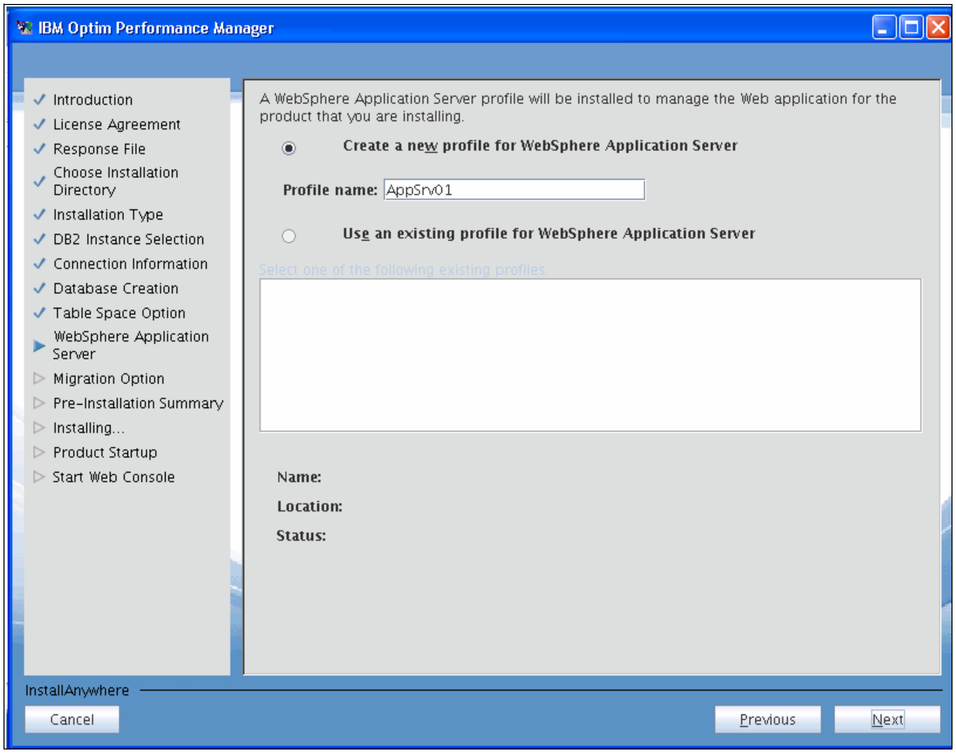


Figure 3-11 WebSphere Application Server profile

12. WebSphere Application Server Global Security (Figure 3-12 on page 77):

In this panel, specify an administrative user to be used to log into the WebSphere Application Server Administrative Console of the profile (`http://hostname:portnumber/ibm/console`) if the global security of the profile is enabled.

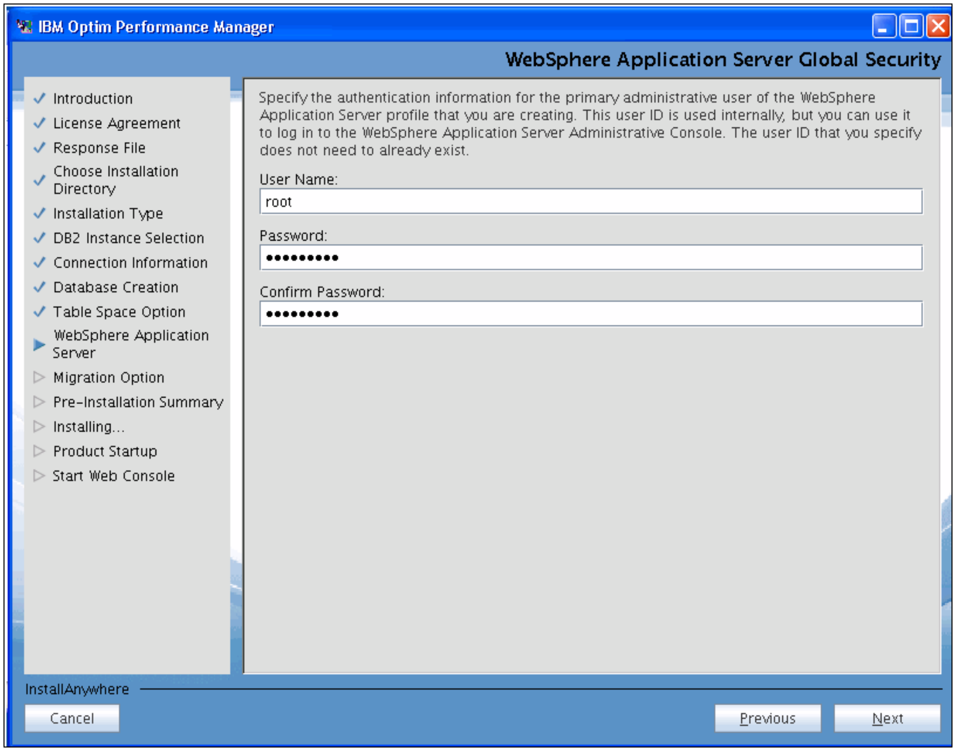


Figure 3-12 WebSphere Application Server profile administrative user

13.Pre-Installation Summary:

Thist panel shows a summary of your installation specifications for your review. Once proceed, a small window pops up informing that the time spend for installation depends on the machine capability.

14.Product Startup (Figure 3-13 on page 78):

When the wizard reaches to this step, the installation has already finished. On the first panel, select whether you want to start IBM Optim Performance Manager when the computer starts. To learn how to start Optim Performance Manger by using the command line, refer to 3.2.5, “Starting and Stopping Optim Performance Manager” on page 89.

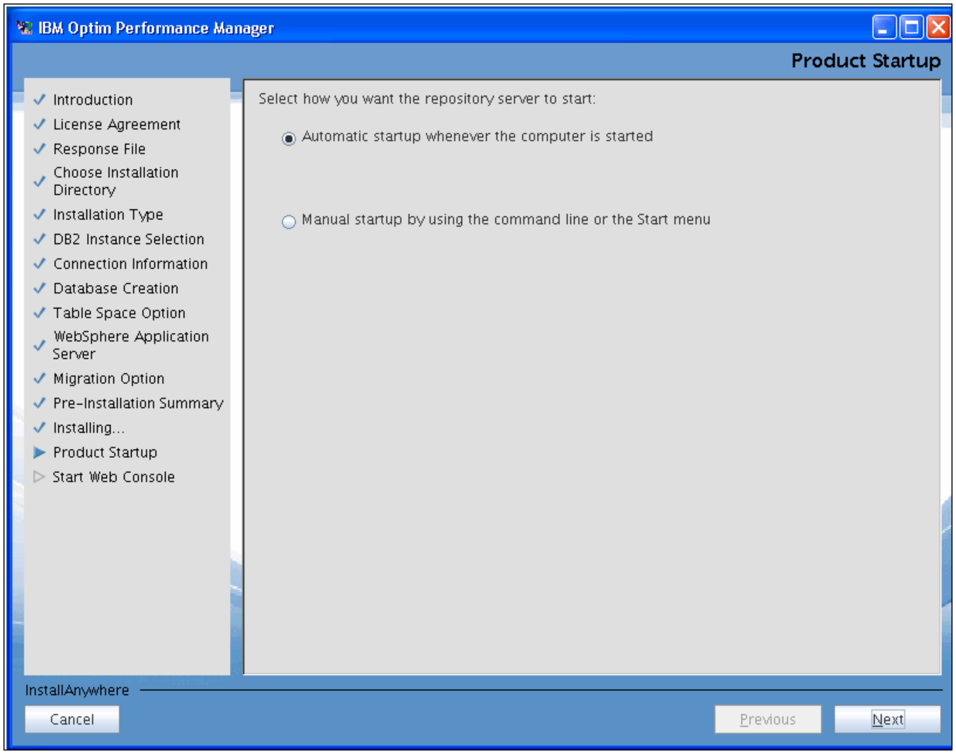


Figure 3-13 Product Startup - option for starting the repository server

On the second panel of Product Startup (Figure 3-14 on page 79), choose whether to start IBM Optim Performance Manager and the associated WebSphere Application Server profile right now. If you select to start the product now, the repository server and the associated WebSphere Application Server profile will be started. If the associated WebSphere Application Server is running, it will be re-started.

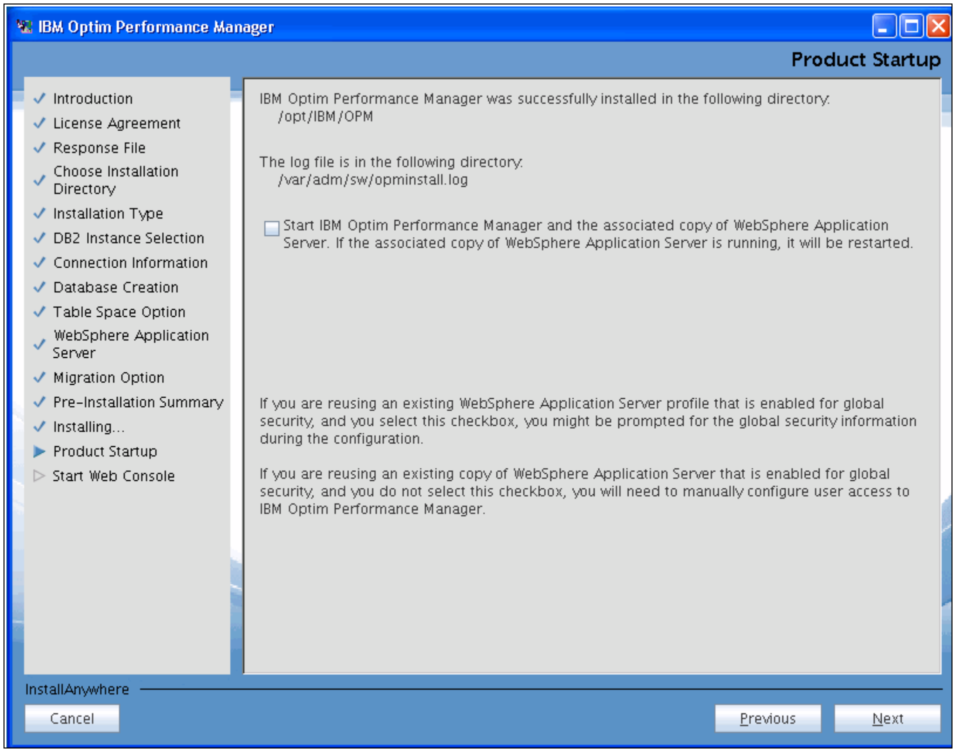


Figure 3-14 Product Startup - option for starting the products now

15. Start Web Console (Figure 3-15 on page 80):

    If you chose to start Optim Performance Manager in last step and it is started successfully, you can select to open the Optim Performance Manager web console now or later. Take a note of the two URLs presented in the panel that are the Optim Performance Manger web console address. If the global security of the associated WebSphere Application Server profile is enabled, you can use either of the URLs. When you use the http URL, you will be automatically redirected to the https URL. If the global security is disabled, you will use the http URL.

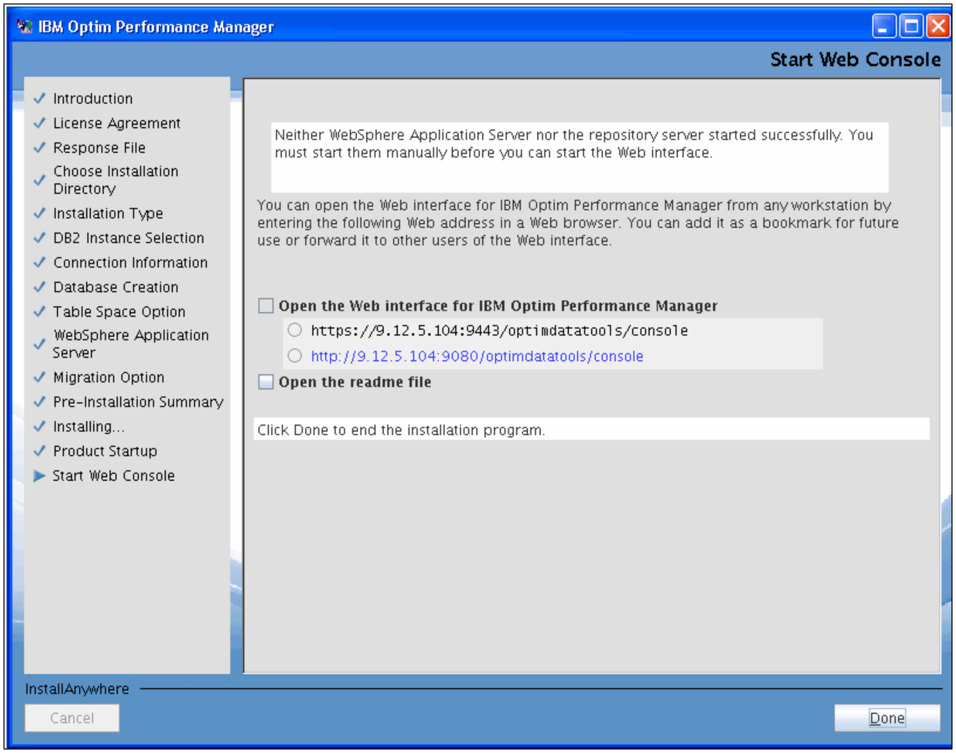


Figure 3-15 Start Web Console

The installation log is in `/var/adm/sw/opminstall.log`.

3.2.2 Activating the Optim Performance Manager license

IBM Optim Performance Manager is a licensed product. IBM provides you the 60-days no charge Try and Buy version to experience the product features and functions. The activation toolkit is for applying the license to the Try and Buy version to continue use the product after 60 days.

The Optim Performance Manager Extended Insight is a separated licensed option of Optim Performance Manager. The Extended Insight server component is installed together with the Optim Manager installation but not activated. To activate the Extended Insight, you can use the Extended Insight activation toolkit to apply the lances.

Optim Performance Manager activation toolkit contains the license of Optim Performance Manager. Having the license, user can apply it during installation or activate license for the Try and Buy edition.

In this section, we discuss how to activate the Optim Performance Manager license using the activation toolkit on AIX.

The Optim Performance Manager activation toolkit for AIX contains the following files.

- `OPM.server.v4.1.0.1.activate-on-aix.bin`
- `OPM.EnterpriseEdition.v4.1.0.1.opm_lic`

Use these steps to apply the license:

1. As root run `OPM.server.v4.1.0.1.activate-on-aix.bin`
2. Choose installation language.
3. Accept the license agreement.
4. Specify the Optim Performance Manager installation directory.

   You may have several copies of Optim Performance Manager installed on the same machine. Choose the one to which you want to apply the license.
5. Review the pre-installation summary.
6. Click **Done** when the installation is finished.

3.2.3 Activating the Optim Performance Manager Extended Insight license

The Extended Insight contains a server component and a client component. The server component is contained in Optim Performance Manager installation package and is installed with Optim Performance Manager in the in Optim Performance Manager installation directory. For example, on AIX, if you install Optim Performance Manager in `/opt/IBM/OPM`, you can see the `/opt/IBM/OPM/pureQuery` subdirectory which contains the major part of Extended Insight server. The property file of the Extended Insight Server named `pdq.properties` is located in `<working directory>/<db2 instance>`. `pdq.properties` is used in configuring the Extended Insight monitoring. `<working directory>` is the working directory of the repository server and `<db2 instance>` is the name of the DB2 instance on which the repository server runs.

> **Tip:** On AIX, the default working directory of Optim Performance Manager repository server is `<user home dir>/opm/v4`, where `<user home dir>` is the home directory of the DB2 instance owner.
>
> You can find out the working directory by running the following command:
>
> `grep <instance name> /var/db2pe/v3/db2pesrv.cfg | grep homedir`
>
> `<instance name>` is the name of the DB2 instance on which the repository server runs. The result shows a parameter `<instance name>.db2pe_homedir` and its value. The value is `<working directory>`.
>
> On Windows, the default working directory is `<OPM install dir>\RepositoryServer`

To use Extended Insight, you must activate the server by applying a license to the Extended Insight server using the Extended Insight activation toolkit.

The Extended Insight activation toolkit for AIX contains the following files:

- `activate_EI.bin`

  Run this file to launch the installation wizard.

- `optim.pm.extendedinsight.pek_4.1.0.1.jar`

  This jar file contains the license for activating the Extended Insight feature.

- `parms_Activate`

  This file contains parameters used by the toolkit. You do not need to do anything with this file.

- `sample_activation.rsp`

  This is a response file that can be used to contain all user input during the installation process so that it can be used later for silent installation on other systems. Using this file is optional.

Use the following step to activate the Extended Insight license on AIX:

1. As root run `activate_EI.bin`.
2. Choose installation language.
3. Select **Activate the Extended Insight license** and **Configure Communication Properties**.

   If you select Configure Communication Properties in this step, you will be directed to step 6 after step 4 and 5. Otherwise, the activation process finishes at step 5.

4. Specify the Optim Performance Manager installation of which the Extended Insight is to be activated.

   The installer detects and lists the valid copies of Optim Performance Manager installed on the system. Choose the copy to which you want to activate Extended Insight feature. If the copy is not in the list, click **Browse** to specify the installation directory.

5. Read and accept license agreement.

6. Configure communication properties. If you selected *Configure Communication Properties* in step 2, you will continue to this step for specifying the host name of Optim Performance Manager and a port number which would be used for communication between Optim Performance Manager and Extended Insight. The installer detects the valid host name of Optim Performance Manager and take it as default value.

   The port number should be an unused on. The default port number is 60000. The value of `pdq.cmx.controllerURL` in `<working directory>/<db2 instance>/pdq.properties` will be updated with the host name and port number you specify in this step:

   ```
   pdq.cmx.controllerURL=<host name>:<port number>
   ```

   `<host name>` is the host name you specified in this step and `<port number>` is the port number you specified.

   You can check and update this property value accordingly later if necessary.

7. Specify the directory of `optim.pm.extendedinsight.pek_4.1.0.1.jar` to finish the installation.

The installation of client component is described in 3.4, “Installing and Configuring Extended Insight Client” on page 129.

3.2.4 Installing DB2 Performance Expert Client

DB2 Performance Expert Client reads the data in the repository database and presents the monitoring information to users. The Optim Performance Manager users can either use the web console clients, the DB2 Performance Expert Client user, or both. Before starting installation, make sure that the prerequisite described in the following web address is met:

http://www-01.ibm.com/support/docview.wss?uid=swg27016380

This section describes the major steps for installing the Performance Expert Client on Windows XP. For all installation options, refer to 2.3, “Installation options” on page 25.

The installation package of Performance Expert Client contains of the following files:

- `db2pe.client.v4.1.0.1.install-on-win32.exe`
- `iehs311win.jar`
- `README.txt`

To install Performance Expert Client on Windows XP, log on with administrator authority and perform the following steps:

1. Run `db2pe.client.v4.1.0.1.install-on-win32.exe` to launch the installer. See Figure 3-16.

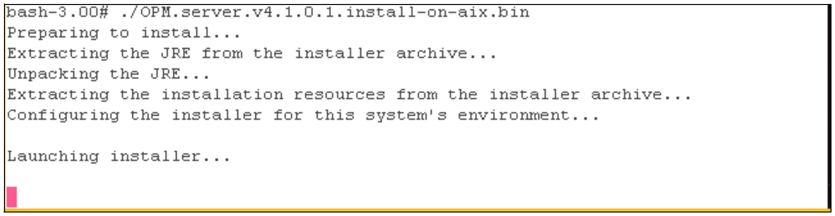


Figure 3-16 Installation introduction

   The Introduction panel provides a simple installation instruction.

2. Software License Agreement: Read and accept the license agreement.
3. Client edition (Figure 3-17 on page 85): Choose the client edition you have purchased. The Performance Expert for Multiplatform edition supports DB2 for Linux, UNIX, and Windows as well as DB2 for z/OS.

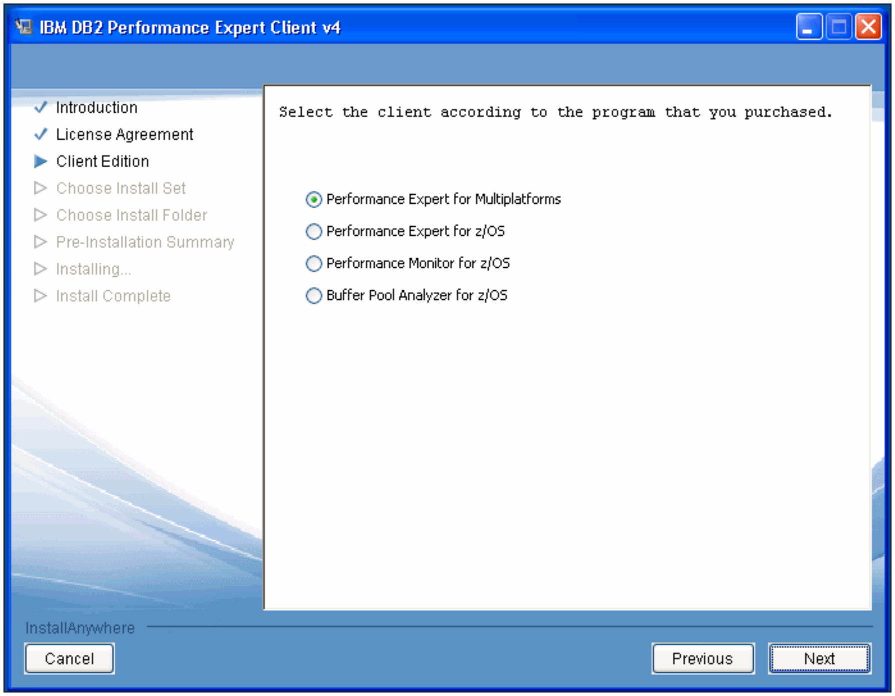


Figure 3-17 Select client edition

4. Choose Install set (Figure 3-18 on page 86): We recommend the *Typical* installation for most of users. Choose **Custom** if you do not want to install “Help” for all supported language and prefer to select particular ones.

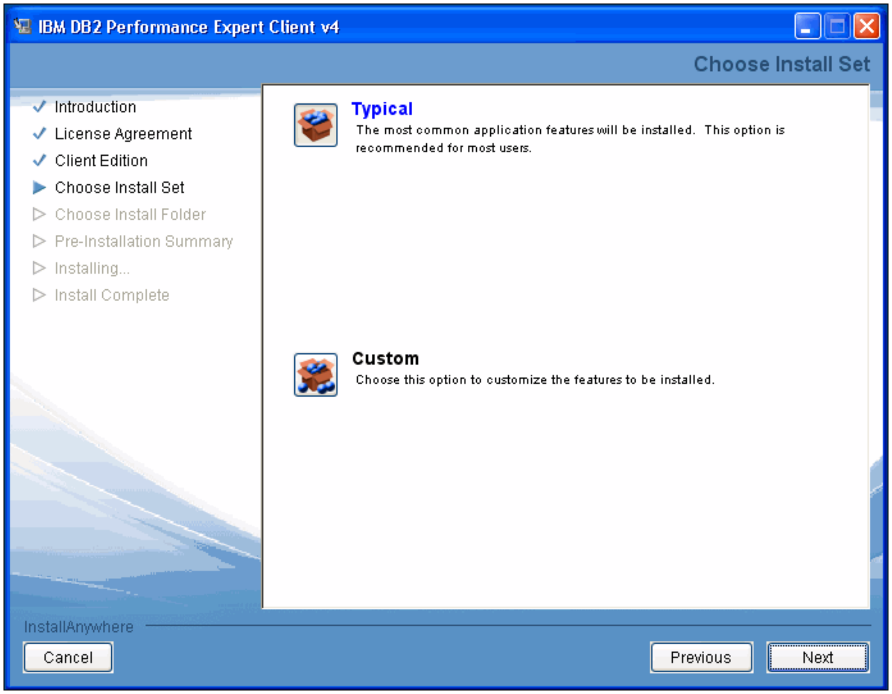


Figure 3-18 Choose install set

5. Installation folder (Figure 3-19 on page 87): Specify the installation directory. Make sure there is enough space available in the specified directory.

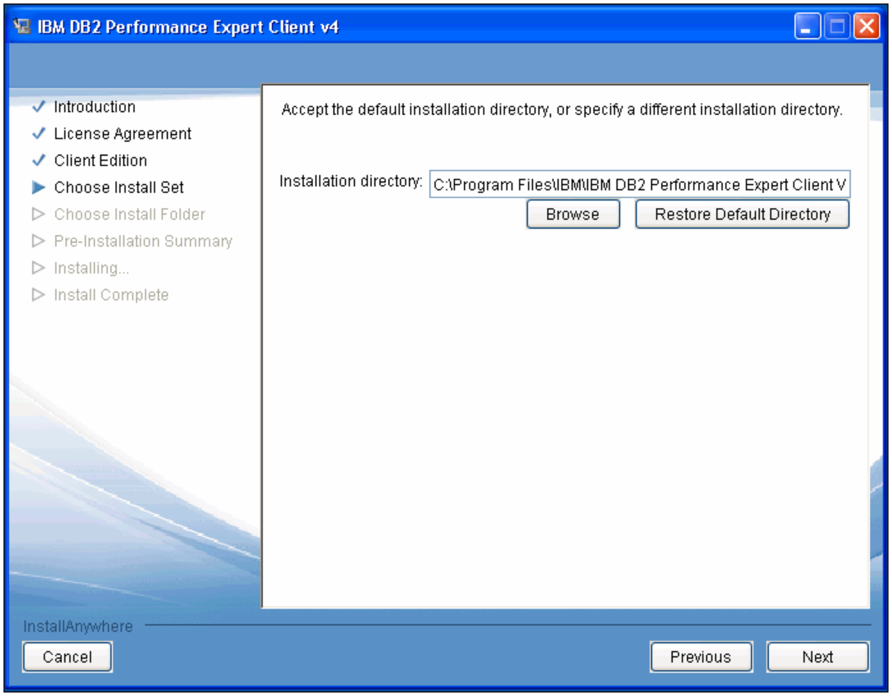


Figure 3-19 Choose install folder

6. Pre-Installation Summary (Figure 3-20 on page 88): Review the information.

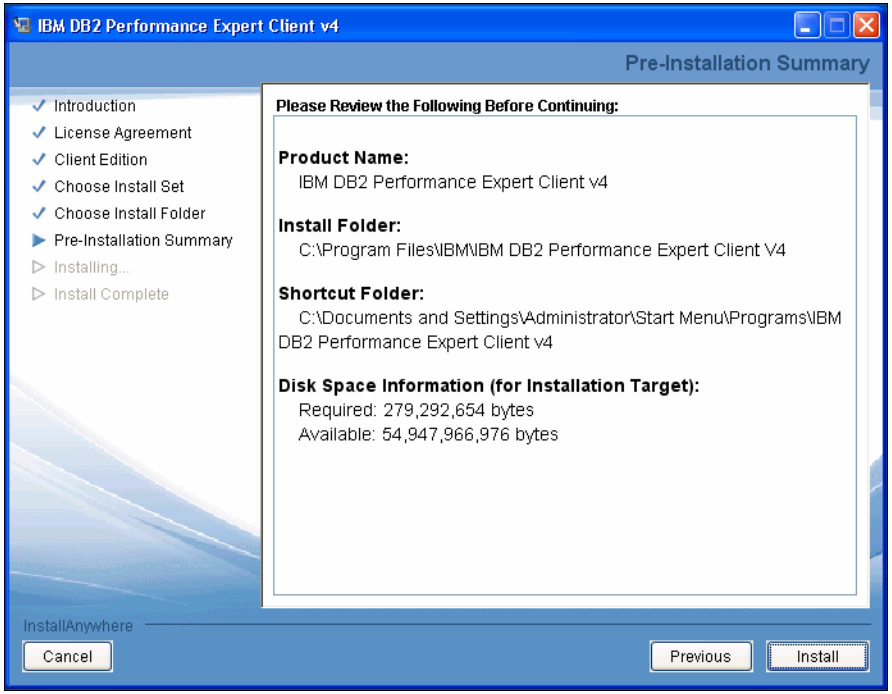


Figure 3-20 Pre-installation summary

7. Install Complete (Figure 3-21 on page 89): Click **Done** when installation finishes.

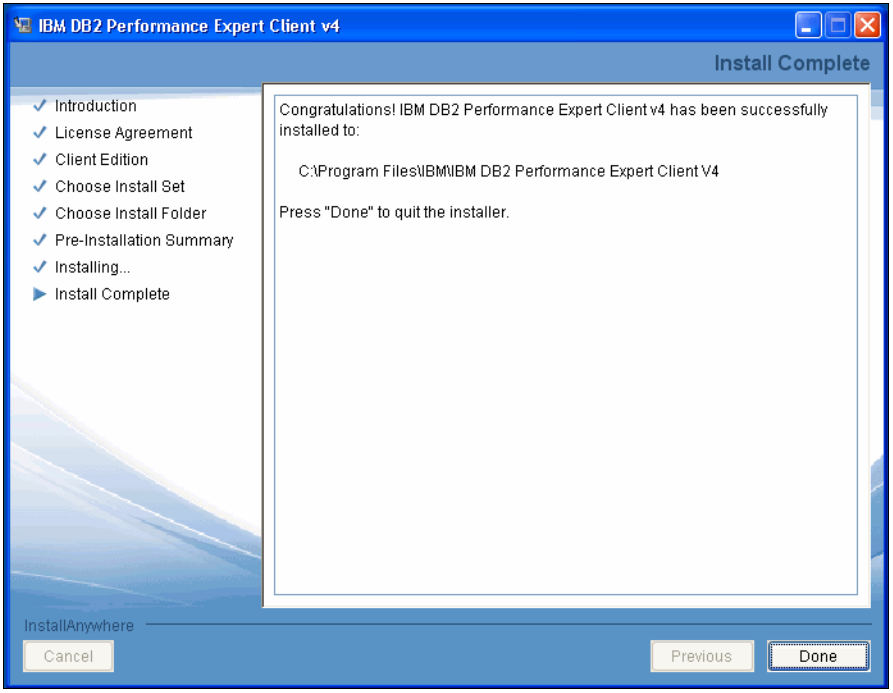


Figure 3-21 Install complete

8. When installation is done, the “Getting Started” window pops up. Click open and then add connections to monitor. Alternatively, you can open this window from the start menu.

3.2.5 Starting and Stopping Optim Performance Manager

This section describes the procedures to start and stop the Optim Performance Manager on AIX including the Optim Performance Manager repository server and the Optim Performance Manager console.

Before starting Optim Performance Manager, the DB2 instance on which the Optim Performance Manager will run should be started.

To start (or stop) the repository server on AIX, finish the following steps:

1. Log on as the DB2 instance owner ID.
2. Go to directory: `<OPM installation directory>/RepositloryServer/bin`, where `<OPM installation directory>` is the installation directory of Optim Performance Manager. On AIX, the default path is `/opt/IBM/OPM`.

3. run `./pestart` to start or run `./pestop` to stop the repository server.

   The starting, stopping, and running logs of the repository server is contained in `<working directory>/<db2 instance>/db2pesrv.log`. Typically, the monitoring status for each monitored data server is contained in the log.

To start (or stop) Optim Performance Manager console on AIX, start (or stop) the associated WebSphere Application Server profile where the Optim Performance Manager console runs. An alternative method is by performing the following steps:

1. Log on as the DB2 instance owner ID or other ID with root authority, dependent on whether the WAS profile runs under DB2 instance owner or root user. If you have installed WAS together with OPM then the WAS profile runs under the DB2 instance owner.
2. Go to the `<OPM installation directory>/bin` directory.
3. run `/WASstart.sh` to start or `/WASstop.sh` to stop the console.

> **Tip:** One of the new functional features of Optim Performance Manager V4.1.0.1 is the ability to run the WebSphere Application Server profile used by Optim Performance Manager as the DB2 instance owner instead of root on AIX or Linux.
>
> On windows, the WebSphere Application Server profile is always running under the SYSTEM account.

The starting, stopping, and running logs of the Optim Performance Manager console is contained in the WebSphere Application Server profile log files under `<WebSphere Application Server installation directory>/AppServer/profiles/<profile name>/logs/server1/`:

- `startServer.log` contains the log information about starting Optim Performance Manager console
- `stopServer.log` contains the log information about stopping Optim Performance Manager console
- `SystemOut.log`, `SystemErr.log`, `native_stdout.log`, and `native_stderr.log` contains the running logs.

Detailed-level log about Optim Performance Manager console access Repository Database is contained in `<OPM installation directory>/logs/datatools.log`, where `<OPM installation directory>` is the installation directory of Optim Performance Manager.

3.3 Configuring Optim Performance Manager

After installation you must configure Optim Performance Manager which comprises of the following steps:

1. Configuring user access

   After installation the DB2 user that you specified during installation can logon to the Optim Performance Manager web console. If other users should have access then you have to give them the privileges. This is an optional step. It can also be done after the next two steps.

2. Adding or importing database connections

   This step defines the databases that you want to monitor with Optim Performance Manager.

3. Configuring the database connections for monitoring

   In this step you configure monitoring using monitoring profiles, define monitoring authorizations, and configure partition sets. Partition sets can be configured only for partitioned database.

> **Note:** In the previous DB2 Performance Expert product you used the **`peconfig`** program to perform steps 2) and 3). Optim Performance Manager still supports the configuration via **`peconfig`** in addition or alternatively to the configuration via web console that we describe in this chapter. Refer to A.10, "Using the configuration program peconfig" on page 464 for more information about using **`peconfig.`**

You can login to the Optim Performance Manager console using the URL that is provided during installation. As an example:

`http://9.12.5.104:9080/optimdatatools/console`

Enter authentication details to access Optim Performance Manager. See Figure 3-22 on page 92. Note that immediately after the installation, only the DB2 user that was provided during the installation can be used if you use the repository database authentication. For more details refer to 2.6, "User authorization" on page 52 that introduces you to the security and authentication concept of Optim Performance Manager.

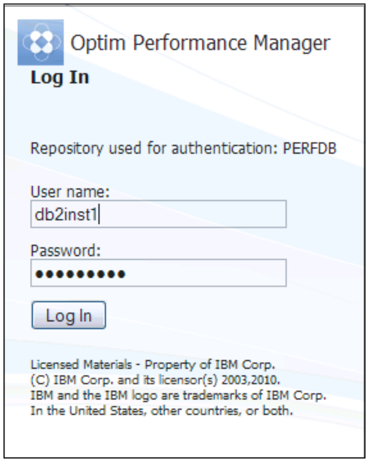


Figure 3-22 Optim Performance Manager Login screen

Once you login you will be at the Optim Performance Manager Welcome screen (Figure 3-23).

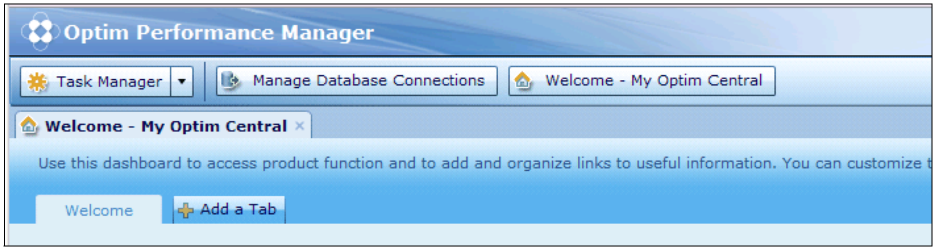


Figure 3-23 Optim Performance Manager Welcome screen

3.3.1 Configure user access

One part of Optim Performance Manager configuration is to manage which users can authenticate to the Optim Performance Manager web console and what privileges they have once they are logged in.

Let us start by reviewing the security method for authenticating users of the web console. The privileges that are assigned to a user control the actions that a user can perform in the web console. We can do this by opening the Console Security tab under Task Manager on the web console (Figure 3-24 on page 93). You must have administrator privileges to use this page.

Figure 3-24 Task Manager - Console Security

After a fresh installation of Optim Performance Manager 4.1.0.1, the authentication method is set to **Repository database authentication** and you can manage user access from the same page (Figure 3-25 on page 94). The user, group, or role that you grant these privileges to must already be defined in the repository database and have the CONNECT privileges on the repository database. If you want to manage user authentication through WebSphere Application Server, select **Web container-managed authentication** and define the user access within WebSphere Application Server Administrative console. For more information about user authentication, refer to the Information Center at

http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.installconfig.doc/pm_configure_user_access_to_opm.html

Tip: If you want to configure user access for users authenticated through LDAP, we recommend to use *Repository database authentication*. This requires that the DB2 instance used for Optim Performance Manager is configured to use LDAP-based authentication through the LDAP security plug-in or using transparent LDAP. If you want to use *Web container-managed authentication* for LDAP users, you must configure WebSphere Application Server to use LDAP through the WebSphere Application Server administrative console. For more information about how to do that, refer to:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_ldap.html

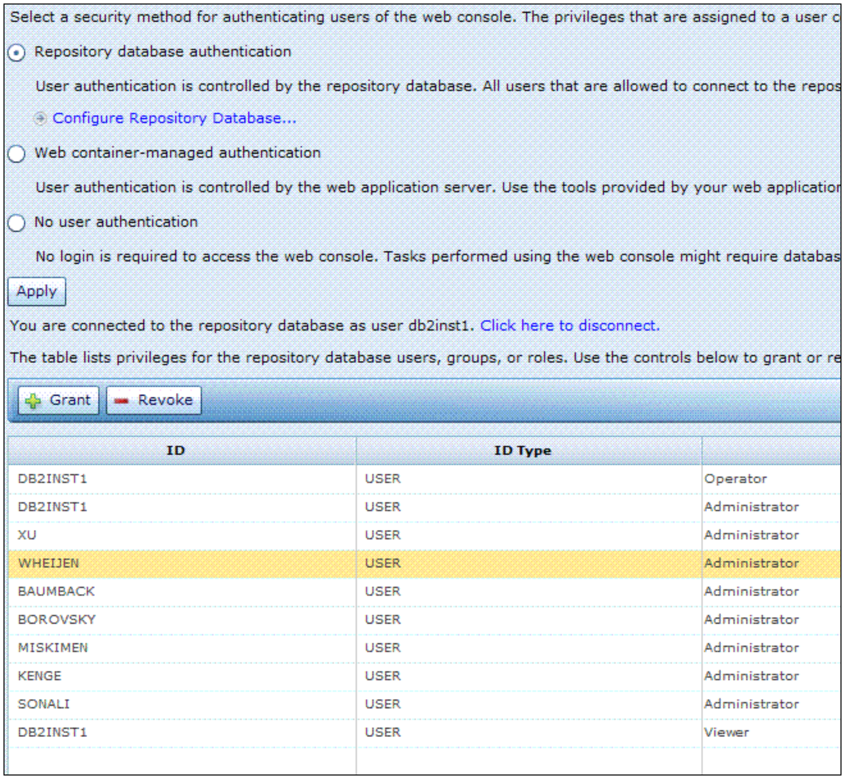


Figure 3-25 Console Security Dashboard

To each user that requires access to the Optim Performance Manager web console, you can grant one of the following user roles:

- Viewer: The Viewer role is the default global privilege for every Optim Performance Manager web console user. A user who is assigned the Viewer role cannot change any global settings. Viewers cannot see the historical monitoring information of any monitored databases that are disconnected.
- Operator: The Operator role in Optim Performance Manager is exactly the same as the Viewer role.
- Administrator: The Administrator role is a global privilege that allows the user to perform any task in the Optim Performance Manager web console. Administrators can also view historical monitoring information of all disconnected databases.

For more details refer to Section 2.6, “User authorization” on page 52 that introduces you to the security and authentication concept of Optim Performance Manager.

3.3.2 Adding or importing database connections

You can add new database connections one by one, or you can import a set of database connections into the database repository.

Adding a database connection using the Connection Wizard

To add a database for monitoring you must have the database authorization and connection information.

Perform these steps to add a database connection:

1. Log on to Optim Performance Manager.
2. From the Optim Performance Manager window, click **Manage Database Connections**.
3. Click **Add** and you will be required to authenticated the Optim Performance Manager repository database (Figure 3-26). The user who connects to the repository database in order to add new connections must have certain privileges to the repository database. For more details about authorization, refer to 2.6, "User authorization" on page 52 that introduces you to the security and authentication concept of Optim Performance Manager.

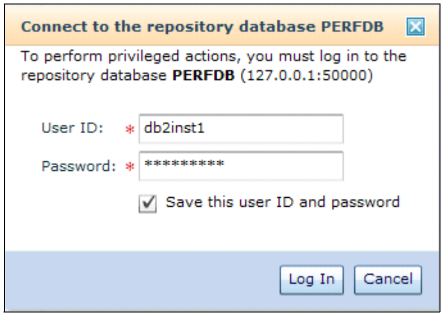


Figure 3-26 Authenticate the repository database

4. The database connection wizard opens up.

   Enter Database connection name, Data server type, Database name, Host name, Port number, User ID and Password. Fields that do not have a red asterisk next to them are optional fields. See Figure 3-27 on page 96.

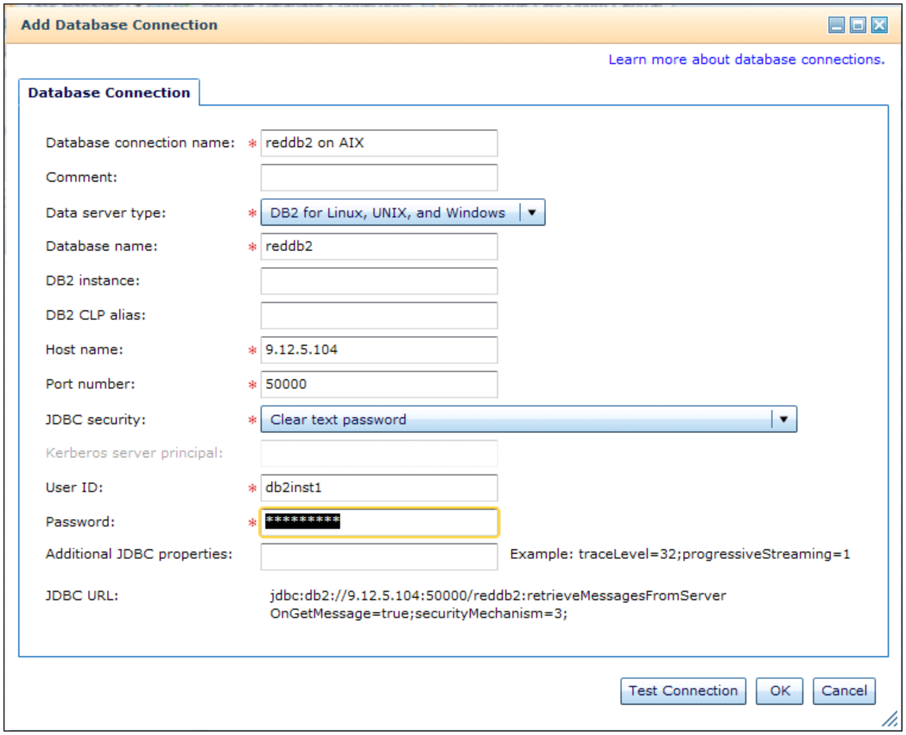


Figure 3-27 Add Database Connection

5. Click **Test Connection** to ensure the authorization and connectivity of the database you want to monitor.

   If authenticated and connected, you will receive a connection successful message.

   Click **OK** to close the pop-up and return to the wizard that guides you through adding a database connection.

After you complete the steps in the wizard, the database connection is added as *Not configured* to the list of database connections under the Manage Database Connections dashboard.

Importing database connections

You can import database connection profiles using files. The two supported file formats are:

- Comma-separated value text (CVS) file:

  Optim Performance Manager comes with a sample CSV text file that contains the information about all the database connections. You can edit and use it to import connections. The sample CSV is located in the `samples` subdirectory of installation directory:

```
<OPM_Working_Directory>/RepositoryServer/samples
```

- Configuration profiles that you can export from DB2 by using the Configuration Assistant menu in the Control Center or by using the `db2cfexp` command, for example:

```
db2cfexp <filename> MAINTAIN
```

Let us consider we have a CSV file named `Import_Databaseconnection.txt` on our local machine which has the information about the database connections to be added. Perform the following steps to import the database connection profile using this CSV file:

1. Under Manage Database Connection tab, click **Import**.
2. An Import Connection wizard is opened (Figure 3-28). Browse for the location of your file and click **Next.**

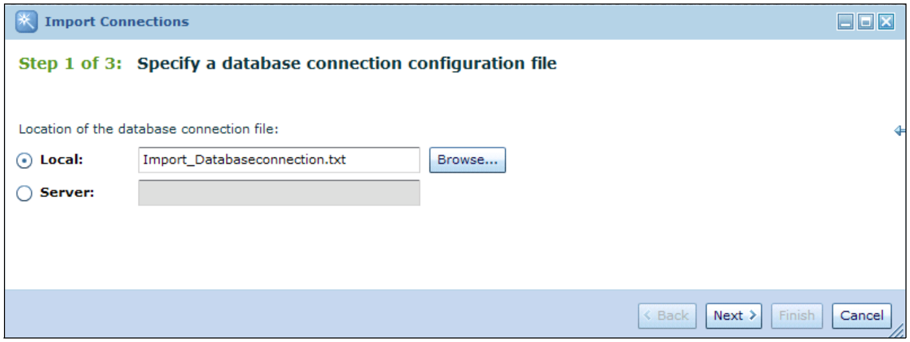


Figure 3-28 Import Connections

3. The next step is how you want to handle a duplicate connection. For the purpose of this book, we say update the existing connection and click **Next**.
4. This display a preview of connections to be imported. See Figure 3-29 on page 98.

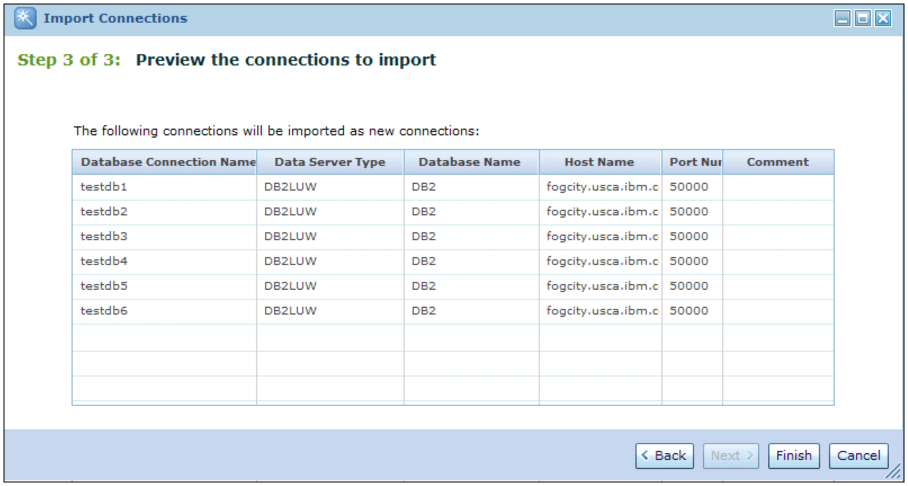


Figure 3-29 Preview the connections to import

5. Click **Finish** and an Import Summary is displayed.

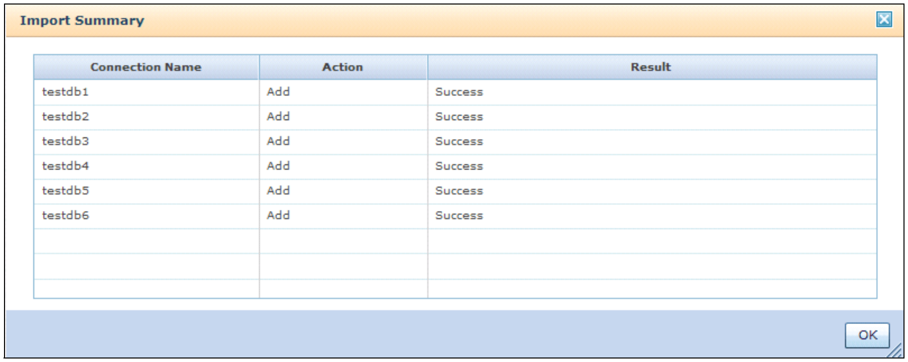


Figure 3-30 Import Summary

After you complete the steps in the Import wizard, the database connections are added as *Not configured* to the list of database connections under the Manage Database Connections dashboard.

3.3.3 Configuring the database for monitoring

After you have added database connection to the Optim Performance Manager, the next step is to configure it for monitoring using the following procedure:

1. Log on to Optim Performance Manager.

2. From the Optim Performance Manager panel, click **Manage Database Connections**. You see a list of database connections that have been added with the Monitoring Status as *Not Configured*.
3. Select the database for which you want to configure monitoring and click **Configure Monitoring**. The Configure Monitoring wizard opens.
4. In the Configure Monitoring panel, enter the following information:
   - Physical database name:

     This is automatically populated with the selected database name.
   - Storage path for collected monitor data:

     By default this is checked to use the default table space path. You can specify a different path. A new path can only be entered if the user selected SMS or DMS table space type during installation. For more details on Storage options refer to 2.5, “Storage options” on page 49 that introduces you to storage concept of Optim Performance Manager.
   - Optim Performance Manager collection user:

     This is automatically populated with the same user that you specified when you added a database connection. You can change to a different user who has appropriate authorization on the database to be monitored. For more details on authorization, refer to 2.6, “User authorization” on page 52.
   - Password:

     Enter the authentication password for the above user.
   - Time zone:

     This parameter refers to the time zones of the server on which the DB2 instance runs. Make sure that this is correct. Optim Performance Manager uses the time zone information to display collected performance data in the time zone of the monitored database. If the time zone is incorrect then wrong timestamps are displayed for collected performance data.
   - Now select a starting point for your monitoring configuration profile that can be either new, use predefined template, or clone from another already configured database. Predefined templates are available for various monitored database systems, for example OLTP, BI or SAP systems. For more details on the various predefined templates, refer to the Information Center at this address:

     `http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.monitor.doc/sys_templates_monitor_profiles.html`

     For the purpose of our book, we select **Use predefined template** and from the drop down we select **Development system**.

– Activate monitoring: Ensure that you have this checkbox clicked if you want to monitor this database immediately after configuration has finished.

Figure 3-31 on page 100 shows the Step 1 for configuring Monitoring settings.

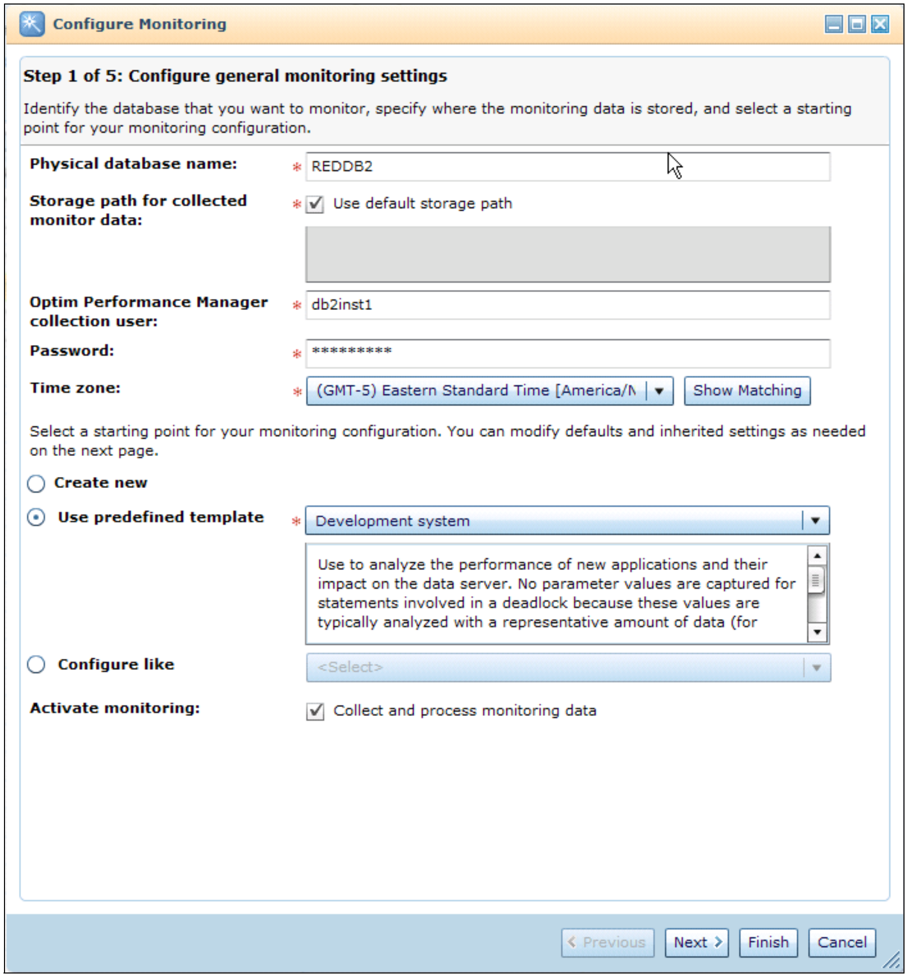


Figure 3-31 Configure Monitoring Step-1

5. Click **Next.**

6. Because we select a predefined template in the previous step, Optim Performance Manager enable associated monitoring profiles - Figure 3-32 on page 101. Depending on what you selected, the system collects different types of data, such as inflight performance, reporting, or extended insight data. You can click the pencil icon next to each profile to get an idea of the

default settings for each preconfigured monitoring profile and edit it if required. Additionally, Optim Performance Manager sets alert thresholds depending on the selected predefined template. For example, for a monitored OLTP system a buffer pool hit ratio over 90% is much more important than for a BI system. You can not edit the alert thresholds using the pencil icons, but you can edit them after configuration using the Performance Alert Configuration dashboard available from Task Manager.

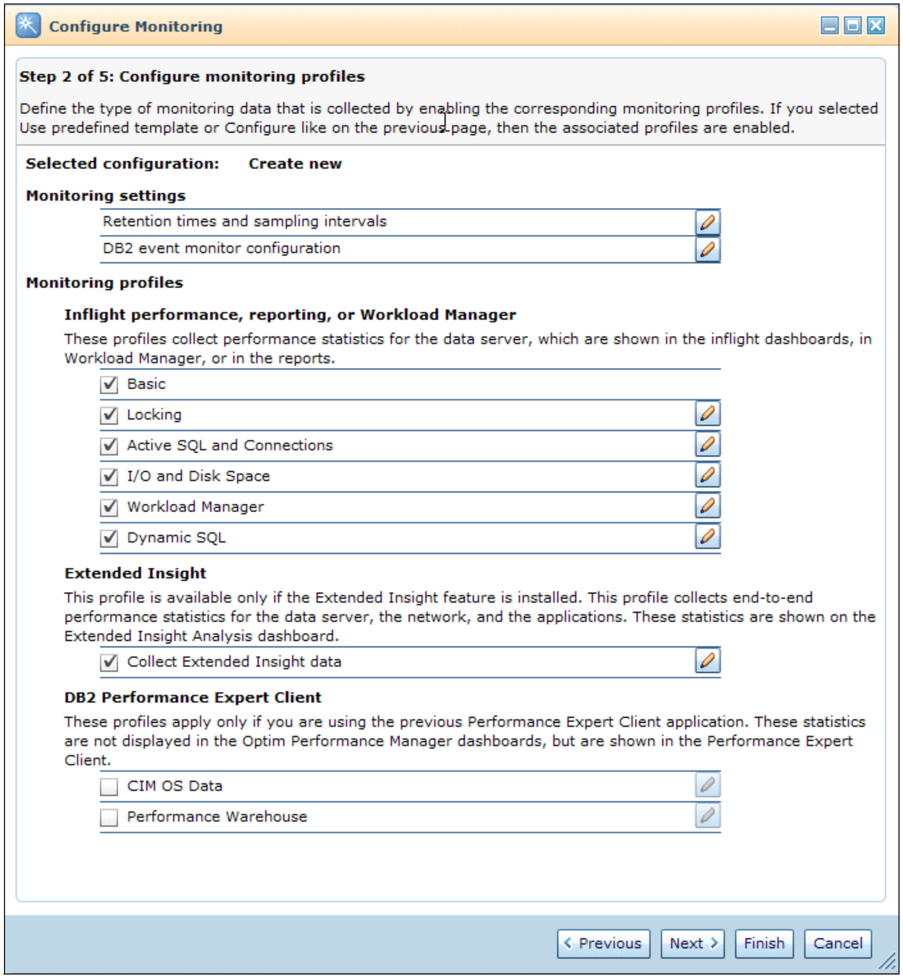


Figure 3-32 Configure Monitoring step 2

Predefined templates are a great way to start with Optim Performance Manager. If you are not familiar with Optim Performance Manager, we suggest that you choose one of the predefined templates. Think of the predefined templates as a fast way of getting started. Once you are familiar with the settings, you can customize them later.

– Monitor Settings:

  - Retention times and sampling intervals:

    Specify how long to keep performance data and how often to collect performance data. The selected predefined template will base these values. You can edit these values at a later time depending on your requirements.

    The *Sampling rate in minutes* specifies the basic interval in that Optim Performance Manager collects snapshot data. In some monitoring profiles you can increase the basic interval to collect data less often.

    The *Data retention in hours* applies to collected inflight performance, report and Workload Manager data. After the retention period is reached Optim Performance Data deletes the collected data automatically. Refer to A.4, "Deleting data from the repository database" on page 453 for detailed information about the automatic deletion.

    The *Storage period* settings apply to collected Extended Insight data. Optim Performance Manager aggregates that data first before it deletes the data completely. Refer to A.3, "Data aggregation concepts" on page 450 about purpose and concept of the data aggregation.

  - DB2 event monitor configuration

    Specify the table space on the monitored database that Optim Performance Manager should use to create event monitor tables for event monitors. If you do not specify one, DB2 chooses the default table space.

    Note: We recommend that you create a dedicated 32K table space for the event monitors. If your monitored database is a partitioned database then the table space must be created across all partitions that you monitor. The table space that you specify here is used for all event monitors that Optim Performance Manager creates unless you specify another table space for dedicated event monitors in the following monitoring profiles: Locking, Workload Manager, Extended Insight.

– Monitoring profiles for inflight performance, reports or Workload Manager:

  These profiles collect performance statistics from the data server and present the data in the inflight dashboards, in Workload Manager, or in the reports. You can edit any of these values from the default thresholds at any point.

- Basic: The Basic profile collects data from the database manager, database, and buffer pool snapshot.
- Locking: The Locking profile uses the Lock event monitor and Lock wait information settings that you can enable separately.
- Active SQL and Connections: The Active SQL and Connections profile collects data from the application snapshot.
- I/O and Disk Space: The I/O and Disk Space profile, by default, collects information for buffer pools. You can also specify to collect I/O information for tables and table spaces.
- Workload Manager: The Workload Manager profile collects data from the statistic event monitor for workload management statistics.
- Dynamic SQL: The Dynamic SQL profile collects data from the dynamic SQL snapshot.

For more details about the monitoring profiles, how to enable them, and sample screen captures of these profiles, refer to 3.3.4, “Monitoring profiles for inflight performance, reporting and Workload manager” on page 109.

– Monitoring profile for Extended Insight:

The Extended Insight profile collects statement and transaction metrics from the Extended Insight Clients and from the data server.

This option is available only if you have activated the Extended Insight feature on the Optim Performance Manager server. If the option is grayed out, check that your Extended Insight activation is completed and is successful. If Extended Insight is activated and it is still grayed out, restart the Optim Performance Manager repository server and the WebSphere Application Server associated with the Optim Performance Manager.

For more details about Extended Insight profile, refer to 3.3.5, “Monitoring profile for Extended Insight” on page 119.

– Monitoring profiles for DB2 Performance Expert Client:

These profiles apply only if you are using Performance Expert Client as client interface to Optim Performance Manager. These statistics are not displayed in the Optim Performance Manager dashboards, but are shown in the Performance Expert Client only.

- CIM OS Data: This is the Common Information Model Operating System Data profile that collects data from the CIMON server that is running on the monitored system and collecting data about the operating system.

- Performance Warehouse: Specify where to store monitor data for Performance Warehouse and how you want the data aggregated for Performance Warehouse.

For more information about the various monitoring profiles, refer to the Information Center at

http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.monitor.doc/sys_templates_monitor_profiles.html

7. Once you are done with configuring monitoring profiles, click **Next**.
8. View Resulting DB2 Settings:

   The switches and configuration settings that are displayed on this panel will be set for this database based on the monitoring profiles that you specified.

   Monitor switches are settings that instruct DB2 to gather more specific information if a given switch is turned on. You can turn on several monitor switches to gather more data. DB2 provides two types of monitors: snapshot and event monitors. Optim Performance Manager creates event monitors in a monitored database depending on the configuration of the monitoring profile.

   In Figure 3-33 on page 105, we see the list of snapshots collected that correspond to the monitoring profile configuration. All monitor switches are turned ON in order to collect the maximum information from the snapshots. Note that all listed event monitors are turned ON. Assuming that you have not set a dedicated table space in the monitoring profiles and have not set a `PCTDEACTIVATE` value in the monitoring profiles, Optim Performance Manager uses the following statements to create event monitors on the monitored DB2 9.7 database:

```
CREATE EVENT MONITOR <name> FOR LOCKING WRITE TO UNFORMATTED EVENT TABLE
(PCTDEACTIVATE 100) MANUALSTART

CREATE EVENT MONITOR <name> FOR PACKAGE CACHE WHERE
UPDATED_SINCE_BOUNDARY_TIME WRITE TO UNFORMATTED EVENT TABLE MANUALSTART

CREATE EVENT MONITOR <name> FOR STATISTICS WRITE TO TABLE
```

   If the unit of work event monitor would also be ON, this would result in the following statement:

```
CREATE EVENT MONITOR <name> FOR UNIT OF WORK WRITE TO UNFORMATTED EVENT
TABLE (PCTDEACTIVATE 100) MANUALSTART
```

   Warning icons indicate any configuration settings that might cause increased overhead. Review these settings and click **Next.**

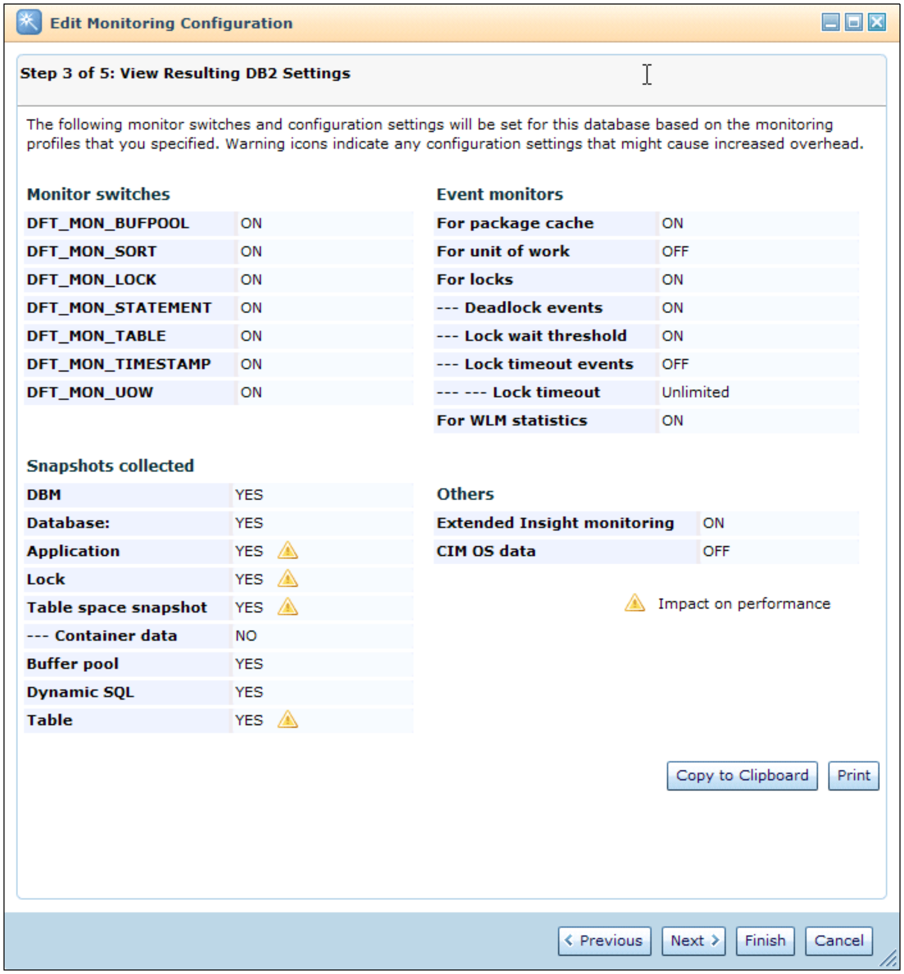


Figure 3-33 Configure Monitoring Step 3

9. Configure monitoring authorizations (Figure 3-34 on page 106):

   When you configure a database for monitoring, you can assign different monitoring privileges for users, groups, or roles that require access to the monitored database. These privileges define the monitoring operations a user is allowed to do.

   – The *CanMonitor* privilege allows this user to look at collected monitoring data of this database. It is checked when you open a dashboard. Select a database and specify user credentials for this database.

   – The *Can Manage Alerts* privilege allows a user to change alert thresholds and notifications for this database. It is checked on the alert notification and configuration dashboards when you select a database and specify user credentials for this database.

For more details on authorization, refer to 2.6, “User authorization” on page 52 that introduces you the security and authentication concept of Optim Performance Manager.

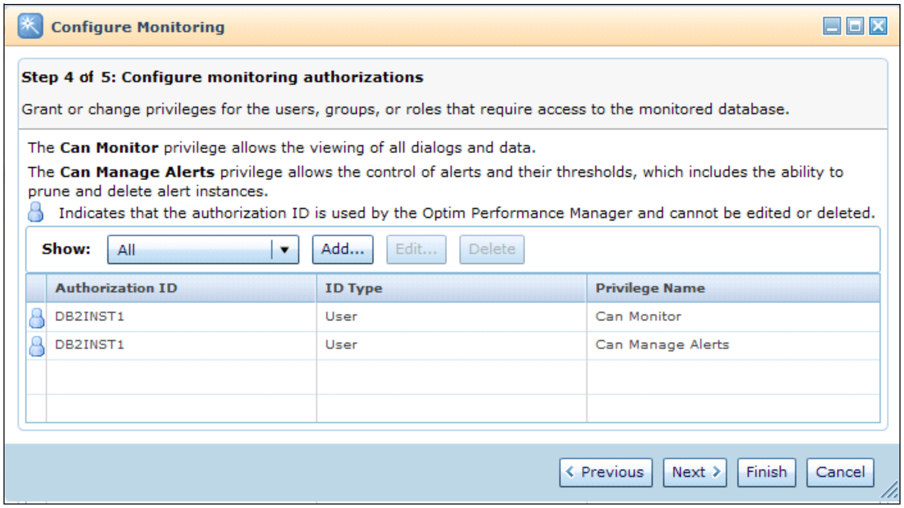


Figure 3-34 Configure Monitoring step 4

10. Configure partition sets:

    This configuration is applicable for partitioned databases only. When you first configure a partitioned database for monitoring, this tab does not appear. When you save the configuration and then re-edit the Configure Monitoring option, the Optim Performance Manager server discovers that it is a partitioned database. Now, you can edit the configuration and add the partitions to monitor. You can also assign each partition a role if a partition has different monitoring requirements. The roles available in Optim Performance Manager are Catalog partition, Coordinator partition, Data partition, and ETL partition.

11. Configuration Summary (Figure 3-35 on page 107):

    This panel shows a summary of the monitoring configuration that you specified for this database. The monitoring settings take effect when this monitoring configuration is saved. Complete the configuration by clicking **Finish**.

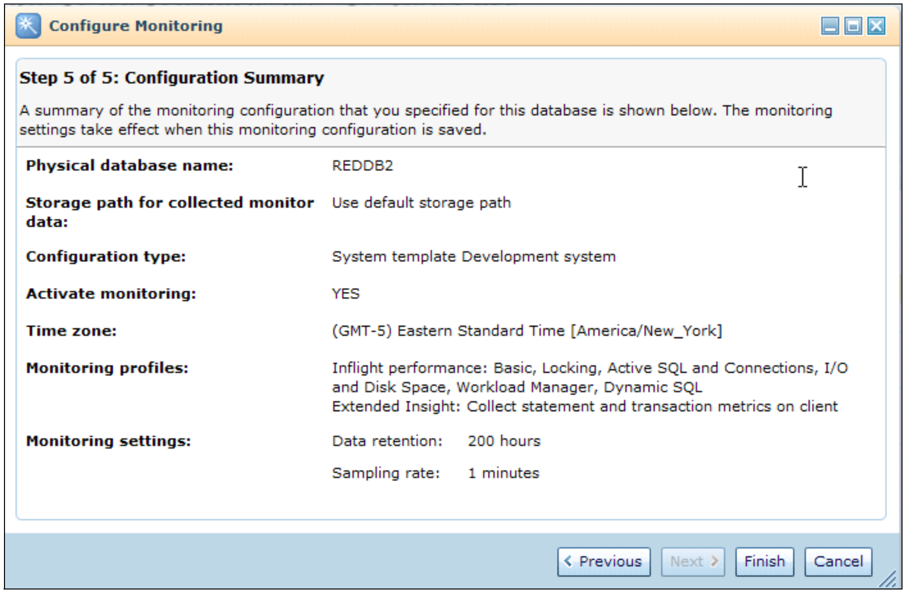


Figure 3-35 Configure Monitoring step 5

It takes a few minutes to complete the configuration. You may see a warning about watchdog procedures not being installed (Figure 3-36).

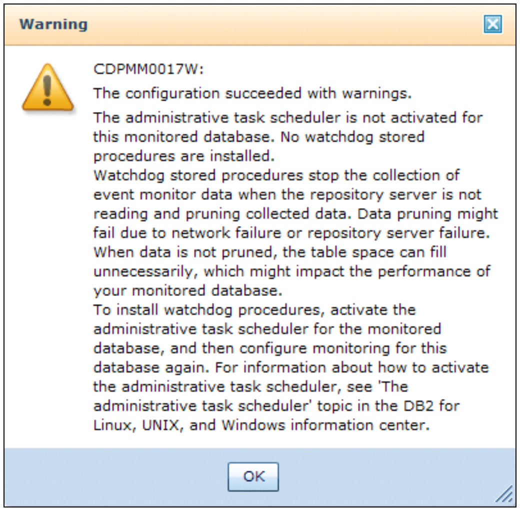


Figure 3-36 Configuration successful with warning

Tip: We recommend to activate the task scheduler because this is an easy way to ensure that the event monitors that Optim Performance Manager started are stopped in case of Optim Performance Manager failures. You can refer to the Information Center for more information about the watchdog procedures and how to install these at

`http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.installconfig.doc/using_watchdog_procedures.html`

After you have done a successful configuration, you can open several dashboards from the successful configuration message box (Figure 3-37).

You can also open various dashboards by clicking **Task Manager** in the Optim Performance Manager console window and then clicking the dashboards that you would like to see.

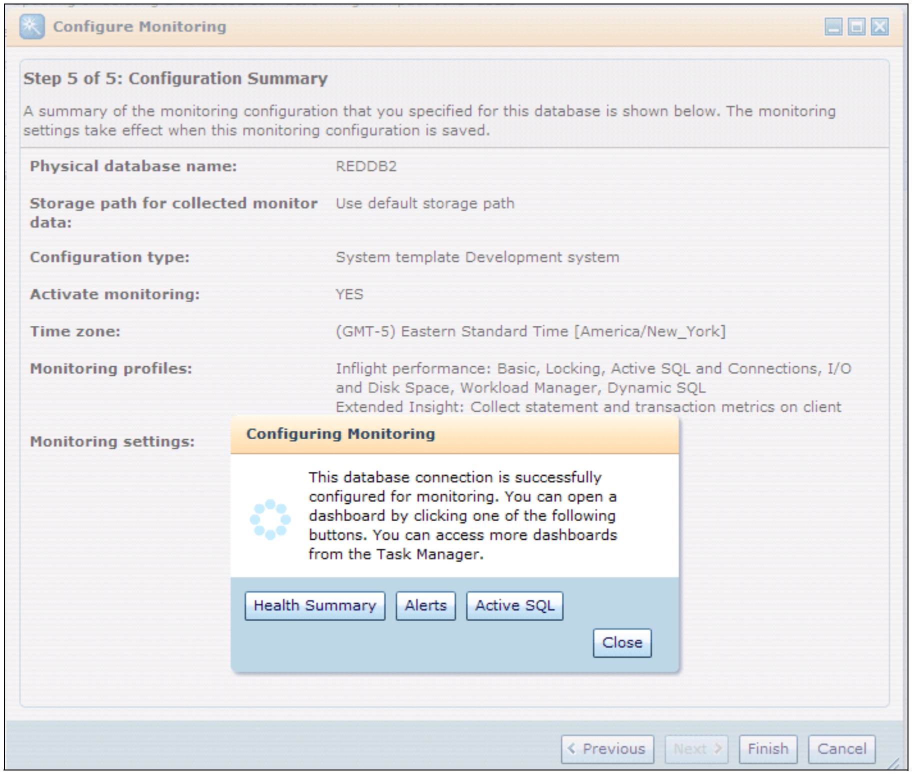


Figure 3-37 Successful configuration

3.3.4 Monitoring profiles for inflight performance, reporting and Workload manager

Monitoring profiles define the performance statistics that Optim Performance Manager should collect from the data server to be shown in the inflight dashboards, in the reports, or for Workload Manager configuration. In 3.3.3, "Configuring the database for monitoring" on page 98, we described shortly which monitoring profiles are available and how to enable and edit them. In this section, we describe the monitoring profiles in more detail and explain which dashboards or reports show the data of which profile. You might familiarize yourself with the dashboards and other features first before reading this section. The dashboards and features are described in Chapter 4, "Getting to know Optim Performance Manager" on page 147.

If a profile is not enabled, no performance data is collected for that profile. If you open a dashboard that would display data for a profile, but the profile is not enabled then you recive a message. Figure 3-38 shows the message that you receive if you open the Locking Dashboard, but the Locking monitoring profile is not enabled.

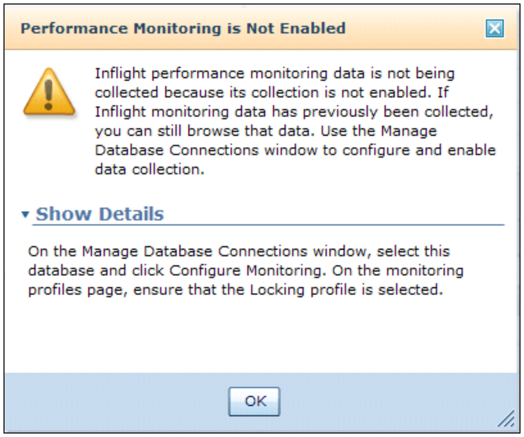


Figure 3-38 Performance Monitoring is not enabled for the selected profile

To enable the monitoring profile, click **Configure Monitoring** for the selected database on the Manage Database connections window and navigate to the **Configure Monitoring Profiles** step as shown in Figure 3-32 on page 101.

Basic profile

If this profile is enabled, Optim Performance Manager collects the basic monitoring information, such as statistics about database activity from the database and database manager snapshot and basic operating system load.

This profile also enables the collection of database and database manager configuration information. The collected data is displayed in various inflight dashboards such as Overview dashboard, Logging dashboard, Utilities dashboard or Workload dashbaord and in the database and database manager configuration reports reports.

Locking profile

If the Locking profile enabled, on the Locking dashboard, you can see the performance monitoring data collected from the lock snapshot, application snapshot, and the lock and deadlock event monitors. To access the locking dashboard, from the Optim Performance Manager console, open **Task Manager** → **Inflight Dashboards** → **Locking**. Select the databases and authenticate it. Figure 3-39 on page 110 shows the Locking Dashboard.

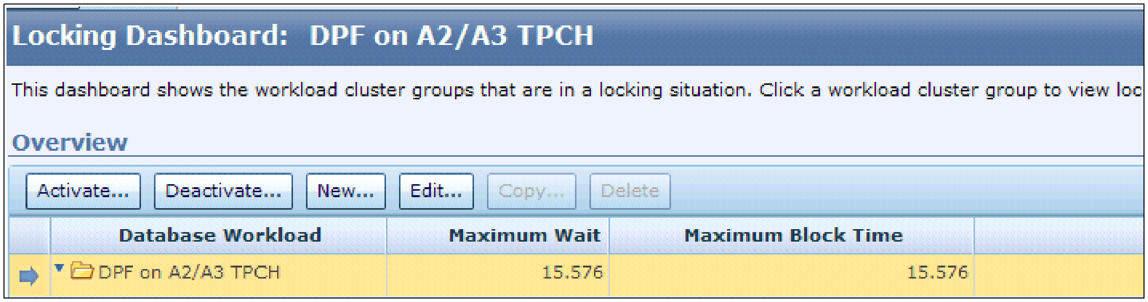


Figure 3-39 Locking Dashboard

If you enable lock wait information as shown in Figure 3-40, Optim Performance Manager uses snapshots to determine locking situations.

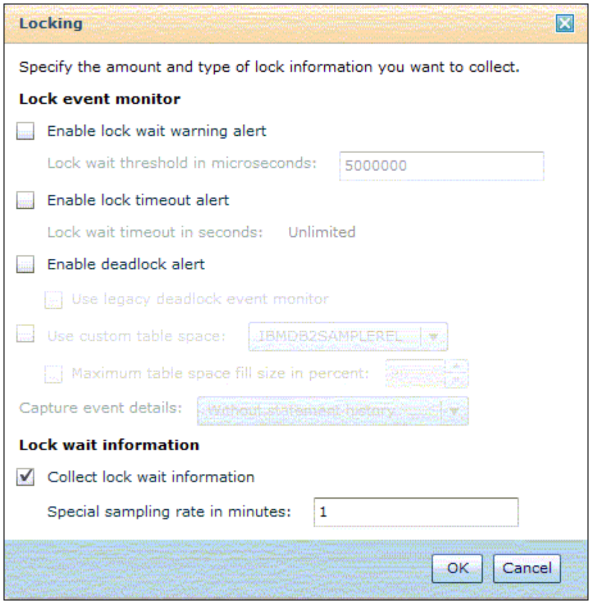


Figure 3-40 Locking Profile Details

On the Locking Dashboard (Figure 3-41 on page 112), Optim Performance Manager presents this information in the Max Wait Time and Max Block Time columns for your database workloads and in the list of Current Waiting Connections and Current Blocking Connections.

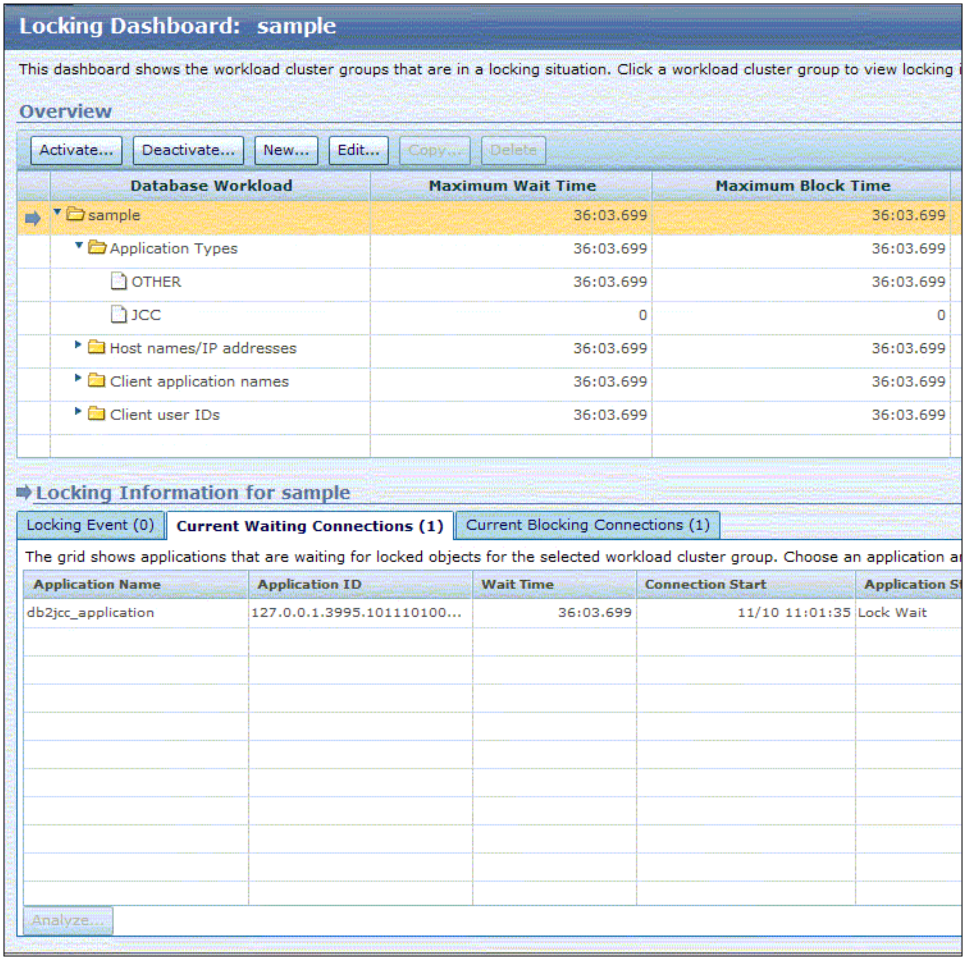


Figure 3-41 Analyzing Locking Situation

For each of these connections you can click **Analyze** to obtain detailed information about the connection and the holding or waiting statement shown in Figure 3-42 on page 113.

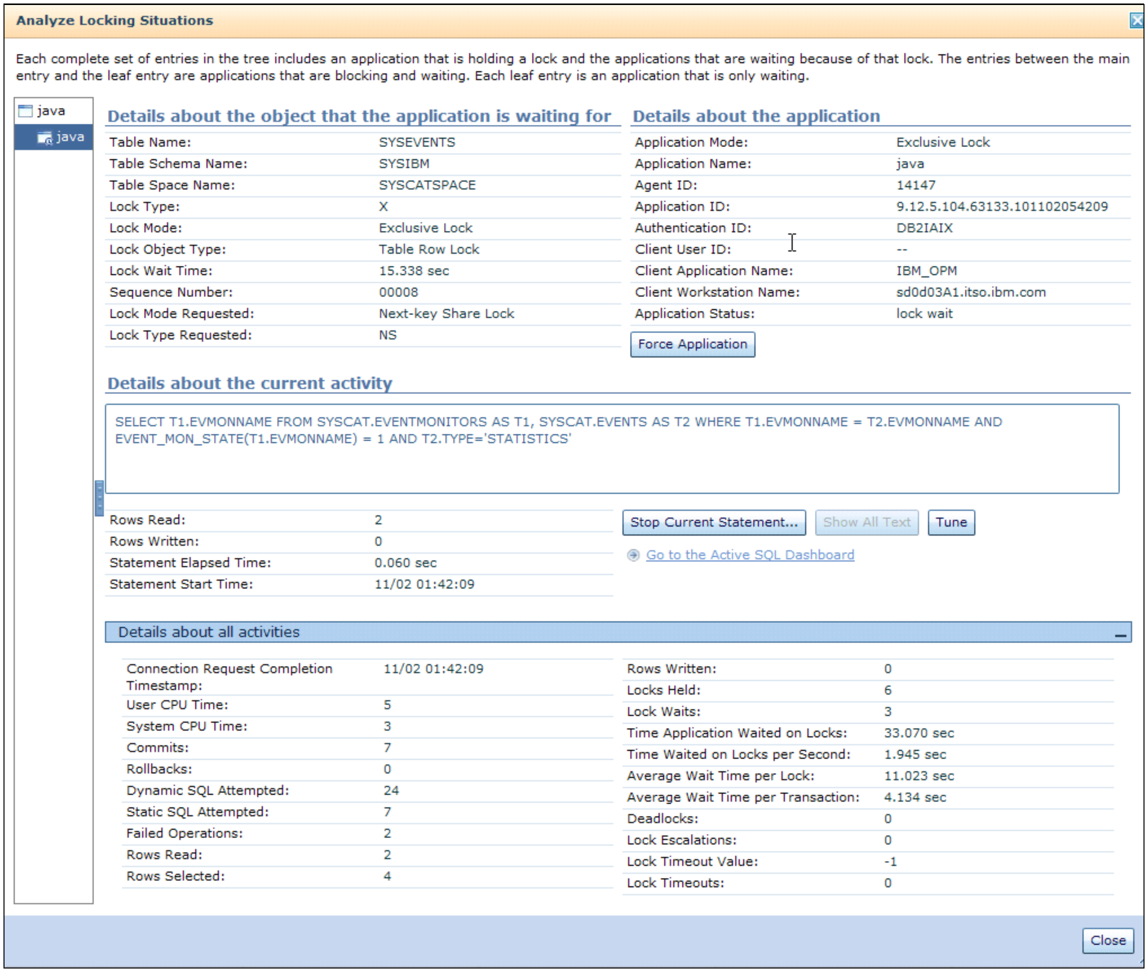


Figure 3-42 Analyze a locking situation

If you enable one or more of the Lock Event monitor options in the monitoring profile, Optim Performance Manager starts the Lock event monitor for DB2 9.7 and higher; or the deadlock event monitor for DB2 prior to Version 9.7. See Figure 3-43 on page 114.

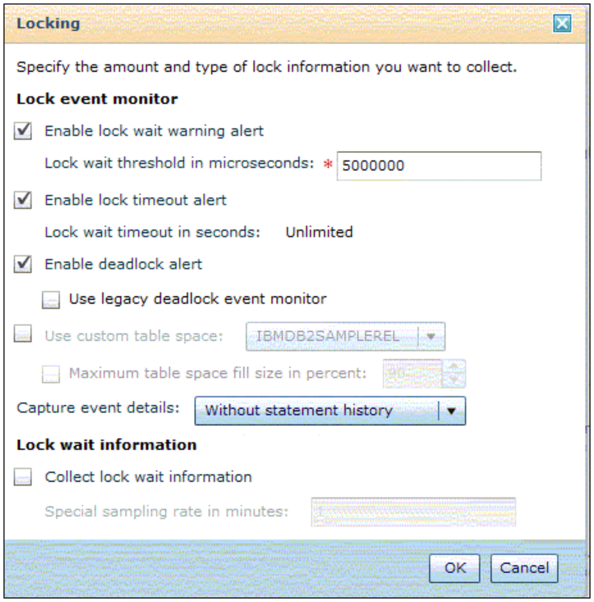


Figure 3-43 Locking monitoring profile

On the locking dashboard, Optim Performance Manager presents the collected events in the Deadlock, Timeout, and Lock Wait Alerts columns for your database workloads. Additionally, it lists all single events in the Locking Event tab (Figure 3-44 on page 115) with the ability to obtain the connection and statement details involved in the event by pressing the Analyze button.

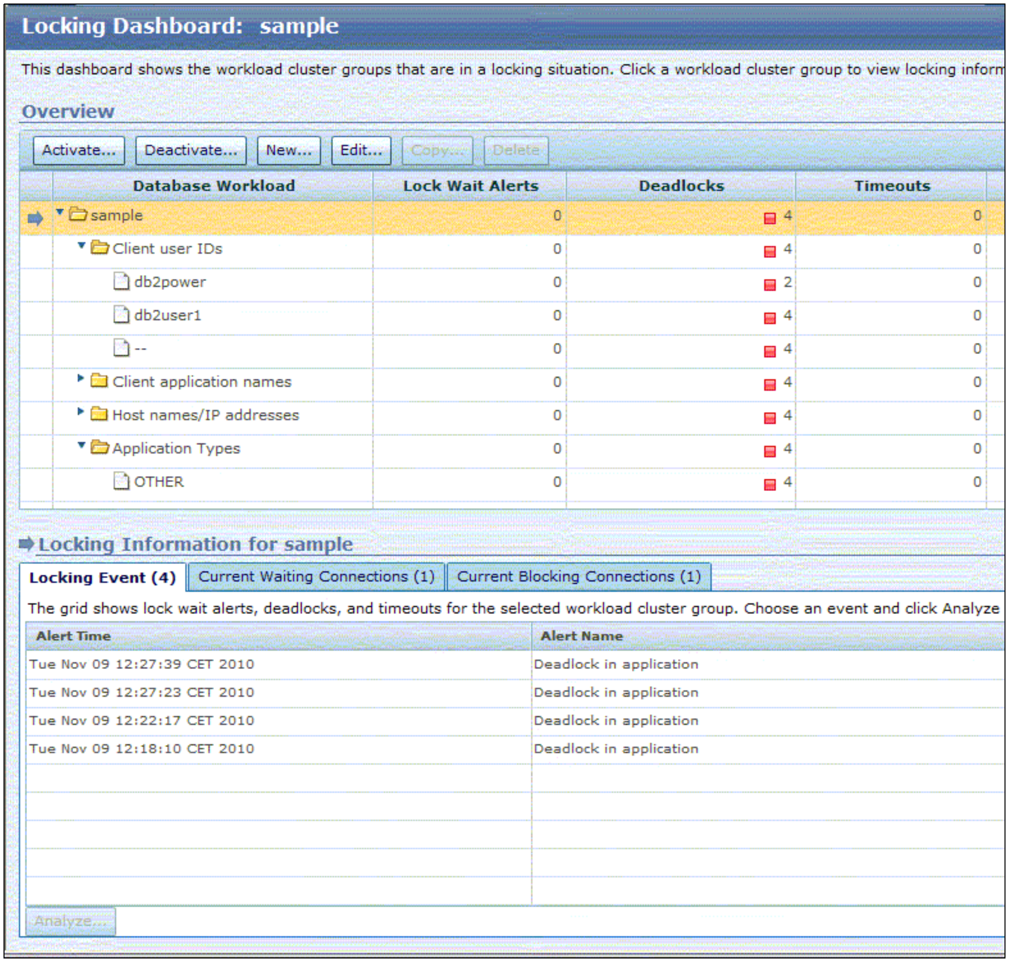


Figure 3-44 Analyze locking events

Active SQL and Connections profile

If the Active SQL and Connections profile is enabled, on the Active SQL dashboard, you can see the performance monitoring data that is collected from the application snapshot. The Database Connection Report is also based on the application snapshot data collected from this profile.

To access the Active SQL dashboard, from the Optim Performance Manager Console open **Task Manager** → **Inflight Dashboards** → **Active SQL**. Select the database and authenticate it. Figure 3-45 shows the Active SQL Dashboard.

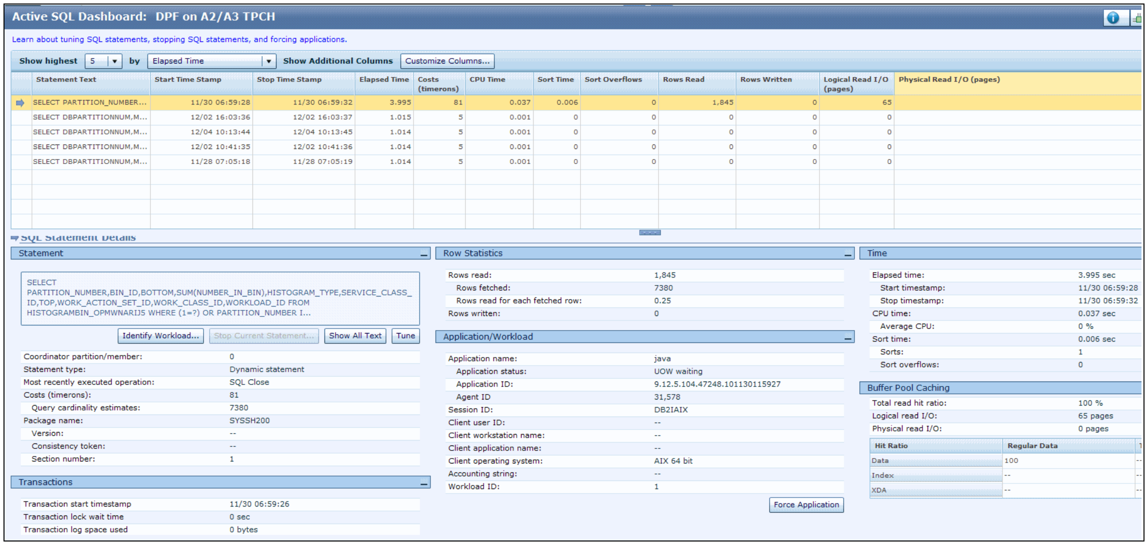


Figure 3-45 Active SQL Dashboard

I/O and Disk Space profile

If the I/O and Disk Space Profile is enabled, on Inflight dashboards, the you can see the performance monitoring data that is collected from the database manager, database, and buffer pool snapshot. Other available options are the table space and table snapshots. The data collected here isalso used by The Disk Space Consumption report.

To access the Inflight dashboards and, from the Optim Performance Manager Console, open **Task Manager** → **Inflight Dashboards** → **I/O and Disk Space**. Select the database and authenticate it. Figure 3-46 on page 117 shows the Buffer Pool and I/O Dashboard.

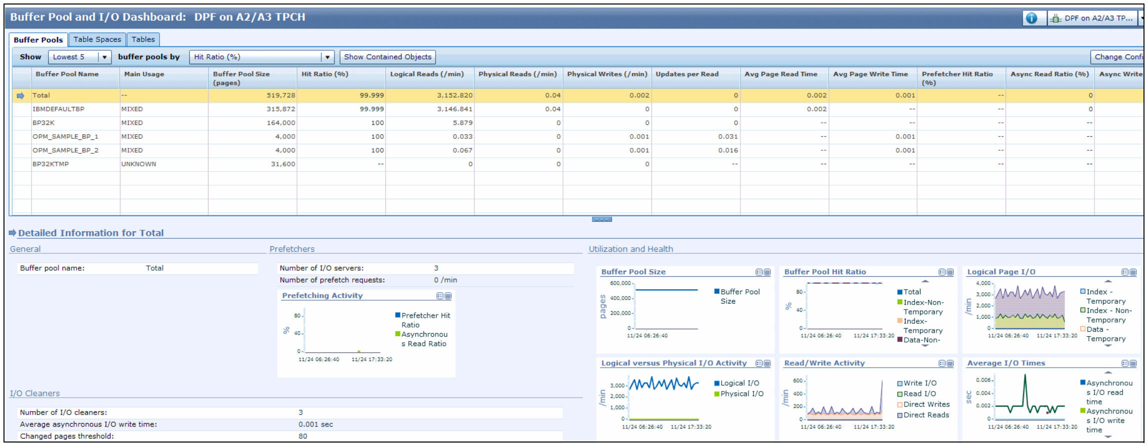
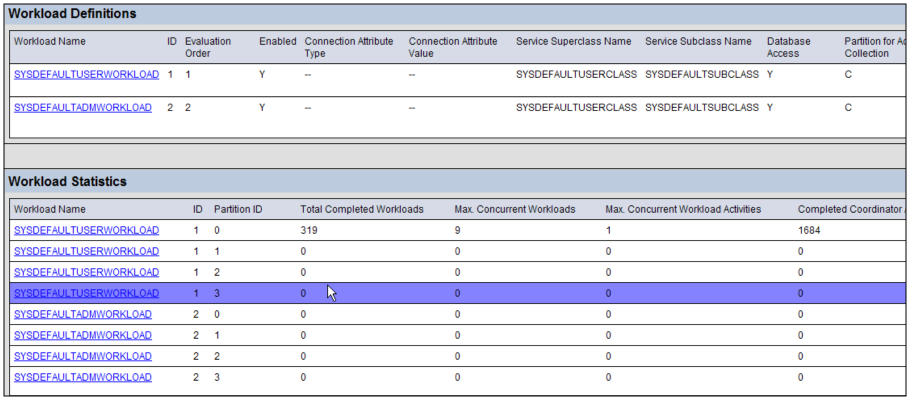


Figure 3-46 Buffer Pool and I/O Dashboard

Workload Manager profile

The Workload Manager profile enables the collection of configuration and statistics for Workload Manager workloads, service classes, and work classes. The collected data is displayed in Workload Manager Configuration and Metrics report. The "Workload Manager Configuration" also shows this information. To access Workload Manager Configuration, from the Optim Performance Manager Console, open **Task Manager** → **Workload Manager Configuration.** For more details about Workload Manager, refer to Chapter 11, "Workload Manager configuration tool" on page 373.

Figure 3-47 and Figure 3-48 on page 118 are examples of Workload Manager Configuration and Metrics report. To create such a report, use **Task Manager** → **Reports.**



Workload Definitions

Workload Name	ID	Evaluation Order	Enabled	Connection Attribute Type	Connection Attribute Value	Service Superclass Name	Service Subclass Name	Database Access	Partition for Ac
SYSDEFAULTUSERWORKLOAD	1	1	Y	--	--	SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	Y	C
SYSDEFAULTADMWORKLOAD	2	2	Y	--	--	SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	Y	C

Workload Statistics

Workload Name	ID	Partition ID	Total Completed Workloads	Max. Concurrent Workloads	Max. Concurrent Workload Activities	Completed Coordinator
SYSDEFAULTUSERWORKLOAD	1	0	319	9	1	1684
SYSDEFAULTUSERWORKLOAD	1	1	0	0	0	0
SYSDEFAULTUSERWORKLOAD	1	2	0	0	0	0
SYSDEFAULTUSERWORKLOAD	1	3	0	0	0	0
SYSDEFAULTADMWORKLOAD	2	0	0	0	0	0
SYSDEFAULTADMWORKLOAD	2	1	0	0	0	0
SYSDEFAULTADMWORKLOAD	2	2	0	0	0	0
SYSDEFAULTADMWORKLOAD	2	3	0	0	0	0

Figure 3-47 Workload Manager Configuration and Metrics report - Work class statistics

Service Superclass Definitions

Service Superclass Name	ID	Enabled	Agent Priority	Prefetch Priority	Outbound Correlator	Partition for Activity Data Collection	Activity Data to Collect	Collect Aggregate Activity Data	Collect Data
SYSDEFAULTSYSTEMCLASS	1	Y	-32768	Default	--	C	N	N	N
SYSDEFAULTMAINTENANCECLASS	2	Y	-32768	Default	--	C	N	N	N
SYSDEFAULTUSERCLASS	3	Y	-32768	Default	--	C	N	N	N

Service Subclass Definitions

Service Superclass Name	Service Subclass Name	ID	Enabled	Agent Priority	Prefetch Priority	Outbound Correlator	Partition for Activity Data Collection	Activity Data to Collect	Collect Aggregate Activit Data
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	11	Y	-32768	Default	--	C	N	N
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	12	Y	-32768	Default	--	C	N	N
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	13	Y	-32768	Default	--	C	N	N

Service Subclass Statistics

Service Superclass Name	Service Subclass Name	Partition ID	Concurrent Activities	Failed Coordinator Activites	Completed Coordinator Activities	Maximum lifetime of Coordinator Activities (ms)	Rejected Coordinator Activities	Max. Estimate (timerons)
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	0	4	1	1024	0	0	0

Figure 3-48 Workload Manager Configuration and Metrics report - Service class statistics

Dynamic SQL profile

This profile is required for the Dynamic SQL statement report.This report identifies the SQL statements that consume the most resources in a given period of time. The report includes a graphical representation of the workload so that you can identify critical or problematic SQL statements. To create this report To create such a report, use **Task Manager** → **Reports** from the Optim Performance Manager web console. Figure 3-49 on page 118 and Figure 3-50 show parts of a report displaying Top 5 SQL Statements by CPU Time.

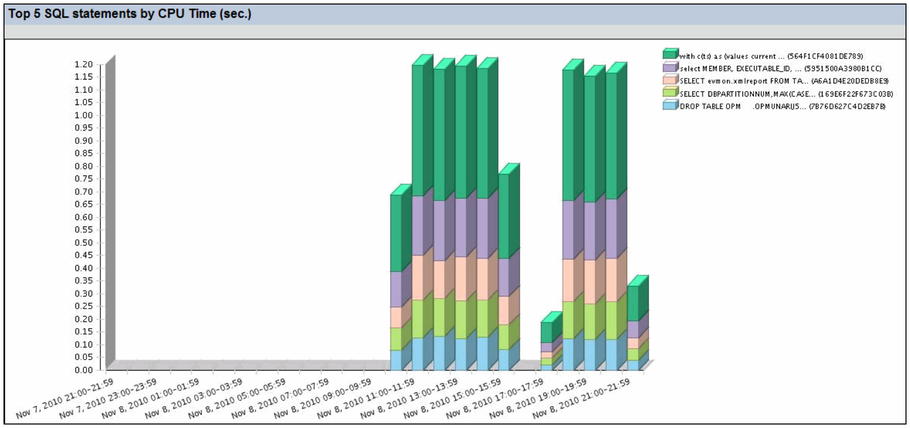


Figure 3-49 Dynamic SQL Report - Top 5 SQL Statements by CPU Time.

Statement Identifier	Statement Text	Partition ID	Number of executions	Total elapsed time (sec)	Average elapsed time (sec)	Number of Logical Reads	Number of physical reads	▼ CPU time (sec)	Total System CPU Time (sec)	Total User CPU Time (sec)	Sorts	Average rows read	Total rows read	Average rows read per selected row	Average rows written	Total rows written
564F1CF4081DE789	with c(ts) as (values current timestamp)select cac...	0	519	12.076983	0.023270	178,190	0	4.533402	0.594186	3.939216	0	4,039	2,096,416	374	340	176,645
5951500A3980B1CC	select MEMBER, EXECUTABLE_ID, STMT_TYPE_ID, INSERT...	0	526	15.497640	0.029463	0	0	2.074248	0.179207	1.895041	0	0	0	0	0	0
7B76D627C4D2EB7B	DROP TABLE OPM .OPMUNARIJ52	0	524	422.914270	0.807088	102,927	0	1.126783	0.076132	1.050651	0	29	15,715	--	20	10,602
A6A1D4E20DEDB8E9	SELECT evmon.xmlreport FROM TABLE (EVMON_FORMAT_UE...	0	495	6.906762	0.013953	15,968	0	1.021705	0.228021	0.793684	0	2	1,030	0	0	0
169E6F22F673C03B	SELECT DBPARTITIONNUM,MAX(CASE WHEN NAME='HOST_NAM...	0	262	265.480034	1.013283	0	0	0.389332	0.077114	0.312218	262	0	0	0	0	0
169E6F22F673C03B	SELECT DBPARTITIONNUM,MAX(CASE WHEN NAME='HOST_NAM...	2	262	263.083908	1.004137	0	0	0.336964	0.064802	0.272162	262	0	0	--	0	0
169E6F22F673C03B	SELECT DBPARTITIONNUM,MAX(CASE WHEN NAME='HOST_NAM...	3	262	262.987637	1.003770	0	0	0.299586	0.062916	0.236670	262	0	0	--	0	0
169E6F22F673C03B	SELECT DBPARTITIONNUM,MAX(CASE WHEN NAME='HOST_NAM...	1	262	262.999949	1.003817	0	0	0.285946	0.058939	0.227007	262	0	0	--	0	0
A6A1D4E20DEDB8E9	SELECT evmon.xmlreport FROM TABLE (EVMON_FORMAT_UE...	2	495	0.464963	0.000939	15,947	0	0.163569	0.012806	0.150763	0	1	988	--	0	0

Figure 3-50 Dynamic SQL Report - Details of the SQL Statements

3.3.5 Monitoring profile for Extended Insight

The Extended Insight monitoring profile enables and configures the collection of transaction and statement response time data that is displayed on the Extended Insight dashboard. If Extended Insight is completely set up and configured, it collects data at two locations:

- It collects respone time data at the location of your application that initiates the transactions on the monitored database. This is performed by the Extended Insight client that must be installed and configured on the computer where the application runs on.
- It collects data from the monitored database itself to obtain details about transaction and statement execution on the database. This is performed by the Extended Insight server within Optim Performance Manager.

The Extended Insight dashboard displays a combination of the data collected from both locations. Review the architecture discussion in 1.2.2, “Optim Performance Manager Extended Insight architecture” on page 13 to have an overview about the parts involved in the data collection. Additionally, read 4.4, “Extended Insight dashboard” on page 178 to have an understanding of the data that the Extended Insight dashboard displays.

If you open the Extended Insight dashboard and this profile is not enabled for the selected database, you receive the message shown in Figure 3-51.

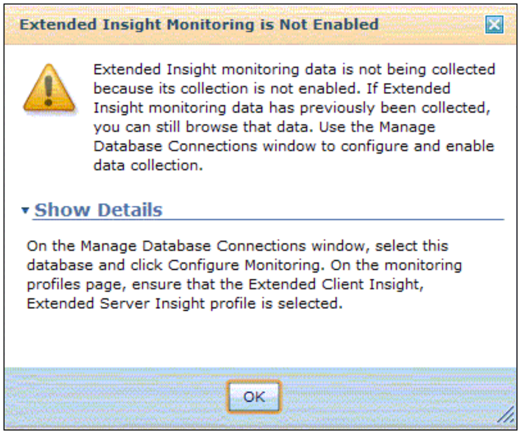


Figure 3-51 Extended Insight profile not enabled message

To enable the Extended Insight profile click **Configure Monitoring** for the selected database on the Manage Database connections window and navigate to the **Configure Monitoring Profiles** step as shown in Figure 3-52 on page 120. You can enable the Extended Insight profile only if the Extended Insight license is activated. See 3.2.3, “Activating the Optim Performance Manager Extended Insight license” on page 81 for how to activate the license.

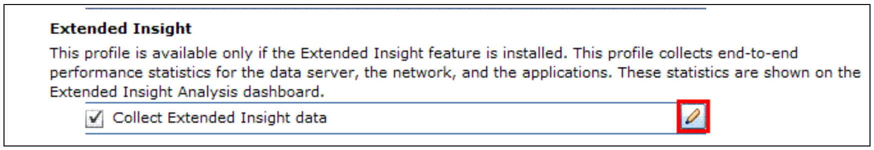


Figure 3-52 Enabling Extended Insight profile

To edit the configuration properties, click the pencil icon. Figure 3-53 shows the Collect Extended Insight data configuration panel.

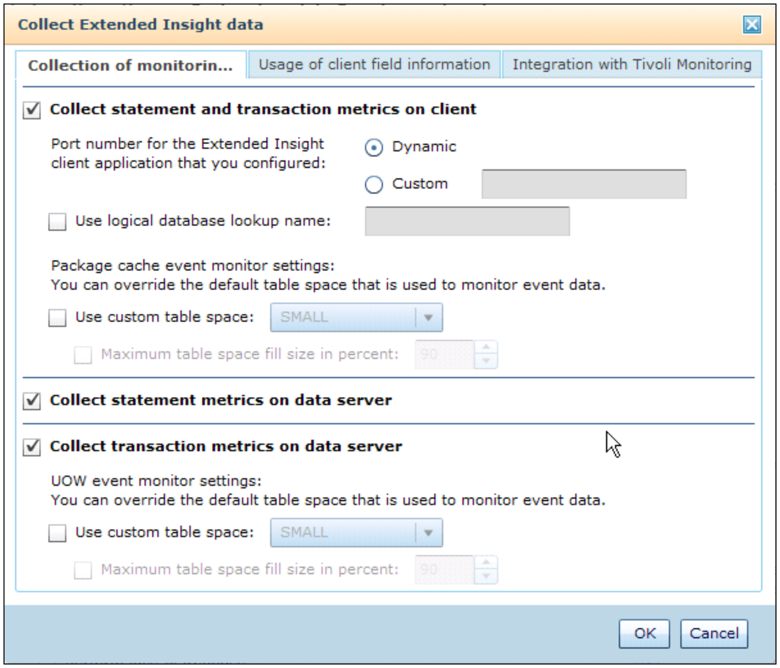


Figure 3-53 Configuring Collect Extended Insight data

There are three main configurations involved here:

- Collection of monitoring data
- Usage of client field information
- Integration with Tivoli Monitoring

Collection of monitoring data

Monitoring data, statement and transaction metrics, can be collected on the Extended Insight client and the Extended Insight monitoring server.

In the Collection of monitoring data tab, you configure the following information:

- Collect statement and transaction metrics on client

  When enabling this task, you can view the end-to-end transaction response time data for this database and it's workloads on the Extended Insight dashboard. It includes, for example, maximum and average transaction response times, and the response time breakdown into client, network, and data server per workload.

  The end-to-end response time data is delivered from the Extended Insight clients that must be installed and configured on the systems that run the applications initiating transactions and executing workload on the monitored database. The installation and configuration of Extended Insight Client is described in 3.4, “Installing and Configuring Extended Insight Client” on page 129.

Figure 3-54 is a sample screen for the data on the Extended Insight dashboard.

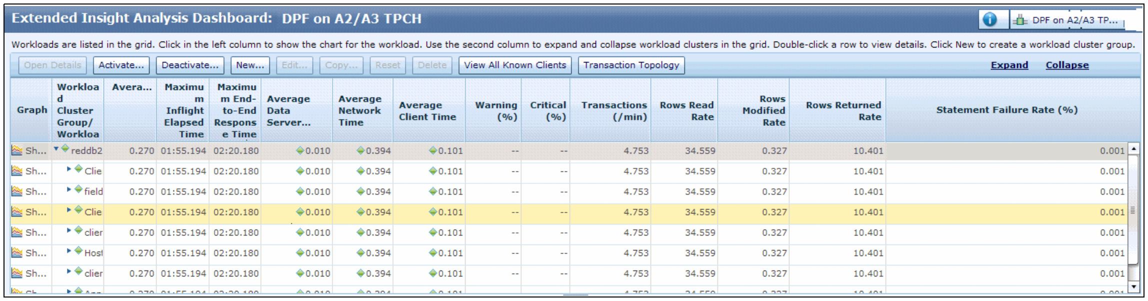


Figure 3-54 Extended Insight overview

Double clicking one workload opens a new window that shows a graphical response time chart, the executed SQL statements, and information about clients who executed the workload. See Figure 3-55.

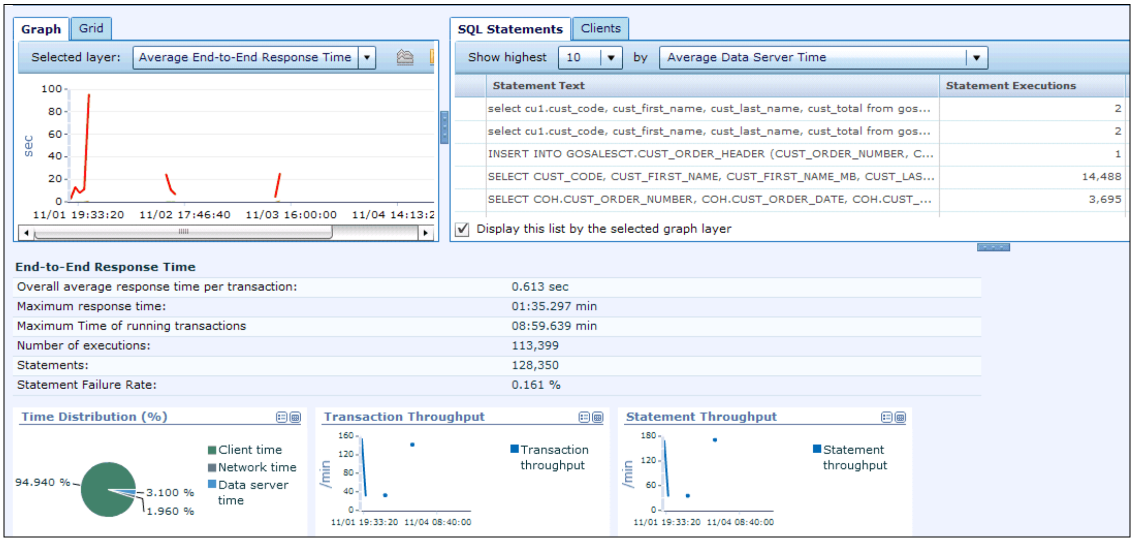


Figure 3-55 Extended Insight Details

Other settings under collect statement and transaction metrics on client are:

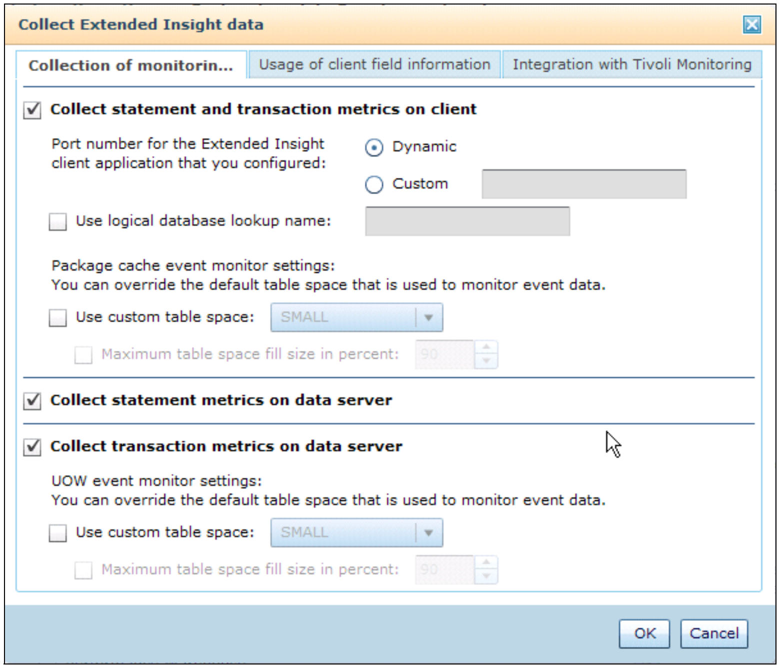


Figure 3-56 Configuring Collect Extended Insight data

– Port number for the Extended Insight client application that you configured

  On this port the Extended Insight monitoring server listens for end-to-end data from Extended Insight clients for this specific monitored database. By default it is determined dynamically. If you specify a port number, the port must be open.

– Use logical name

  This is an optional field. A definition of a logical name is required only if the applications for that you set up Extended Insight use different IP address or database alias to connect to the monitored database as Optim Performance Manager. This can happen in the following cases:

  - Your application use JCC type 2 to connect to the monitored database and have the monitored database cataloged in the local DB2 database catalog using a different database alias than Optim Performance Manager.
  - Your application uses a DB2 connect gateway to connect to a z/OS DB2 database.

> **Tip:** Extended Insight is also available for DB2 z/OS as part of IBM Tivoli Omegamon for Performance Expert for z/OS v5.1. We cover only Linux, UNIX, and Windows platforms.

- You use network address translation (NAT) in your company and a NAT is between the application and the monitored database, but not between Optim Performance Manager and the monitored database (or the other way round). In that case the application would use a different IP address than Optim Performance Manager to connect to the monitored database.
- You have multiple network adapters on the monitored data server. The application uses a different network adapter than Optim Performance Manager to connect to the monitored data server.

For all these cases, specifying a logical name ensures correct communication between Extended Insight client and Optim Performance Manager server in order to provide Extended Insight data for the monitored database.

If you specify a logical name, then you must specify it on Extended Insight client as well, either within your application in the connection URL or in the `pdq.properties` file.

To specify a logical name in the application in the connection URL, enrich the URL as follows:

```
dbc:db2://<host>:<port>/<dbname>:monitoredDataSourceName=<logical name>
```

To specify a logical name in `pdq.properties`, set the property as follows:

```
monitoredDataSourceName=<logical name>
```

– Use custom table space

This option is available only if the monitored database is on DB2 9.7 Fix Pack 1 or higher. Specify the table space on the monitored database that Optim Performance Manager should use to create event monitor tables for the package cache event monitors. If you do not specify one, Optim Performance Manager uses the one specified in the DB2 event monitor configuration settings. If that one is empty, then DB2 chooses the default tables space.

> **Tip:** We recommend that you specify a dedicated 32K table space defined across all partitions for the event monitors instead of letting DB2 choosing the default table space.

Optim Performance Manager uses the statement text retrieved from the package cache to show the complete statement text on the Extended Insight dashboard. DB2 may flush statements from the package cache before Optim Performance Manager could collect them. In that case Optim Performance Manager will not show the complete statement text on the Extended Insight dashboard. To avoid this, Optim Performance Manager uses the package cache event monitor to collect statements.

Refer to the Information Center for more information on DB2 package cache at

`http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000266.htm`

- Collect statement metrics on data server

  This option is available only when the monitored database is DB2 9.7 Fix Pack 1 or higher. When it is enabled, the Statement Details area of Extended Insight dashboard gets an additional *Statement Server Execution Details* tab that displays time spent and execution data about the statements. Optim Performance Manager collects the data from the monitored database using the `MON_GET_PKG_CACHE_STMT` table function. See Figure 3-57 on page 125 for a sample.

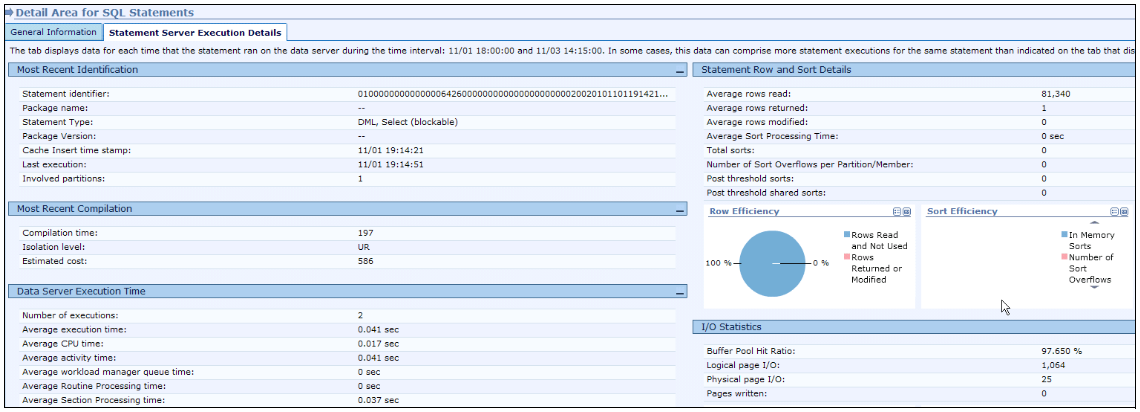


Figure 3-57 Statement Server Execution Details

- Collect transaction metrics on data server

  This option is available only when the monitoring database is DB2 9.7 Fix Pack 1 or higher. When it is enabled, you see details of the transactions from a data server execution perspective. Figure 3-58 illustrates more on the type of data seen when this is enabled.

For example on the Extended Insight Overview dashboard, as seen in Figure 3-58, there are additional columns (in comparison to Extended Insight Client only setup).

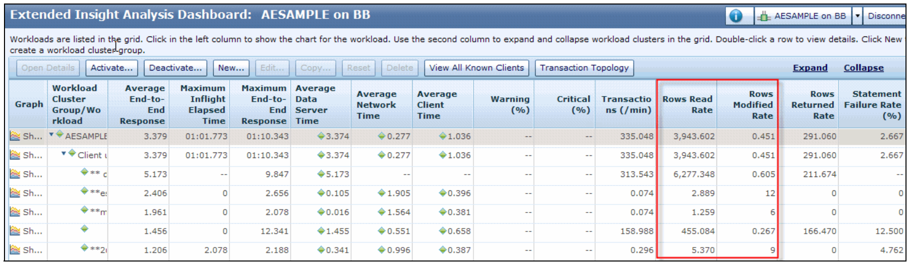


Figure 3-58 Extended Insight Analysis Dashboard

The response time distribution chart has all the layers from the *Average Data Server Time per Transaction* category. After selecting each layer, detailed data for this layer is displayed as a rainbow chart (Figure 3-59 on page 126).

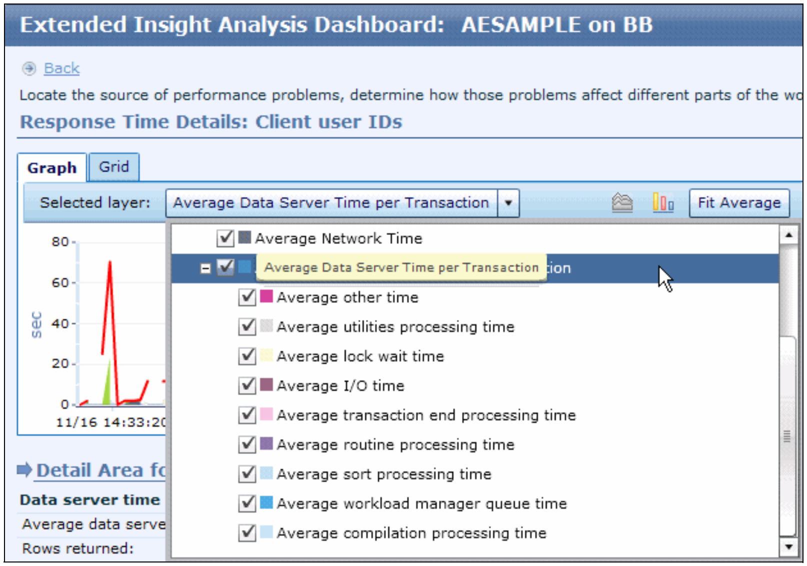


Figure 3-59 Extended Insight Analysis Dashboard

In case of a partitioned database, there is a TopPartitions tab with Extended Insight server transaction metrics on a per partition basis.

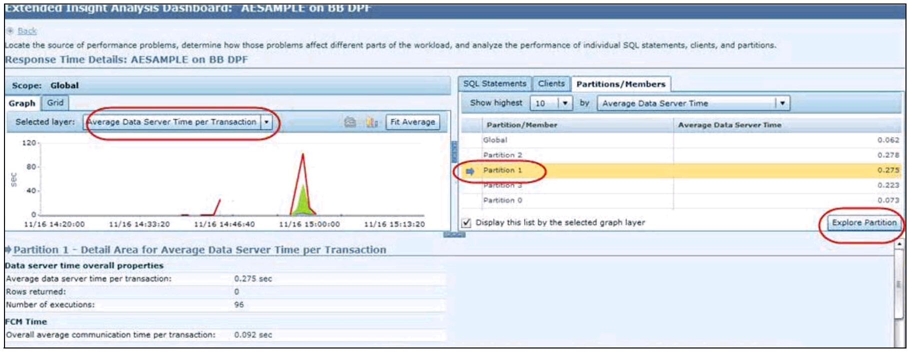


Figure 3-60 Extended Insight Analysis Dashboard

Selecting **Average Data Server Time per Transaction** in the left tab, and selecting the partition on the right side, and then clicking on the explore partition in Figure 3-60, you see more detail data for that partition as shown in Figure 3-61 on page 127.

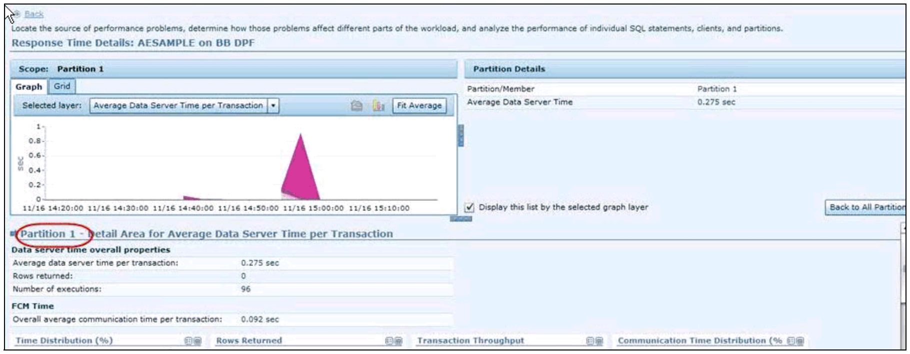


Figure 3-61 Extended Insight Dashboard - Explore partition

Usage of client field information

This functionality allow Optim Performance Manager 4.1.0.1 to be configured to restrict the set of client information fields, or to mask portions of the client information fields, that are used collectively as a key to aggregate the statement and transaction statistics. Configuration of client information field masking can be changed during runtime. It can take some time (depending on monitored client application configuration) to consume changed configuration of masking fields. By default, the masking is disabled. You can enable it by checking **Usage of client field information**. See Figure 3-62 on page 128.

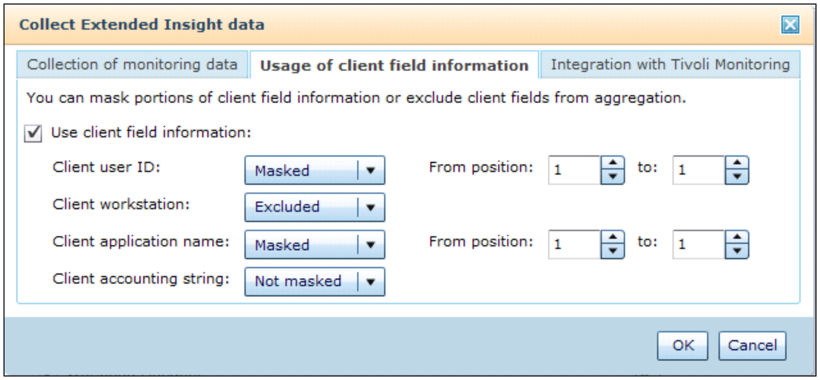


Figure 3-62 Usage of client field information

Client information fields are explained in 4.4.3, "Workload cluster groups and Workload clusters" on page 187. The default value for each client information field is *not* masked. Possible values for each client info fields are:

- Masked:

  The client information fields configured as Masked are included into the set of aggregation keys but part of the field is masked. Figure 3-63 shows that the Client user IDs field is Masked from field 1 to field 2.

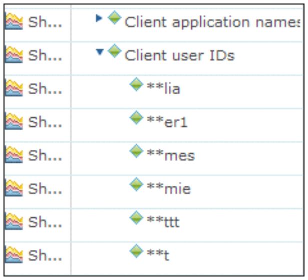


Figure 3-63 Masking applied to the Client user ID field in Extended Insight dashboard

- Excluded

  The client information fields configured as Excluded are excluded from the aggregation keys set. There is no information about excluded client information field collected in Optim Performance Manager and user is unable to do any clustering based on it. The Optim Performance Manager Extended Insight dashboard shows a blank for the client information field that is excluded. Figure 3-64 on page 129 shows that the Client user IDs field is excluded from masking.

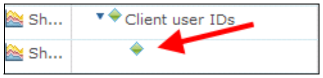


Figure 3-64 Excluding Client user ID field from masking

- Not masked

  The client information filed configured as *not* masked are included into the aggregation keys set. It behaves in the same way as in Optim Performance Manager 4.1. Figure 3-65 on page 129 shows that the Client user IDs field is *not* masked.

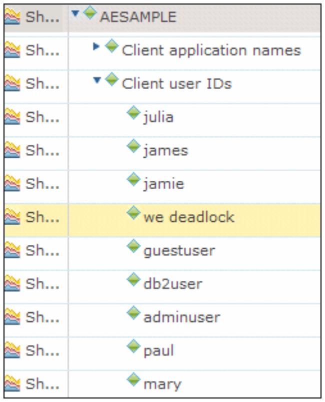


Figure 3-65 Masking not applied to Client User ID field

Integration with Tivoli Monitoring

For details about this setting, refer to Chapter 10, “Integration with Tivoli monitoring components” on page 323.

3.4 Installing and Configuring Extended Insight Client

To use Extended Insight monitoring with Optim Performance Manager, you must first activate Extended Insight on the Optim Performance Manager server using the activation tool as explained in 3.2.3, “Activating the Optim Performance Manager Extended Insight license” on page 81.

We can now go ahead with installing and configuring Extended Insight on the client computer that the applications run that you want to monitor with Extended Insight. Extended Insight client collects end-to-end response time information for database transactions that the applications initiate and execute on the monitored database and sends it to the Extended Insight monitoring server within Optim

Performance Manager. Since communication to the Extended Insight monitoring server is established during Extended Insight configuration, perform the configuration step after you have configured the database for monitoring and enabled the Extended Insight monitoring profile for in the Optim Performance Manager web console. Refer to 3.3.5, “Monitoring profile for Extended Insight” on page 119 for information how to do that.

3.4.1 Installing Optim Performance Manager Extended Insight client

Extended Insight client provides an installation wizard and a configuration wizard to install and configure the product on your client computer. If you want to install and configure the product on a system that does not have a graphical user interface, you can do so by using the console mode. For more details on the installation using the console mode, refer to the Information Center at

http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.ei.installconfig.doc/ei_install_wizard_or_console.html

The following are the steps we performed to install the Optim Performance Manager Extended Insight Client using the wizard on a Linux system:

1. From the directory of the Extended Insight installation image, launch the installation wizard by double click **`IBM_OPMEI_V4_1_0_1_Linux_x86.bin`**.
2. The installation wizard displays the title screen. For the purpose of the book we select English and click **OK**.
3. Proceed through the Introduction panel and the Software License Agreement panel.
4. From the Choose Install Directory panel (Figure 3-66 on page 131), specify the installation path where you want to install the product and click **Next**. Use the default directory unless you need to install it somewhere else.

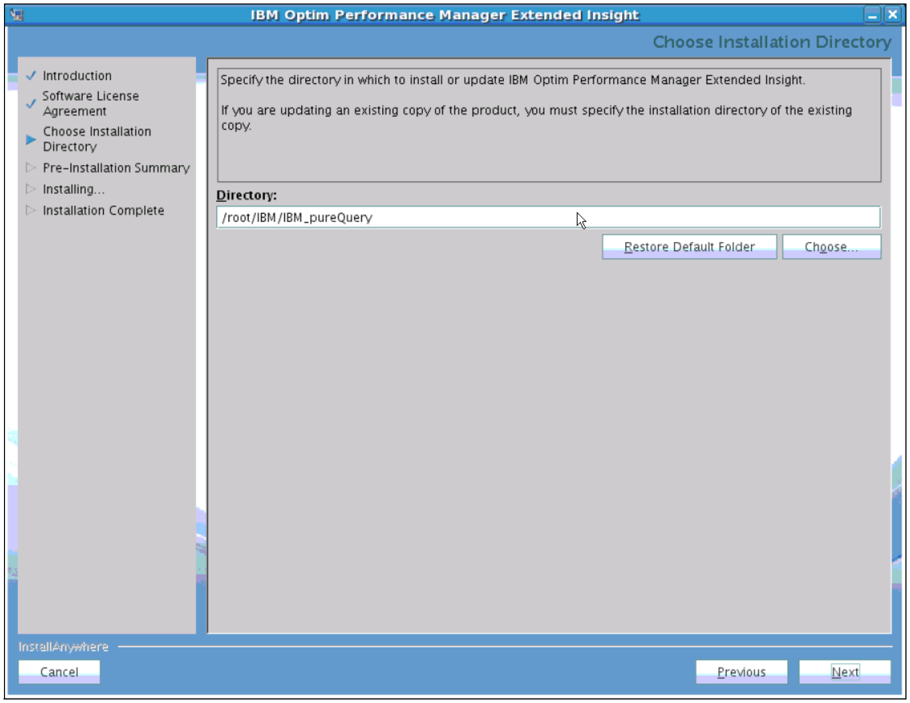


Figure 3-66 Choose Install Directory screen

5. Review the information on the Pre-Installation Summary screen, and click **Install**.

   It will take a few minutes for the installation to complete. The Installing screen has an indicator at the bottom that shows the progress.

6. When installation is complete, you see a screen that indicates installation was successful. Select **Open the configuration tool** and click **Done** (Figure 3-67 on page 132). If you want to run the configuration wizard later, uncheck the checkbox and click **Done.**

   Extended Insight is now installed on the client.

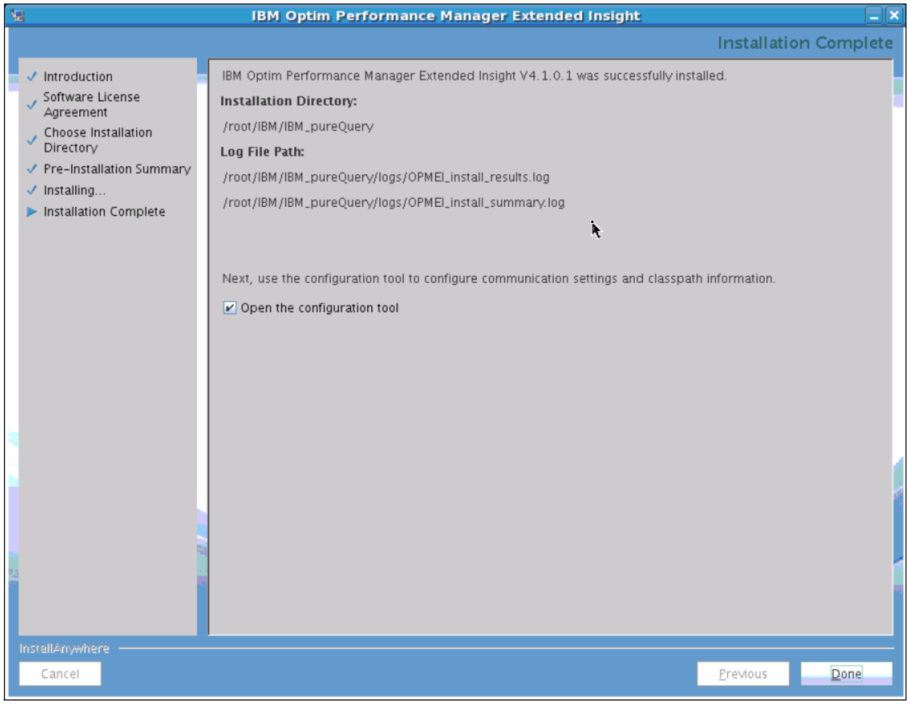


Figure 3-67 Installation complete

3.4.2 Configuring Optim Performance Manager Extended Insight Client

Once the Optim Performance Manager Extended Insight client is installed, you must configure it for each application that you want to monitor with Extented Insight. The configuration comprises the following:

- Ensures that the DB2 CLI or JDBC driver that the application uses to connect to the monitored database can load and access the Extended Insight client at application runtime.
  - During runtime, the DB2 CLI or JDBC driver calls Extended Insight client to provide data about transactions and statements, for example, start and end times.
  - Extended Insight client aggregates the provided transaction and statement data on a per minute basis.
  - If the application is a WebSphere application, the Extended Insight client checks periodically during runtime the connection pool state and collects connection pool information

- Establishes communication with the Extended Insight monitoring server for the monitored database within Optim Performance Manager.
  - During runtime Extended Insight client sends the aggregated transaction and statement data as well as connection pool information for WebSphere applications after each minute to the Extended Insight monitoring server.

If you ran the configuration program as part of the installation of Extended Insight client, the communication settings and your applications are already configured for extended insight monitoring. If you need more control over these configuration settings, or if you did not run the configuration tool, you must configure the Optim Performance Manager Extended Insight manually.

There are two approaches to configure the Extended Insight client:

- If you install the product by using the installation wizard, the installation wizard gives you the option to start the configuration tool automatically.
- You can run a command from the `EI_installation_dir/configuration` directory to start the configuration tool in the GUI mode:
  - UNIX: **`./cfgtool.sh`**
  - Windows: **`cfgtool.bat`**

  You can use **`./cfgtool.sh -i console`** for Unix and **`./cfgtool.bat -i console`** for Windows to run the configuration tool in console mode.

Here we demonstrate how to configure CLI, JDBC, and WebSphere applications for Optim Performance Manager Extended Insight monitoring using the installation wizard. We launched the configuration tool by selecting the Open the configuration tool at the end of installation.

Follow these steps to configure the database applications such as CLI, JDBC and WebSphere:

1. From the initial Configuration panel (Figure 3-68), select applications that you would like to configure at this point. For demonstration, we select all applications.

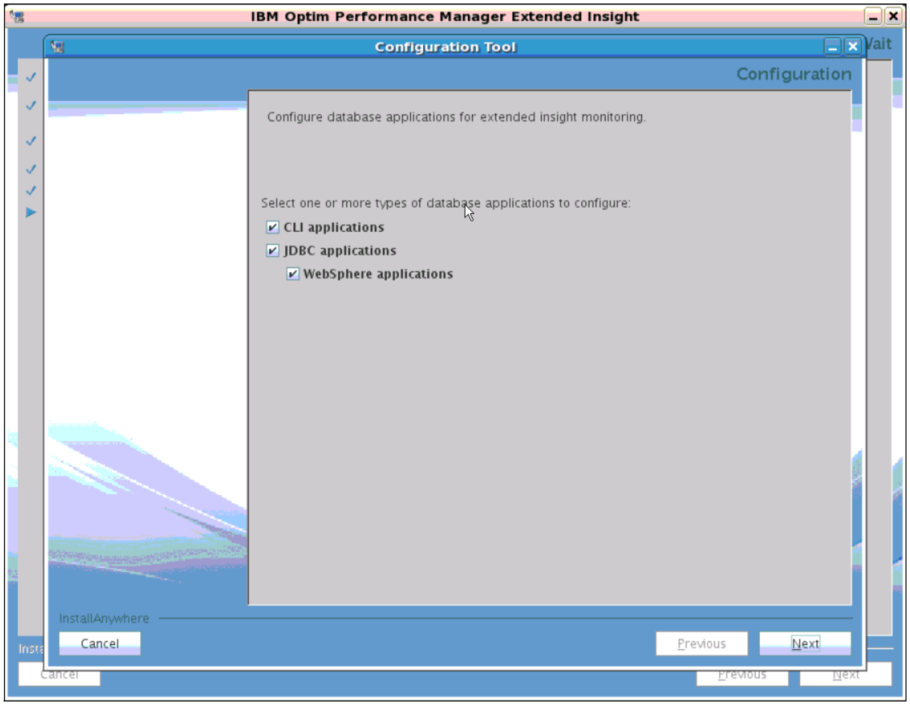


Figure 3-68 Configuration

2. In the Configuration Tool panel (Figure 3-69 on page 135), enter the host name or IP address of the Optim Performance Manager and the port number of the Extended Insight controller, that you already specified during activation of Extended Insight, see 3.2.3, “Activating the Optim Performance Manager Extended Insight license” on page 81. If you don’t have the port number by hand then you can obtain it from the `pdq.cmx.controllerURL` parameter in the `pdq.properties` file which is located on the Optim Performance Manager machine at:

   – UNIX:

   ```
   <OPM_Working_Directory>/opm/v4/<instancename>/pdq.properties
   ```

   Windows:

   ```
   <OPM_Working_Directory>\RepositoryServer\instances\<instancename>\pdq.properties
   ```

   The controller port is used by the Extended Insight client to establish communication between the Extended Insight client and Optim Performance Manager. It is saved together with the host name or IP address of Optim Performance Manager in the `pdq.properties` file of Extended Insight client.

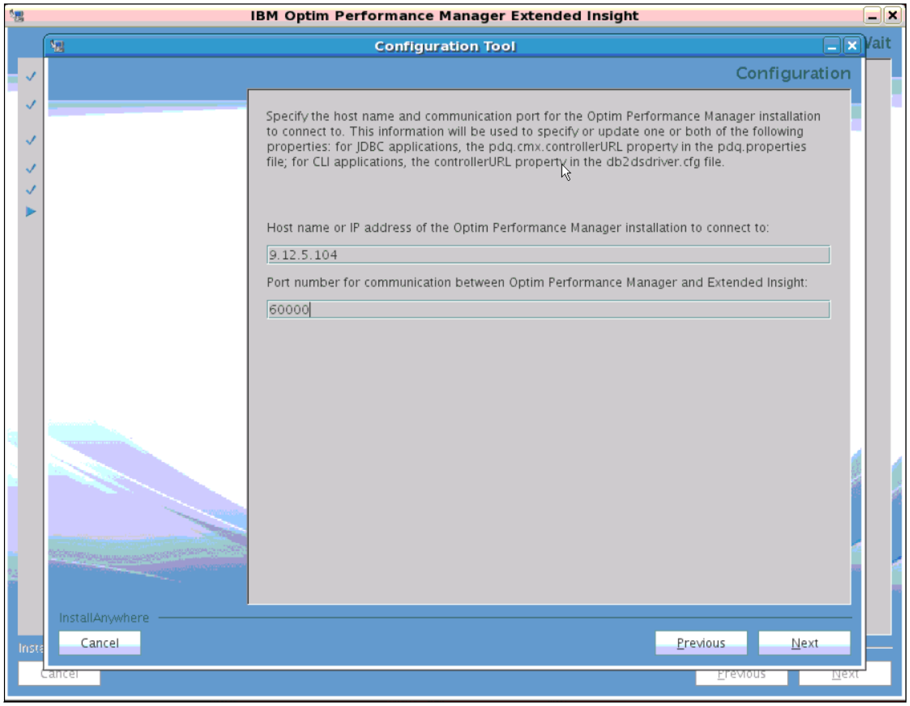


Figure 3-69 Configuration Tool wizard — Host name or IP address and port number

3. If you are configuring Extended Insight for CLI applications, in Configuration File panel (Figure 3-70 on page 136), you must identify the CLI driver that is being used by the application that you want to monitor. By specifying the correct CLI driver, you can identify which `db2dsdriver.cfg` file to configure. The configuration will not succeed if you do not configure the correct `db2dsdriver.cfg` file.

   The configuration tool attempts to find the `db2dsdriver.cfg` files for you. If the configuration tool cannot find the one that you need to configure, you can enter the location for it. For more information about the contents and location of the `db2dsdriver.cfg` file, refer to the Information Center at

   http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.swg.im.dbclient.config.doc/doc/c0054555.html

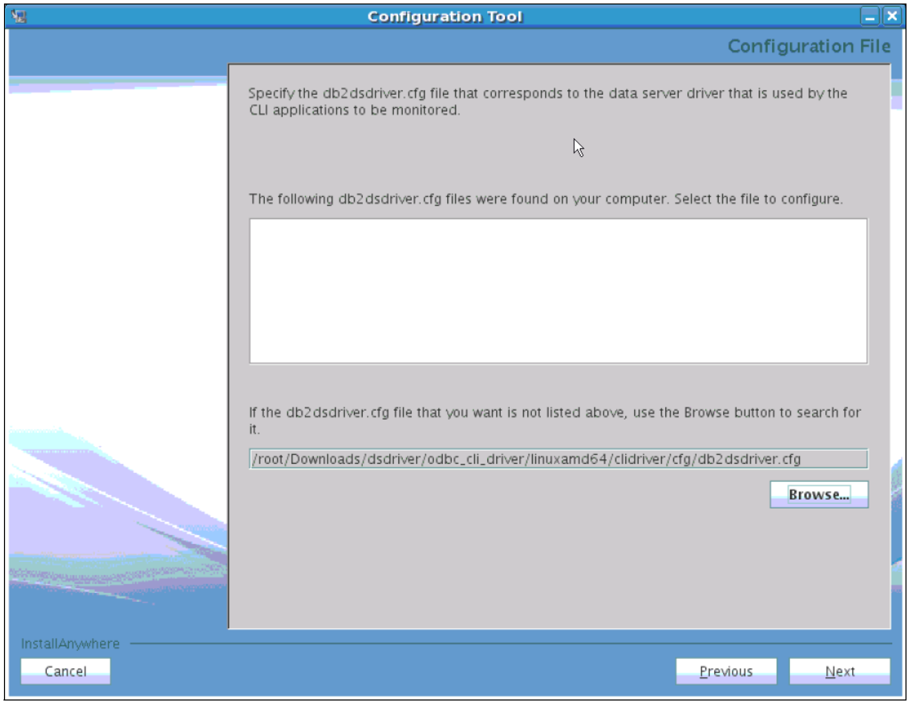


Figure 3-70 Configuration File panel

Once the `db2dsdriver.cfg` file is specified, The configuration tool modifies the `db2dsdriver.cfg` file and completes the CLI configuration. More information about what is changed and the contents of this configuration file before and after the configuration are well explained in the developerWorks® tutorial at

`http://www.ibm.com/developerworks/data/tutorials/dm-1010optimextendedinsight/index.html`

The next step is configuring Extended Insight for WAS applications.

4. In this step you configure Extended Insight for WebSphere Application Server applications. You must provide credentials to connect to the WebSphere Application Server. This information is required to update the WebSphere JDBC provider so that the WebSphere applications can be monitored. The SOAP port is required, the other fields are required only if you have security turned on for your copy of WebSphere Application server. The SOAP port is generally the default port 8880. If you do not use the default ports, check the `portdef.props` file in your WebSphere Application Server profile for the port number. See Figure 3-71 on page 137.

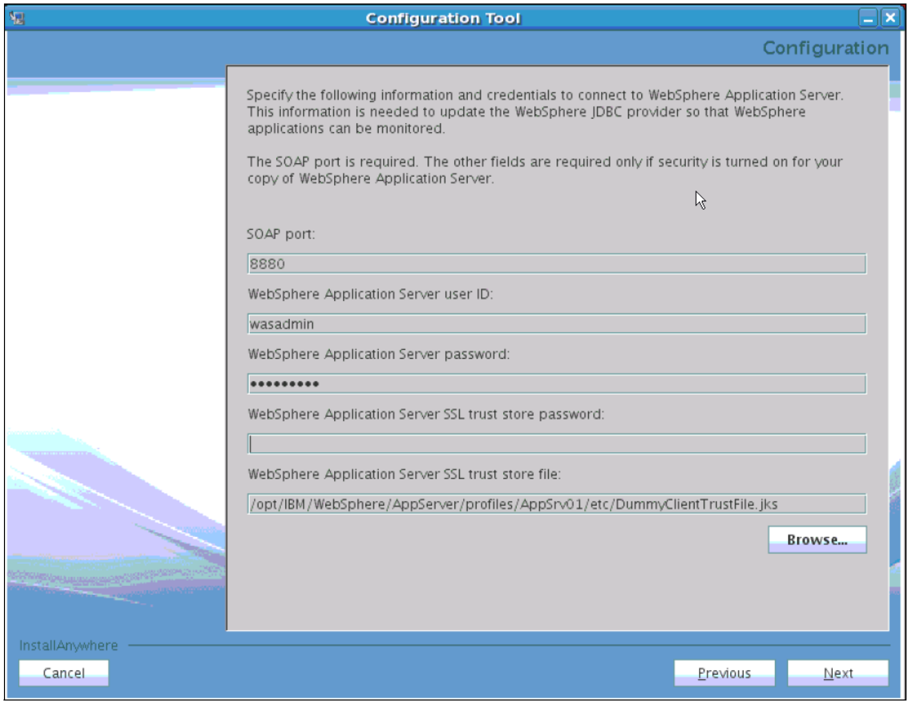


Figure 3-71 WebSphere configuration

5. Specify the JDBC providers that correspond to the databases to collect extended insight data for (Figure 3-72 on page 138). The classpath for the JDBC provider will be updated so that Extended Insight can monitor your applications.

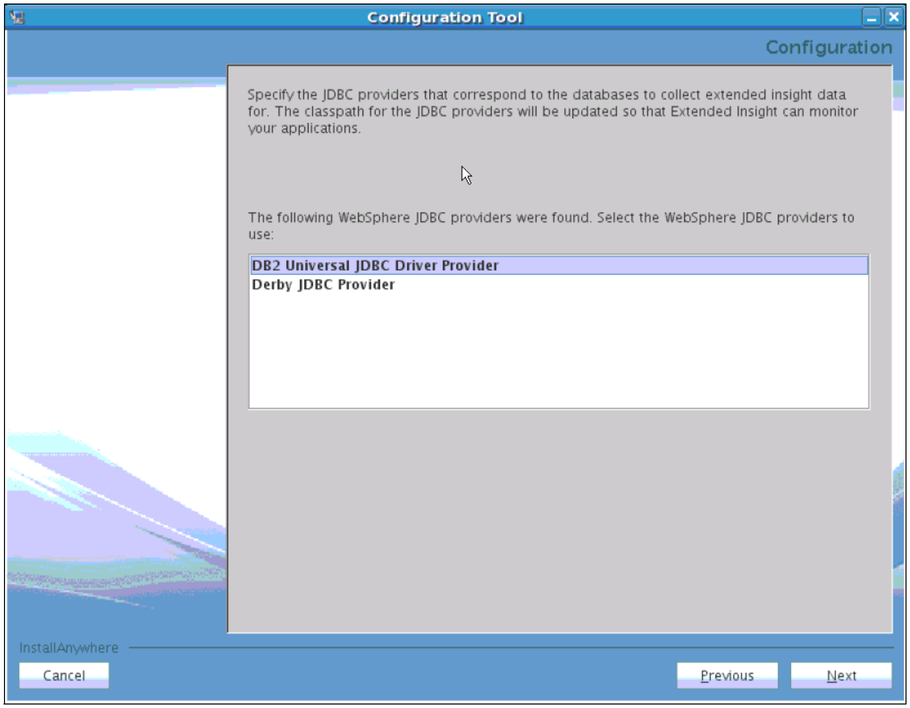


Figure 3-72 Select JDBC provider

6. This step is optional. If you want to verify that the database is properly configured for monitoring, the configuration tool can do that. Enter information for the database that you want to monitor. See Figure 3-73 on page 139.

   For CLI applications, note that even though this information is requested at this point, it is not added into the `db2dsdriver.cfg` file. You have to manually add this information while setting up to run the workload. More information about what to add manually in the configuration file is well explained in Step 2 of *Run a workload to validate that data is being collected* in the developerWorks tutorial at

   `http://www.ibm.com/developerworks/data/tutorials/dm-1010optimextendedinsight/index.html`

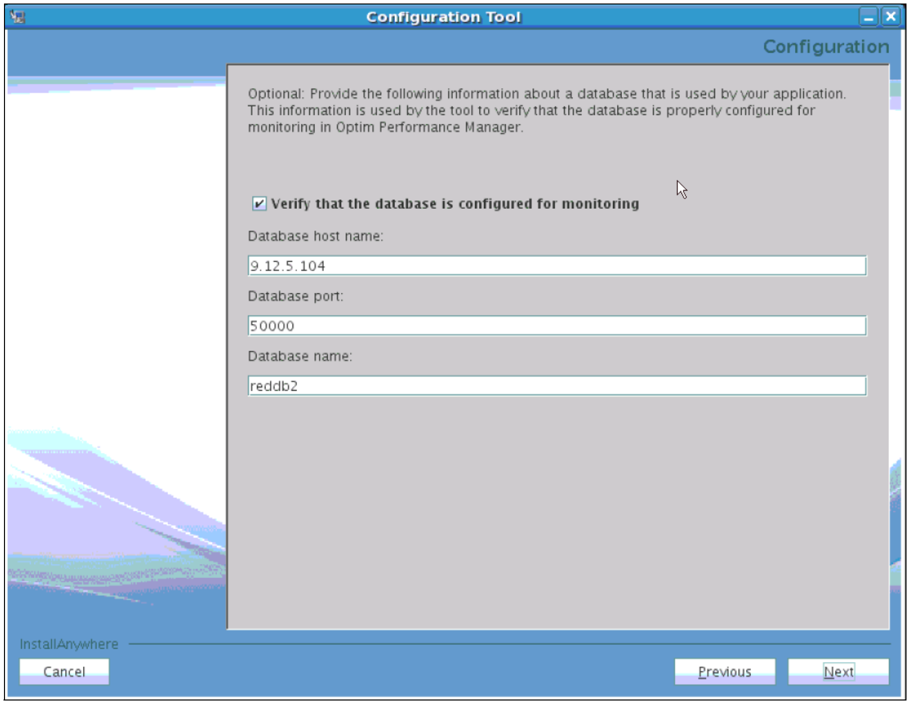


Figure 3-73 Verify if the database is configured for Monitoring

If the database is not added for monitoring or the information that you added in this panel for verification is not correct, you will see an error as shown in Figure 3-74 on page 140.

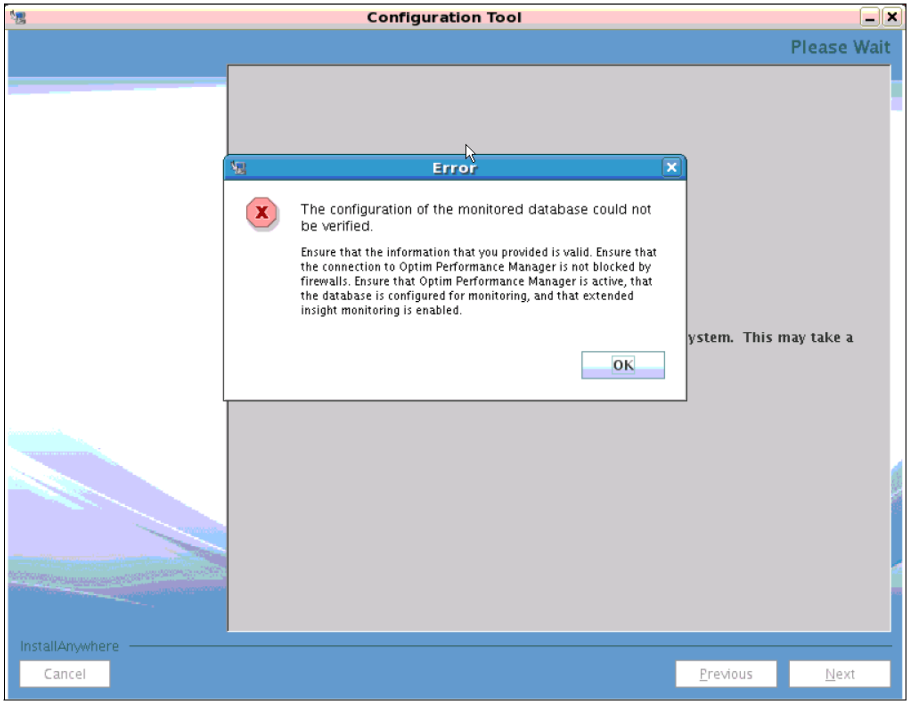


Figure 3-74 Error while verifying if the database is correctly configured for monitoring

The error suggests to verify the following:

- Ensure that you have provided the correct database host name, database port, and database name of the database that you are monitoring.
- Ensure that the controller port number is not blocked.
- Ensure that your Optim Performance Manager server is up and running. On UNIX and Linux, you can use the `pestatus` command from the `<OPM>/RepositoryServer/bin` directory to check whether Optim Performance Manager is running. On Windows, check the Optim Performance Manager service using the Control panel.
- Ensure that database is configured for monitoring. You can check the Manage Database Connections Dashboard to see the monitoring status of this database (Figure 3-75).

Name ▲	Data Server Type	Database Name	Host Name	Port Number	
reddb2	DB2 for Linux, UNIX, and Windows(9.7.2)	REDDB2	TREASUREISLAND.USCA.IBM.C(	50000	Configured - Enabled

Figure 3-75 Manage database connections - verify the monitoring status

– Ensure that you have enabled Extended Insight profile.

7. Review the information on the summary panel (Figure 3-76 on page 141).

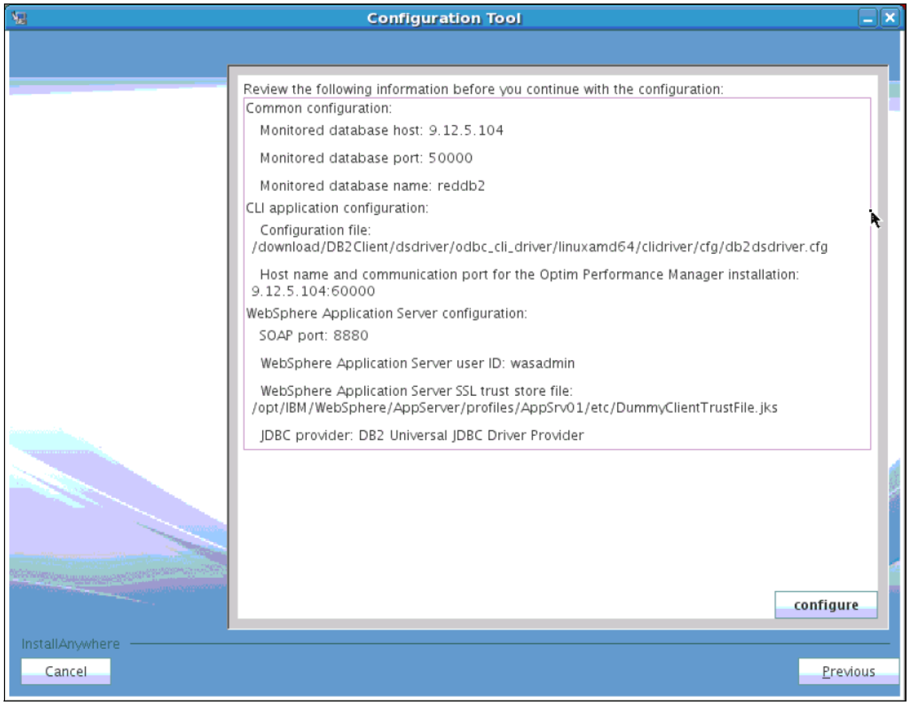


Figure 3-76 Summary Screen for Configuration

Review the information before you continue with the configuration and Click **configure**. If you decide to change any information you can go back to the previous screens using the Previous button.

It will take a few minutes to configure. Once the configuration is complete you will see the Finish screen as shown in Figure 3-77 on page 142.

The Finish screen indicates that the tool has configured JDBC, CLI and WebSphere applications successfully.

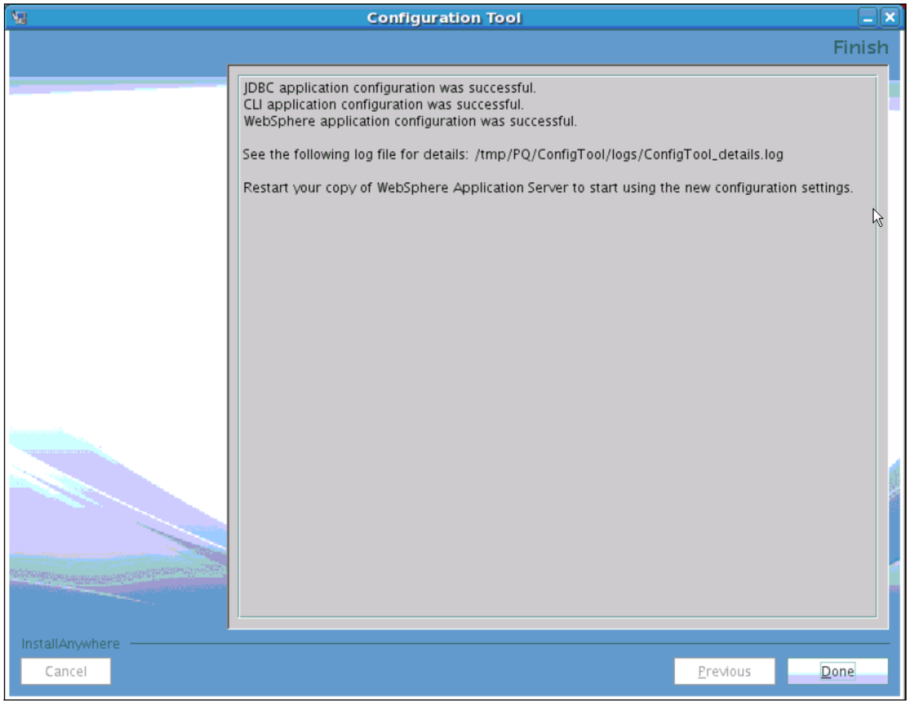


Figure 3-77 Finish Screen

8. If you are using WebSphere Application Server version 6, an additional configuration step is required. You must add a custom property `enableEndToEndMonitoringFeature` and set it to true in the WebSphere Application Server version 6.

   Figure 3-78 on page 143 shows the WebSphere Application Server console.

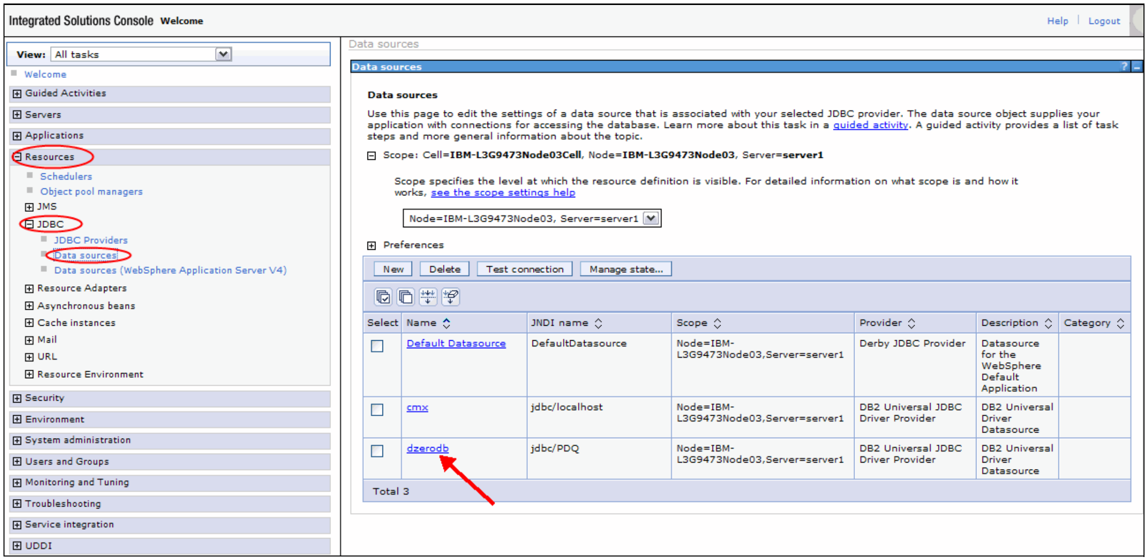


Figure 3-78 WebSphere Application Server 6 console - Select the datasource

Click **Resources** → **JDBC** → **Data sources**. Select the datasource that you want to configure for this new property. In the datasource details panel (Figure 3-79 on page 143), select **Custom properties**.

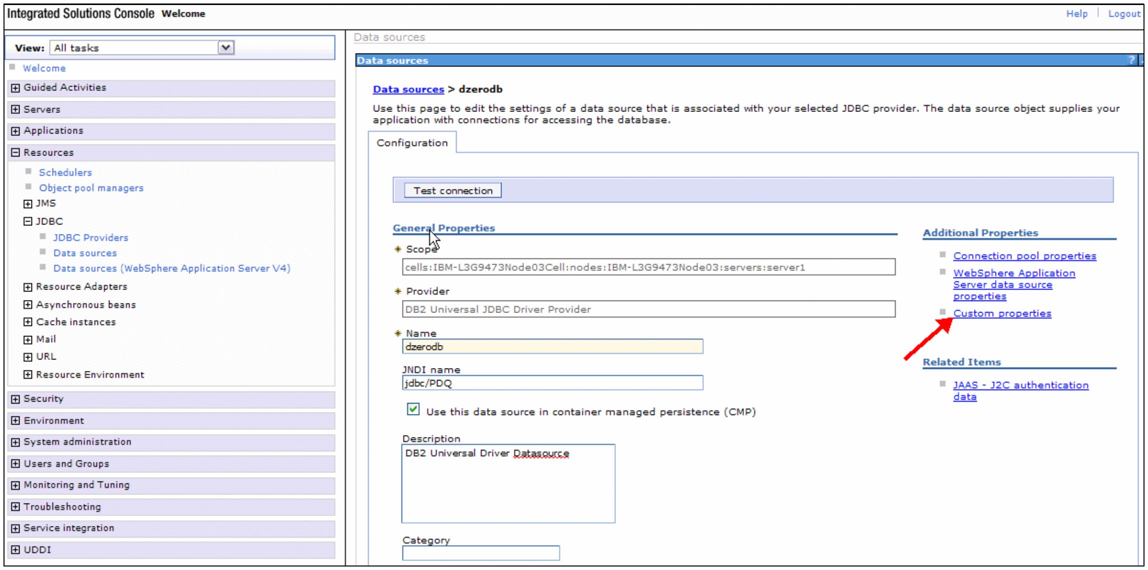


Figure 3-79 WebSphere Application Server 6 console - Custom properties

In the References panel (Figure 3-81 on page 145), select **New** to create a new property.

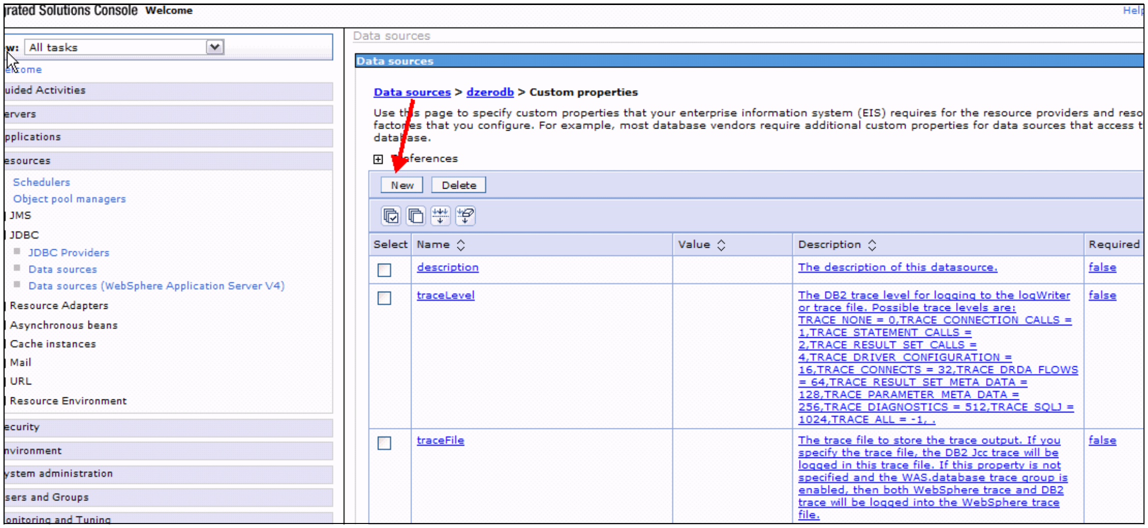


Figure 3-80 WebSphere Application Server 6 - Select New

Enter the name of the property as `enableEndToEndMonitoringFeature`, set the value to true, and click **Apply**. You have to restart the WebSphere Application Server for this new configuration to take affect.

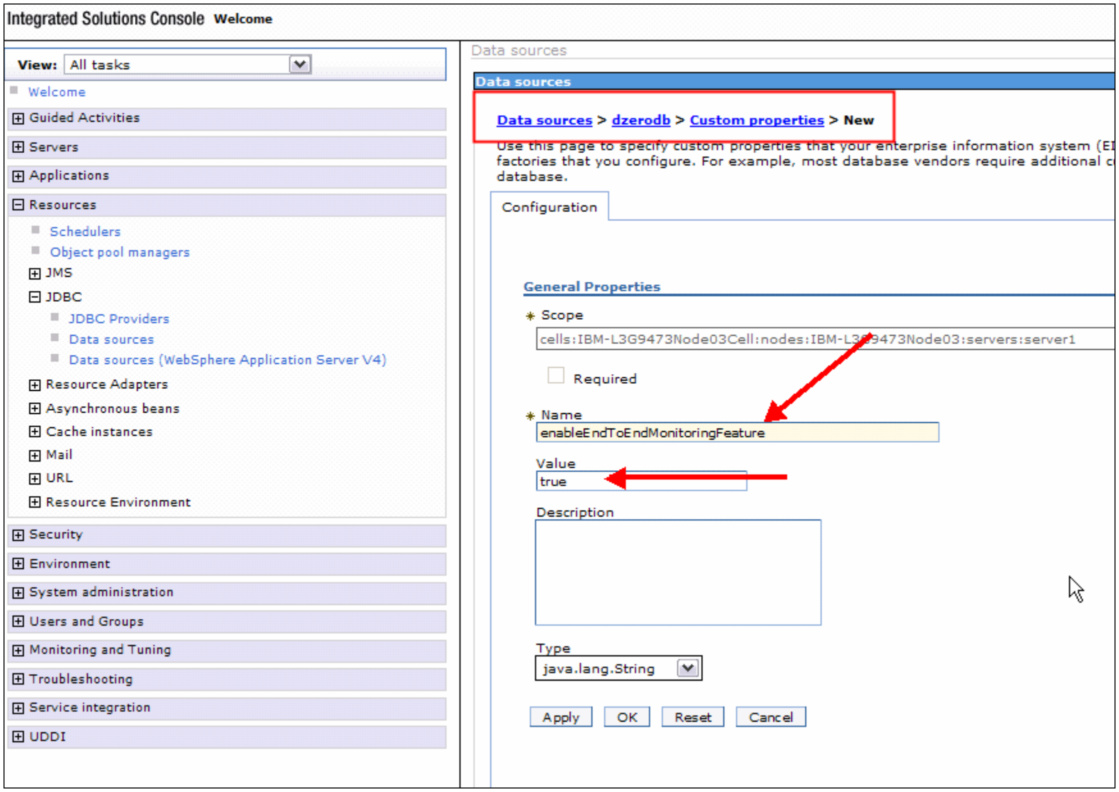


Figure 3-81 WebSphere Application Server 6 - New Custom property

Optim Performance Manager Extended Insight is now installed and configured. You can run the Extended Insight sample program against a monitored database to validate your configuration and see how the monitoring works. For information about how to run the sample program, refer to:

- For JDBC Application: See the readme file in the `EI_installation_dir/samples` directory.
- For CLI Application: See the tutorial, which has a sample application and steps to run the application at

  http://www.ibm.com/developerworks/data/tutorials/dm-1010optimextendedinsight/index.html

4

Getting to know Optim Performance Manager

Optim Performance Manager delivers new paradigm in terms of how it is used to monitor and manage database and database application performance issues. It enables:

- Top-down database performance management: An ability to start database performance management at the application level (top level) and trace its performance data as it traverses through database client and network to the database server.
- Proactive database performance management: An ability to integrate with DB2 Workload Management feature, which helps to prevent performance issues by creating a more predictable database server environment by assigning resources to various workloads according to their priorities.
- Bottom-up database performance management: The traditional way of database performance management, which starts at the database server level (bottom component)

It uses web interface and guided workflows to navigate user through the series of performance dashboards. Each dashboard surfaces collected metrics for a different database performance category, such as: memory, I/O, locking, SQL.

Optim Performance Manager provides predefined report templates that you can use to generate reports for trend detection, proactive monitoring, establishing baselines, and more.

New in Optim Performance Manager is the administrative and management tooling for DB2 workload management (WLM). This new tooling provides the ability to proactively administer, manage, and monitor the workload in context to their workload management settings.

Integration between Optim Performance Manager Extended Edition and Tivoli Composite Application Manager provides a consolidated view of the business transactions across the enterprise while providing comprehensive detail to help diagnose database-specific areas.

If you install the optional DB2 Performance Expert Client component, you can do a smoother migration from DB2 Performance Expert to Optim Performance Manager.

In this chapter we describe individual product dashboards and reports and discuss how they can be used to identify, diagnose, prevent, and solve database performance problems. The integration of Optim Performance Manager with the Tivoli Composite Application Manager as well as Workload Manager tool is described in separate chapters.

4.1 Health summary

This section describes several Optim Performance Manager dashboards, which give information about the overall health of the monitored database.

4.1.1 Health summary page

You might start by viewing the Health Summary page (**Task Manager** → **Health Summary**) to see an overview of the health of all monitored databases. From the Health Summary page (Figure 4-1), you can determine which databases have problems, and you can drill down for more details.

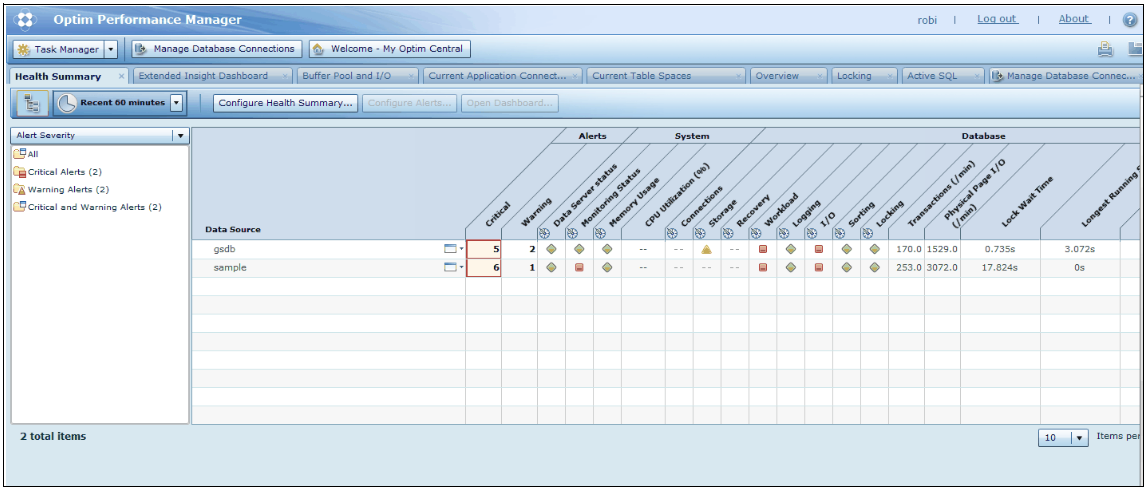


Figure 4-1 Health summary page

The Health Summary shows an overview of the severity and number of alerts for your data sources. The types of alerts that the Health Summary displays depends on the type of data source and the alerts that are configured for each data source.

Use the group pane (Figure 4-2) to select the data sources to view alert severity, host, and port from the default groups. If the custom group feature is available for your product, you can also create custom groups that are shared with all users of the web console.

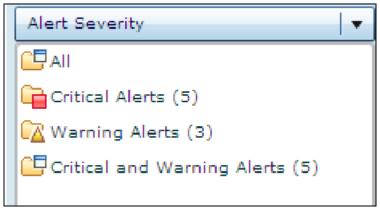


Figure 4-2 Health summary page group pane

Use the column controls (Figure 4-3 on page 150) to control which columns to display in the grid, move columns around, filter and sort the data in the grid as needed. You can also click the column headings to sort the data in the grid according to any column value.

Figure 4-3 Health summary page column controls

Use the time control (Figure 4-4 on page 150) to set the time period for the health data that is displayed. You can view recent data that is refreshed at configurable intervals, or view historical data for a specific time period. For recent data, the refresh rate controls the rate at which the Health Summary page is updated. For historic data, specify the history timeframe in the timezone of your web browser.The alert data is collected at a rate that you can set individually for each monitored database.

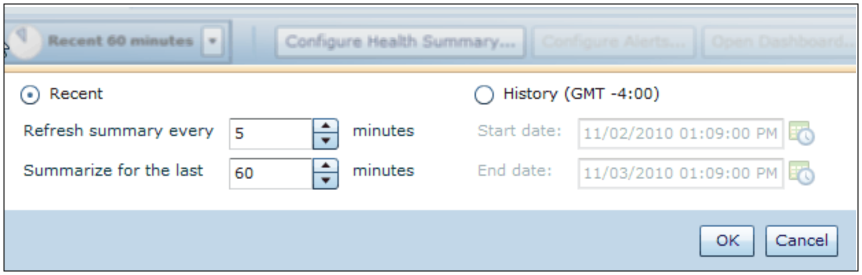


Figure 4-4 Health summary page time control

Each column in the Health Summary represents an alert category contributed by a data source that shares the Health Summary interface. The alert icons (Figure 4-5 on page 151) in the grid reflect alert severity for each alert category and data source.

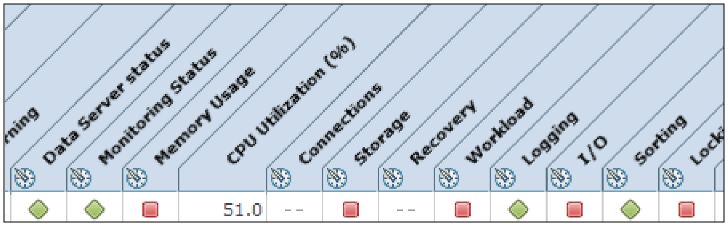


Figure 4-5 Alert categories and alert icons

Click an alert icon in the grid to see a list of all of the alerts of that specific category for that data source. You can then select a specific alert to troubleshoot.

To better understand or resolve the cause of an alert, open the appropriate dashboard by clicking **Open Dashboard**.

4.1.2 Overview dashboard

After the initial review of the health of monitored data sources, you can drill down from the Health summary page into the overview dashboard for the selected data source. See Figure 4-6 on page 151.

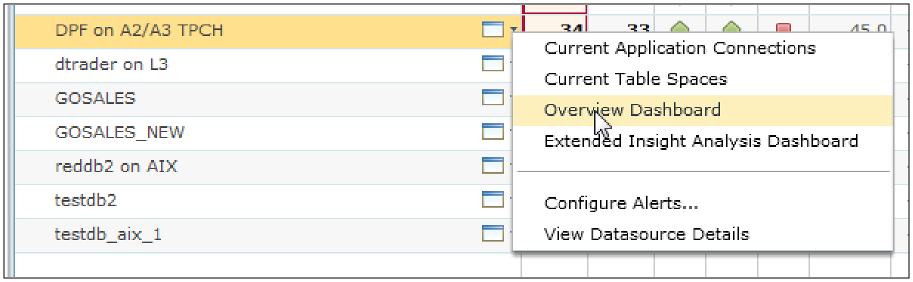


Figure 4-6 Overview dashboard invocation

Use the Overview dashboard (Figure 4-7 on page 152) to view the key performance indicators for multiple problems areas for a database, such as, workload, sorting, caching, locking, I/O and disk space.

Use the time slider and time controls to control the time period for the performance data that is shown on the dashboard.

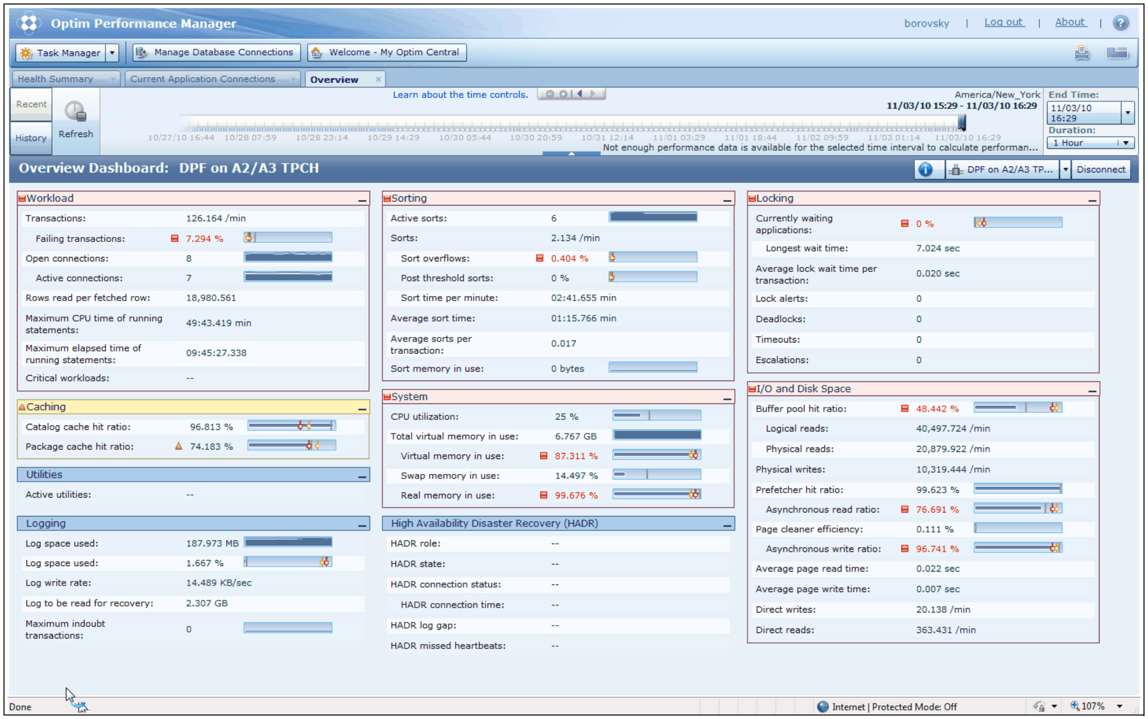


Figure 4-7 Overview dashboard

You can see problem areas for a database quickly when you open the Overview dashboard. If a section heading displays a yellow warning icon or red critical icon, an alert threshold was crossed. The small graphs show where the alert thresholds are set. Figure 4-8 on page 153 shows the I/O and Disk Space section in the Overview dashboard.

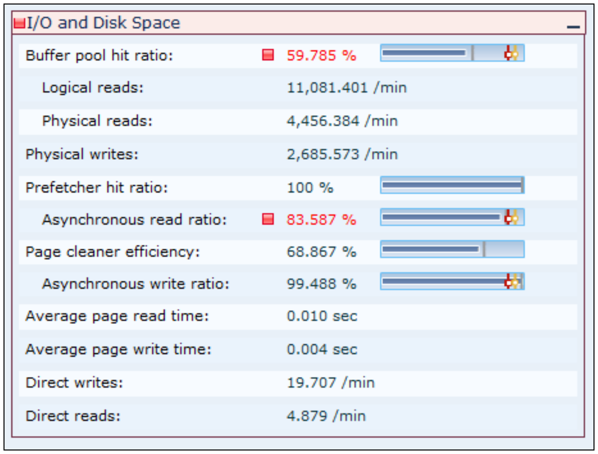


Figure 4-8 I/O and Disk Space section

You can obtain more information for each key performance indicator on the dashboard in the following ways:

- Read about the value for the indicator

  Read the hover help to determine what type of value is displayed for each key performance indicator. For example, the value might be the average, current, most recent, or maximum value for the time period that you selected with the time slider.

  Figure 4-9 on page 153 shows a key performance indicator hover help.

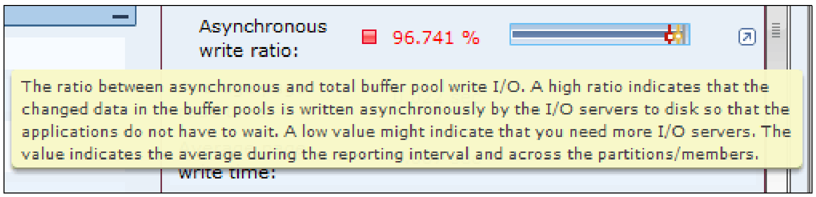


Figure 4-9 Key performance indicator hover help

- Drill down to a problem-oriented dashboard

  Double-click a value for a performance indicator to drill down to the more detailed problem-oriented dashboards, where you can review statistics that are relevant to a specific problem.

If an alert is indicated but the average value for the time period that you selected does not exceed the threshold, then one or more data points exceeded the threshold. Click the alert icon to see when the alerts occurred. You can also use the time slider and time controls to shorten the reporting interval to identify the data point that triggered the alert.

Identifying problems by using the small graphs

Many performance indicators have a small graph. You can use these graphs to determine how your system is behaving over time and identify bottlenecks or peaks. Performance indicators might have a bar graph or an area graph.

Bar graph

The blue horizontal bar indicates the average value during the reporting interval and across the partitions. For a partitioned database, the grey vertical bar indicates the maximum or minimum value that was reached during the reporting interval for any partition. The order of critical alert and warning alert markers depend on whether a high or low value indicates a problem for the type of metric.

Figure 4-10 on page 154 shows a small bar graph.

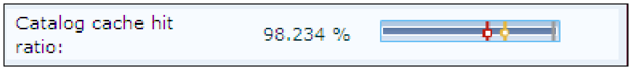


Figure 4-10 Bar graph

Area graph

The blue area indicates the changes in value during the reporting interval. For partitioned database, in most cases, indicates the change in the average value across the partitions. For partitioned database, the gray line indicates the highest value for any partition at specific times during the reporting interval.

Figure 4-11 on page 154 shows a small area graph.

Figure 4-11 Small area graph

You can interact with the small graphs in the following ways:

- Move the cursor over a bar graph to see an enlarged view of the small graph. This enlarged view includes the threshold and metric values for the time period that you selected.

  Figure 4-12 shows an enlarged small graph.

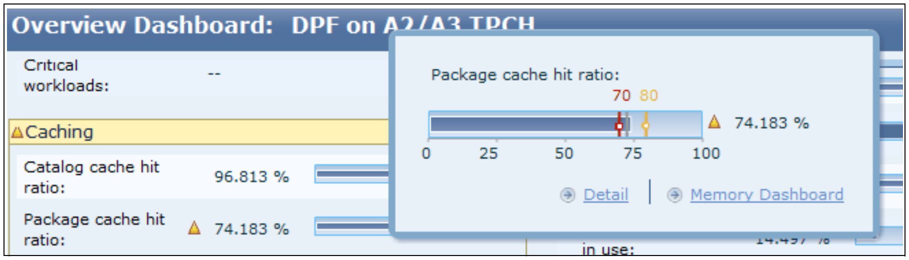


Figure 4-12 Enlarged small graph

- Click a small graph to see a detailed graph window. In the detailed graph window, you can view details for the performance indicator and, where applicable, set values for warning and critical alerts. From a detailed graph window, you can also click on a link to open the dashboard for the performance indicator.

  Figure 4-13 on page 155 shows a detailed view of a small graph.

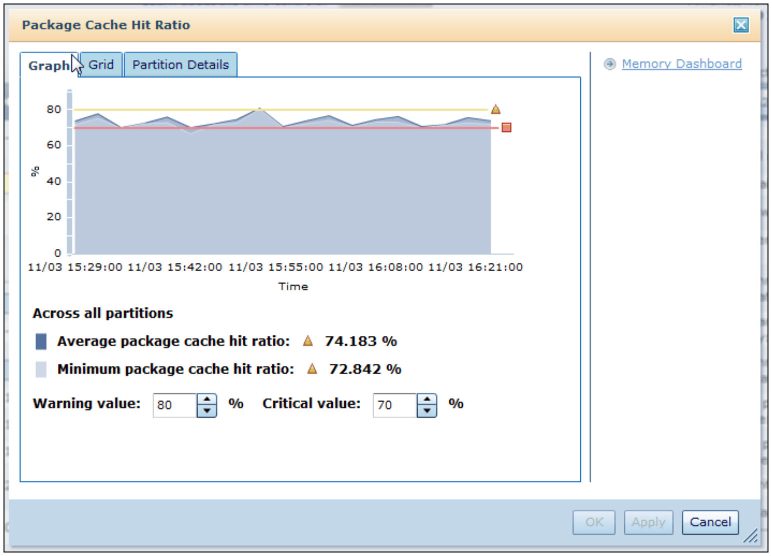


Figure 4-13 Detailed view of a small graph

- Double-click a small graph to open the dashboard for the performance indicator.

Time slider and time controls

Use the time slider and time controls (Figure 4-14) to control the time period for the performance data that is displayed on the dashboard.

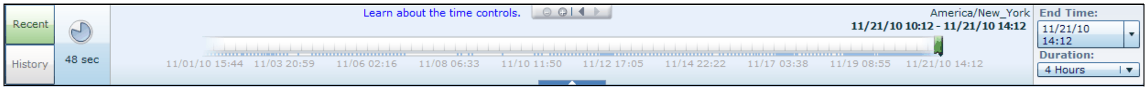


Figure 4-14 Time slider and time controls

Time line

The time line (Figure 4-15 on page 156) is initially expanded to show the length of time that performance data has been collected. For example, if performance data has been collected for six weeks, then the leftmost time stamp is six weeks ago. If you use the zoom in and zoom out controls at the top of the time line, you can change the span of the time line. Changing the span can make the time slider easier to manipulate. However, changing the span of the time line does not change the data on the dashboard. Use the time slider control to control the data that is shown on the dashboard.

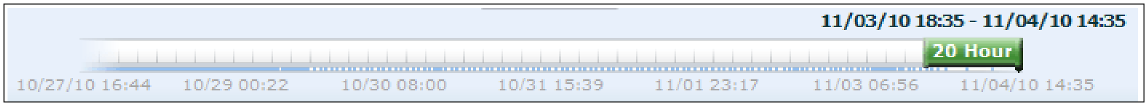


Figure 4-15 Time line

Time slider

This control shows the time interval for the data that is displayed on the dashboard. You can move this control to show different data on the dashboard. For example, you can slide the control to the left to show older data. You can control the amount of data that is shown on the dashboard by increasing or decreasing the time interval with the *Duration* control.

The time slider (

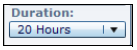
) control helps you isolate and analyze what was taking place on the database at a specific point in time. For example, by sliding the time slider, you can remove distracting data points that might not be related to the performance problem that you are investigating.

A dashboard shows performance data only if the position of the time slider on the time line contains at least two data points. Otherwise, you should increase the time interval for the time slider or move the time slider to another position. An inactive database can result in gaps in the collection of performance data.

Zoom and data point controls

Use the zoom controls () to change how much of the time line is shown. For example, if the time line initially shows 60 days of data, you can zoom in so that the time line shows only 5 days of data. In this way, you can more easily manipulate the position of the time slider on the time line. Use the data point controls to move the time slider from one data point to the next. The blue lines at the bottom of the time slider indicate the points where data was collected. You can move from one data point to the next data point or to the previous data point.

Duration control

Click the Duration control (

) to control how much data is shown at one time on the dashboard. The duration is reflected in the time slider. For example, if the time slider indicates that two hours of data is shown on the dashboard, you can use this button to change the duration to one hour.

End Time control

Click End Time (Figure 4-16) to specify the end time for the time slider. For example, if you want to analyze performance data for a time period that ended at 10:00 a.m. the previous day, you can click this button and specify that day and time. The time slider will move to show you the requested time frame, and the data in the dashboard will reflect that time frame

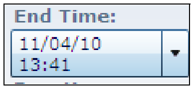
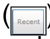


Figure 4-16 End time control

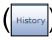
Clock button

The Clock button () indicates the time that remains until the next refresh interval. Refresh interval is always 60 seconds. The content however is refreshed based on the sampling rate that was set when the database was configured for monitoring. You can click the Clock button to enable or disable the automatic refreshing of the data on the dashboard. Automatic refresh is useful for viewing the latest performance data.

Recent button

In Recent mode, the latest performance data is displayed and the time slider is at the right end of the time line. Click the Recent button () to ensure that you are seeing the most recent data. Timeframe of the displayed performance data is based on the time zone of the monitored database.

History button

In History mode (

), you can position the time slider to display the performance data for a previous time and date. The length of the time line indicates the amount of historical data that is available. If more data should be available, use the Zoom out button until you can see the whole time line. The retention period, which is set when the database was configured for monitoring, controls the amount of historical data that is available. Timeframe of the displayed performance data is based on the time zone of the monitored database.

4.1.3 Current application connections

From the Health summary page you can continue to review the health of the monitored database by drilling down into current application connections page. This page displays the applications which are currently connected to a database, and indicates the status of the connection by idle time. The page also provides information about the connection, such as rows read and rows written.

Depending on the connection status you can choose to force the application. For example, to allow for system maintenance or to enhance system performance, you can force an application connection that has been idle for a long time.

For databases that have many connected applications, you can filter your view by criteria such as name, status, and idle time. You can select individual connection and click **View SQL details** to review the SQL statement that this connection is currently executing.

Figure 4-17 on page 158 shows a Current Application Connections view. The current application connections page has been integrated into Optim Performance Manager from Data Studio Health Monitor.

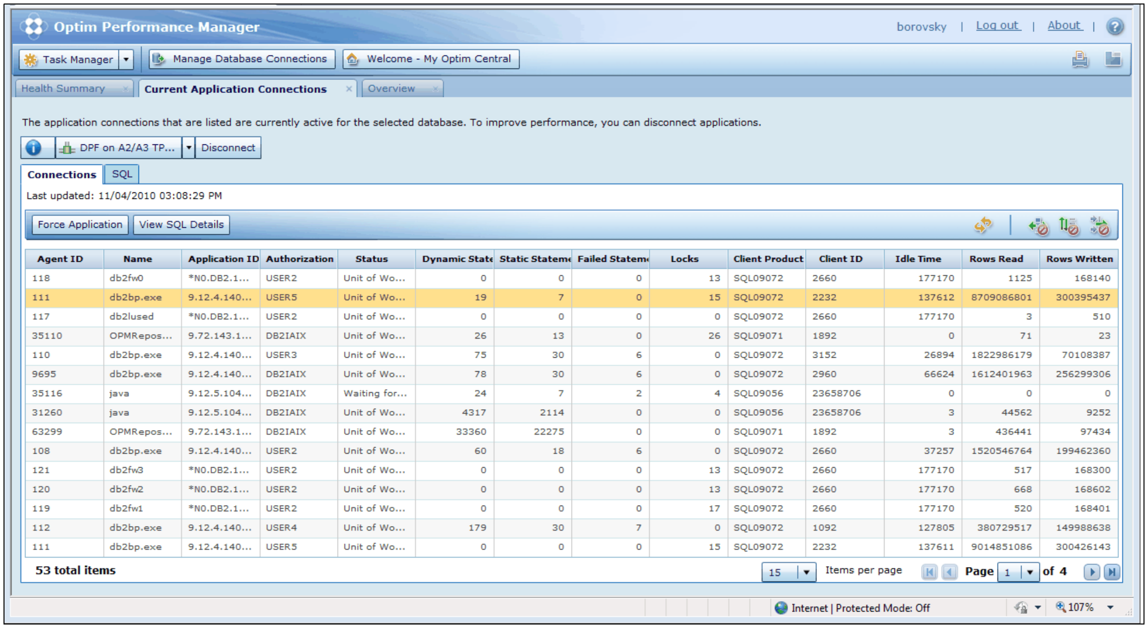


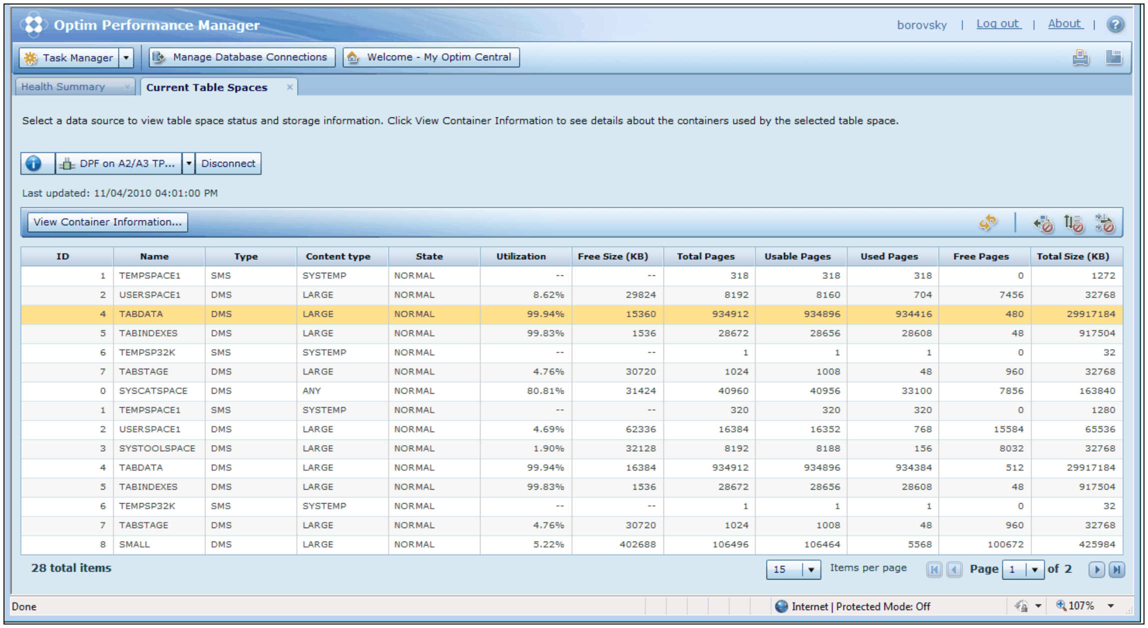
Figure 4-17 Current Application Connections

4.1.4 Current table spaces

To finalize the initial health check of the monitored database, you can drill down into Current Table Spaces page (Figure 4-18 on page 159) from the heath summary page. This page displays information about the current status of table spaces and table space containers for the selected database.

For databases that contain many table spaces you can filter your view by criteria such as name, state, and utilization.

The Current Table Spaces page has been integrated into Optim Performance Manager from Data Studio Health Monitor.



ID	Name	Type	Content type	State	Utilization	Free Size (KB)	Total Pages	Usable Pages	Used Pages	Free Pages	Total Size (KB)
1	TEMPSPACE1	SMS	SYSTEMP	NORMAL	--	--	318	318	318	0	1272
2	USERSPACE1	DMS	LARGE	NORMAL	8.62%	29824	8192	8160	704	7456	32768
4	TABDATA	DMS	LARGE	NORMAL	99.94%	15360	934912	934896	934416	480	29917184
5	TABINDEXES	DMS	LARGE	NORMAL	99.83%	1536	28672	28656	28608	48	917504
6	TEMPSP32K	SMS	SYSTEMP	NORMAL	--	--	1	1	1	0	32
7	TABSTAGE	DMS	LARGE	NORMAL	4.76%	30720	1024	1008	48	960	32768
0	SYSCATSPACE	DMS	ANY	NORMAL	80.81%	31424	40960	40956	33100	7856	163840
1	TEMPSPACE1	SMS	SYSTEMP	NORMAL	--	--	320	320	320	0	1280
2	USERSPACE1	DMS	LARGE	NORMAL	4.69%	62336	16384	16352	768	15584	65536
3	SYSTOOLSPACE	DMS	LARGE	NORMAL	1.90%	32128	8192	8188	156	8032	32768
4	TABDATA	DMS	LARGE	NORMAL	99.94%	16384	934912	934896	934384	512	29917184
5	TABINDEXES	DMS	LARGE	NORMAL	99.83%	1536	28672	28656	28608	48	917504
6	TEMPSP32K	SMS	SYSTEMP	NORMAL	--	--	1	1	1	0	32
7	TABSTAGE	DMS	LARGE	NORMAL	4.76%	30720	1024	1008	48	960	32768
8	SMALL	DMS	LARGE	NORMAL	5.22%	402688	106496	106464	5568	100672	425984

Figure 4-18 Current Table Spaces

4.2 Alerts

To understand database performance, you monitor a set of key performance indicators for your connected DB2 data server databases. Potential areas for concern are identified by critical alerts (red squares) and warnings (yellow triangles) on the performance manager dashboards and windows.

The Health Summary and Alerts list provide an overview of alerts across all your monitored databases. The Health Summary shows active alerts by category. The

Alerts list also provides information about active alerts, but also provides information about all the alerts that occurred during the monitored period.

4.2.1 Alert list

To display the Alert List page (Figure 4-19 on page 160), from the OPM console, click **Task Manager** → **Alert List**.

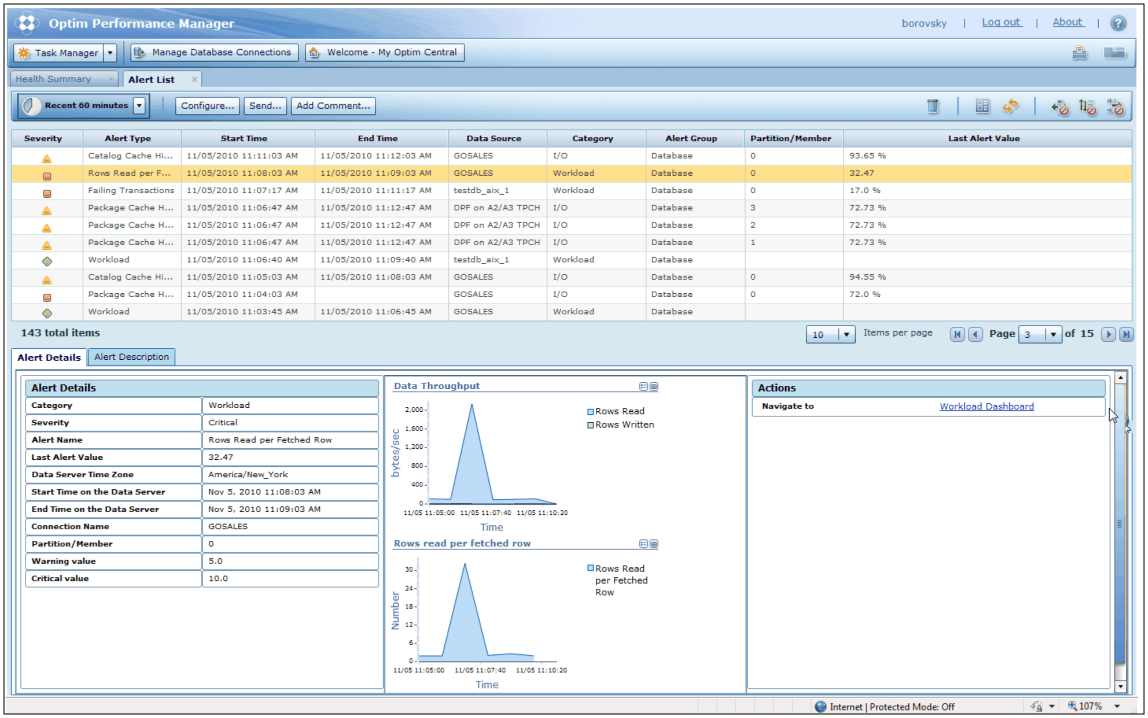


Figure 4-19 Alert List

The types of alerts that the Alert List displays depends on the type of data source and the alerts that are configured for each data source.

Alerts are enabled with default threshold values, which are set when you configure a database for monitoring. The thresholds determine when an alert is triggered. The key performance indicators are checked periodically by using a default sampling rate. If a threshold is breached, then an alert is generated and an indicator appears on the Health Summary, the Alerts list, and the associated dashboard. You can customize the alert thresholds as needed for your environment.

If you want to share details about an alert with system administrators or other users, you can send the alert in an email or add comments to the alert.

- To send a link to an alert in an email, select the alert and click **Send**. The email contains a link to the alert in Health Summary.

> **Note:** Email communication requires that the Email Service is configured with a valid SMTP host name and port.

- To add comments to an alert, select the alert and click **Add Comment**. Comments added to alerts are visible to all users.

To view detailed information about an alert, click the alert in the grid and view the details in the bottom of the page. To view additional details and suggestions for troubleshooting the alert, click **View details** ( ).

Figure 4-20 shows a sample alert detail panel.

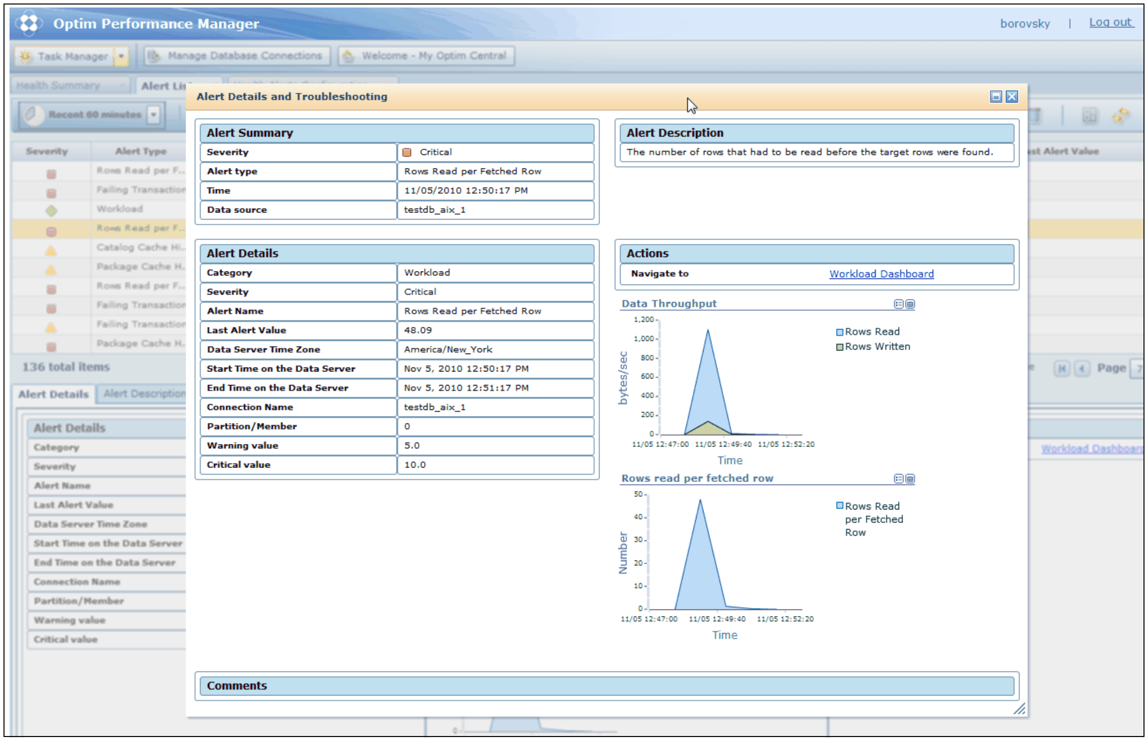


Figure 4-20 Alert details panel

Optim Performance Manager alerts could be classified into:

- Health alerts

  Health alerts are triggered by the Data Studio Health Monitor component of Optim Performance Manager. They provide database health status information, such as, data server status, storage space state and utilization, HADR state. To configure health alerts, select **Task Manager** → **Health Alerts Configuration**. Figure 4-21 on page 162 shows a sample Health Alerts Configuration panel.

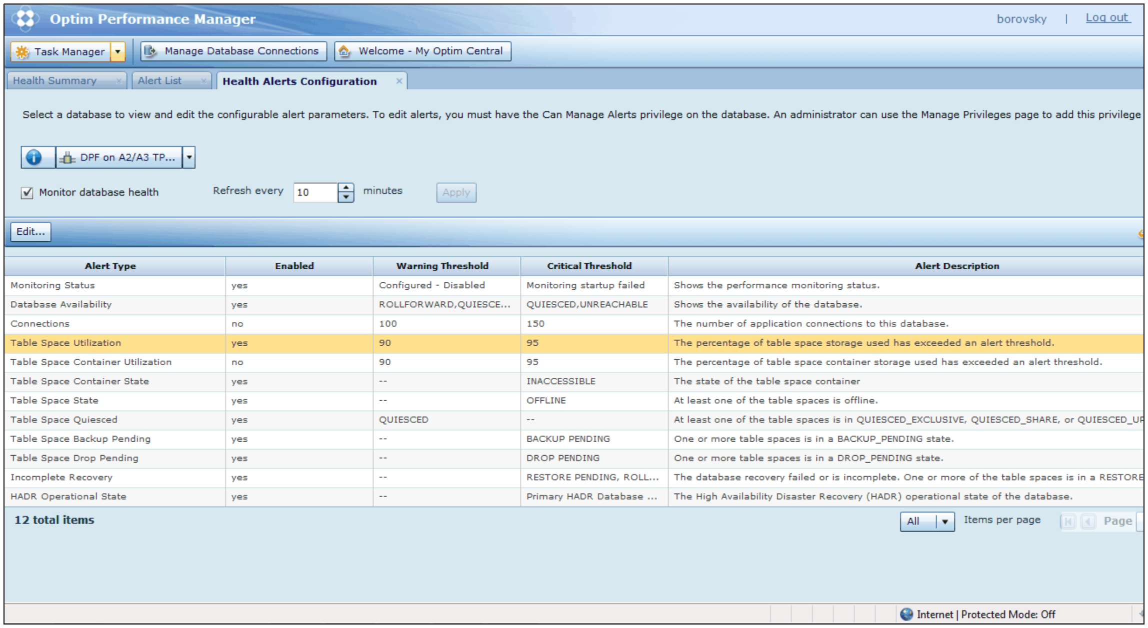


Figure 4-21 Health Alerts Configuration

- Performance alerts

  Performance alerts represent database key performance indicators from different categories such as memory, I/O, locking, logging, and SQL. Example of performance alerts are Rows read per fetched row, Package cache hit ratio, Currently waiting applications, and Sort overflows.

  To configure performance alerts, select **Task Manager** → **Performance Alert Configuration**. Figure 4-22 on page 163 shows a sample Performance Alert Configuration panel.

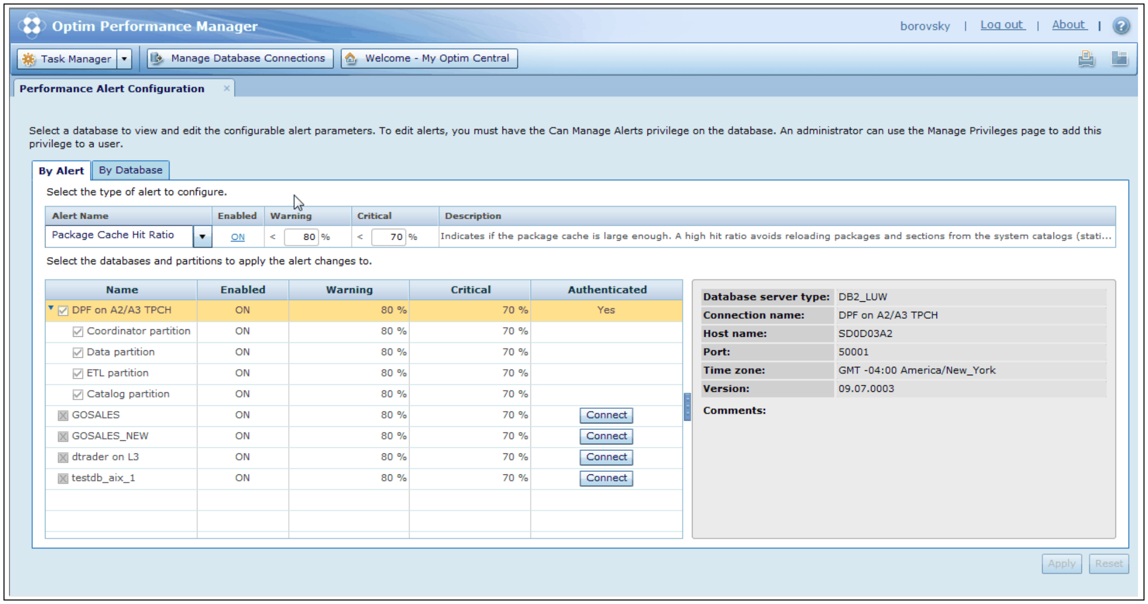


Figure 4-22 Performance Alert configuration panel

> **Note:** For partitioned DB2 databases, you can set different thresholds to different partition roles (catalog, data, coordinator). This requires that you have assigned partition roles to individual database partitions when you have configured your database for monitoring.

4.2.2 Alert configuration

From the Alert List, you can configure the alert settings for a specific alert category and data source by selecting the alert and clicking **Configure Alerts**. This configuration is only applicable for performance alerts.

Alert configuration panel allows you to configure alert settings in two ways:

- By Alert

  You can configure individual alert settings for one or more databases. You can modify, enable, or disable setting as well as warning and critical threshold levels for this particular alert.

  Figure 4-23 on page 164 Alert configuration panel - By Alert

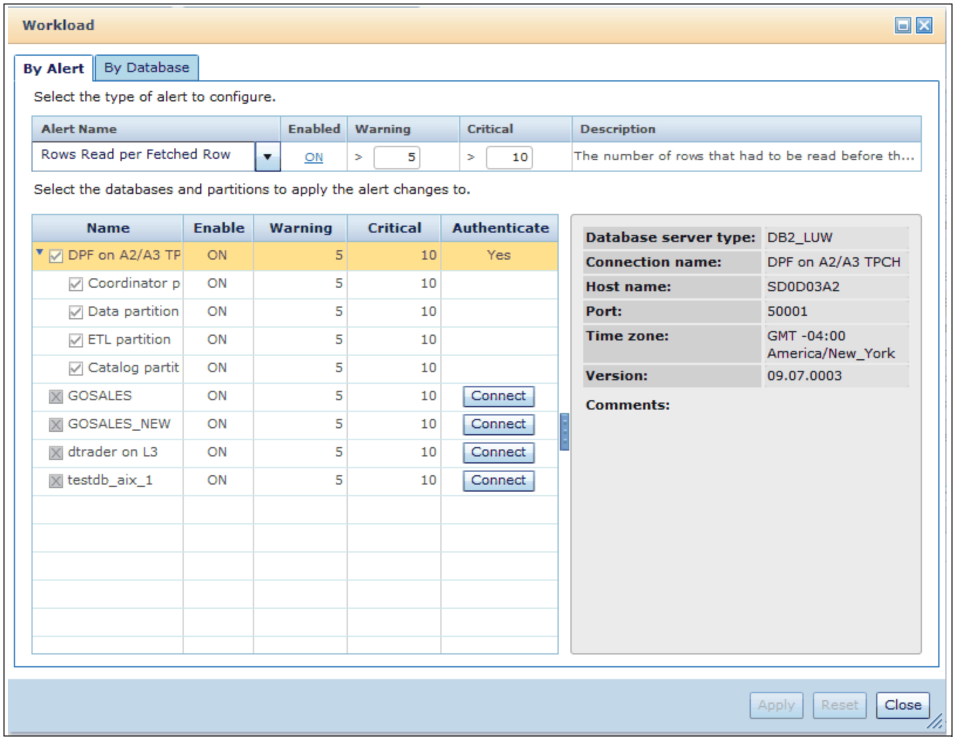


Figure 4-23 Alert configuration panel - By Alert

- By Database

  You can configure settings for any alert for a particular database. Use **Copy Settings** to replicate alert definitions to other monitored databases.

  Figure 4-24 on page 165 shows the Alert configuration panel - By Database.

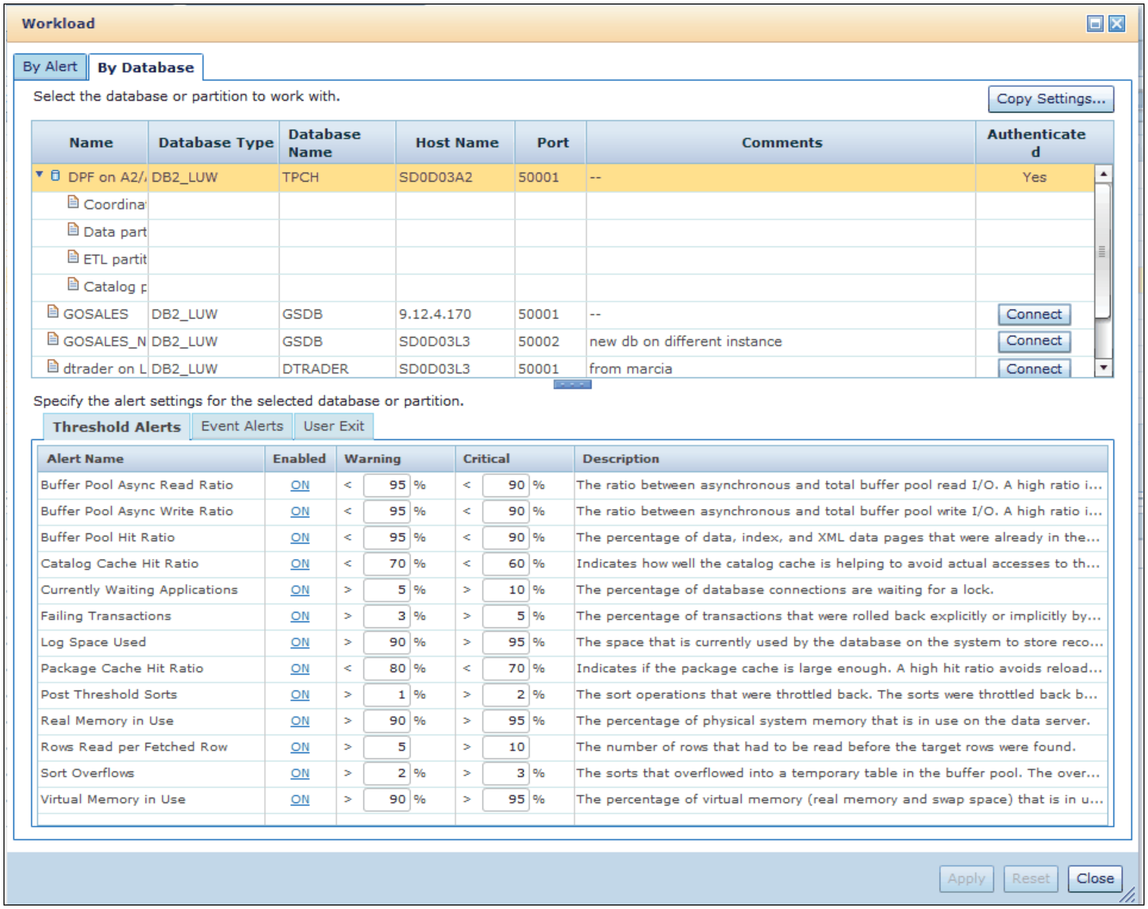


Figure 4-24 Alert configuration panel - By Database

Optim Performance Manager can automatically purge alerts which are no longer important. Select **Task Manager** → **Purge Alert Configuration** to specify time interval after which alerts will be deleted.

Figure 4-25 on page 166 shows the purge alerts configuration panel.

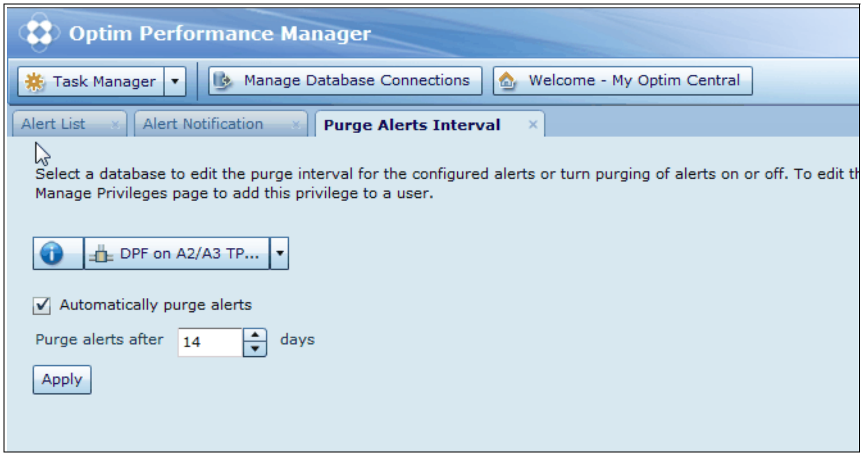


Figure 4-25 Purge alerts configuration

4.2.3 Alert notification

After you have configured alerts for your databases, you can setup alert notifications for individual alerts per monitored database. This allows Optim Performance Manager to send alert notifications to different individuals based on their responsibility for a particular monitored database or an alert category within the monitored database.

You can define alert notification parameters such as email addresses, SNMP trap generation, and alert frequency. To configure alert notifications, click **Task Manager** → **Alert Notification.** Figure 4-26 on page 167 shows the Alert notification panel.

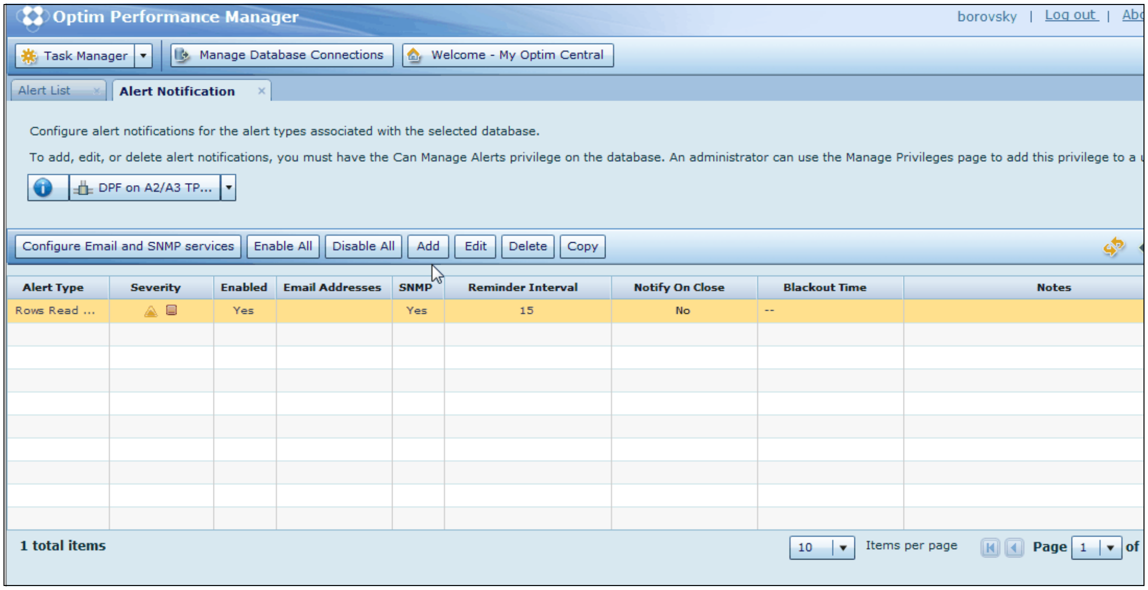


Figure 4-26 Alert notification panel

Optim Performance Manager can use two alert notification methods:

- Email
- SNMP trap

To configure these notification methods, click **Configure Email and SNMP services** and provide details for each of the methods, including:

- Address and port number of the outbound SMTP mail server that will be used to send email notifications.
- SNMP Management server that will receive the SNMP traps.

After you have configured alert notification services you can proceed to add and configure individual alert notification details. From the Alert Notification panel click **Add**. Figure 4-27 on page 168 shows the panel to add new alert notification. On this panel you can specify:

- Alert severity type: Warning or Critical.
- Email address: Specify email addresses of the recipients of the alert notification.
- SNMP trap generation: Optim Performance Manager alerts can generate SNMP traps, which can be forwarded to SNMP manager.
- Blackout period: Time interval during which alert notification is disabled.

Select the **Enabled** checkbox to activate the alert notification for the alert.

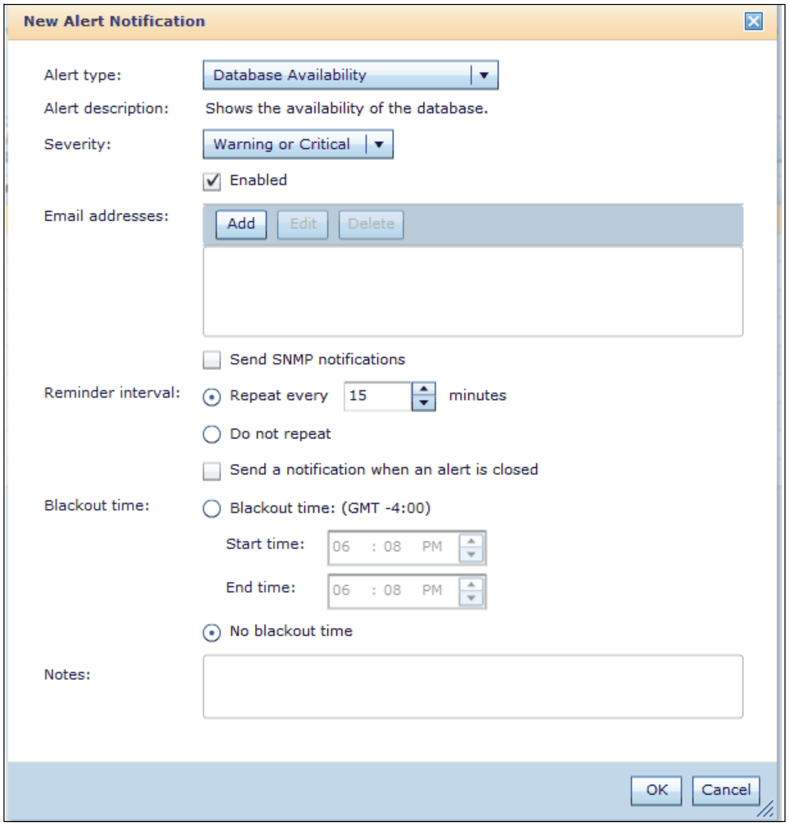


Figure 4-27 Adding new alert notification

4.3 Inflight dashboards

After initial review of the database health from the Health Summary page and the Overview dashboard, which were described in 4.1, “Health summary” on page 149, we can continue to drill down for more detailed database performance information. Optim Performance Manager delivers this information in a series of database performance dashboards, which are grouped under the Inflight dashboards category. They provide information about a database that relates to a particular category of potential performance issues including buffer pool and i/o, locking, logging, memory, active SQL, system, utilities, and workload.

This section gives an overview of individual Inflight dashboards. To navigate to Inflight dashboards, select **Task Manager** → **Inflight dashboards**.

4.3.1 Workload dashboard

The Workload dashboard (Figure 4-28 on page 169) shows information about the workload on the database. It provides detailed information about the sort performance, throughput of transactions, SQL statements and rows, as well as information about database connections. You can use this dashboard to check the utilization of the database.

Figure 4-28 Workload dashboard

> **Note:** For partitioned DB2 databases, Workload Dashboard provides ability to display data for configured partition sets using *Select partition set for analysis* drop down menu.

4.3.2 System dashboard

The System dashboard (Figure 4-29 on page 170) shows the system resources of the system on which the database is running. It provides information about the CPU and memory utilization. If you have launched Optim Performance Manager from Tivoli Enterprise Portal Console, you can click on Advanced System

Information link of the System dashboard, which will take you to Tivoli panels that provide more detailed system information.

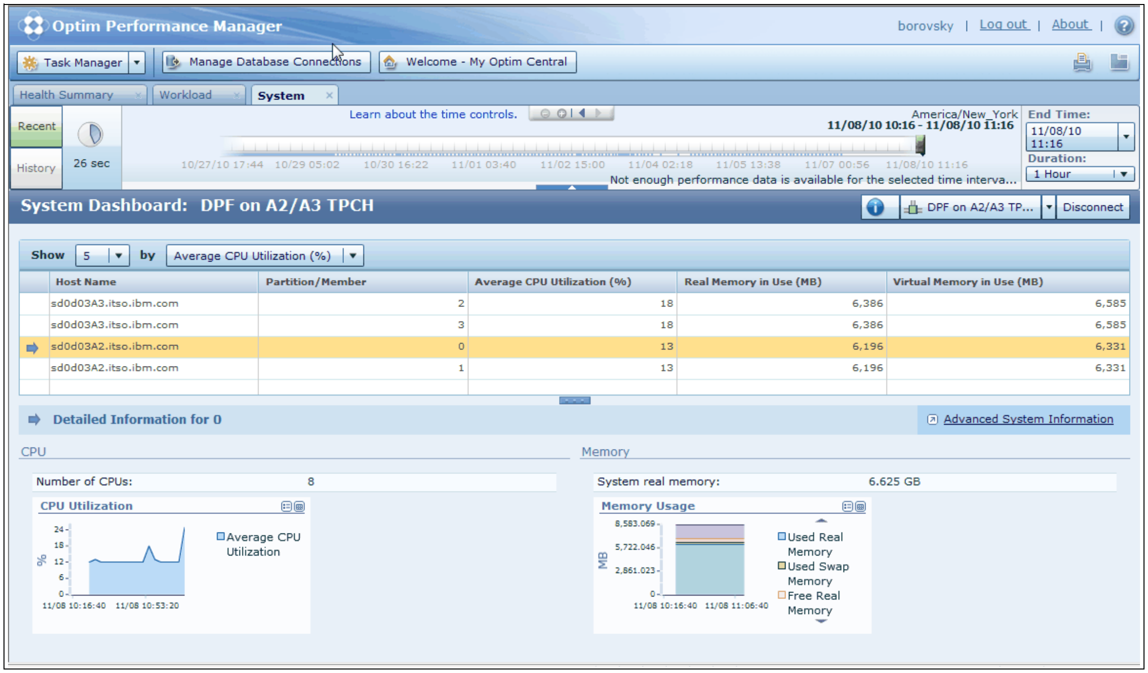


Figure 4-29 System dashboard

> **Note:** For partitioned DB2 databases system dashboard displays performance data for every partition.

4.3.3 Buffer pool and I/O dashboard

This dashboard (Figure 4-30 on page 171) shows database I/O at the buffer pool, table space, and table level. You can see the highest and lowest buffer pools by selecting the metric that you want to learn more about. You can find the buffer pools with low hit ratios and high activity and consider increasing their size to improve performance. You can also find table spaces and tables with low hit ratios and high activity and look for statements that access those tables that might need tuning.

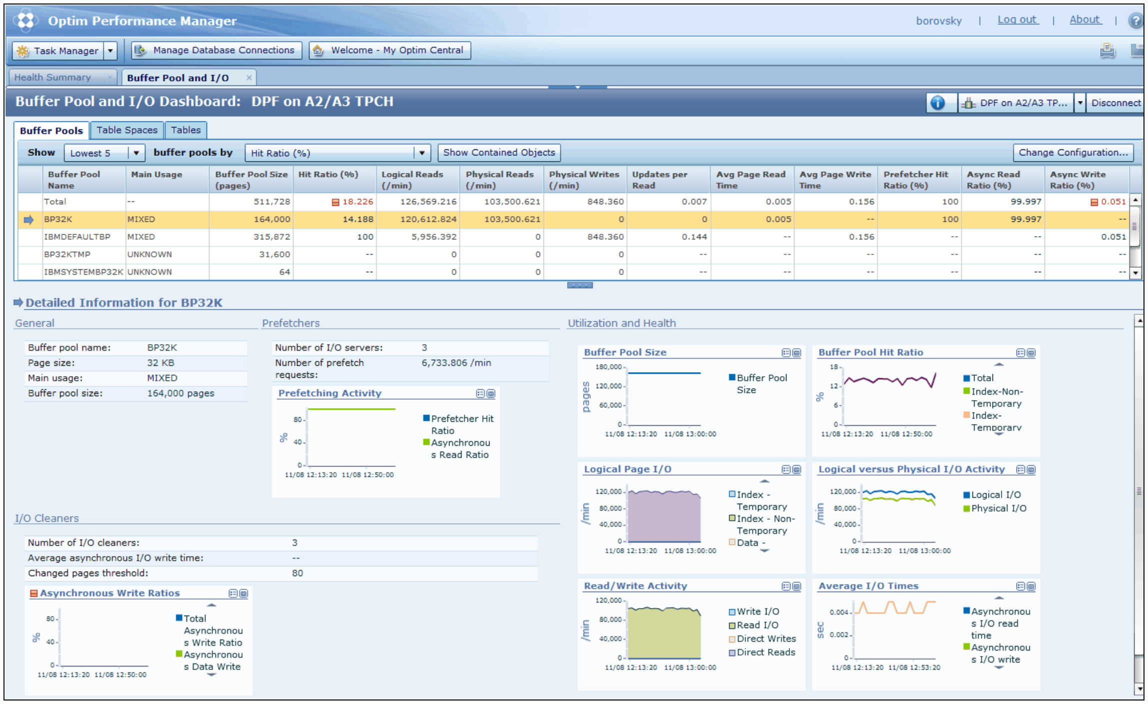


Figure 4-30 Buffer Pool and I/O dashboard

Click an item in the grid to view details about that item in the Detailed Information area. Select a buffer pool and click **Show Contained Objects** to view the table spaces that use that buffer pool. Select a table space and click **Show Contained Objects** to view the table objects that the space contains.

> **Note:** For partitioned DB2 databases buffer pool and I/O dashboard displays performance data which is aggregated over all partitions.

4.3.4 Memory dashboard

This dashboard (Figure 4-31 on page 172) shows the memory consumption of the selected database. It shows memory usage by instance, by database, and by application, and shows memory that is shared between applications.

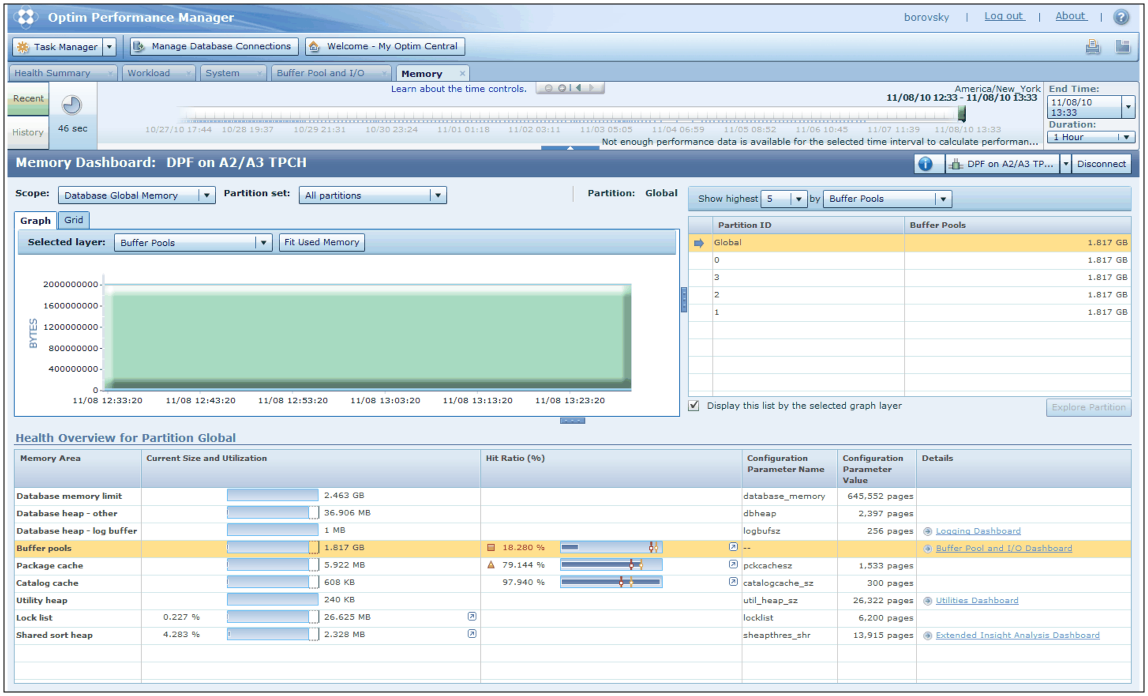


Figure 4-31 Memory dashboard

Select the Graph tab to display the statistics of the memory scope that you selected as a distribution chart. The layer (memory area) of the highest value is highlighted by default. Use the Selected layer menu to select a specific memory area to be highlighted both in the chart and in the bottom of the Health Overview table.

In a multi-partition environment, a table to the right of the Graph displays the partitions that have the highest values of the selected memory area. You can use the menu above this table to specify a memory area as well. To view the memory consumption of a specific partition or member, double click a partition, or select one of them and click **Explore Partition**.

The bottom Health Overview table shows the health status of a database. By using the Health Overview table, you can see how the memory areas are configured, the allocated size of the areas and how much of the allocated memory is in use, and check the efficiency of the memory area by examining at the hit ratio. By looking at this information, you can determine if a memory area is healthy or not, whether you want to increase the size of the memory area, and whether you can decrease the size to release some memory.

4.3.5 Locking dashboard

This dashboard (Figure 4-32) shows deadlocks, timeouts, and locking conflicts. You can use this dashboard to determine which workload applications, users, or servers have the most locking problems, and you can drill down to find the waiting or blocking connections and events.

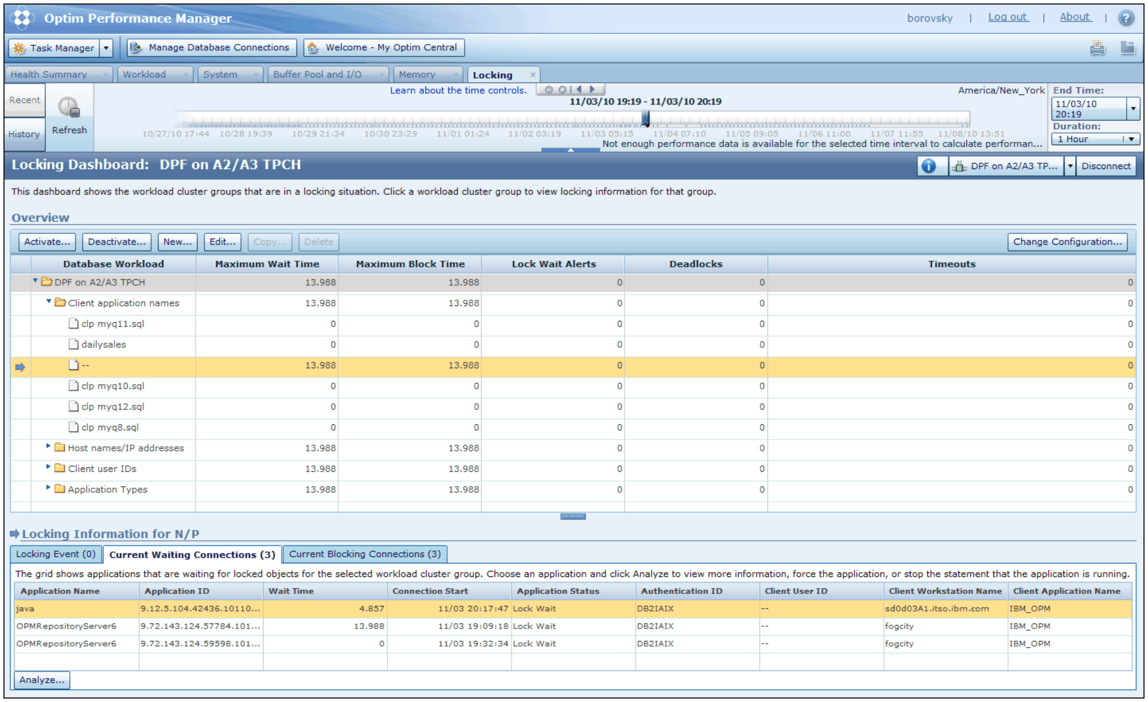


Figure 4-32 Locking dashboard

To analyze a locking event:

- In the Overview grid, click the workload cluster group, workload cluster, or the database for which you want to view locking information.
- In the Locking Information grid, click the connection (waiting or blocking) or event that you want to analyze.
- Click **Analyze**.

Figure 4-33 on page 174 shows the Analyze Locking Situation panel. This panel displays the complete lock tree for waiting or blocking connection. Each complete set of entries in the tree includes an application that is holding a lock and the applications that are waiting because of that lock. The entries between the main entry and the leaf entry are applications that are blocking and waiting. Each leaf entry is an application that is only waiting. In case of a deadlock event, waiting connections are also blocking connections. Deadlock analysis is described in detail in Chapter 7, “Analyzing locking problems” on page 251.

You can also use this panel to force an application, and stop or tune the statement that the application is running.

Note: Workload cluster groups and workload clusters are shared with the Extended Insight dashboard. For their description, refer to section 4.4, “Extended Insight dashboard” on page 178

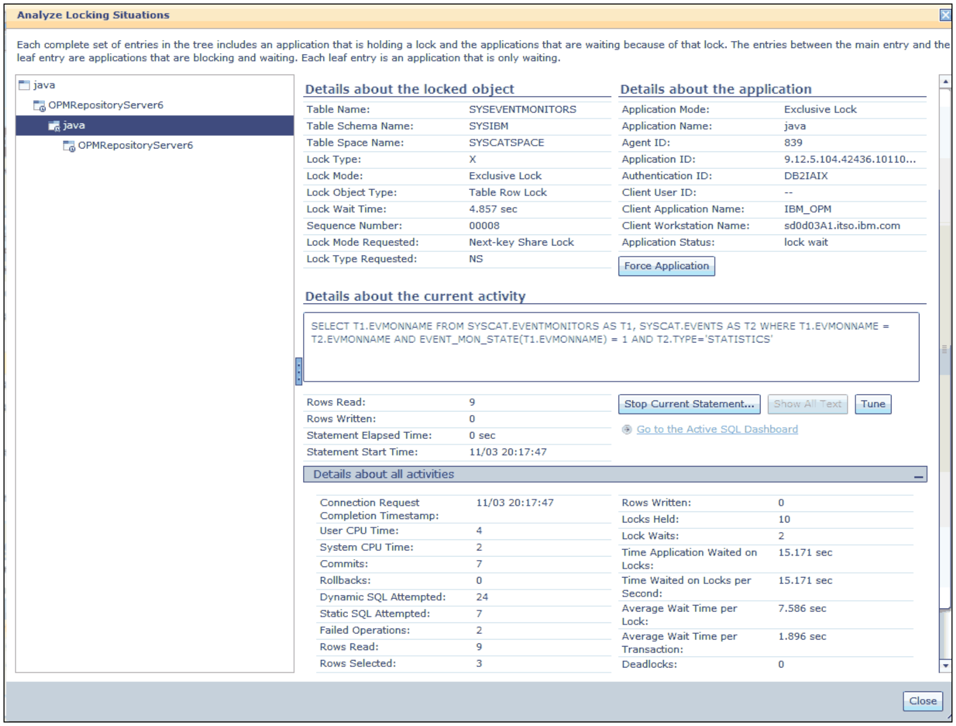


Figure 4-33 Analyze locking situation panel

4.3.6 Logging dashboard

This dashboard (Figure 4-34 on page 175) shows the configuration and logging activity. You can use this dashboard to determine how recovery is impacted and whether tuning is needed for the log files or log buffer. High average log read/write times, which are in the Logging activity and I/O performance section of

Logging dashboard, can indicate whether disk configuration for the database log files can be responsible for the database performance degradation.

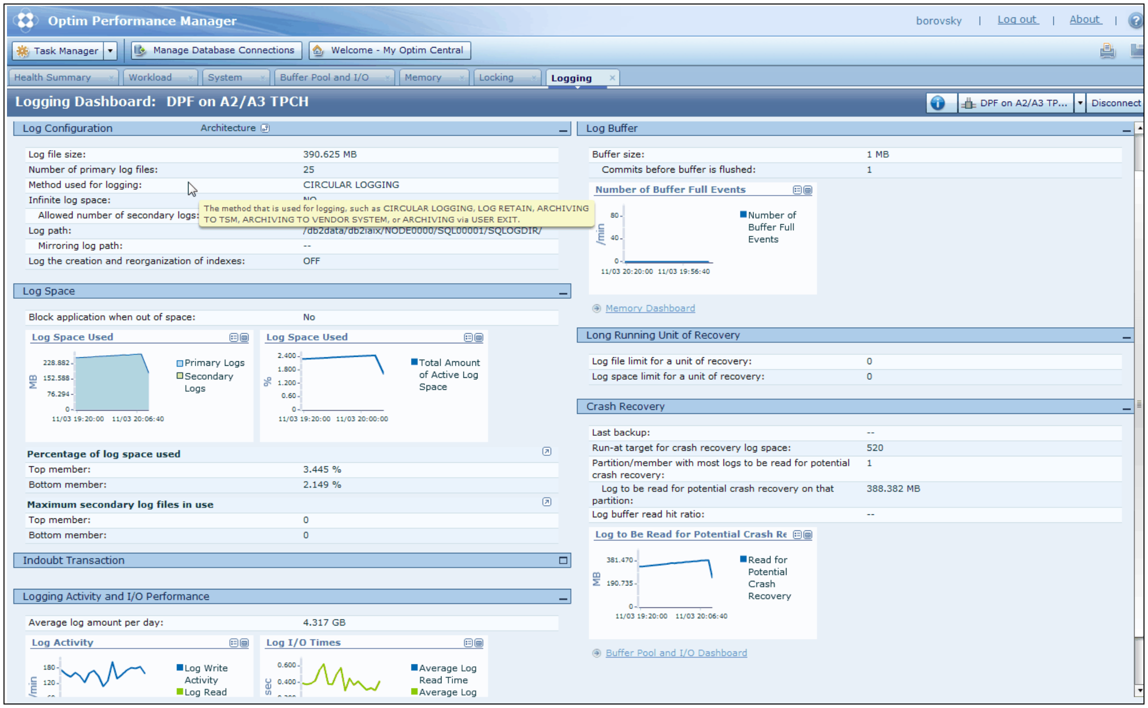


Figure 4-34 Logging dashboard

> **Note:** For partitioned DB2 databases, Logging Dashboard provides ability to display data for configured partition sets using *Select partition set for analysis* drop down menu.

4.3.7 Active SQL dashboard

This dashboard (Figure 4-35 on page 176) shows SQL statements which were running at the time when Optim Performance Manager snapshot was taken. You can select an SQL statement from the list and look through the key performance indicators (KPIs) in the SQL Statement Details area to see if the statement impacts your system negatively. From this dashboard, you can stop queries and force applications if needed. You can tune these SQL statements with Optim Query Tuner, if it is available.

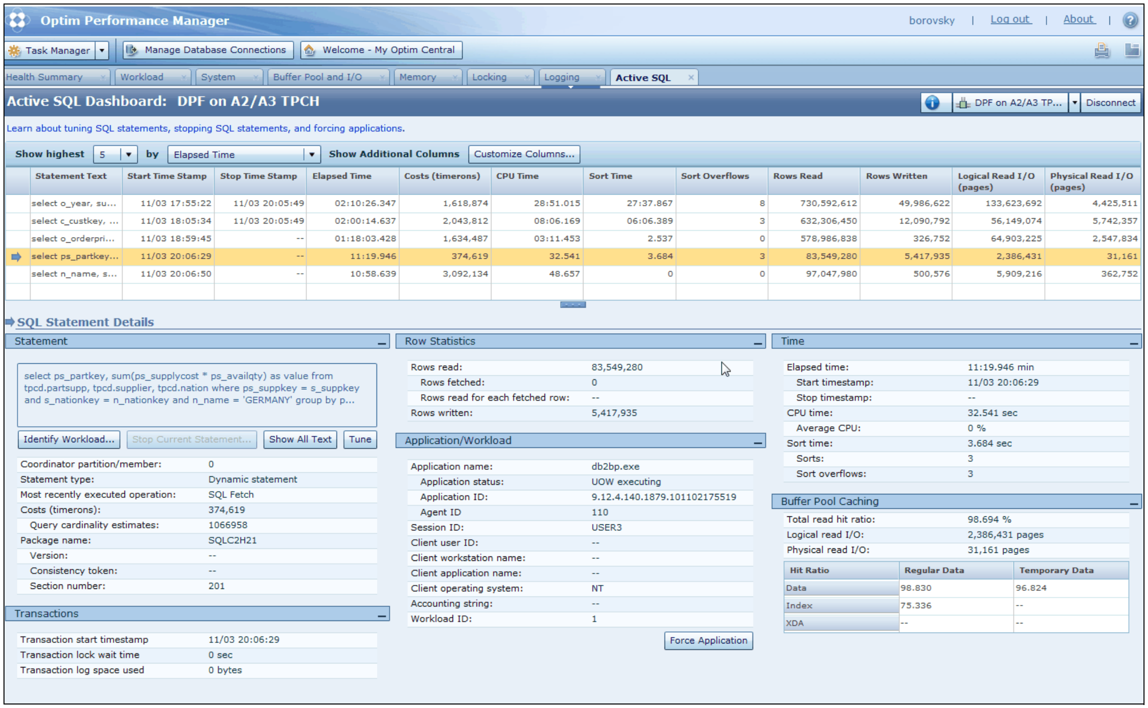


Figure 4-35 Active SQL dashboard

> **Note:** Data presented in this dashboard is based on application snapshot metrics. These snapshots are taken at intervals specified during the Optim Performance Manager configuration of the monitored database in the *Active SQL and Connections* profile, for instance, every 2 minutes. SQL statements which begin and end between the snapshots are therefore not captured in this dashboard.

Tuning SQL statements

The process of tuning an SQL statement consists of first analyzing the statement, and then changing the SQL statement based on the results of the analysis.

Check the key performance indicators in the SQL Statement Details area for statements that you might want to tune. For example, a high value of sort time might indicate that an index is missing, and a high value of rows read per fetched row might also indicate that an index is missing. If one of these two values is high and the total execution time is unusually high, then tuning the statement is recommended. To tune the statement, start Optim Query Tuner product. and click

Tune on the Active SQL dashboard. Optim Query Tuner can provide recommendations about indexes, materialized query tables (MQTs), or statistics to improve the execution of the SQL statement.

Stopping SQL statements

Check the key performance indicators in the SQL Statement Details area for statements that you might want to stop. For example, check the values for CPU time, sort time, and physical reads. If they are very high, then other workloads might be impacted. In such cases, you might want to stop the statement. Some applications might be allowed to use a great amount of resources, so the decision to stop the statement also depends on the application that is executing the statement. Look at the Application/Workload section to see which application is executing the statement so that you can decide if the resource usage is unusual. If you decide that you want to release the resources to the system, you can stop the statement by clicking **Stop Current Statement**. This will roll back the current statement.

Forcing applications

Similarly to stopping currently executing SQL statement, you can also use Active SQL dashboard to force an application by clicking **Force Application**. Forcing an application cancels the database connection that is being used by the application.

> **Tip:** The impact of canceling a connection might be very big. Therefore, consider using the Stop Current Statement button instead to reduce the impact.

4.3.8 Utilities dashboard

This dashboard (Figure 4-36 on page 178) shows the status and progress of utilities such as `RUNSTATS`, `LOAD`, and `BACKUP`. You can get information about utilities that are in progress and that completed, and you can identify utilities that failed. You can use this information to determine if some utilities should run at different times to avoid high workloads.

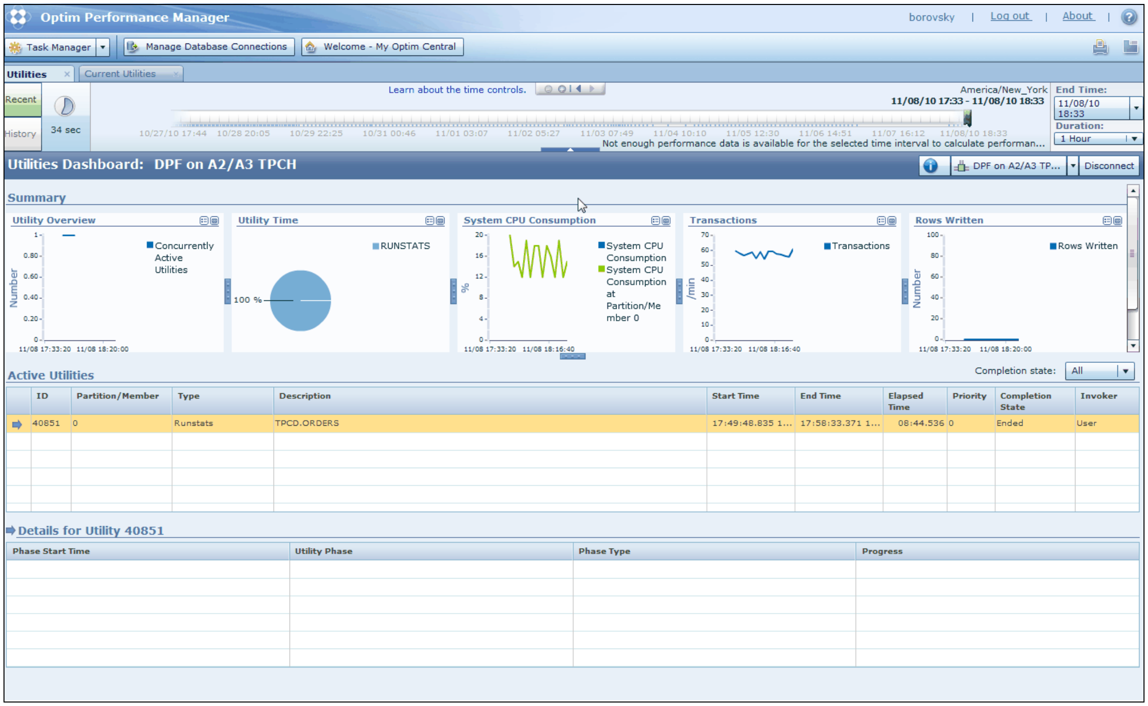


Figure 4-36 Utilities dashboard

4.4 Extended Insight dashboard

The Optim Performance Manager Extended Insight dashboard displays end-to-end data about the entire database application system, which includes clients, application servers, data servers, and the network. Monitoring begins when you initiate a transaction, continues as that transaction is processed by each component, such as the client, network, and data server, and ends when the application finishes processing and produces the results.

Figure 4-37 on page 179 illustrates software layers which are involved in the database transaction. Optim Performance Manager Extended Insight is able to identify how much time a database transaction spends in each layer (Application server, database driver, network, database server). For monitored databases that are at DB2 9.7 Fix Pak 1 or above, Optim Performance Manager Extended Insight can further break down the transaction response times at the database server layer, that is, average compilation time, sort processing, I/O processing, lock wait, and so on.

This can help database administrators to determine which software layer is responsible for the slow database transaction response time.

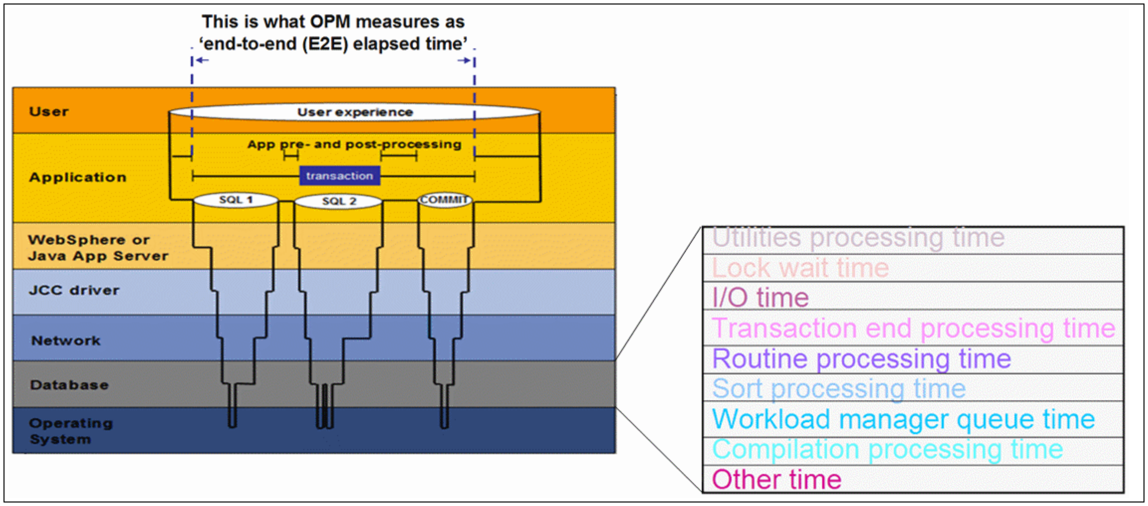


Figure 4-37 Extended Insight end-to-end monitoring

Extended Insight dashboard consists of two panels:

- Overview panel
- Details panel

4.4.1 Extended Insight dashboard overview panel

Figure 4-38 on page 180 shows a sample Extended Insight dashboard overview panel.

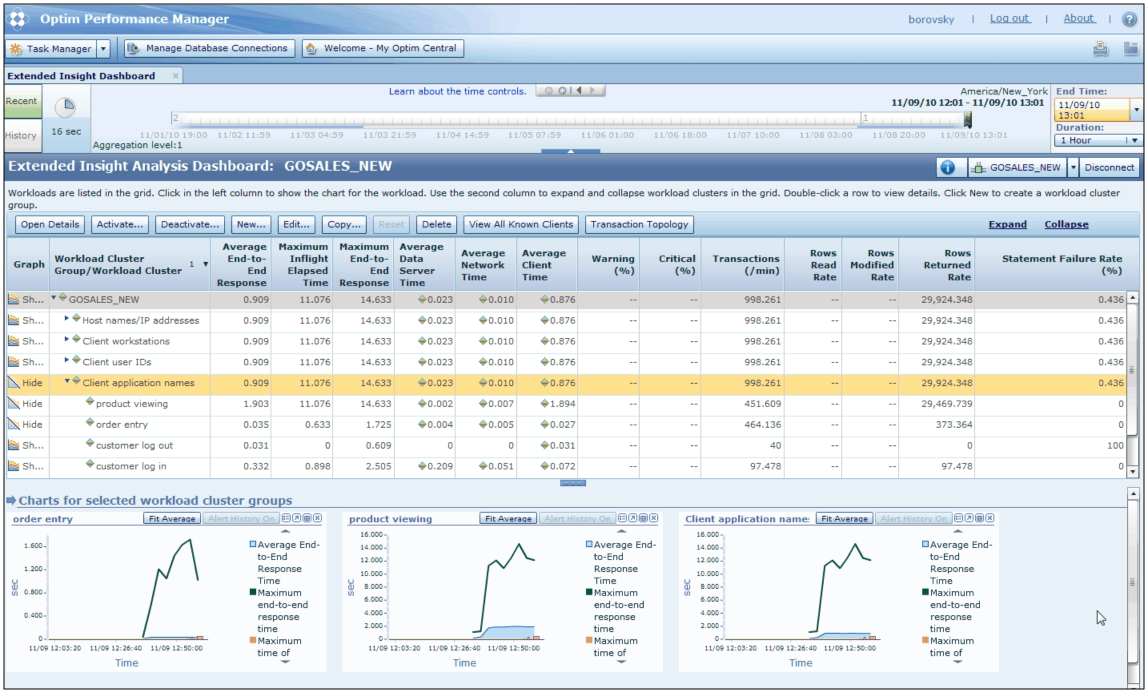


Figure 4-38 Extended Insight dashboard overview panel

You can use the overview panel to view:

- ► Statistics in the grid

  The Extended Insight Analysis overview grid lists statistics for the workload cluster groups, workload clusters, and database that you are monitoring. Key statistics in the grid are:

  – Average end-to-end response time
  – Maximum end-to-end response time
  – Average data server time
  – Average network time
  – Average client time

  You can double-click a row in the grid to view response time details for a workload cluster group, a workload cluster, or the entire database.

- ► Charts

  You can view charts for a selected workload cluster group, workload cluster, or database by clicking the icon in the left column of the grid. Figure 4-39 on page 181 shows the chart for the selected workload cluster.

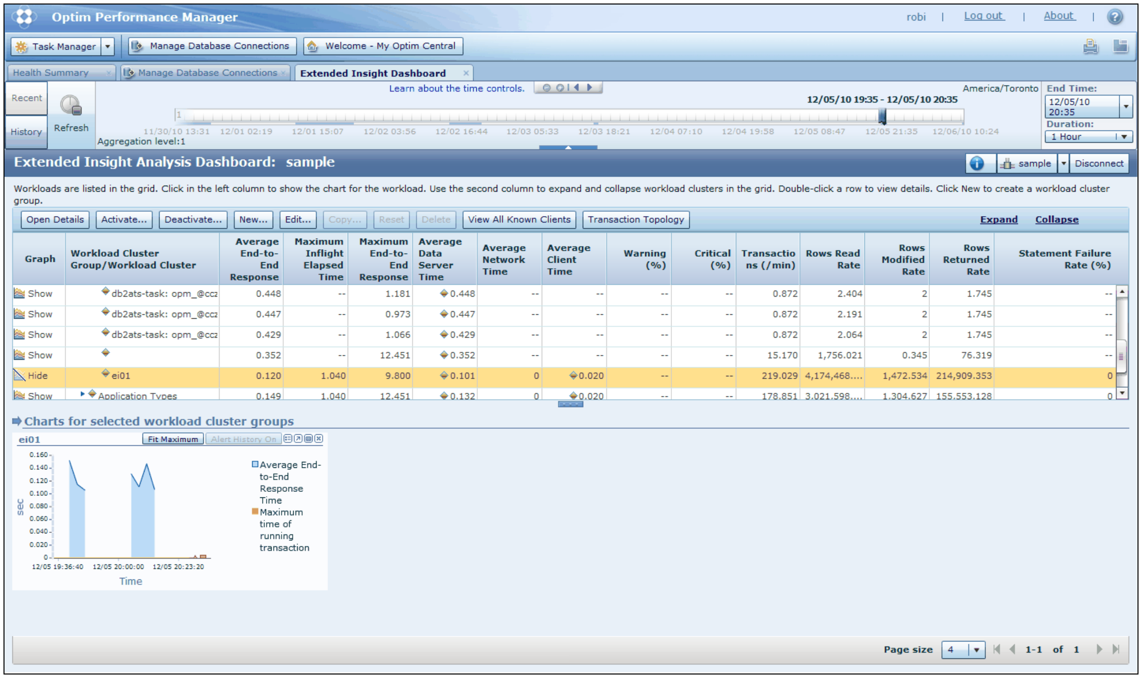


Figure 4-39 Chart for the selected workload cluster

You can double click on the chart to obtain more detailed information about the transaction end-to-end response times for the selected workload cluster group, workload cluster, or the entire database. This detailed information also contains response time histogram, which groups transaction end-to-end response times into discrete ranges. Bar for each range of values represents the percentage of transactions, which completed within the particular range. Figure 4-40 on page 182 is an example of a detailed information displayed for the selected workload cluster chart.

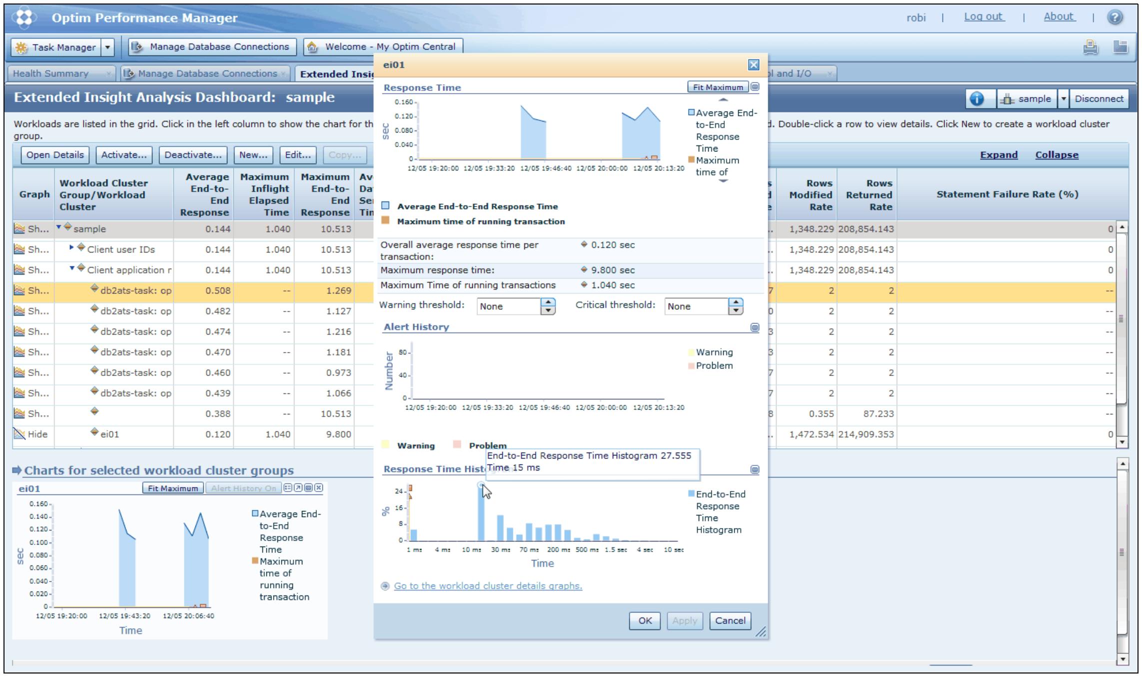


Figure 4-40 Detailed information for the selected workload cluster chart

4.4.2 Extended Insight dashboard details panel

Use the Extended Insight dashboard details view to locate the source of performance problems, determine how those problems affect different parts of the workload, and analyze the performance of individual SQL statements, clients, and partitions.

To view details panel, double click on the workload cluster group or workload cluster in the overview panel. Figure 4-41 shows an Extended Insight dashboard details panel.

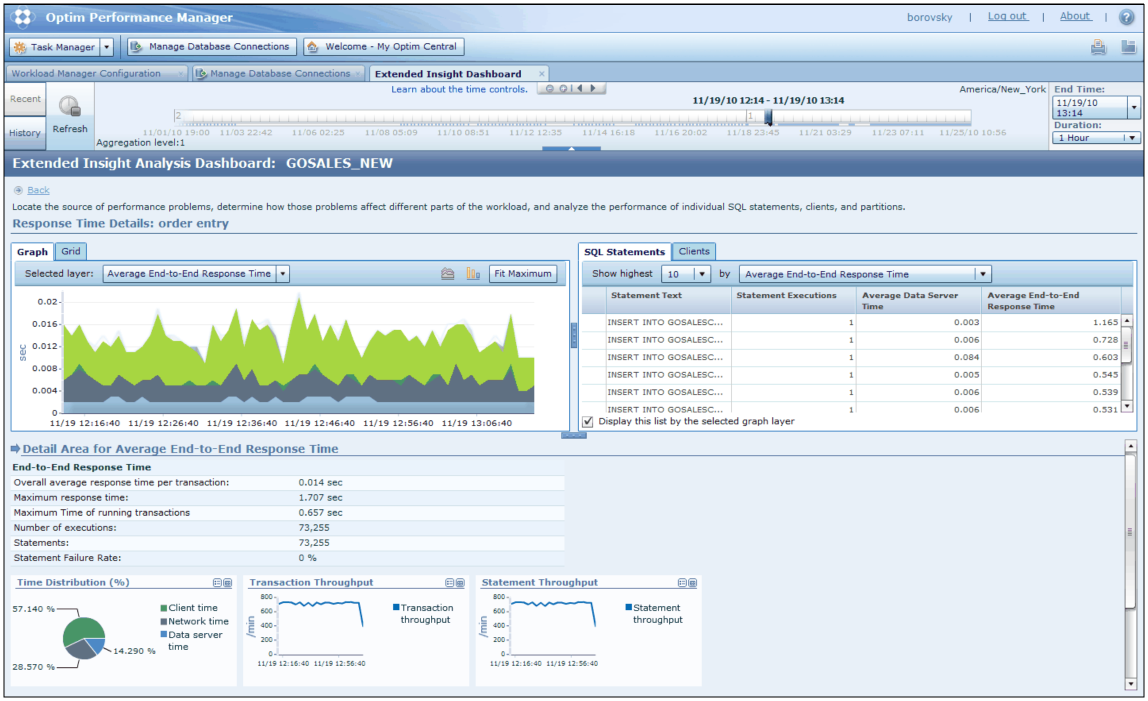


Figure 4-41 Extended Insight dashboard details panel

The details panel contains several tabs.

Graph tab

The Graph tab shows response times for the components of the workload cluster group, workload cluster, or database for the time frame that is selected on the time slider. The graph shows time layers for each component, such as the data server, client, and network. Use the Selected layer list or click a layer in the graph to view details for that layer in the bottom pane of the dashboard. Figure 4-41 is an example of a details panel with the Graph tab for the selected layer of Average data server time per transaction. The details area of this panel displays more information relevant to the selected layer.

SQL Statements tab

The SQL Statements tab lists the SQL statements that were run by the transactions of the workload cluster group, workload cluster, or database for the time frame that is selected on the time slider. Click a statement from the list to view details for that statement in the details area of the dashboard. Details area consists of two tabs:

- General information tab

This tab shows the general SQL statement information such as statement text, end-to-end time spent metrics.

Figure 4-42 shows the details panel with the SQL statement general information.

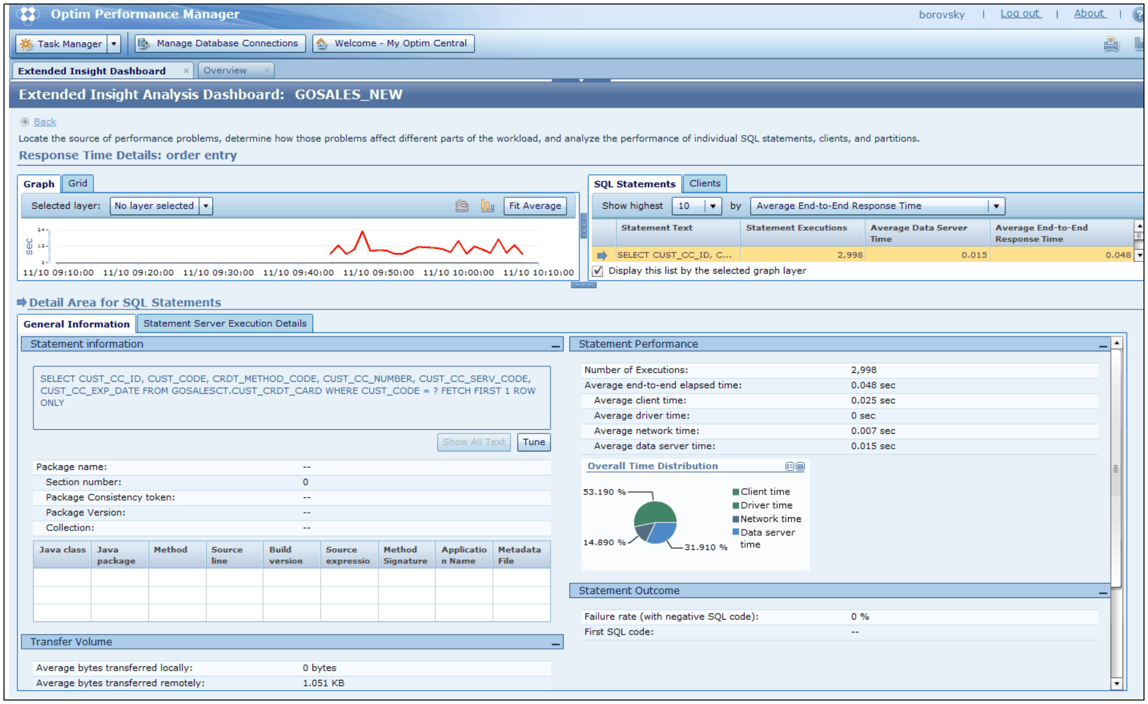


Figure 4-42 Details panel - SQL statement general information tab

- Statement server execution details

  This tab (Figure 4-43) shows the details of this SQL statement from a data server execution perspective. It displays information such as row and sort performance, I/O statistics, locking statistics, and data server execution times.

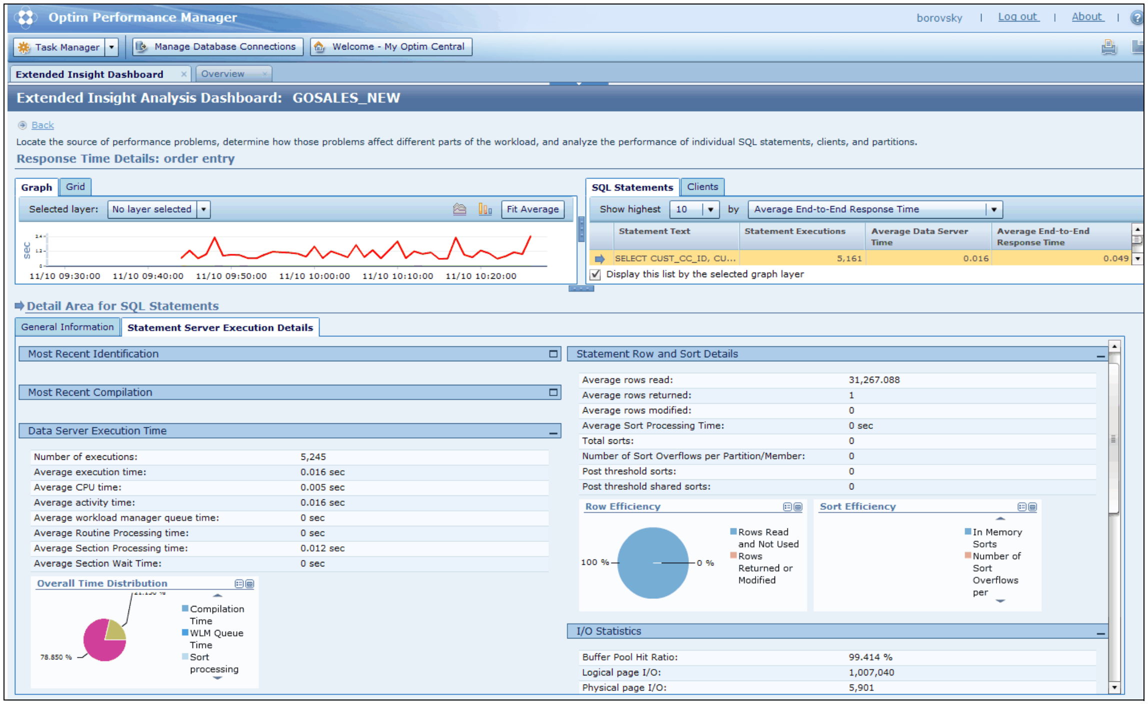


Figure 4-43 Details panel - SQL statement data server execution statistics tab

> **Note:** Statement server execution details are available for DB2 V9.7 Fixpak 1 or higher databases. It also requires that during the OPM configuration process of the monitored database you have selected to collect statement metrics on data server.

Clients tab

The Clients tab (Figure 4-44 on page 186) lists the clients that ran the transactions for the workload cluster group, workload cluster, or database for the time frame that is selected on the time slider. Click a client from the list to view the details for that client in the bottom pane of the dashboard.

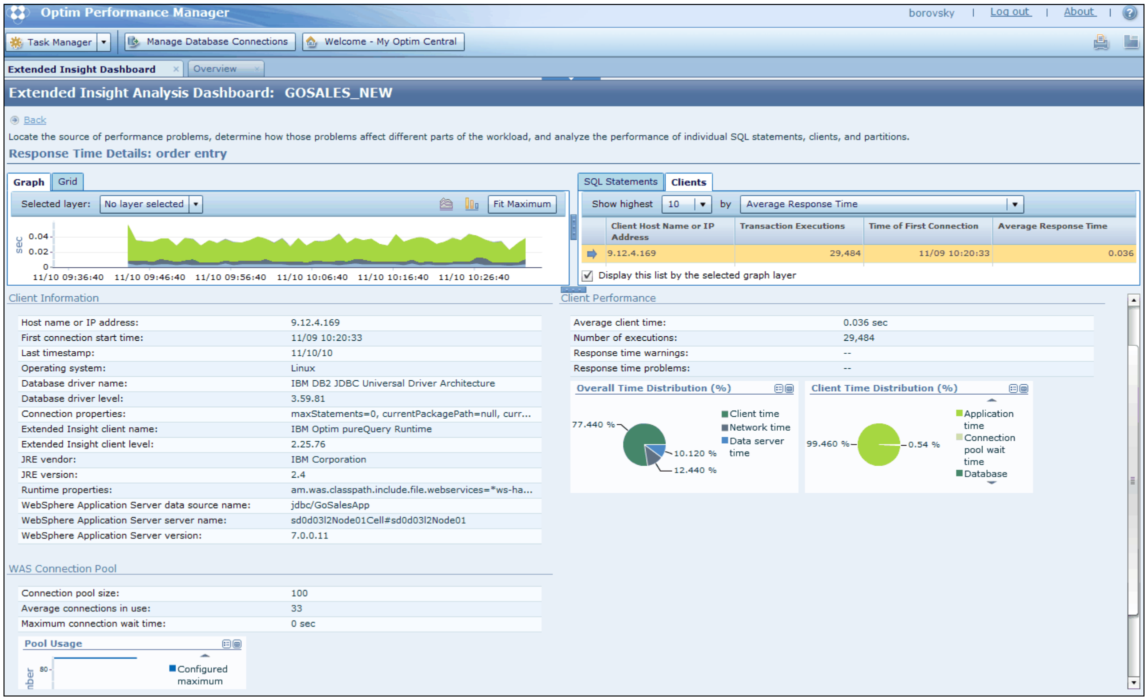


Figure 4-44 Details panel - Clients tab

Partitions/Members tab

The Partitions/Members tab (Figure 4-45 on page 187) lists the partitions of a data server where transactions for the workload cluster group, workload cluster, or database were run during the time frame that is selected on the time slider. Click a partition from the list to view details for that partition in the bottom pane of the dashboard.

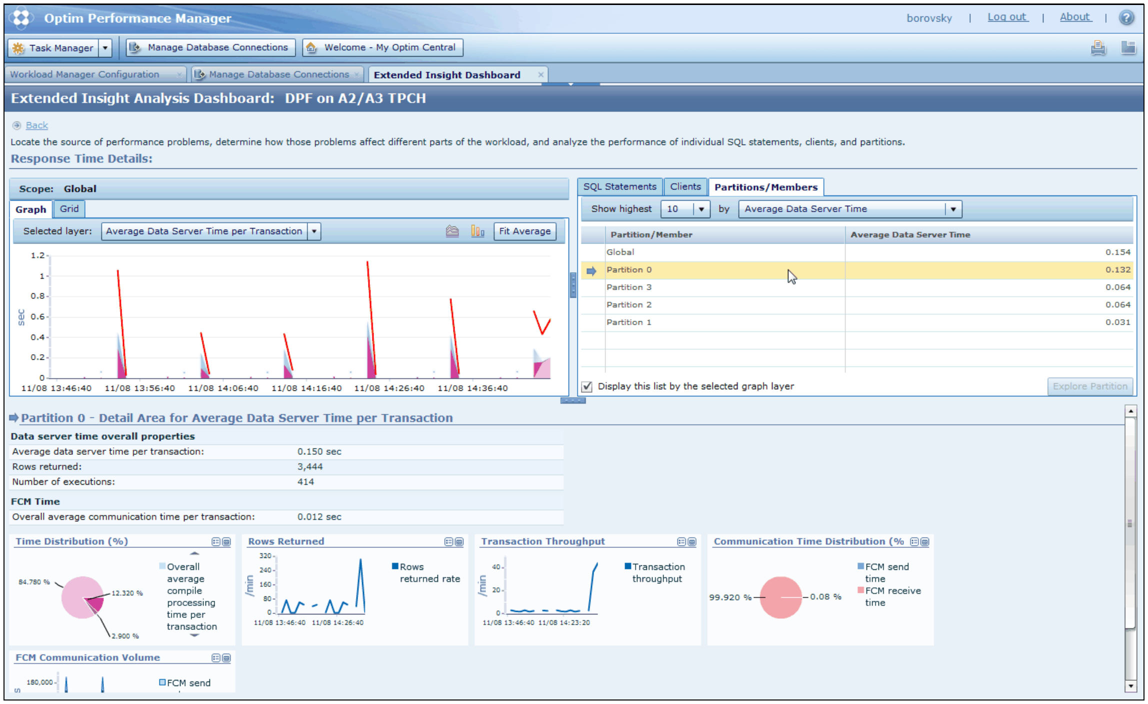


Figure 4-45 Details panel - Partitions/Members tab

4.4.3 Workload cluster groups and Workload clusters

It can be difficult to determine where and when performance problems and bottlenecks occurred. To isolate the source of performance problems, you can group and monitor transactions that come from specific components by using workload cluster groups.

Extended Insight monitoring is based on predefined or user defined workload cluster groups that contain workload clusters. The grouping of transactions into workload cluster groups is based on the connection attributes that are set for a connection to the monitored database. For example, in the predefined workload cluster group named *Client user IDs*, you get one workload cluster for each user ID that is set in the corresponding connection attribute. In each workload cluster of this group, all the transactions that originate from the same user ID are grouped together. The performance metrics are also aggregated within the group.

Connection attributes represent client information fields which are used at the DB2 server for determining the identity of the application or user currently using the connection. For more detailed description of connection attributes, refer to section 8.2.3, “Understanding workload clusters” on page 291.

You can define values for the following connection attributes:

- Authorization ID
- Client user ID
- Client application name
- Client workstation
- Client accounting string
- IP address (DB2 Version 9.7 and higher)
- Application type

Connection attributes can be set either at the database server or at the application or application server level. For detail information on how to set these attributes refer to DB2 documentation site:

`http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp`

You can activate or deactivate a workload cluster group for monitoring at any time from Extended Insight dashboard by clicking **Activate** or **Deactivate**. When you activate a workload cluster group for monitoring, you can view detailed data for all transactions in the group and for each of the workload clusters in the group. Activation or deactivation process doesn't stop the collection of database performance data. It only impacts the aggregation of the data on the dashboard.

You can create workload cluster groups specific to your needs or use the predefined workload cluster groups that are already activated. OPM EI provides specific support for monitoring a set of common application frameworks:

- SAP: Monitoring response times per SAP end user, SAP transaction, SAP source module and SAP application server.
- Cognos: Monitoring response times per Cognos report, Cognos, report user, Cognos report package, and Cognos end user system and Cognos server
- DataStage: Monitoring response times per DataStage job, DataStage job user, DataStage project, and DataStage server
- InfoSphere Warehouse

To create new workload cluster group, click **New** on the Extended Insight dashboard.

Figure 4-46 on page 189 shows a Create new workload cluster group panel. In this example, let us assume that we want to create new workload cluster group called `a_user_group`, which will contain database application performance data from a client user ID `a user`. On screen shown in Figure 4-46 provide a name for the new workload cluster group name, and click **Next**.

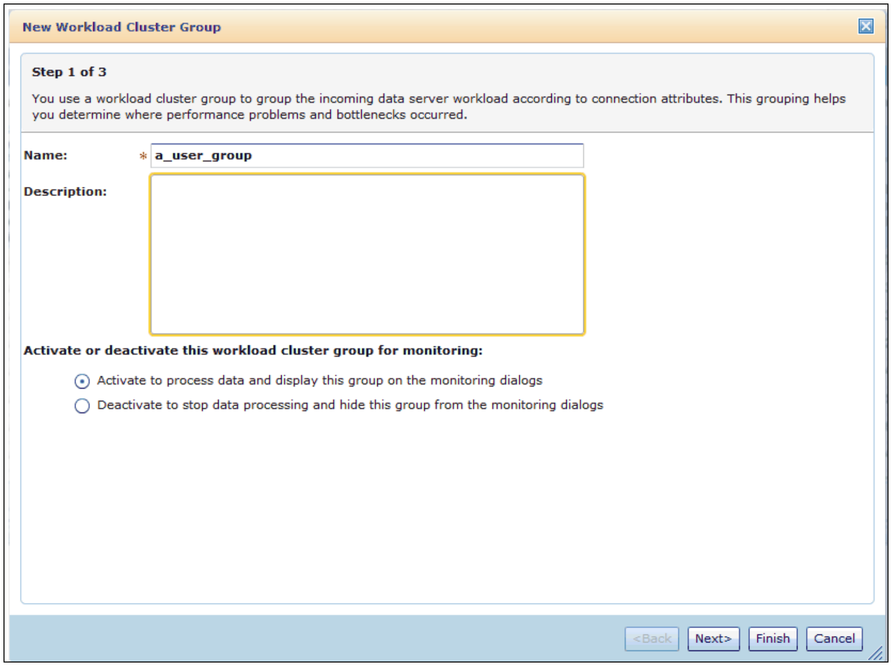


Figure 4-46 Create new workload cluster group panel

Figure 4-47 on page 190 shows the New workload cluster group details panel. In this panel, select the connection attribute for this workload cluster group (Client User ID) and the filtering condition, which will only display data for the specific Client User ID (a user).

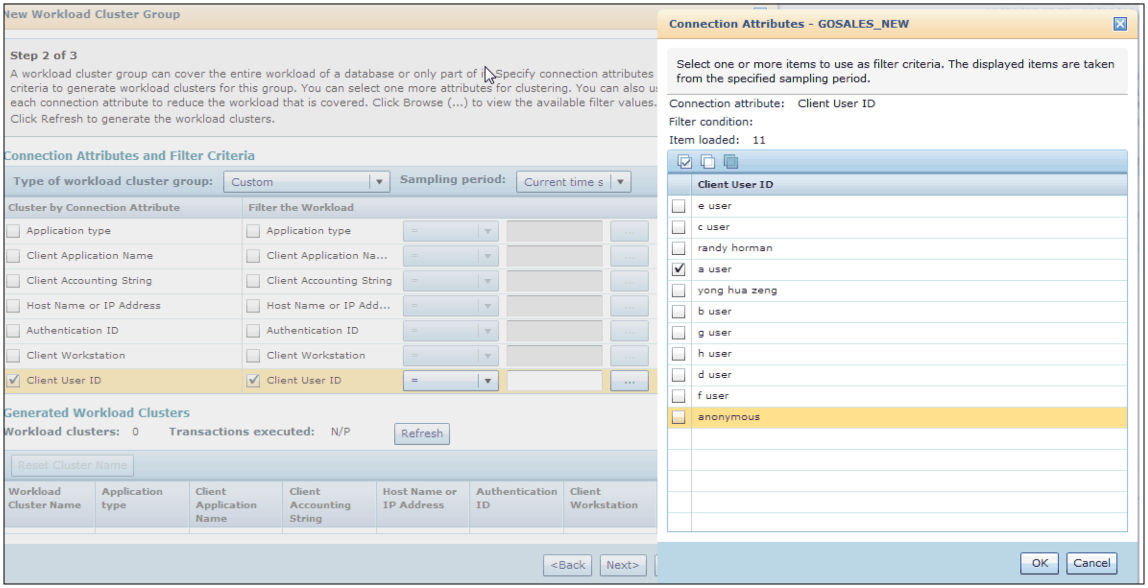


Figure 4-47 New workload cluster group details

Figure 4-48 on page 191 shows the new workload cluster group threshold settings panel. You can specify response time thresholds for the entire workload cluster group or for individual workload clusters. When the workload cluster group is activated for monitoring, you are informed if thresholds were violated. Click **Finish**.

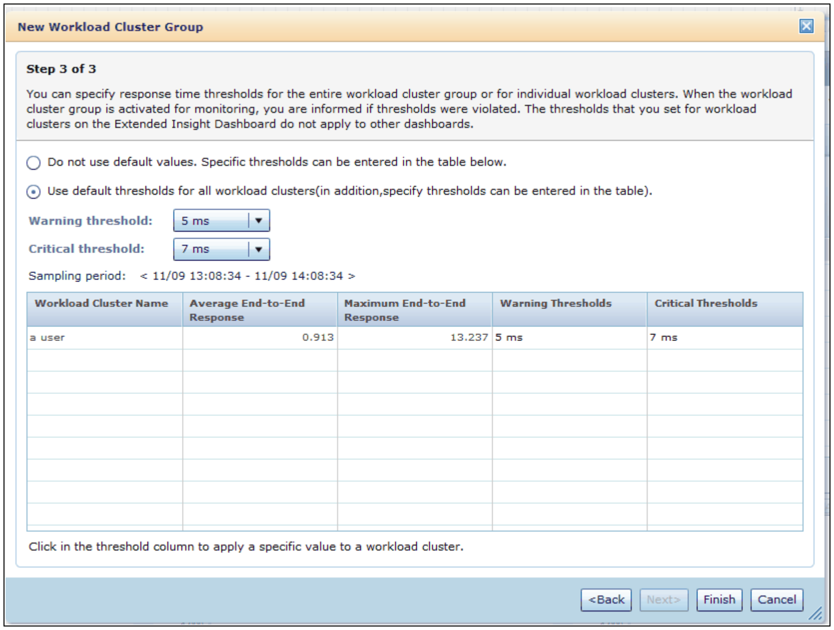


Figure 4-48 New workload cluster group threshold settings

After the new workload cluster group is defined and activated, you can view its database performance metrics on the Extended Insight dashboard.

Figure 4-49 on page 192 shows the Extended Insight dashboard with the new workload cluster group.

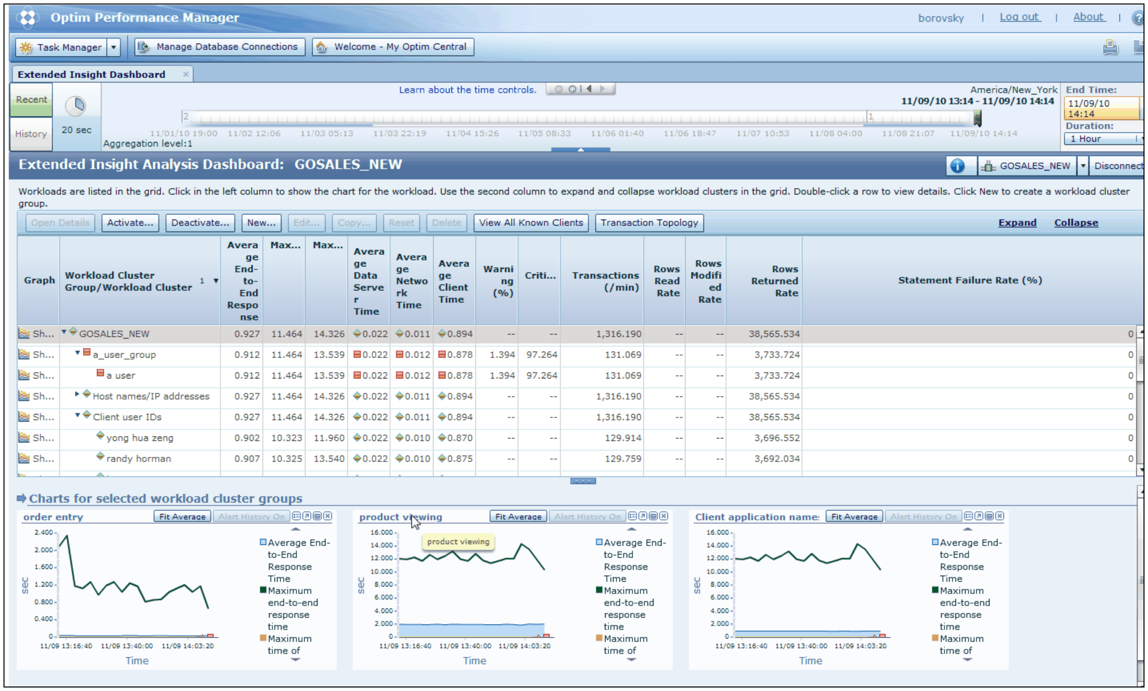


Figure 4-49 Extended Insight dashboard with the new workload cluster group

4.4.4 pureQuery Runtime integration

An important step in tuning SQL is to determine the source of the SQL so it can be modified. This can be difficult, especially if the SQL was generated by a third party framework such as Hibernate or JPA. To help identify the source code, the Extended Insight dashboard can display the pureQuery metadata, such as Java class, package, application name, method name, and source line number, as shown in Figure 4-50 on page 193.

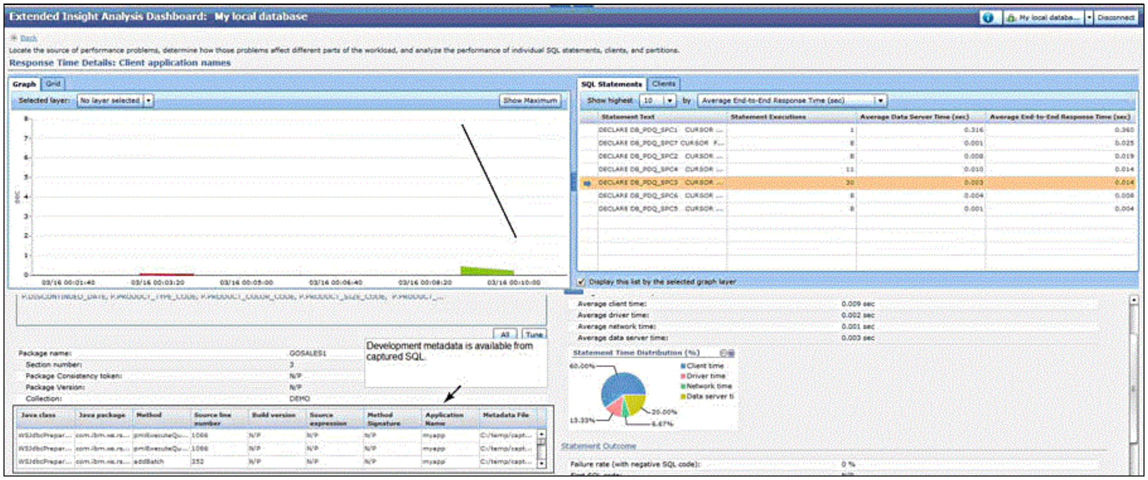


Figure 4-50 Integration of pureQuery Runtime metadata

This feature enables the database administrator and the developer to collaborate by quickly identifying the source SQL. This feature requires a license for the pureQuery Runtime product.

For more information about the pureQuery product, refer to *Using Integrated Data Management To Meet Service Level Objectives*, SG24-7769.

4.5 Reports

Optim Performance Manager provides predefined report types that you can use to generate reports to review and analyze your data in different ways. The following is a list of predefined reports:

- Database configuration report
- Database manager configuration report
- Database connection report
- Disk space consumption report
- Dynamic SQL statement report
- Workload manager configurations and metrics report

In this section we introduce and give examples of each report. From Optim Performance Manager web console, click **Task Manager** → **Reports** to navigate to individual reports.

4.5.1 Database Configuration report

The Database Configuration report (Figure 4-51) shows an overview of the database configuration. If you use the DB2 Database Partitioning Feature (DPF), the report groups information by partition types so that you can compare the configuration of your coordinator, data, ETL, catalog, monitoring, or other partition types. This grouping of information by partition types requires that you have assigned roles (ETL, data, catalog, and so on) to database partitions during the Optim Performance Manager configuration process for your monitored DB2 DPF database.

You can generate additional reports to check whether all partitions of the same partition type have the same configuration, or to check which configuration parameters for a given partition or group of partitions changed over time.

The Database Configuration report contains information about capacity management, communications, logging and recovery, and database management. You can generate more detailed reports for a given partition role or partition member.

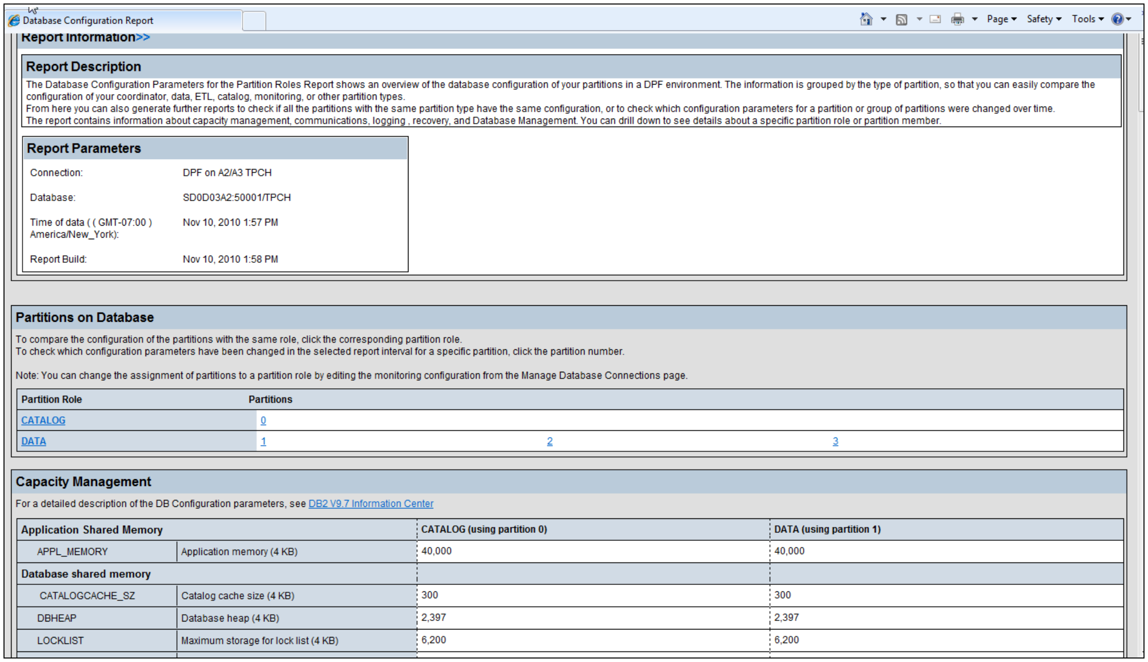


Figure 4-51 Database Configuration report

4.5.2 Database Manager Configuration report

The Database Manager Configuration report (Figure 4-52 on page 195) shows an overview of the current database manager configuration and which parameters have been changed in a given time frame to help you determine whether a problem might have been caused by configuration changes.

The Database Manager Configuration report contains details about system management, system monitoring parameters, instance administration, capacity management, and communications.

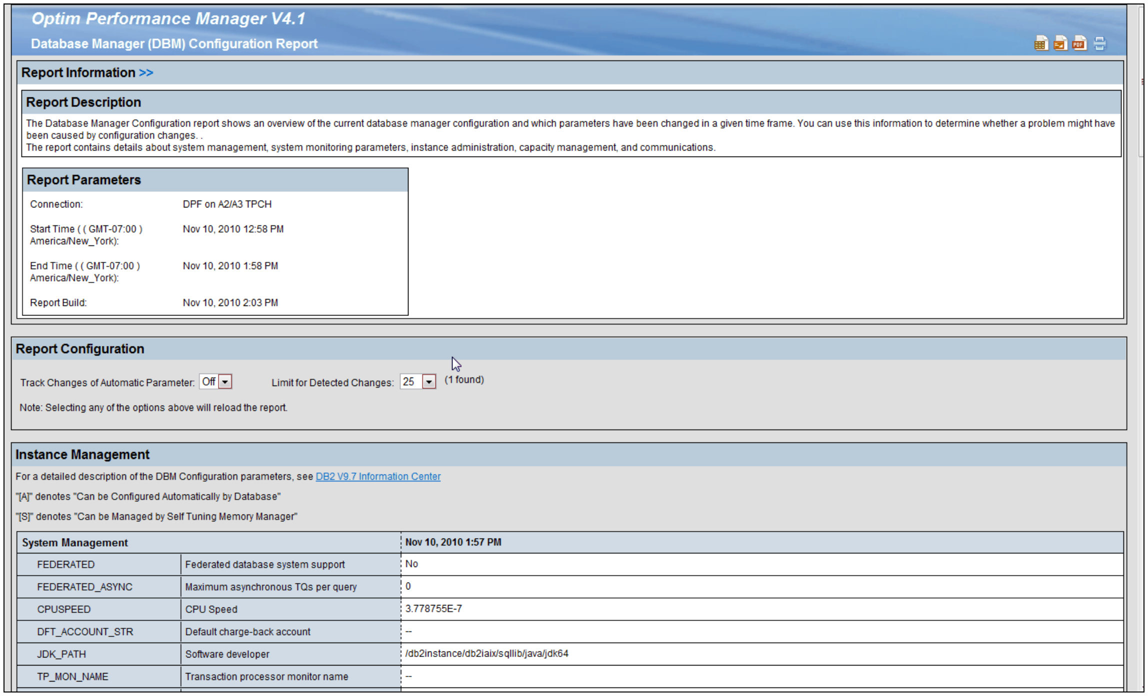


Figure 4-52 Database Manager Configuration report

4.5.3 Database Connection report

Database Connection report shows an overview of the active database connections at a specific time. The report displays key performance indicators, such as lock wait times, physical and logical reads and writes, and other connection statistics to help you identify applications that are not performing well or applications that are causing problems in other database applications. For DB2 partitioned databases, this report displays the connection information for all partitions. You can click on individual partition in the Active connection section of the report to see the connection information for the selected partition.

You can navigate between snapshots in time to identify when problems occurred or to see the activity of the database application over time.

Figure 4-53 on page 196 shows the Active connections in a Database connection report.

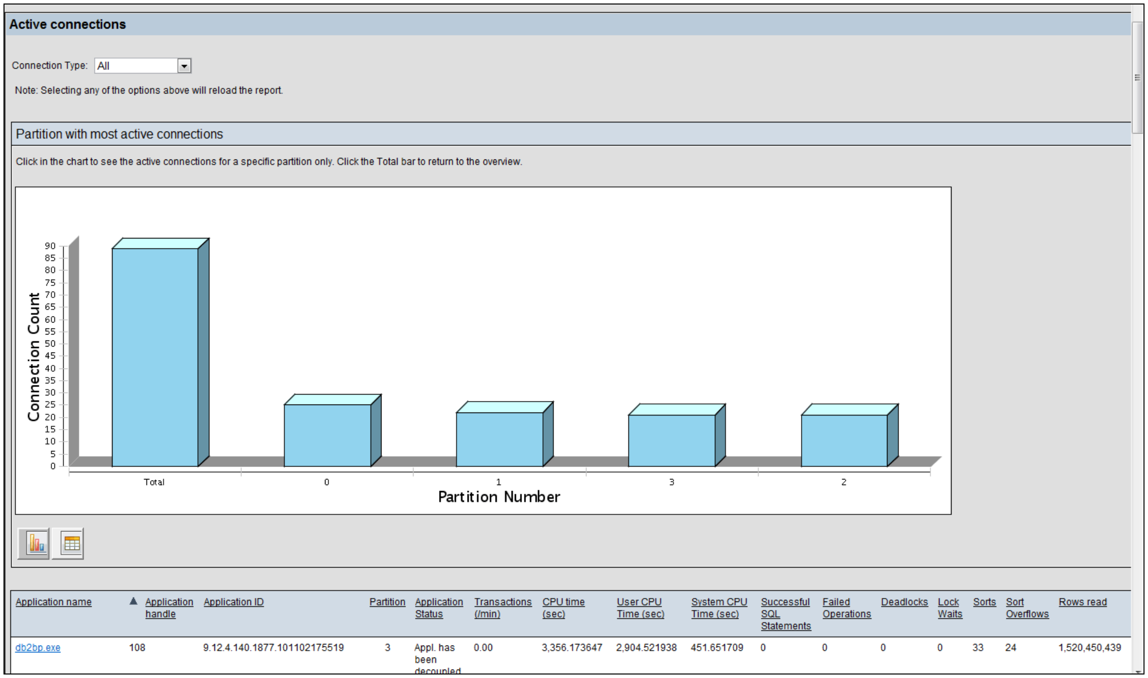


Figure 4-53 Database connection report

By clicking on individual database connection you can drill down to see the details of the connection in a Database Connection Detail report (Figure 4-54 on page 197). It shows information about the selected connection, such as complete identification details, timing information, SQL activity, locks, cache, buffer pool, sorts, and agent-related activity.

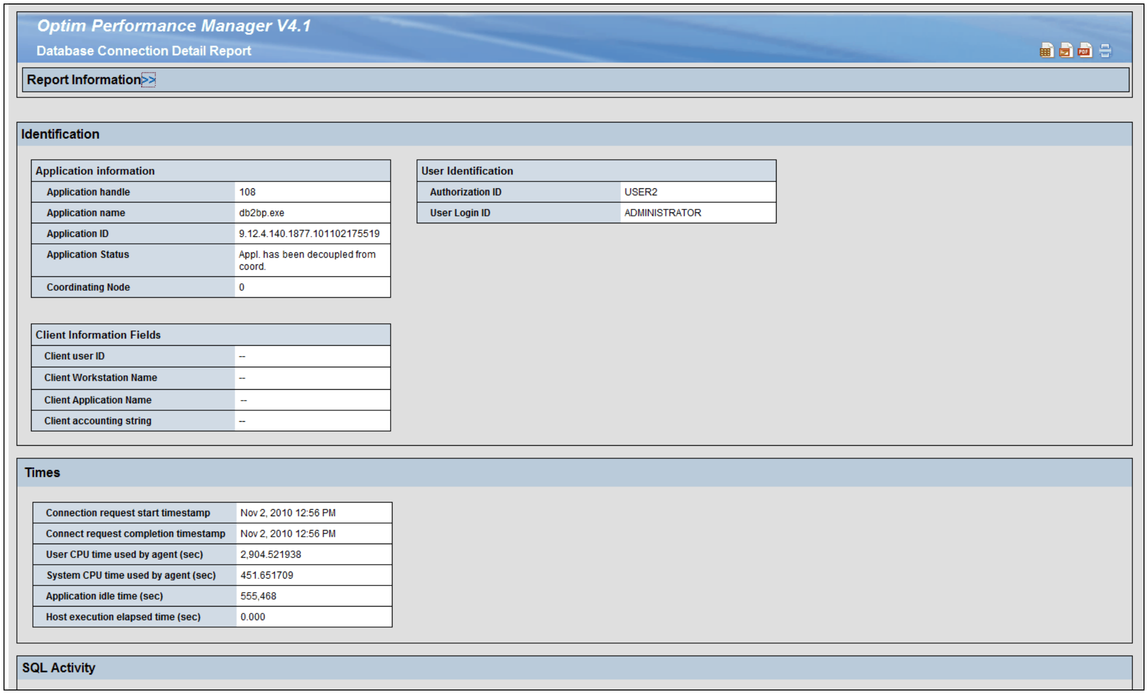


Figure 4-54 Database connection detail report

4.5.4 Disk Space Consumption report

The Disk Space Consumption report (Figure 4-55 on page 198) shows an overview of the current disk space usage by table space. You can analyze information about growth rates to plan for future disk space requirements or table space configuration changes.

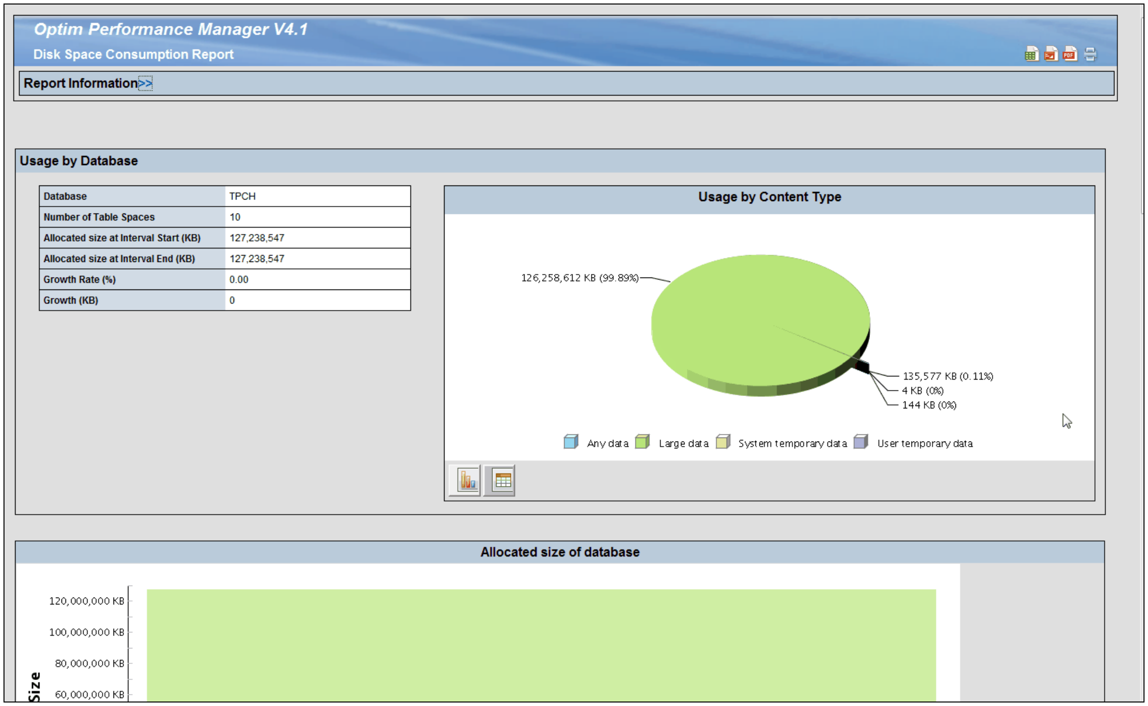


Figure 4-55 Disk space consumption report

You can drill down to review the details of a given table space, such as containers and tables, container layout, and data skew, in the Disk Space Consumption Detail report (Figure 4-56 on page 199).

The Disk Space Consumption Detail report shows details about table space configuration, container details, ranges, table space layout, and active tables under a specific table space. The report includes information for the table space, such as general information about the table space, size of the table space, storage information, and the variation in size of the table space over time.

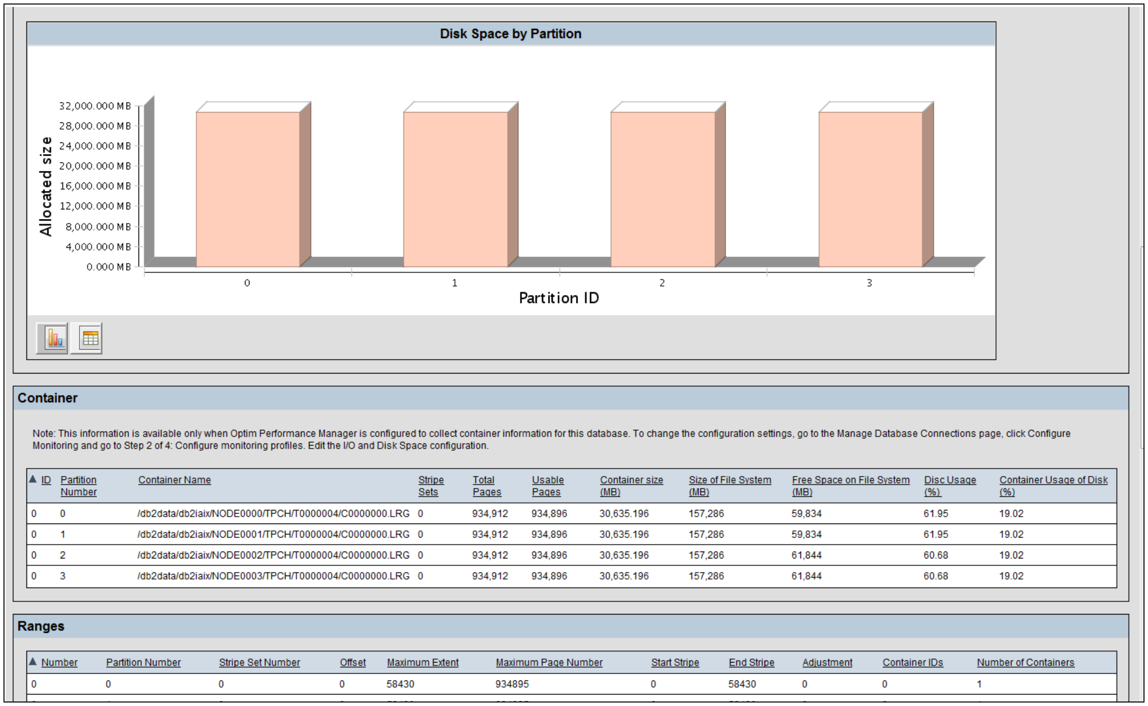


Figure 4-56 Disk Space Consumption Detail report

4.5.5 Dynamic SQL Statement report

The Dynamic SQL Statement report (Figure 4-57 on page 200) identifies the SQL statements that consume the most resource in a given period of time. The report includes a graphical representation of the workload over time so that you can easily identify critical or problematic SQL statements.

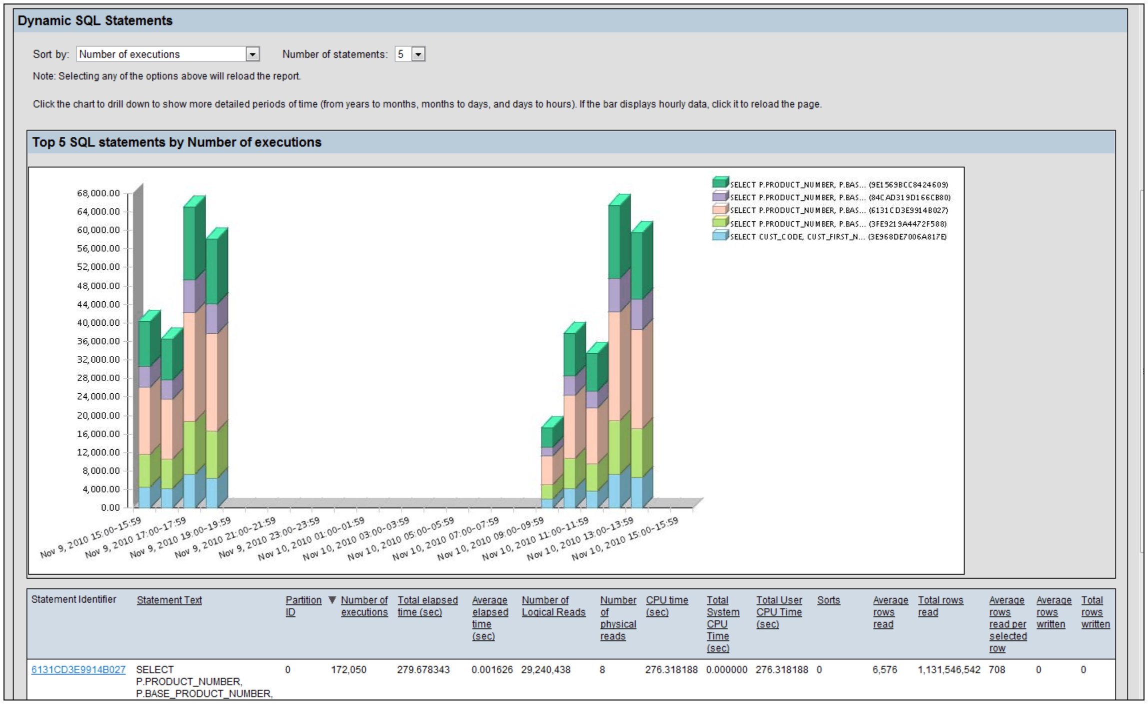


Figure 4-57 Dynamic SQL statement report

You can drill down to further analyze the activity and behavior of a given SQL statement in the Dynamic SQL Statement Detail report (Figure 4-58 on page 201).

The Dynamic SQL Statement Detail report shows an analysis of a specific SQL statement. The report includes detailed information, such as the complete statement text, general statement relation information, response time analysis, sort performance, I/O activity, and buffer pool activity.

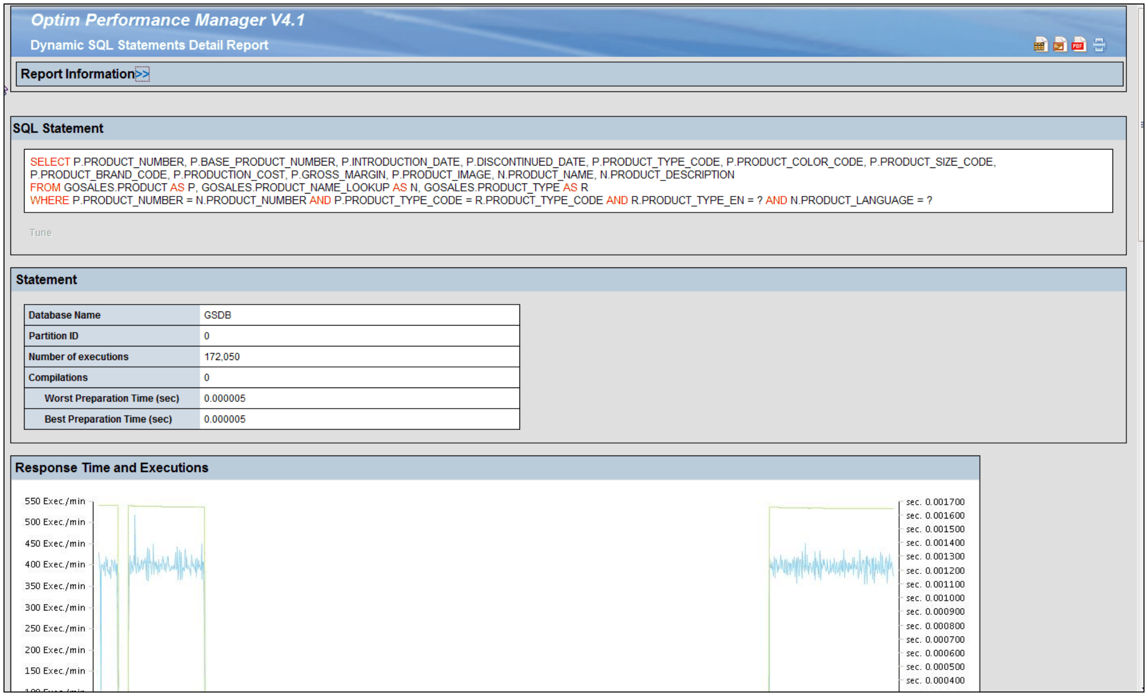
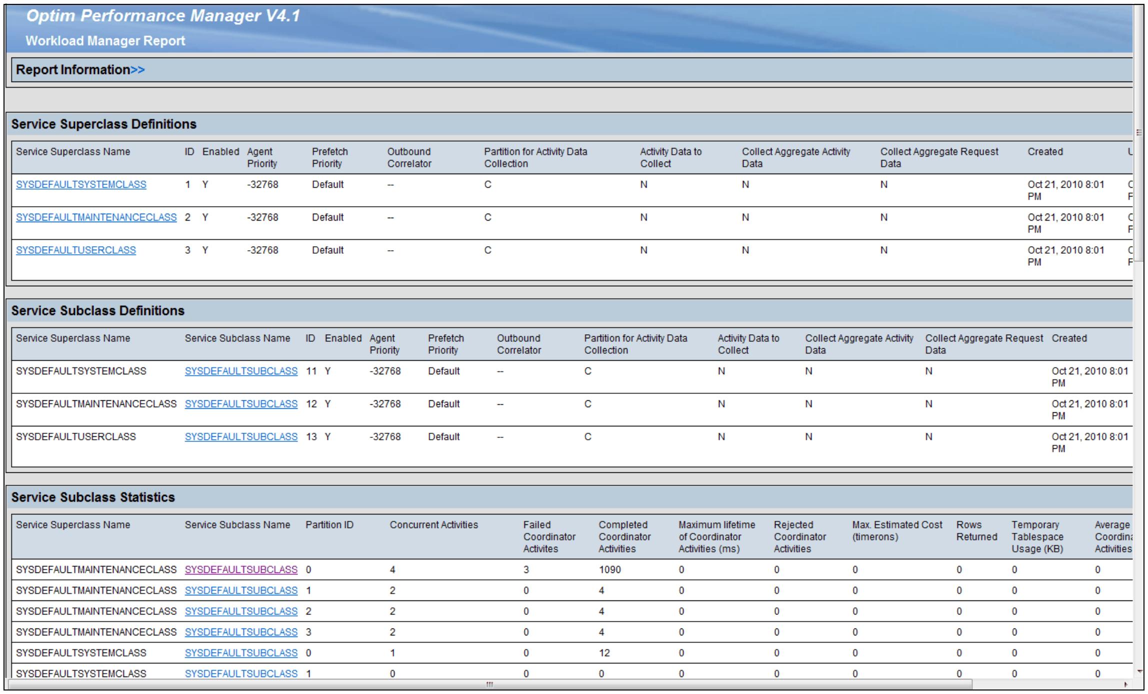


Figure 4-58 Dynamic SQL statement detail report

4.5.6 Workload Manager Configuration And Metrics report

This report provides an overview of your Workload Manager configuration and an overview of statistics related to your workload manager objects: service superclasses, service subclasses, workloads, and work classes. You can click the links in this report to view detailed statistics related to your workload manager objects. It also presents a summary of the statistics summed up for the service subclasses, workloads and workclasses.

Figure 4-59 on page 202 shows a WorkloaD Manager Configuration And Metrics report.



Optim Performance Manager V4.1

Workload Manager Report

Report Information>>

Service Superclass Definitions

Service Superclass Name	ID	Enabled	Agent Priority	Prefetch Priority	Outbound Correlator	Partition for Activity Data Collection	Activity Data to Collect	Collect Aggregate Activity Data	Collect Aggregate Request Data	Created
SYSDEFAULTSYSTEMCLASS	1	Y	-32768	Default	--	C	N	N	N	Oct 21, 2010 8:01 PM
SYSDEFAULTMAINTENANCECLASS	2	Y	-32768	Default	--	C	N	N	N	Oct 21, 2010 8:01 PM
SYSDEFAULTUSERCLASS	3	Y	-32768	Default	--	C	N	N	N	Oct 21, 2010 8:01 PM

Service Subclass Definitions

Service Superclass Name	Service Subclass Name	ID	Enabled	Agent Priority	Prefetch Priority	Outbound Correlator	Partition for Activity Data Collection	Activity Data to Collect	Collect Aggregate Activity Data	Collect Aggregate Request Data	Created
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	11	Y	-32768	Default	--	C	N	N	N	Oct 21, 2010 8:01 PM
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	12	Y	-32768	Default	--	C	N	N	N	Oct 21, 2010 8:01 PM
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	13	Y	-32768	Default	--	C	N	N	N	Oct 21, 2010 8:01 PM

Service Subclass Statistics

Service Superclass Name	Service Subclass Name	Partition ID	Concurrent Activities	Failed Coordinator Activites	Completed Coordinator Activities	Maximum lifetime of Coordinator Activities (ms)	Rejected Coordinator Activities	Max. Estimated Cost (timerons)	Rows Returned	Temporary Tablespace Usage (KB)	Average Coordina Activities
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	0	4	3	1090	0	0	0	0	0	0
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	1	2	0	4	0	0	0	0	0	0
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	2	2	0	4	0	0	0	0	0	0
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	3	2	0	4	0	0	0	0	0	0
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	0	1	0	12	0	0	0	0	0	0
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	1	0	0	0	0	0	0	0	0	0

Figure 4-59 WorkloaD Manager Configuration And Metrics report

4.6 Performance Expert client

Performance Expert client is an optional graphical user interface of Optim Performance Manager product. It originated with the DB2 Performance Expert product, and provided a client interface for it. It is a Java based application, which has to be installed on every user workstation that requires access to Optim Performance Manager product.

The new Optim Performance Manager web based client interface requires zero footprint on user's workstation. It contains majority of the functionality of the Performance Expert client.

If you have migrated to Optim Performance Manager from DB2 Performance Expert product, you may want to use Performance Expert client alongside with the new web interface for a smoother migration.

You can also perform the following tasks by using DB2 Performance Expert Client only:

- In addition to the features of the Workload Manager tool and the Workload Manager report, you can monitor activities for a specified time frame by starting Workload Manager activity traces.
- In addition to the features of the Reporting feature, you can do long-term performance analysis through Performance Warehouse.
- In addition to the features of the Optim Performance Manager plug-in for Tivoli Enterprise Portal, you can monitor and analyze operating system performance after installing the optional Common Information Model (CIM) server component.
- Real time database monitoring.
- More detailed monitoring of partitioned DB2 databases.
- User defined threshold sets.
- Trace the SQL statements of one or multiple connections for a specified timeframe by starting SQL Activity traces.
- Profile SQL PL stored procedures.

5

Monitoring I/O utilization

This chapter discusses the use of Optim Performance Manager to better manage I/O performance. The basic steps involved include:

- Identify high I/O that affect your DB2 response time using alerts on the different dashboards of the Optim Performance Manager.
- Drill-down into problem detail and analyze the high I/O utilization on your monitored database.
- Resolve the high I/O causes and improve the performance of your applications by using expert advice.
- Prevent problems by monitoring historical trends for planning and building performance from the ground up.

5.1 Symptoms of high I/O utilization

Performance refers to the way that a system behaves in response to a particular application. It can be measured in terms of system response time and resource utilization. Performance is generally affected by:

- The resources that are available on your system.
- How well those resources are used and shared.

I/O performance can play an important part in the health of a database system. Possible reasons for high I/O could be any of the following:

- Too much activity on a single disk
- Small buffer pool
- No indexes
- Bad plans
- Poor SQL
- Hardware issues

It is important to isolate the scope of the high I/O activity. In this section, we discuss how to narrow down a high I/O activity seen on the operating system level down to the database object and the application consuming I/O.

5.2 Monitoring high I/O through various alerts

The environment that we are using here is two physical AIX machines with a partitioned database with four partitions.

You can see the I/O utilization of your database system using operating system commands such as `iostat` that reports central processing unit (CPU) statistics, asynchronous input and output (AIO), and input and output statistics for the entire system. The `iostat` command is used to monitor system I/O devices (physical and logical) that are loaded, by observing the time for which these devices are active. You can also see the I/O activity details for your monitoring database using the Buffer pool and I/O dashboard in the Optim Performance Manager. This dashboard lets you check buffer pools, tables paces, and tables performance. It helps identifying hot objects and moving them to dedicated buffer pools. It also lets you check the appropriate size of a buffer pool as well as check the disk space and container definition of table spaces.

We start by seeing the Health Summary dashboard on the Optim Performance Manager. From Figure 5-1 on page 207 we see critical alerts for memory usage, storage, workload, I/O, and locking.

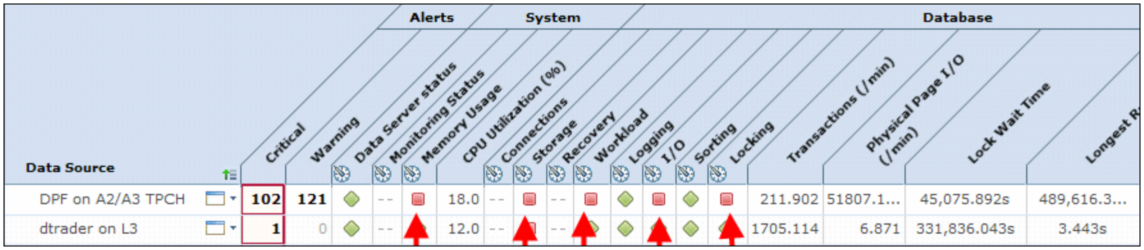
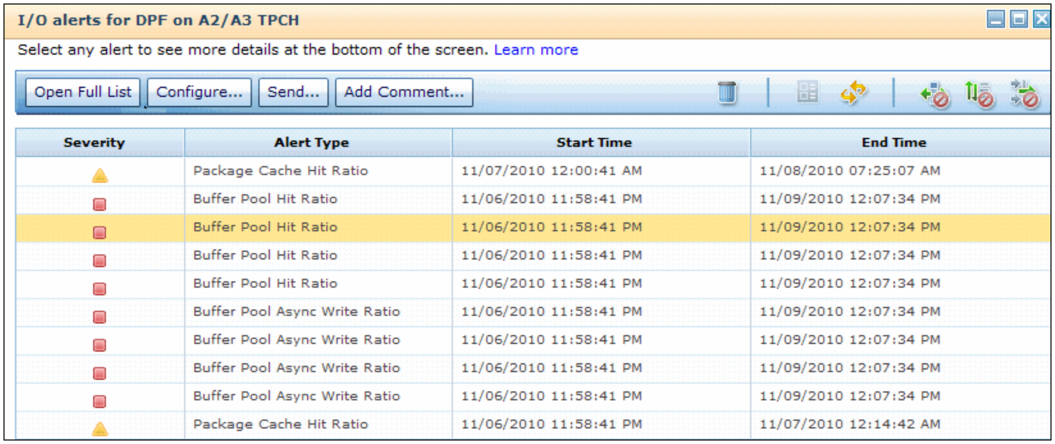


Figure 5-1 Part of the Health Summary dashboard

Clicking the red I/O alert icon gives you an overview of alerts of the monitored databases. The alert list provides information about active I/O alerts and about all the I/O alerts that occurred during the monitored period.

Figure 5-2 shows some of the I/O alerts that have occurred for our monitored database `DPF`. Let us go through them:

- Buffer Pool Hit Ratio: This is an I/O alert.
- Buffer Pool Async Write Ratio: This is an I/O alert.
- Buffer Pool Async Read Ratio: This is an I/O alert.



Severity	Alert Type	Start Time	End Time
	Package Cache Hit Ratio	11/07/2010 12:00:41 AM	11/08/2010 07:25:07 AM
	Buffer Pool Hit Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Buffer Pool Hit Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Buffer Pool Hit Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Buffer Pool Hit Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Buffer Pool Async Write Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Buffer Pool Async Write Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Buffer Pool Async Write Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Buffer Pool Async Write Ratio	11/06/2010 11:58:41 PM	11/09/2010 12:07:34 PM
	Package Cache Hit Ratio	11/06/2010 11:58:41 PM	11/07/2010 12:14:42 AM

Figure 5-2 I/O alerts for the A2/A3 TPCH database - from the Health Summary dashboard

In Figure 5-3 on page 208, “Rows Read per fetched row” is one of the Workload alerts that have occurred for our monitored database `DPF` during the same time frame as the I/O alert which is also of interest to us. This alert is from the Workload category, not from the I/O category. However, this does affect I/O so we talk about this alert here.

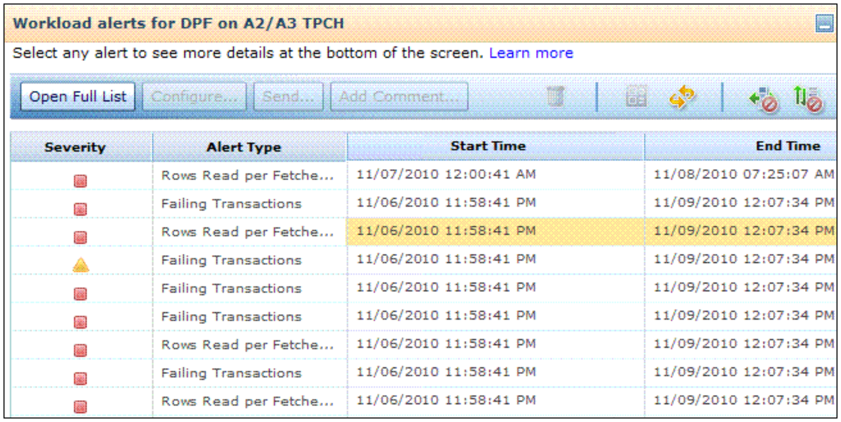


Figure 5-3 Workload alerts from the Health Summary dashboard

Before we drill down for more details, we take a look at the Overview Dashboard (Figure 5-4) to see if there are obvious hints about problems in I/O and Disk Space, System which includes CPU utilization and memory areas, Logging, Locking, and Sorting sections.

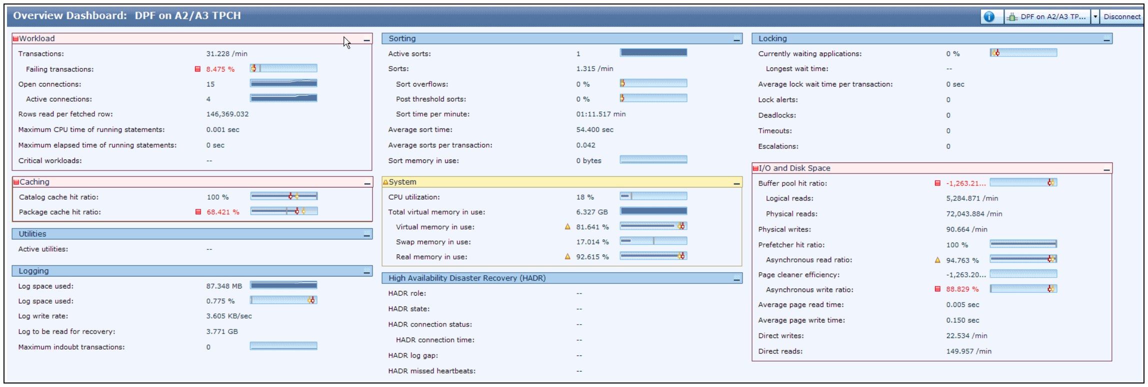


Figure 5-4 Overview Dashboard

From the Overview Dashboard, other than I/O, all the other sections look good. There are alerts for failing transactions. This is caused by failing statements in a workload that is indicated by a package cache alert. We discuss the workload and package cache alert in Chapter 6, “Monitoring CPU and memory usage” on page 217. Here we focus on the I/O alerts.

Figure 5-5 on page 209 shows the I/O and disk space information shown in the Overview Dashboard. The buffer pool hit ratio is negative, many physical reads is high, and the asynchronous write ratio is 88.8%. Let us understand what each of these mean.

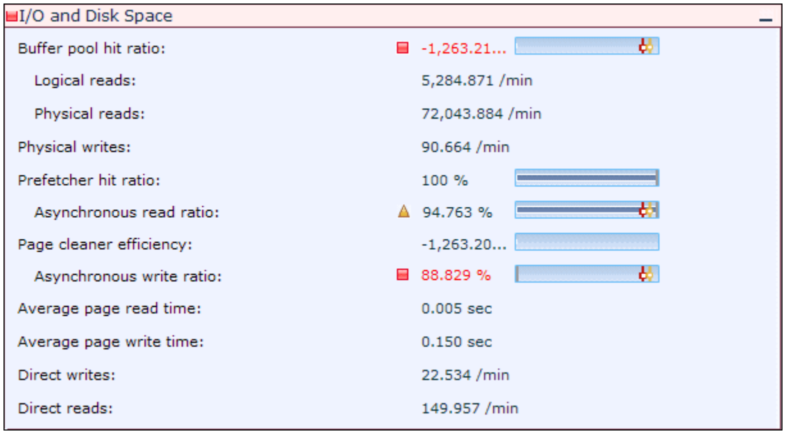


Figure 5-5 I/O and disk space information in the Overview Dashboard

Alerts are enabled with default threshold values which are set when you configure a database for monitoring. Thresholds determine when an alert is triggered. The key performance indicators are checked periodically by using a default sampling rate. If a threshold is reached, an alert is generated and an indicator appears on the Health Summary, the Alerts list, and the associated dashboard.

Now we look at the I/O alerts from the Health Summary dashboard for our database.

Buffer pool hit ratio

A good metric for buffer pool monitoring is the overall buffer pool hit ratio. We see from the Overview dashboard that the buffer pool hit ratio is -1,263% which is terrible. A low hit ratio indicates that some of the data that the application requested was not in the buffer pool. As a result, the data was read from the disk rather than from the buffer pool. Reading the data from the disk requires more time and resources.

In our case, we see the ratio is not only low but in negative. A negative hit ratio means that the prefetcher has brought pages into the buffer pool that are not subsequently referenced. The pages are not referenced because either the query stops before it reaches the end of the table space or DB2 must take the pages away to make room for newer ones before the query can access them. For our scenario, the second reason could be true because we see a workload alert for high Rows Read per fetched row. This alert means that DB2 is reading huge number of rows for every fetched row and, therefore, could be taking pages away to make room for newer ones. All pages are read although they are not really needed and all do not fit in the buffer pool, therefore, must be freed. The same reason is true for negative page cleaner efficiency too which is discussed in "Buffer pool async write ratio" on page 212. Another reason could be a partitioned database system where the data is not balanced and is spanned across partitions which could lead to low hit ratio, but again that alone does not justify this terrible ratio.

We check the Buffer Pool and the I/O Dashboard for details about the I/O activities on the buffer pool, table spaces, and tables on our database. Figure 5-6 on page 210 shows the number of logical reads and the physical reads per minute for the buffer pool, `BP32K`, which is the buffer pool used by the application.

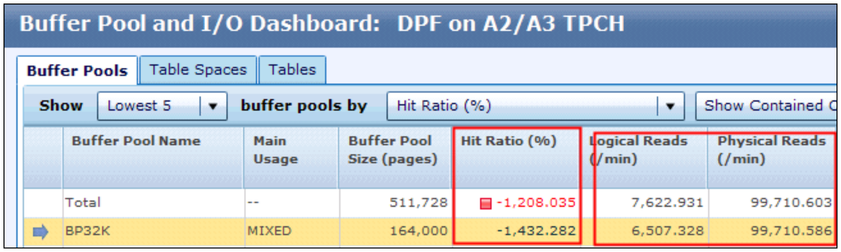


Figure 5-6 Part of the Buffer Pool and I/O Dashboard - Buffer Pools tab

We see that the physical reads are much higher than the logical reads that indicates that DB2 had to perform a lot of read access to data on disk. These high physical reads gives a good idea on the amount of physical versus logical reads taking place on the database, for tracking purposes, and establishing a baseline. This baseline can be useful to identify if an I/O bound system is a result of an increased number of physical reads. We can increase the buffer pool size and rerun our workload. If the physical reads have decreased and logical reads have increased, we know that buffer pool size is the cause of the I/O issue.

From the data so far we can make a note to try to increase the buffer pool size. However, increasing the buffer pool size may address only one of the causes of the high I/O issue. We continue analyzing for other reasons that could be causing our I/O issue.

We check the table spaces that use this buffer pool by selecting the Table Spaces tab (Figure 5-7). An alternative approach to come to this panel is from the Buffer Pool and I/O Dashboard, select the buffer pool, and click **Show Contained Objects**.

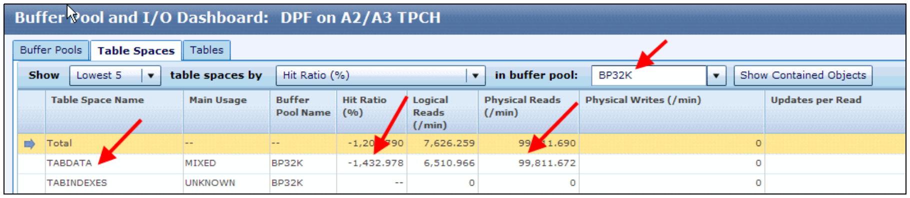


Figure 5-7 Part of the Buffer Pool and I/O Dashboard - Table Spaces tab

We see that there are two table spaces associated with this buffer pool namely, `TABDATA` and `TABINDEXES`. The low buffer pool hit ratio and the high physical reads are associated with the `TABDATA` table space. Note that the `TABINDEXES` table space has no activity.

To see the buffer pool activity details of the tables associated with the `TABDATA` table space, select the Tables tab (Figure 5-8) or selecting **`TABDATA`** and click **Show Contained Objects**.

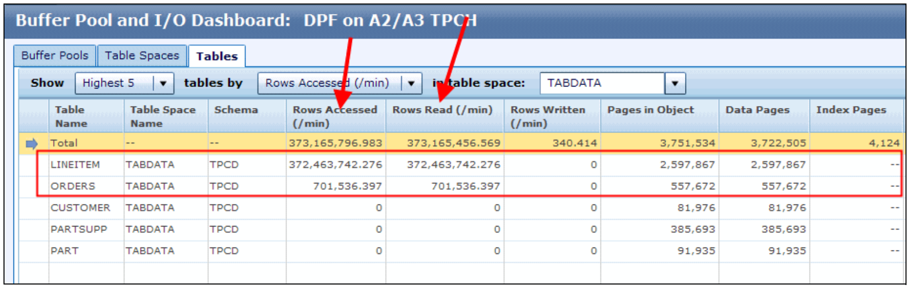


Figure 5-8 Part of Buffer Pool and I/O Dashboard - Tables tab

We see the top two tables on the list for the `TABDATA` table space are `LINEITEM` and `ORDERS`. These are two heavy activity tables where most of the I/O is being done within the database. These are generally referred to as “Hot Tables” too.There are few things to note here:

- Rows Accessed per minute
- Rows Read per minute
- Index Pages

The number of rows accessed per minute is the sum of rows read and rows written. Because we have no rows written, the number of rows accessed is the same as the rows read. It is important to note that the number of rows read is high for a period of two hours. (We selected the time slider to show the data for the past two hours.) There are no index pages which gives an indication that there is no index associated with the tables and that we could benefit from creating an index.

Buffer pool async write ratio

This is the ratio between the asynchronous writes and the total buffer pool write I/O. A high ratio indicates that the changed data in the buffer pools is written asynchronously by the I/O servers to the disk and that the applications are not waiting. A low value may indicate that you do not have enough I/O servers.

In Figure 5-5 on page 209, the *asynchronous write ratio* is about 89% which is a hint of high amount of I/O to the physical disk. This towards the higher side but is not too bad at this point.

The *Page cleaner efficiency* shows a negative percentage. Note that it is not showing us an alert because at the time of writing, Optim Performance Manager did not provide the possibility to specify alert thresholds for this metric. The negative page cleaner efficiency is resulted from the negative buffer pool hit ratio.

The buffer pool consists of pages that are either *in use*, meaning the pages are being updated or read, or *dirty*, meaning the pages have not yet been written to disk. Once the dirty pages are written to disk, they remain in the buffer pool, but their status changes to “clean for reuse” or “for continual use” by other database transactions. This is where the page cleaner (`NUM_IOCLEANERS`) comes into play. The page cleaners write changed (dirty) pages from the buffer pool to disk. As a result, application transactions are faster because DB2 agents do not have to wait idle for I/O.

If your database usage is only for querying, it is safe to leave `NUM_IOCLEANERS` set at its default of 1. If your application will be doing updates, inserts, and other action transactions, we recommend to set this value to at least the number of physical disks in the database. Due to the high number of physical reads in our example, the space must be freed in the buffer pool by the page cleaners in order to read the data in. This cannot be done efficiently by the page cleaners due to the high amount of physical reads and we could benefit from an index.

Buffer pool async read ratio

This is the ratio between the asynchronous reads and the total buffer pool read I/O. A high ratio indicates that the majority of data requested was read by prefetchers. Prefetching is the retrieval of data (one or more extent pages) from disk in anticipation of their use. This can significantly improve performance in SELECT statements by reducing the time waiting for I/O to complete. Setting the `PREFETCHSIZE` tells DB2 to place that number of pages in the buffer pool in anticipation of its use. The default is 32.

A low async read ratio may indicate that the prefetch size is too small. Figure 5-5 on page 209 shows the ratio is almost 95% which is really good for asynchronous read ratio. The high value in our example indicates that the majority of the data requested was read by prefetchers. When DB2 has to scan many pages to find result sets, usually because indexes are missing or sub-optimally defined, DB2 uses asynchronous prefetch I/O. A high percentage of asynchronous I/O indicates that DB2 is doing a lot of scanning to find result sets. Scans occur when indexes are missing or are sub-optimally defined.

Rows read per fetched row

Though *Rows read per fetched row* is not an I/O alert, this alert does affect high or low I/O depending on its value. Let us look at the Workload section from the Overview Dashboard shown in Figure 5-9 on page 213.

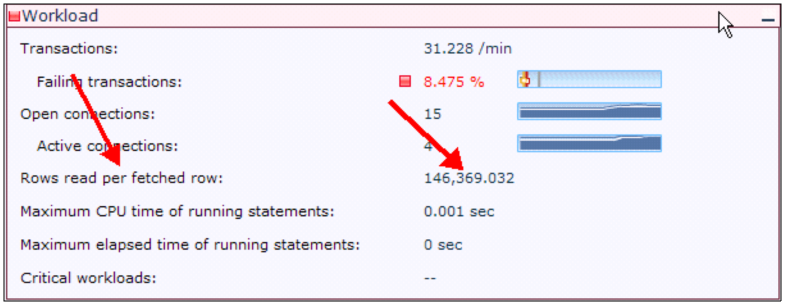


Figure 5-9 Part of Overview Dashboard - Workload section

Rows read per fetched row is the number of rows that were read from the table for each row that was returned to the application. A high number indicates a table scan, which can be improved by indexing. In our scenario we see a terrible number for Rows read per fetched row which gives another clue that we can benefit from using an index on the high activity table.

5.3 Resolving high I/O problem to improve performance

From our analysis, we saw a few possible causes for the high I/O in our system: a small buffer pool and lack of indexes on the heavy activity tables. Here we demonstrate the performance effect on creating an index on the heavy access table.

You can see the running statements from the Active SQL dashboard. If Optim Query Tuner has been set up on the same machine where user runs the browser to access the Optim Performance Manager web console, users can perform SQL tuning in Optim Query Tuner. We discuss SQL query tuning in more detail in Chapter 8, “Extended Insight analysis” on page 271.

In our scenario, the Query Tuner recommended to create an index on the `LINEITEM` table. Indexes can be stored in a different table space from the table data. When indexes and data are in same table spaces, both data and index pages use the same extent size and prefetch quantity. Creating indexes in a separate table space reduce the I/O traffic to the index table space. You can also create index table spaces on faster physical devices. In addition, you can assign the index table space to a different buffer pool, which might keep the index pages in the buffer longer because they do not compete with table data pages. In our example, we create the index in the `TABINDEXES` table space, the `bp32k` buffer pool using the following command:

```
CREATE UNIQUE INDEX tpcd.l_okln ON tpcd.lineitem (l_orderkey ASC,
l_linenumber ASC) PCTFREE 3 ;
ALTER TABLE tpcd.lineitem ADD PRIMARY KEY (l_orderkey, l_linenumber);
```

After you create a new index, run the RUNSTATS utility to collect index statistics. These statistics allow the optimizer to determine whether using the index can improve access performance.

We then re-run our workload for about four hours and check the Health Summary dashboard on the Optim Performance Manager. The critical alerts for Storage, Workload, and I/O remains as shown in Figure 5-10.

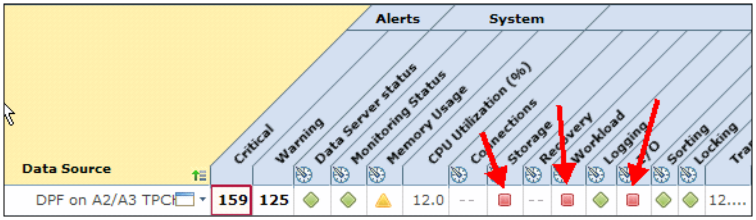


Figure 5-10 Part of the Health Summary Dashboard

We click the red square alert icon to see the details of the I/O alert. Figure 5-11 on page 215 shows that the buffer pool hit ratio is still one of the top alerts in our partitioned database.

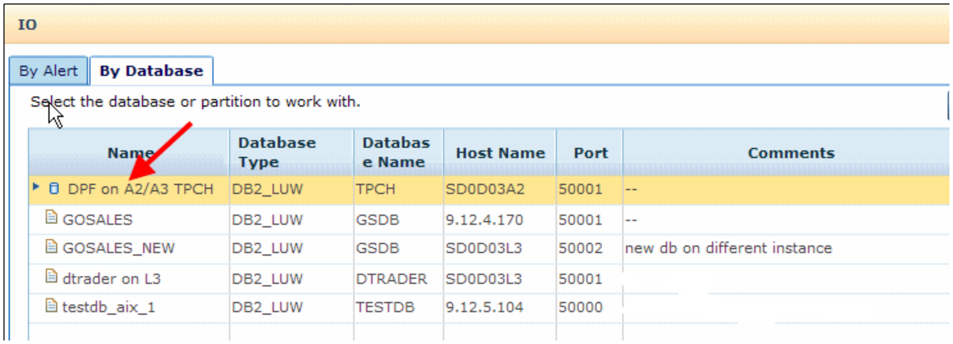


Figure 5-11 I/O alerts for the DPF on A2/A3 TPCH database from the Health Summary dashboard

From the Overview Dashboard I/O and Disk Space section (Figure 5-12), we see improvements made by this added index:

- Buffer pool hit ration: From a negative number to 85.579%.
- Physical reads in comparison to the logical reads per minute:
  - Before index creation: 5K logical reads and about 72K physical reads per minute
  - After index creation: 180K logical reads and about 25K physical reads per minute

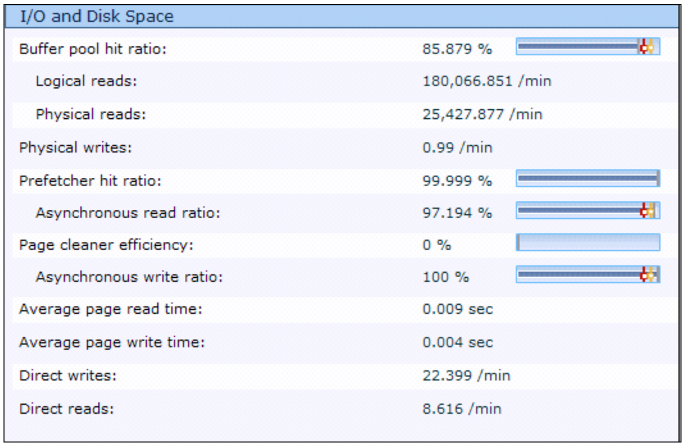


Figure 5-12 Details about I/O and disk space from the Overview dashboard

5.4 Preventing high I/O

In a production environment, database performance tuning is a complicated task and should not be entered into for the reason of only gaining performance. First determine if there is a performance problem. If so, identify it and work from there. When designing a database, you should know what kind of performance you expect from it beforehand. Poor planning can affect performance by a factor of over 60% compared to good planning.

Within DB2, large amounts of disk I/O is a commonly seen attribute to poor performance. Minimizing the number of times DB2 has to retrieve data from disk increases performance. (Some disk I/O, such as logging, is unavoidable.) Consequently, DB2 uses buffer pools to improve performance. Increasing the buffer pool size is always the first attempt and usually shows significant impact, especially if you started a new database with the default parameters. Increase the buffer pool with caution and check the resulting performance because every database has a point when adding memory to the buffer pool does not show much effect any more. Tune the `BUFFPAGE` (buffer pool size) until you receive a good hit ratio. When you have got your hit ratio up, continue tuning other parameters such as `PREFETCHSIZE`, `NUM_IOCLEANERS`, `CHNGPGS_THRESH`, and `NUM_IOSERVERS` until you get the expected results. (Use the Optim Performance Manager Dashboards and `iostat` to monitor I/O performance.)

Ensure that the tables are well maintained by running the `REORGCHK_TB_STATS` procedure and reorganize the table as needed. Although the optimizer decides whether to use an index to access table data, you must decide which indexes might improve performance and create these indexes. You must also run the RUNSTATS utility to collect new statistics about the indexes in the following circumstances:

- After you create an index
- After you change the prefetch size
- At regular intervals to keep the statistics current.

If your table has a lot of inserts but few updates, consider the APPEND option. It tells DB2 to write new records always to the end of the table and never to look for free space in between existing records (for example, space freed after deleting records). Use `ALTER TABLE schema.tablename APPEND ON/OFF` to enable or disable this option.

6

Monitoring CPU and memory usage

In this chapter, we describe how you can use Optim Performance Manager to analyze a high CPU utilization on your monitored system. We show the dashboards and reports displaying CPU metrics and how you can use the CPU and related metrics to find the cause of a high CPU utilization. We also show how Performance Expert client can give additional insight on CPU utilization.

Another topic discussed is how to use Optim Performance Manager to monitor the memory that DB2 allocates and uses for the instance, databases, and applications. We describe the information you see on the memory dashboard and introduce the DB2 memory model in high level. This helps you to understand how DB2 uses memory in order to detect any memory bottlenecks easily. Additionally, we introduce briefly the use of the database manager configuration and database configuration reports because the configuration influences the memory allocation and usage.

6.1 Monitoring CPU utilization

High CPU utilization situations can slow down the response time of your database system. If the CPU utilization is higher than normal or close to 100%, you should investigate this situation and find the reason for that. Possible reasons, for example, can be

- Execution of long running or not well-tuned SQL statements
- Execution of DB2 utilities such as LOAD
- Memory problems such as under-sized DB2 memory areas or over-committed system memory

You can see the CPU utilization of your database system using operating system commands such as `vmstat` or the Optim Performance Manager dashboards.

In this scenario, we monitor a partitioned database with four partitions on two physical machines. We notice a higher-than-normal CPU utilization of 71% shown on the Health Summary dashboard (Figure 6-1). In addition, there are critical red alert indicators for Memory usage, I/O, and Sorting. These are good candidates for being responsible for the high CPU utilization.

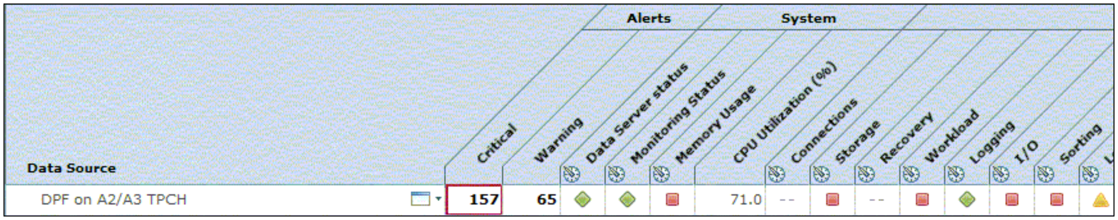


Figure 6-1 Part of the Health Summary dashboard

We open the Overview dashboard of this database to have more information about the utilization of CPU, memory, I/O, and Sorting. First we look at the CPU utilization and system memory usage information on the System section as shown in Figure 6-2 on page 219. The System section shows the following information about CPU utilization:

- The current average CPU utilization over all partitions as percentage value. This is 71 %.
- The average CPU utilization over all partitions and over the displayed monitoring time frame as bar in the bar chart. For our example, we set six hours as monitoring time frame to be displayed on the dashboard.

- The average maximum CPU utilization of the partitions over the displayed monitoring timeframe as vertical line in the bar chart. This indicates that we have higher CPU utilization on some partitions than others.

The memory usage is also high, the Real Memory is nearly all used, but Virtual Memory still has some room.

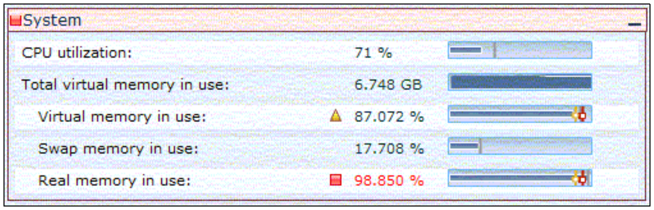


Figure 6-2 System section of Overview dashboard

Click on the bar chart next to the CPU utilization value to see the CPU utilization over time for the displayed monitoring time frame. In Figure 6-3, we see a continuous higher CPU utilization one some partitions than on others. The CPU utilization differs over time with higher than normal peaks in the first third of the time frame and at the end.

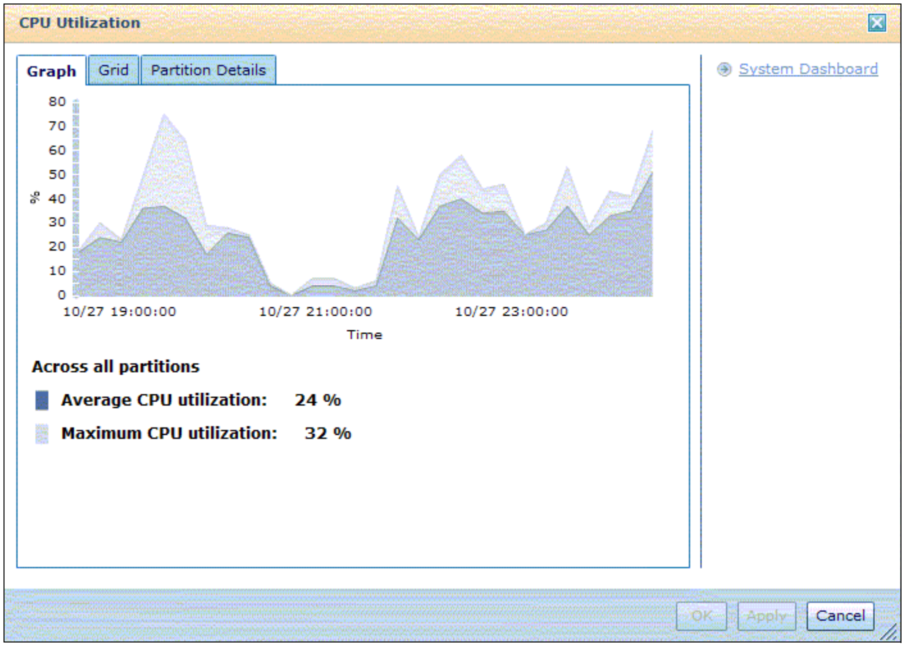


Figure 6-3 CPU utilization overlay on Overview dashboard

The System Dashboard link at the right side of the CPU utilization graph takes you to the System dashboard where you can look at the CPU utilization of each partition. Since partition 0 and partition 1 are on the same machine, the CPU utilization is the same for both partitions, same for partition 2 and partition 3. Figure 6-4 on page 220 shows that the CPU utilization on partition 0 and 1 is higher over time as for partition 2 and 3 (Figure 6-5). The peaks that we have seen in Figure 6-3 are coming from partition 0 and 1.

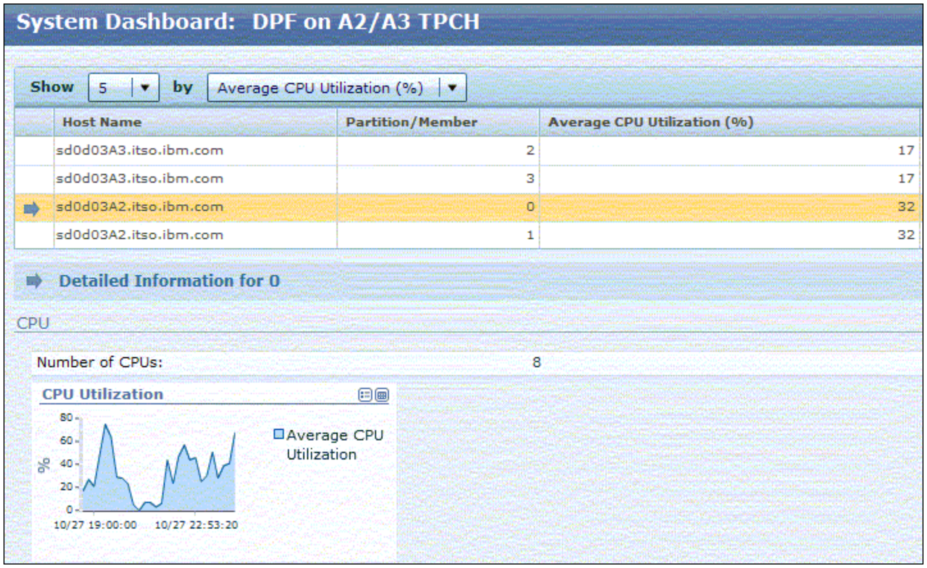


Figure 6-4 CPU utilisation on partition 0 and 1

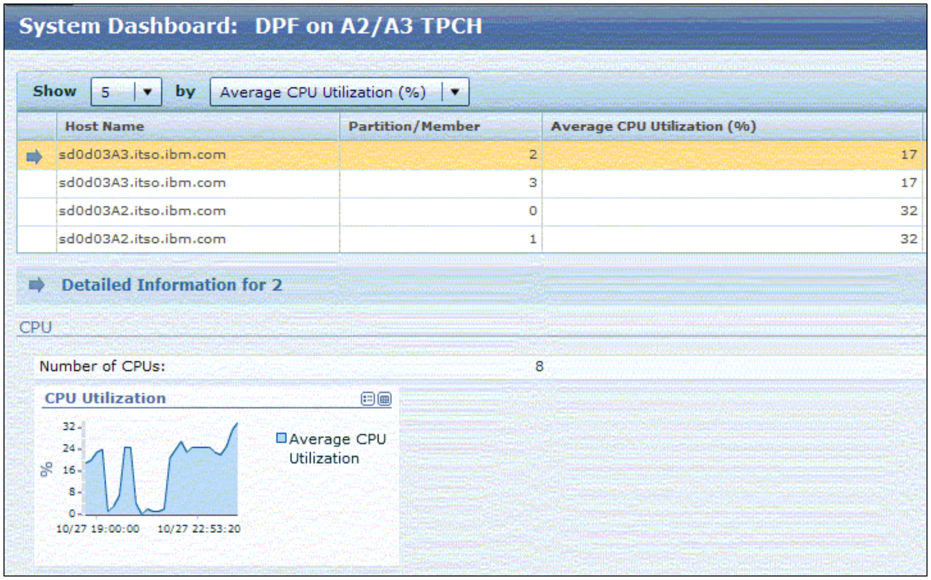


Figure 6-5 CPU utilization on partition 2 and 3

Let us go back to the Overview dashboard to see what hints we have for the high CPU utilization. The Workload section (Figure 6-6) shows, during the monitoring time frame, at least one statement took more than 17 minutes of the CPU time. The *Rows read per fetched row* number is high too. These provide us the clue that long running SQL statements are executed and because of the high rows read per fetched row ratio, the heavy queries could benefit from using an index.

In Figure 6-6, the red alert icon on the Caching section tells that the Package cache hit ratio is below the critical threshold. A low package cache hit ratio could indicate that a lot of time is spent in the SQL compiler that can lead to high CPU utilization. However, in this example, because we see more obvious reasons like the high CPU time of statements, the probability that the low package cache hit ratio is the primary reason for high CPU utilization is lower.

In the Utilities section in Figure 6-6, we see that four utilities were run in the monitoring time frame. This is also a good hint since utilities such as LOAD require high CPU time.

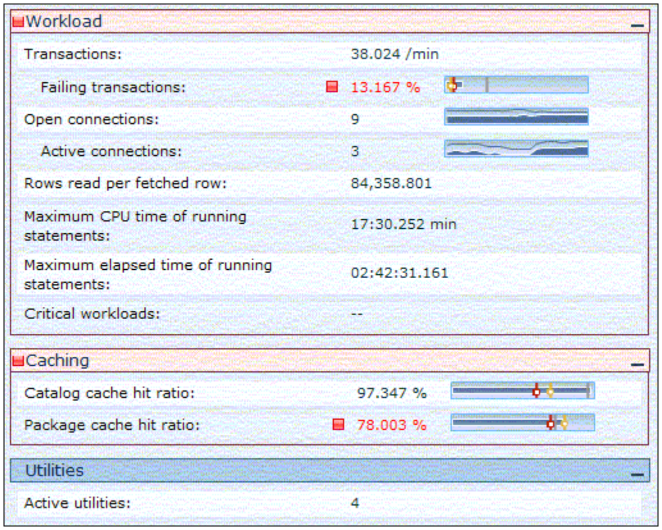


Figure 6-6 Metric sections on Overview dashboard

The Sorting section (Figure 6-7 on page 222) shows that there are sort overflows within the displayed monitoring time frame. The sort overflows can be an indicator of undersized DB2 memory areas. A high sort number can be reduced by tuning heavy statements. Because we already have indicators that the executed statements could benefit from tuning, we do not worry about the sort overflows for now.

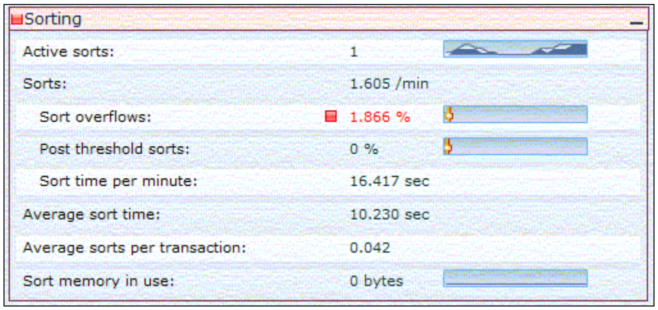


Figure 6-7 Sorting section on Overview dashboard

The I/O and Disk Space section (Figure 6-8) shows a very low buffer pool hit ratio. The reasons could be that either the buffer pools are too small or too many activities are on the buffer pool due to the execution of untuned statements. If an index is missing then probably a table scan took place that reads the entire table into the buffer pool. Both reasons can lead to the high CPU utilization.

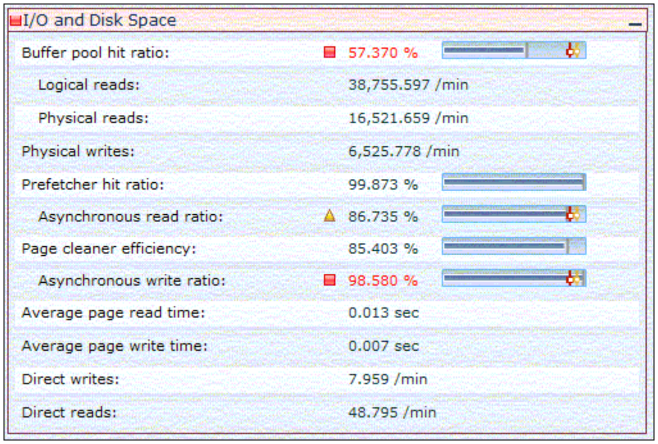
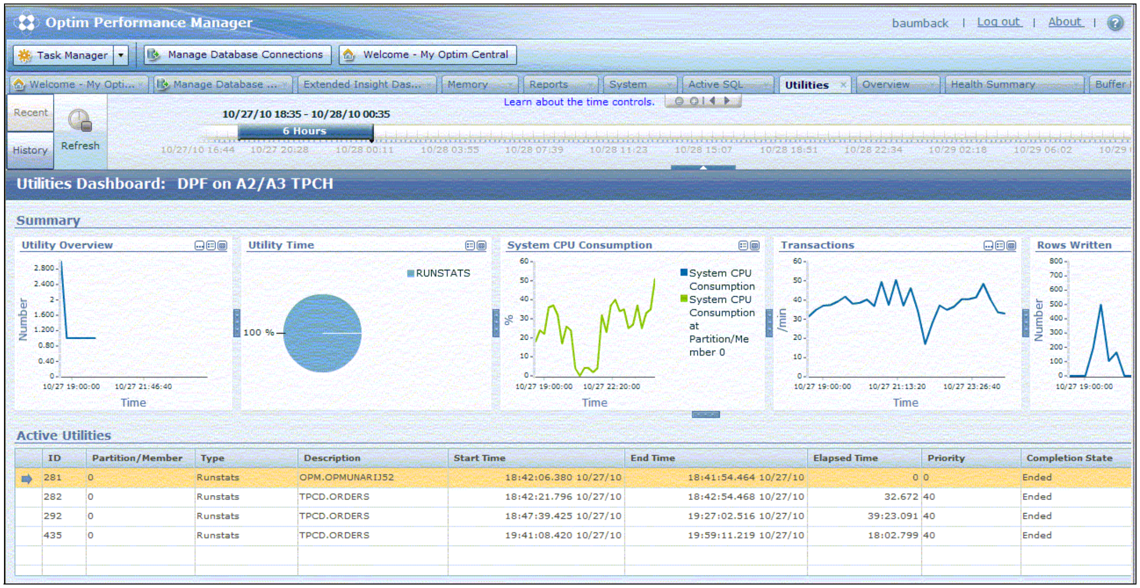


Figure 6-8 I/O section on Overview dashboard

To drill down further for the root cause of the high CPU utilization, we will check quickly what utilities were executed to find out whether they contribute to the high CPU utilization. After that, we look at the SQL statements that are running, then we check the buffer pool behavior.

6.1.1 Monitoring utility execution

The Utility dashboard in Figure 6-9 shows that in our 6 hours monitoring time frame, only the RUNSTATS utilities were executed on partition 0. From the Utility Overview graph you can see that these RUNSTATS jobs ran at the beginning of our monitoring interval. They add CPU consumption on partition 0 during the time frame of the first peak, but are not the main contributor to the CPU peaks that we see on System dashboard in Figure 6-4 on page 220 for partition 0 in the first third and at the end of the monitoring interval.



ID	Partition/Member	Type	Description	Start Time	End Time	Elapsed Time	Priority	Completion State
281	0	Runstats	OPM.OPMUNARIJ52	18:42:06.380 10/27/10	18:41:54.464 10/27/10	0	0	Ended
282	0	Runstats	TPCD.ORDERS	18:42:21.796 10/27/10	18:42:54.468 10/27/10	32.672	40	Ended
292	0	Runstats	TPCD.ORDERS	18:47:39.425 10/27/10	19:27:02.516 10/27/10	39:23.091	40	Ended
435	0	Runstats	TPCD.ORDERS	19:41:08.420 10/27/10	19:59:11.219 10/27/10	18:02.799	40	Ended

Figure 6-9 Utility dashboard

6.1.2 Monitoring statement execution

Here we look at the running SQL statements using the Active SQL dashboard, the Dynamic SQL statement report, and Performance Expert Client.

The Active SQL dashboard shown in Figure 6-10 on page 225 let you display the top N statements by your favorite metric that run in the specified monitoring interval. In our example we display the top 10 statements by CPU time. The table on the dashboard lists these statements including execution metrics. If you click a statement then you see more execution details in the lower part of the dashboard including:

- ► The complete statement text.
- ► Launching to the statement in Query Tuner.
- ► Stopping the statement execution if the statement is still running.
- ► Force the connections.
- ► Identify the workload the statement belongs to in the Extended Insight dashboard by clicking the Identify Workload button.

  At the time of the writing, clicking Identify Workload only opens the Extended Insight dashboard for the same database and monitoring time frame, but does not preselect any workload cluster group or workload cluster this statement belongs to.

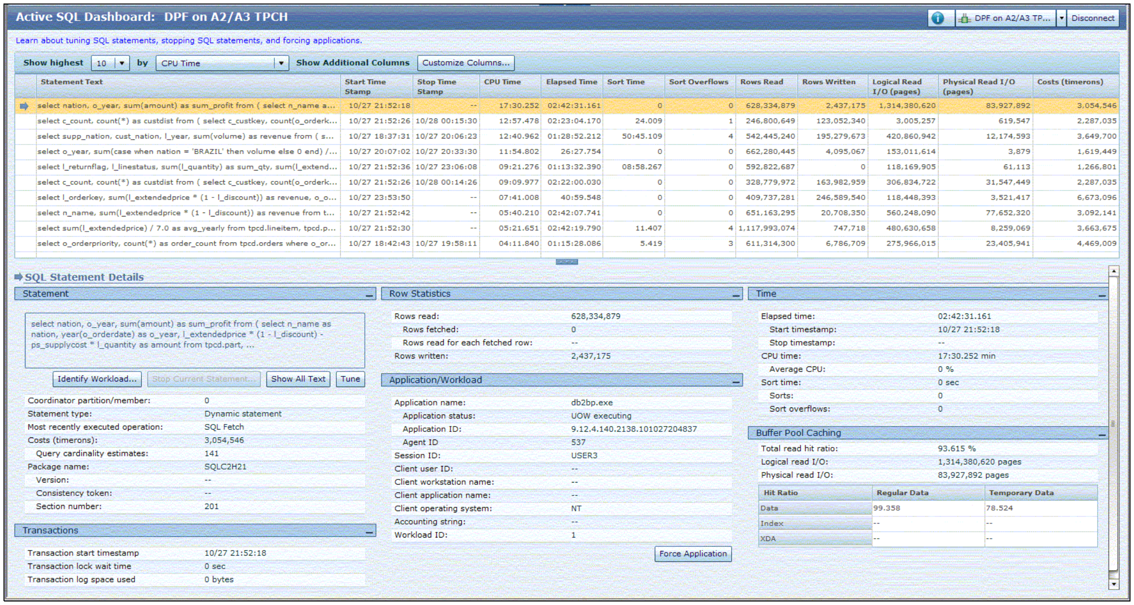
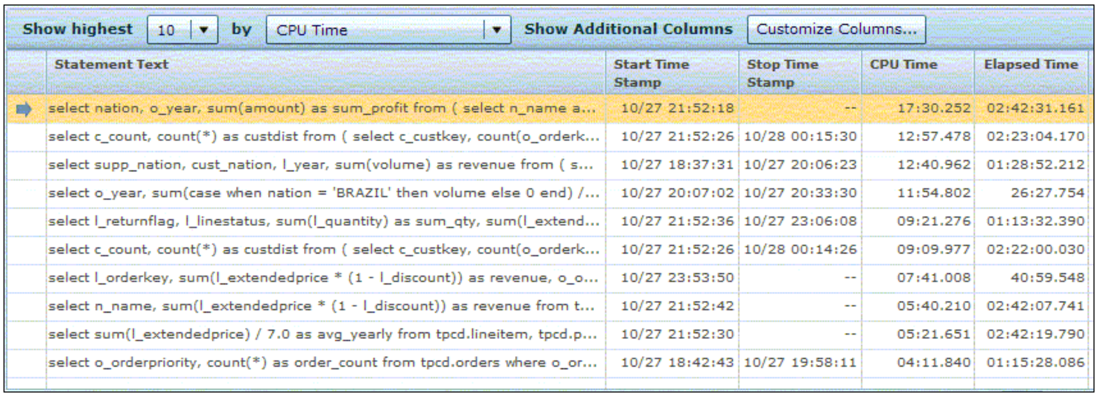


Figure 6-10 Active SQL dashboard

Let us take a closer look what we see on this dashboard. The listed statements shown in Figure 6-11 on page 225 are sorted by CPU time. The CPU time consists of user CPU and system CPU time. The first statement consumes most CPU and has not stopped yet at the end of the monitoring interval. Therefore, it contributes to the CPU peak we see at the end of the monitoring interval. The same is true for the other listed statements that have not finished yet.



Show highest 10 by CPU Time Show Additional Columns Customize Columns...

Statement Text	Start Time Stamp	Stop Time Stamp	CPU Time	Elapsed Time
select nation, o_year, sum(amount) as sum_profit from (select n_name a...	10/27 21:52:18	--	17:30.252	02:42:31.161
select c_count, count(*) as custdist from (select c_custkey, count(o_orderk...	10/27 21:52:26	10/28 00:15:30	12:57.478	02:23:04.170
select supp_nation, cust_nation, l_year, sum(volume) as revenue from (s...	10/27 18:37:31	10/27 20:06:23	12:40.962	01:28:52.212
select o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) /...	10/27 20:07:02	10/27 20:33:30	11:54.802	26:27.754
select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extend...	10/27 21:52:36	10/27 23:06:08	09:21.276	01:13:32.390
select c_count, count(*) as custdist from (select c_custkey, count(o_orderk...	10/27 21:52:26	10/28 00:14:26	09:09.977	02:22:00.030
select l_orderkey, sum(l_extendedprice * (1 - l_discount)) as revenue, o_o...	10/27 23:53:50	--	07:41.008	40:59.548
select n_name, sum(l_extendedprice * (1 - l_discount)) as revenue from t...	10/27 21:52:42	--	05:40.210	02:42:07.741
select sum(l_extendedprice) / 7.0 as avg_yearly from tpcd.lineitem, tpcd.p...	10/27 21:52:30	--	05:21.651	02:42:19.790
select o_orderpriority, count(*) as order_count from tpcd.orders where o_or...	10/27 18:42:43	10/27 19:58:11	04:11.840	01:15:28.086

Figure 6-11 Statements on Active SQL dashboard

Looking closer at the execution metrics in Figure 6-12, we see that high sorting and high I/O activity took place. For some statements the sort time is still 0, especially for statements that have not finished yet at the end of our monitoring interval. The reason is that DB2 updates the sort information only if either the execution of subsections or the entire statement has finished. The reported Rows Written also results from the sorting activities.

Tip: If you want to verify that DB2 spills out sorts to disk and therefore report the Rows Written activity for SELECT statements then you can open the I/O dashboard and check the activity going on in temporary table spaces.

Sort Time	Sort Overflows	Rows Read	Rows Written	Logical Read I/O (pages)	Physical Read I/O (pages)	Costs (timerons)
0	0	628,334,879	2,437,175	1,314,380,620	83,927,892	3,054,546
24.009	1	246,800,649	123,052,340	3,005,257	619,547	2,287,035
50:45.109	4	542,445,240	195,279,673	420,860,942	12,174,593	3,649,700
0	0	662,280,445	4,095,067	153,011,614	3,879	1,619,449
08:58.267	0	592,822,687	0	118,169,905	61,113	1,266,801
0	0	328,779,972	163,982,959	306,834,722	31,547,449	2,287,035
0	0	409,737,281	246,589,540	118,448,393	3,521,417	6,673,096
0	0	651,163,295	20,708,350	560,248,090	77,652,320	3,092,141
11.407	4	1,117,993,074	747,718	480,630,658	8,259,069	3,663,675
5.419	3	611,314,300	6,786,709	275,966,015	23,405,941	4,469,009

Figure 6-12 Execution metrics on Active SQL dashboard

We run our workload using the DB2 command line processor for this example. Though the Extended Insight client does not support this type of workload, because the Extended Insight server monitoring is enabled for this monitored database, we can obtain the time spent per transaction information from the data server. We can use this information to verify our assumption about sorting and I/O contributing mostly to the high CPU utilization. Note that the Extended Insight dashboard reports how the execution time is spent, not how the CPU time is spent. However, because execution requires CPU, we can use this approach to verify our assumption.

On the Extended Insight dashboard, we open the Response Time Details for the whole database workload for our six hours monitoring interval and display the Data Server Time layer as shown in Figure 6-13. Most interested to us is the Time Distribution chart in the detail area of the dashboard. This shows that for all queries that run in this six hours interval, more than 65% of the execution time is spent for I/O and more than 32% of the execution time is spent for sorting. This confirms our assumption.

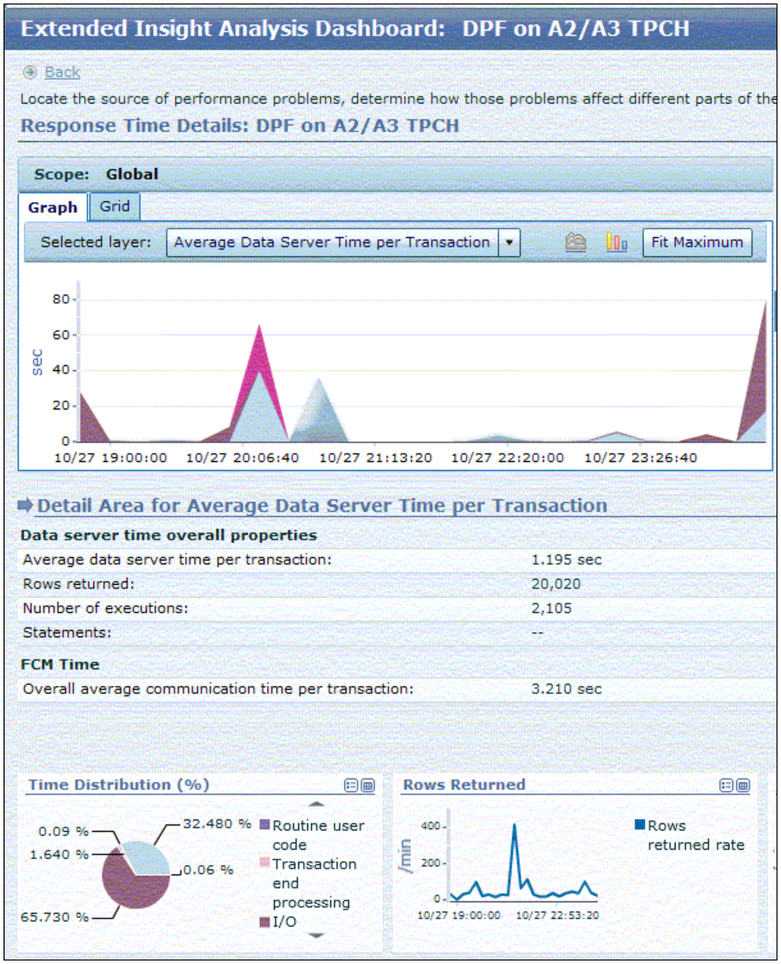
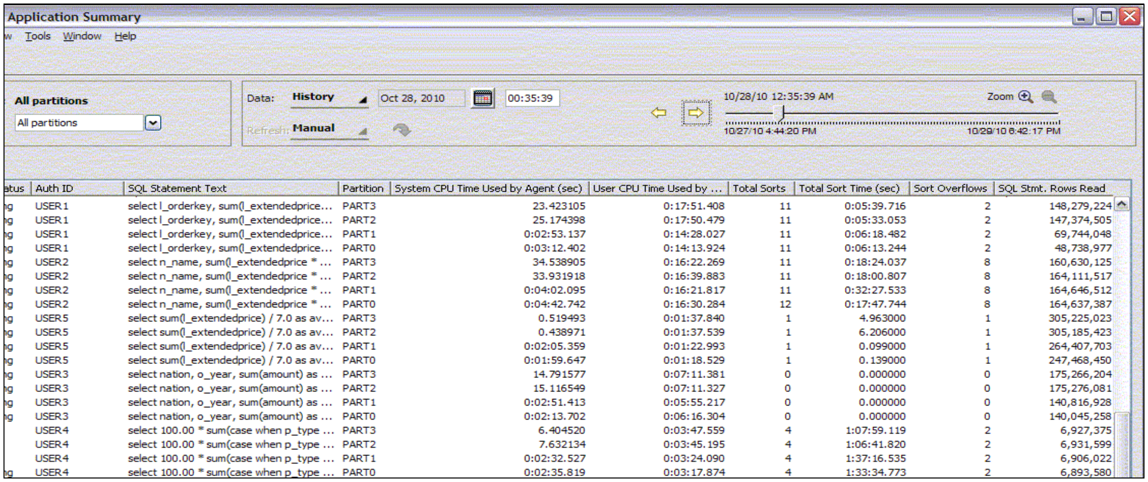


Figure 6-13 Data Server Time layer on Extended Insight dashboard

We monitor a partitioned database with four partitions. In Optim Performance Manager version 4.1.0.1, the Active SQL dashboard shows the statements and execution details aggregated of all partitions only. There is no way to see how a statement execution performed on a single partition. We use Performance Expert Client to look the execution information for a single partitions. We want to compare the CPU utilization on partitions 0 and 1 with the utilization on partitions 2 and 3. Let us see whether Performance Expert gives us additional insight why the CPU utilization is higher on partition 0 and 1 than on 2 and 3.

Open Performance Expert Client, launch the Application Summary panel, and select All partitions view which shows all four partitions. In the time slider, we select the end timestamp of the monitoring interval because we know that the

CPU utilization was high at the end of the six hours interval. The Application Summary panel (Figure 6-14) shows the connected applications on the selected timestamp including the executing SQL statements.



atus	Auth ID	SQL Statement Text	Partition	System CPU Time Used by Agent (sec)	User CPU Time Used by ...	Total Sorts	Total Sort Time (sec)	Sort Overflows	SQL Stmt. Rows Read
ng	USER1	select l_orderkey, sum(l_extendedprice...	PART3	23.423105	0:17:51.408	11	0:05:39.716	2	148,279,224
ng	USER1	select l_orderkey, sum(l_extendedprice...	PART2	25.174398	0:17:50.479	11	0:05:33.053	2	147,374,505
ng	USER1	select l_orderkey, sum(l_extendedprice...	PART1	0:02:53.137	0:14:28.027	11	0:06:18.482	2	69,744,048
ng	USER1	select l_orderkey, sum(l_extendedprice...	PART0	0:03:12.402	0:14:13.924	11	0:06:13.244	2	48,738,977
ng	USER2	select n_name, sum(l_extendedprice * ...	PART3	34.538905	0:16:22.269	11	0:18:24.037	8	160,630,125
ng	USER2	select n_name, sum(l_extendedprice * ...	PART2	33.931918	0:16:39.883	11	0:18:00.807	8	164,111,517
ng	USER2	select n_name, sum(l_extendedprice * ...	PART1	0:04:02.095	0:16:21.817	11	0:32:27.533	8	164,646,512
ng	USER2	select n_name, sum(l_extendedprice * ...	PART0	0:04:42.742	0:16:30.284	12	0:17:47.744	8	164,637,387
ng	USER5	select sum(l_extendedprice) / 7.0 as av...	PART3	0.519493	0:01:37.840	1	4.963000	1	305,225,023
ng	USER5	select sum(l_extendedprice) / 7.0 as av...	PART2	0.438971	0:01:37.539	1	6.206000	1	305,185,423
ng	USER5	select sum(l_extendedprice) / 7.0 as av...	PART1	0:02:05.359	0:01:22.993	1	0.099000	1	264,407,703
ng	USER5	select sum(l_extendedprice) / 7.0 as av...	PART0	0:01:59.647	0:01:18.529	1	0.139000	1	247,468,450
ng	USER3	select nation, o_year, sum(amount) as ...	PART3	14.791577	0:07:11.381	0	0.000000	0	175,266,204
ng	USER3	select nation, o_year, sum(amount) as ...	PART2	15.116549	0:07:11.327	0	0.000000	0	175,276,081
ng	USER3	select nation, o_year, sum(amount) as ...	PART1	0:02:51.413	0:05:55.217	0	0.000000	0	140,816,928
ng	USER3	select nation, o_year, sum(amount) as ...	PART0	0:02:13.702	0:06:16.304	0	0.000000	0	140,045,258
	USER4	select 100.00 * sum(case when p_type ...	PART3	6.404520	0:03:47.559	4	1:07:59.119	2	6,927,375
	USER4	select 100.00 * sum(case when p_type ...	PART2	7.632134	0:03:45.195	4	1:06:41.820	2	6,931,599
	USER4	select 100.00 * sum(case when p_type ...	PART1	0:02:32.527	0:03:24.090	4	1:37:16.535	2	6,906,022
ng	USER4	select 100.00 * sum(case when p_type ...	PART0	0:02:35.819	0:03:17.874	4	1:33:34.773	2	6,893,580

Figure 6-14 Application Summary in Performance Expert Client

The interesting information we obtain from Performance Expert Client is the user and system CPU time per statement by partition, see Figure 6-15 on page 228. The user CPU time per statement is nearly the same on all partitions, but the system CPU time used is much higher on partition 0 and 1 that is in line with what we have already seen on the System dashboard in Figure 6-4 on page 220.

SQL Statement Text	Partition	System CPU Time Used by Agent (sec)	User CPU Time Used by ...
select l_orderkey, sum(l_extendedprice...	PART3	23.423105	0:17:51.408
select l_orderkey, sum(l_extendedprice...	PART2	25.174398	0:17:50.479
select l_orderkey, sum(l_extendedprice...	PART1	0:02:53.137	0:14:28.027
select l_orderkey, sum(l_extendedprice...	PART0	0:03:12.402	0:14:13.924
select n_name, sum(l_extendedprice * ...	PART3	34.538905	0:16:22.269
select n_name, sum(l_extendedprice * ...	PART2	33.931918	0:16:39.883
select n_name, sum(l_extendedprice * ...	PART1	0:04:02.095	0:16:21.817
select n_name, sum(l_extendedprice * ...	PART0	0:04:42.742	0:16:30.284
select sum(l_extendedprice) / 7.0 as av...	PART3	0.519493	0:01:37.840
select sum(l_extendedprice) / 7.0 as av...	PART2	0.438971	0:01:37.539
select sum(l_extendedprice) / 7.0 as av...	PART1	0:02:05.359	0:01:22.993
select sum(l_extendedprice) / 7.0 as av...	PART0	0:01:59.647	0:01:18.529
select nation, o_year, sum(amount) as ...	PART3	14.791577	0:07:11.381
select nation, o_year, sum(amount) as ...	PART2	15.116549	0:07:11.327
select nation, o_year, sum(amount) as ...	PART1	0:02:51.413	0:05:55.217
select nation, o_year, sum(amount) as ...	PART0	0:02:13.702	0:06:16.304
select 100.00 * sum(case when p_type ...	PART3	6.404520	0:03:47.559
select 100.00 * sum(case when p_type ...	PART2	7.632134	0:03:45.195
select 100.00 * sum(case when p_type ...	PART1	0:02:32.527	0:03:24.090
select 100.00 * sum(case when p_type ...	PART0	0:02:35.819	0:03:17.874

Figure 6-15 CPU time in Application Summary

Partition 0 is the coordinator partition. On this partition, DB2 combines the result sets from all partitions. Therefore, the subsection execution of the plan on this partition requires more CPU time than on other partitions. We confirm that by looking at the subsection details in Performance Expert Client. Figure 6-16 shows the subsection details of one query on partition 0 and Figure 6-17 shows the subsection details on the same query on partition 2.

Subsections

Subsection Number	Rows Read	Rows Written	System CPU Time used by Subsection (sec)	User CPU Time used by Subsection (sec)
0	0	0	0.000053	0.000214
1	25	0	0.002992	0.003590
2	0	0	0.001571	0.000873
3	25,017,237	361,196	28.266892	0:02:45.844
5	1,064	0	0.001831	0.001644
4	115,026,932	367,710	0:01:45.399	0:03:30.278

Figure 6-16 Subsection details on partition 0

Subsections

Subsection Number	Rows Read	Rows Written	System CPU Time used by Subsection (sec)	User CPU Time used by Subsection (sec)
1	0	0	0.000523	0.000465
2	3,795,192	3,683,559	2.721730	24.066228
3	18,520,953	45,623	3.652644	33.777163
4	125,058,360	81,315,258	7.536225	0:03:54.384

Figure 6-17 Subsection details on partition 2

The next step would be to launch Query Tuner from the Active SQL dashboard for each of the high CPU consuming statements to see the access plan and to obtain tuning recommendations. We do not show that in detail in this scenario because we cover this topic as well in 8.1, "Application running slowly caused by index issue" on page 272. To confirm our assumptions we show a portion of the access plan of one query here in Figure 6-18 which illustrates that table scans are performed on the database followed by sorts.

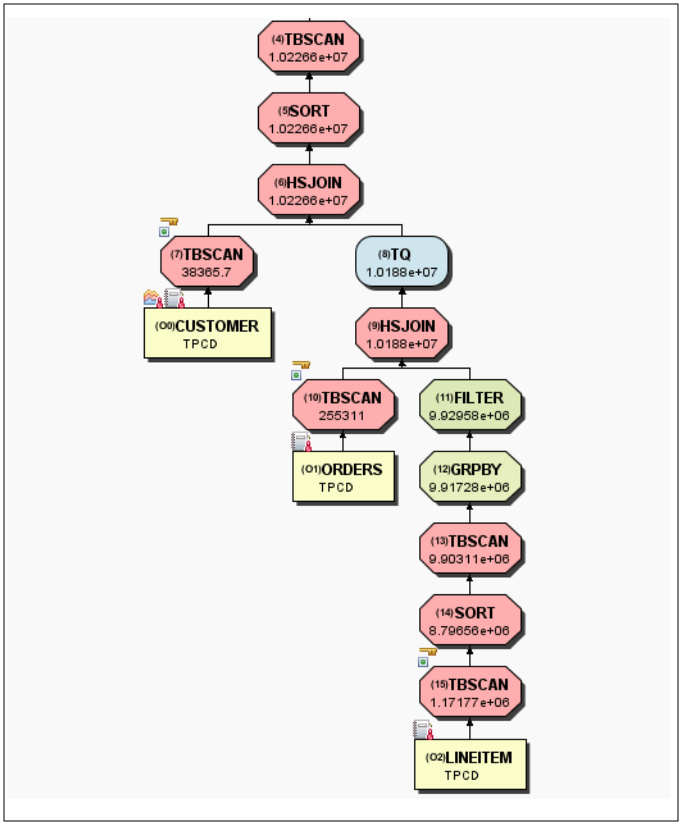


Figure 6-18 Part of the access plan showing table scans

Along with the access plan, Query Tuner recommends indexes as shown in Figure 6-19.

Custom and recommended indexes

Indexes by Table	Creator	Object Name	Columns	Estimated Disk Space	Custom
ORDERS					
Index	DB2OE	IDX1010281836420	O_ORDERKEY(ASC) ,O_TO...	753.860375 M	No
LINEITEM					
Index	DB2OE	IDX1010281836310	L_ORDERKEY(ASC) ,L_QU...	2210.1885 M	No
CUSTOMER					
Index	DB2OE	IDX1010281836360	C_CUSTKEY(ASC) ,C_NAM...	75.376 M	No

Figure 6-19 Index recommendations from Query Tuner

Using the Dynamic SQL Statement report

In this section we show you how to use the Dynamic SQL report to analyze the SQL statements. This is another method to identify untuned queries. The

Dynamic SQL Statement report has the following characteristics compared to the Active SQL dashboard:

- It shows aggregated statement execution information, whereas the Active SQL dashboard reports each statement execution separately.
- It does not show statement start and end times like the Active SQL dashboard because the statement execution information are aggregated over executions.
- It considers all executed dynamic SQL statements, whereas the Active SQL dashboard considers only statements that were running when Optim Performance Manager took the snapshot on the monitored database which is ones per minute by default. Short running statements that start and finish between two snapshots are not collected and, therefore, not considered by the Active SQL dashboard. Hence this dashboard is helpful for business intelligent (BI) environments but less useful for OLTP environments.
- It shows statement execution information per partition whereas the Active SQL dashboard reports statement execution aggregated over all partitions only.
- You can export reports into Microsoft Power Point chart, Excel spreadsheet, and PDF formats.

To generate Dynamic SQL Statement report, select **Reports** from Task Manager and specify the input parameters, see Figure 6-20 on page 232. We choose six hours as the time interval and specify that we want to see the top 10 statements by CPU time.

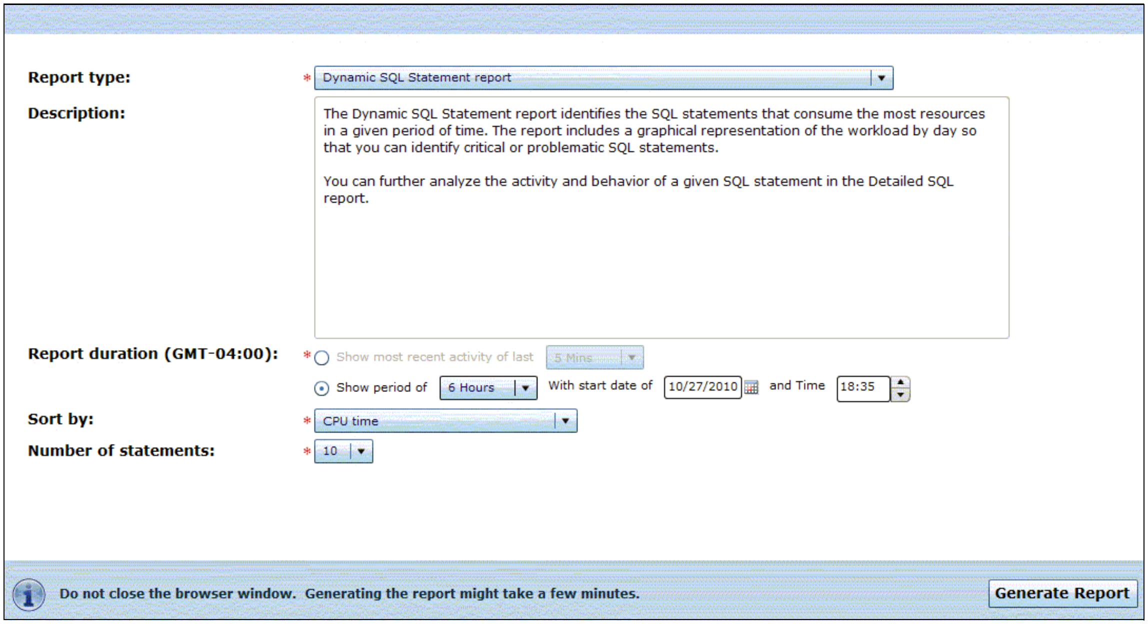


Figure 6-20 Dynamic SQL Statement Report launch panel

The report first shows graphically the top 10 statements by CPU time per hour, see Figure 6-21.

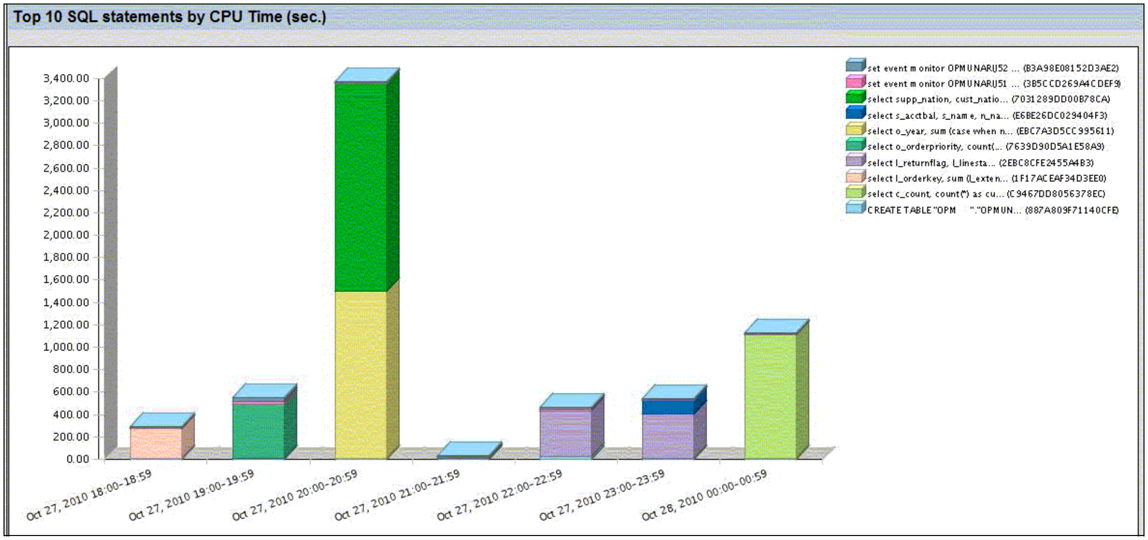


Figure 6-21 Top 10 statements by CPU time per hour

Next, the aggregated execution details of the top 10 statements are shown. The statements are sorted by CPU time. As in Performance Expert Client, the report distinguishes between user and system CPU time and shows the information per single partition. Figure 6-22 shows the execution details for the top CPU statement.

Statement Text	Partition ID	Number of executions	Total elapsed time (sec)	Average elapsed time (sec)	Number of Logical Reads	Number of physical reads	▼ CPU time (sec)	Total System CPU Time (sec)	Total User CPU Time (sec)
select supp_nation, cust_nation, l_year, sum(volum...	0	0	5,370.743362	--	993,400	2,040	539.764355	142.153539	397.610816
select supp_nation, cust_nation, l_year, sum(volum...	1	0	5,370.730606	--	991,116	380	498.705839	108.668071	390.037768
select supp_nation, cust_nation, l_year, sum(volum...	2	0	5,370.701343	--	990,809	322	409.286927	9.047416	400.239511
select supp_nation, cust_nation, l_year, sum(volum...	3	0	5,370.731160	--	991,020	304	406.583160	9.011266	397.571894

Figure 6-22 Execution details in dynamic SQL statement report

Hint: In Figure 6-22, the number of executions for the displayed statement is 0. The reason is that this statement runs only ones in the six-hours reporting interval and has not finished yet at the end of the reporting interval. DB2 updates the execution metrics in the dynamic SQL statement cache while the statement is running, but DB2 does not update the number of executions before the statement has finished.

Within the report you can further drill down the statement to see how it behaves during the reporting interval. Click the link in front of the statement and the statement details opens in a new browser window. The statement details include the complete statement text, more execution details, and graphs. One graph shows, over the reporting interval, the number of executions per minute combined with the average execution time per minute. The other graph shows,over the reporting interval, the average user and system CPU time per execution. If a statement is executed very often, these graphs are helpful in understanding whether the execution time differ over time. In hour example, since most statements were run only ones during the reporting interval, this drill-down report does not give us new information.

Hint: In Performance Expert Client you obtain the Dynamic SQL Statements view in the Statistics Details panel. Performance Expert users often use this view to find the top statements. The Dynamic SQL Statement report uses the same source data as the Dynamic SQL Statements view in Performance Expert Client and is the replacement in the Optim Performance Manager web console.

6.2 Monitoring buffer pool behavior

In this high CPU utilization scenario we just do a quick check on the buffer pool behavior because we have seen on the Overview dashboard (shown in Figure 6-8 on page 223) that we have a low buffer pool hit ratio along with the high CPU utilization and the long running statements. For deeper buffer pool and I/O analysis, refer to Chapter 5, “Monitoring I/O utilization” on page 205.

The Overview dashboard shows you the buffer pool hit ratio across all buffer pools whereas the Buffer Pool and I/O dashboard shows you the buffer pool behavior for each single buffer pool. Figure 6-23 shows the Buffer Pool and I/O dashboard with the six hours monitoring interval that we used in our scenario. The BP32K and the IBMDEFAULTBP buffer pool both have low hit ratios that constitutes to the low overall hit ratio of 57.37%.

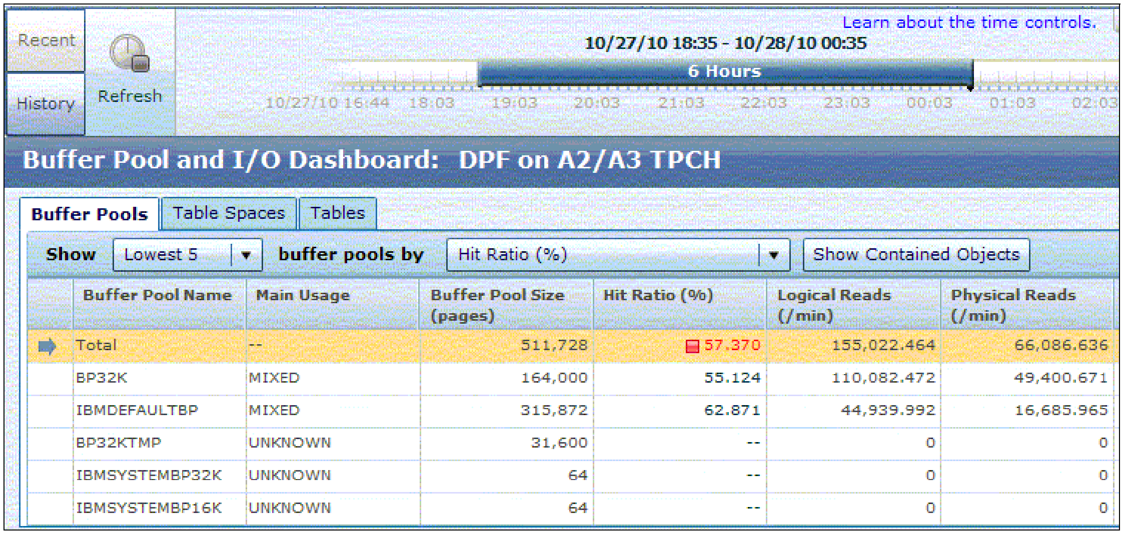
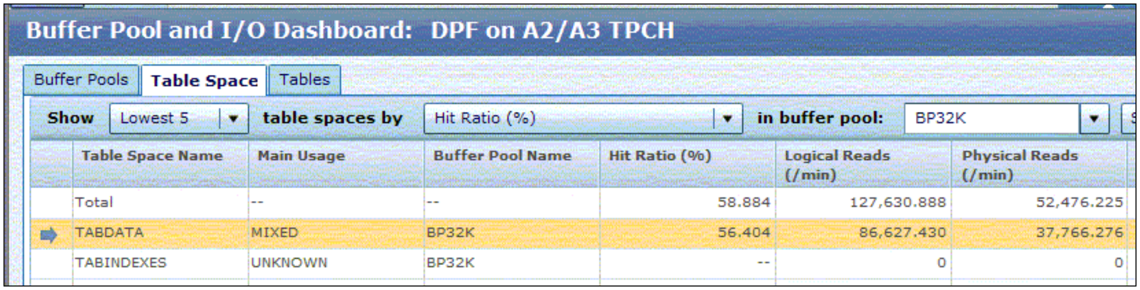


Figure 6-23 Hit ratios per buffer pool on Buffer Pool and I/O dashboard

Select the BP32K buffer pool and click **Show Contained Objects** to lists the table spaces that use the BP32K buffer pool. See Figure 6-24 on page 235. The TABDATA table space contains all the tables that our SQL queries use and the hit ratio is low with around 56%. The TABINDEXES table space is supposed to contain the indexes. No activity is done in the buffer pool for the TABINDEXES table space which is an indication of missing indexes.


Buffer Pool and I/O Dashboard: DPF on A2/A3 TPCH

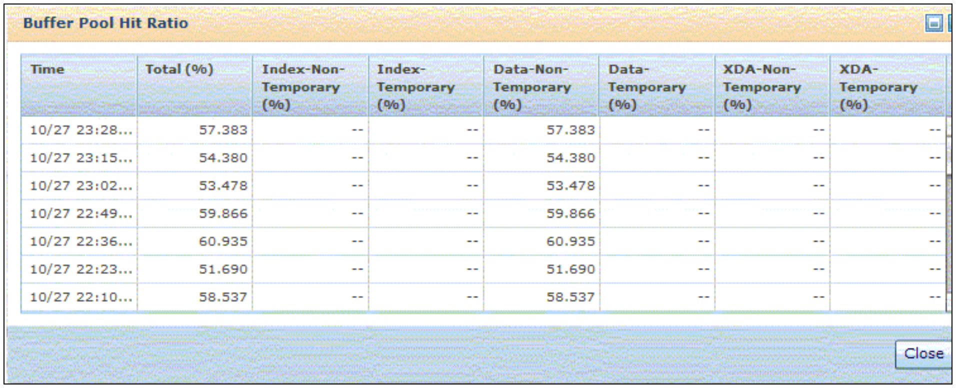
Buffer Pools | Table Space | Tables

Show Lowest 5 table spaces by Hit Ratio (%) in buffer pool: BP32K

Table Space Name	Main Usage	Buffer Pool Name	Hit Ratio (%)	Logical Reads (/min)	Physical Reads (/min)
Total	--	--	58.884	127,630.888	52,476.225
TABDATA	MIXED	BP32K	56.404	86,627.430	37,766.276
TABINDEXES	UNKNOWN	BP32K	--	0	0

Figure 6-24 Hit ratios per table spaces using the BP32K buffer pool

A detailed look at the hit ratio graph grid view confirms that only data reads are performed and no index reads at all, as shown in Figure 6-25.

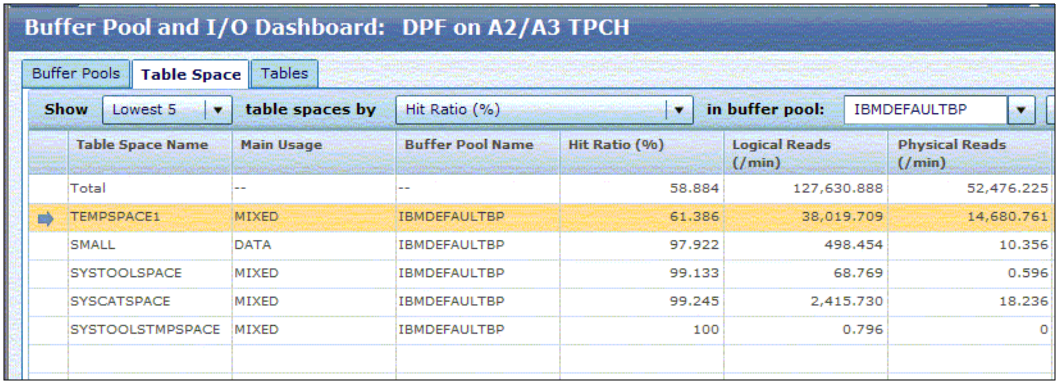

Buffer Pool Hit Ratio

Time	Total (%)	Index-Non-Temporary (%)	Index-Temporary (%)	Data-Non-Temporary (%)	Data-Temporary (%)	XDA-Non-Temporary (%)	XDA-Temporary (%)
10/27 23:28...	57.383	--	--	57.383	--	--	--
10/27 23:15...	54.380	--	--	54.380	--	--	--
10/27 23:02...	53.478	--	--	53.478	--	--	--
10/27 22:49...	59.866	--	--	59.866	--	--	--
10/27 22:36...	60.935	--	--	60.935	--	--	--
10/27 22:23...	51.690	--	--	51.690	--	--	--
10/27 22:10...	58.537	--	--	58.537	--	--	--

Close

Figure 6-25 Hit ratio separation for TABDATA table space

IBMDEFAULTBP buffer pool also had a low hit ratio of around 63%. We use same method to check quickly which table spaces use IBMDEFAULTBP buffer pool. Select **IBMDEFAULTBP** and click **Show Contained Objects** results in the list of table spaces shown in Figure 6-26 on page 236. The main contributor to the low hit ratio in IBMDEFAULTBP is the table space TEMPSPACE1 with around 61%. This table space is used for the sorting activity of the queries, which is high due to the missing indexes.



Buffer Pool and I/O Dashboard: DPF on A2/A3 TPCH

Buffer Pools | Table Space | Tables

Show Lowest 5 table spaces by Hit Ratio (%) in buffer pool: IBMDEFAULTBP

Table Space Name	Main Usage	Buffer Pool Name	Hit Ratio (%)	Logical Reads (/min)	Physical Reads (/min)
Total	--	--	58.884	127,630.888	52,476.225
TEMPSPACE1	MIXED	IBMDEFAULTBP	61.386	38,019.709	14,680.761
SMALL	DATA	IBMDEFAULTBP	97.922	498.454	10.356
SYSTOOLSPACE	MIXED	IBMDEFAULTBP	99.133	68.769	0.596
SYSCATSPACE	MIXED	IBMDEFAULTBP	99.245	2,415.730	18.236
SYSTOOLSTMPSPACE	MIXED	IBMDEFAULTBP	100	0.796	0

Figure 6-26 Hit ratios per table space using the IBMDEFAULTBP buffer pool

6.3 Monitoring memory usage

In this section we describe how you can use the Memory dashboard to monitor the memory that DB2 allocates and uses for the instance, databases, and applications. We describe the information you see on the memory dashboard and use that to introduce the DB2 memory model. This help you to understand how DB2 uses memory in order to detect any memory bottlenecks easily.

Since the database manager configuration and database configuration influence how the memory is used, we also describe using the *Database Manager Configuration report* and the *Database Configuration report* to check how your database manager and databases are configured and which parameters have been changed over time.

6.3.1 Monitoring DB2 instance memory usage

DB2 uses the *INSTANCE_MEMORY* parameter to limit the memory usage on each partition. This parameter is set to AUTOMATIC by default. The INSTANCE_MEMORY database manager configuration parameter controls the maximum amount of memory that can be used by each DB2 logical partition, including all shared memory and private memory usage for the agents associated to that particular logical partition. DB2 considers that between 75% and 95% of the RAM can be used for all the database partitions within a physical node.

The partitioned database that we used in the CPU scenario has two partitions on each physical node. Each physical node has 8 GB of RAM. The instance memory limit for each partition on the physical node is around 2.7 GB (217263

4K pages in the INSTANCE_MEMORY parameter) which results to around 5.4 GB for both partitions. This is at the lower bound of the RAM that DB2 uses as memory limit in our scenario.

On the Memory dashboard you can look at these values by selecting the **Instance** scope, shown in Figure 6-27. The blue line in the graph represents the available RAM on the machine, whereas the red line shows the defined instance memory limit per partition.

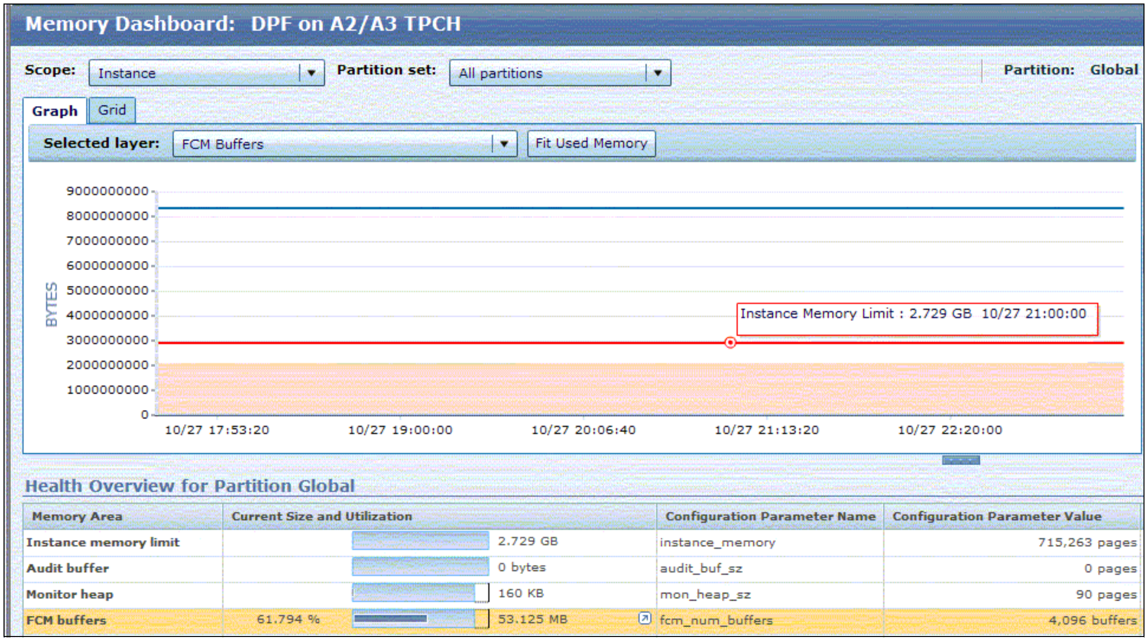


Figure 6-27 Instance Memory usage and memory limit on memory dashboard

Click **Fit Used Memory** to remove the limit lines and adjust the scale of the graph to the memory that is really used. See Figure 6-28 on page 238.

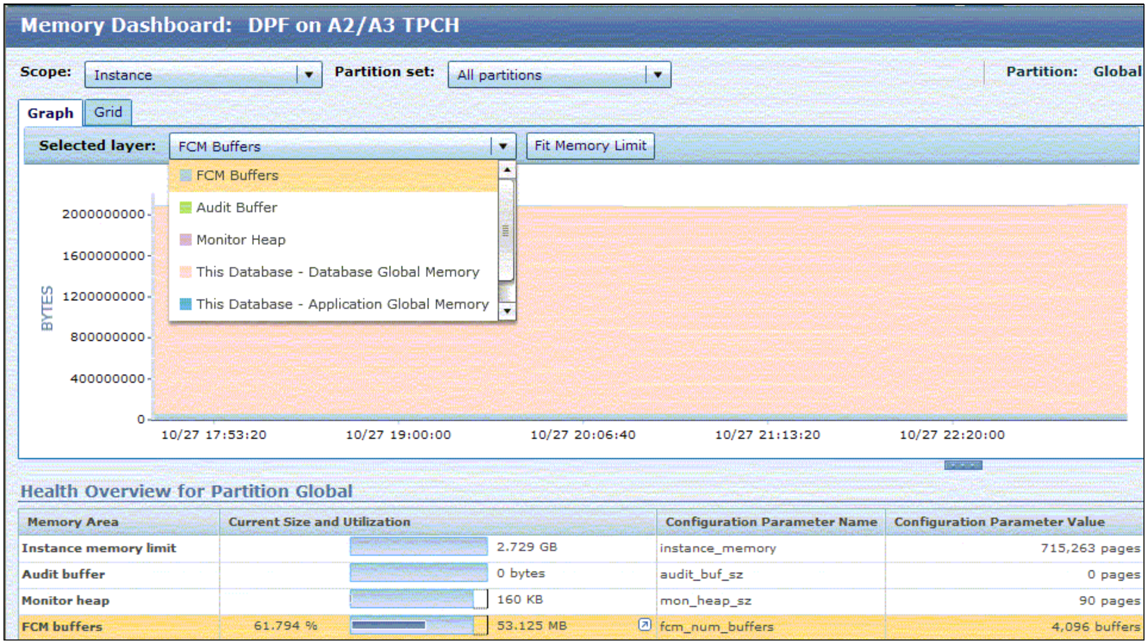


Figure 6-28 Actual Instance memory usage on memory dashboard

In the graph you see the memory areas that belong to the database manager shared memory are Audit Buffer, Monitor Heap, FCM Buffers. These memory areas are allocated when the instance is started. The graph also shows the database global memory and application memory of the database you selected on this dashboard. If your instance contains multiple databases and you want to know the entire amount of memory that DB2 allocated for this instance, you have to select the databases one by one on the memory dashboard and calculate the sum of the used memory manually.

The table beneath the graph only shows details for the memory areas belonging to database manager shared memory. For the details of the database global memory and application memory, change the scope by using the **Scope** drop down box.

6.3.2 Monitoring database memory usage

DB2 uses the *DATABASE_MEMORY* parameter to limit the memory used for the database global memory allocations on each partition. This parameter is set to AUTOMATIC by default. The DATABASE_MEMORY database configuration parameter represents the maximum amount of memory reserved for the database global memory allocations. On DB2 9.7, this value is equal to individual memory pool allocations within the database global memory such as buffer pool, utility heap size with additional memory to accommodate for dynamic heap growth. DATABASE_MEMORY should be less than INSTANCE_MEMORY. Database global memory is allocated when the database is activated.

On the Memory dashboard you can look at this value by selecting the **Database Global Memory** scope as shown in Figure 6-29. The red line shows the defined database memory limit set in the DATABASE_MEMORY parameter.

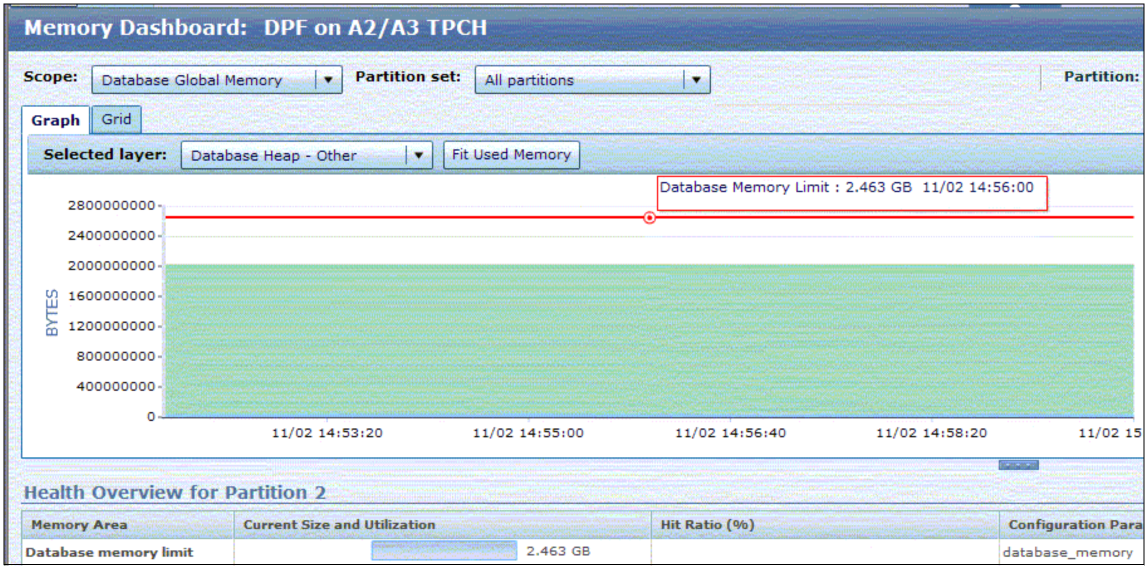


Figure 6-29 Database memory limit on memory dashboard

The colored part below the database memory limit show the database global memory that is used by the database over time. The memory limit is not reached yet. If there are memory shortages in one of the heaps or buffer pools, there is still room to increase size for the heaps or buffer pools.

Click **Fit Used Memory** to remove the limit line and to adjust the scale of the graph to the memory that is really used. See Figure 6-30 on page 240. Each color in the graph represents one memory area of the database global memory. The largest part of the memory is used by buffer pools, therefore, it is hard to see any other column in the graph. The other memory areas are rather small.

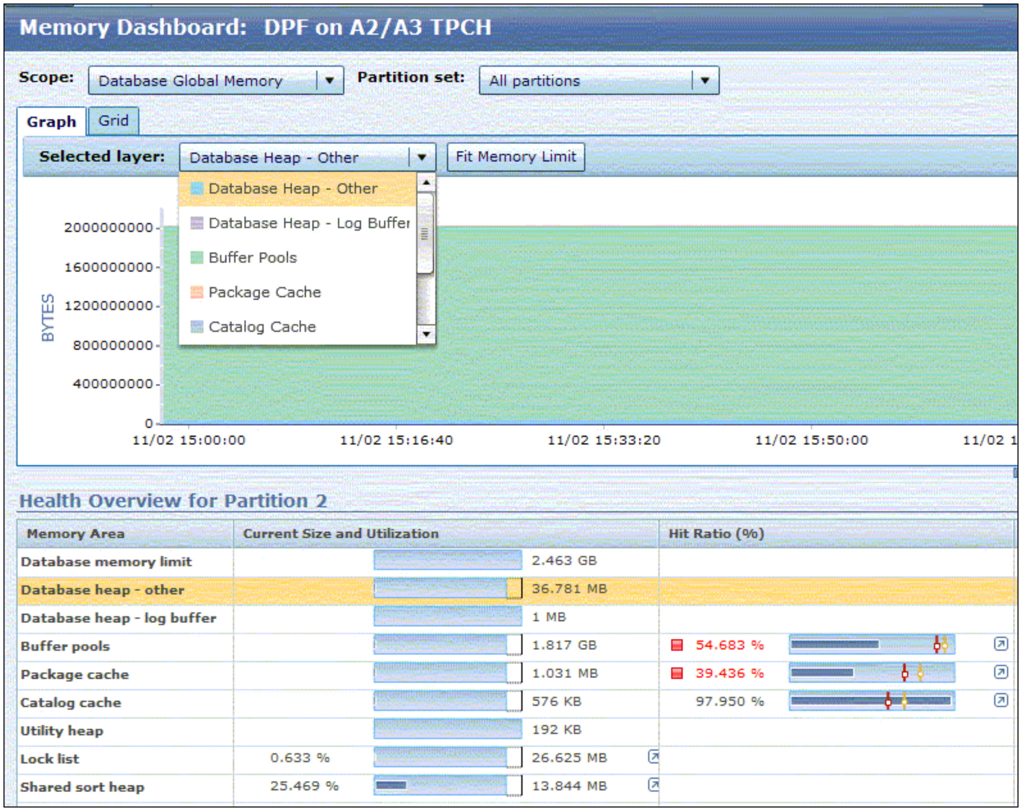


Figure 6-30 Memory areas in data base global memory

The table beneath the graph shows the details for each memory that consist of:

- Current Size and Utilization:

  The current size is the size of the memory area at the end of the monitoring interval. If you are looking at recent data then this is the most recent value from the database.

  The utilization shows the utilization percentage of the memory area at the end of the monitoring interval. Because DB2 does not return the actual utilization value for every memory area, only a few memory areas' utilization can be calculated, for example the lock list. A high utilization is an indication that increasing the memory area could be helpful. For example, if the Lock list is used by 100% then DB2 changes row locks to table locks which can slow down concurrency.

- Hit Ratio (%):

This field shows the average hit ratio during the monitoring interval. The hit ratio tells the percentage the data is available in the memory areas when the data is needed. Only a few memory areas' hit ratio can be calculated and make sense to calculate. A low hit ratio is an indication that this memory area could be too small and the data is not in the memory area when needed.

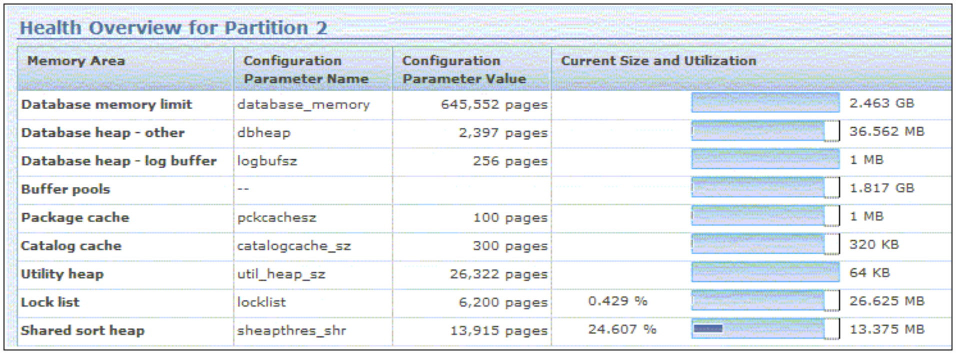
- Configuration Parameter Name:

  This field shows the database manager configuration parameter or the database configuration parameter that can be configured to control the size of the memory area.

- Configuration Parameter Value:

  This is the value of the configuration parameter at the end of the monitoring interval. If you are looking at the recent data then this is the most recent value from the database.

Figure 6-30 on page 240 shows the current size, utilization, and hit ratio information which are only a portion of the whole memory dashboard. In Figure 6-31, we reorder the columns of the table to present the configuration parameter name and value.

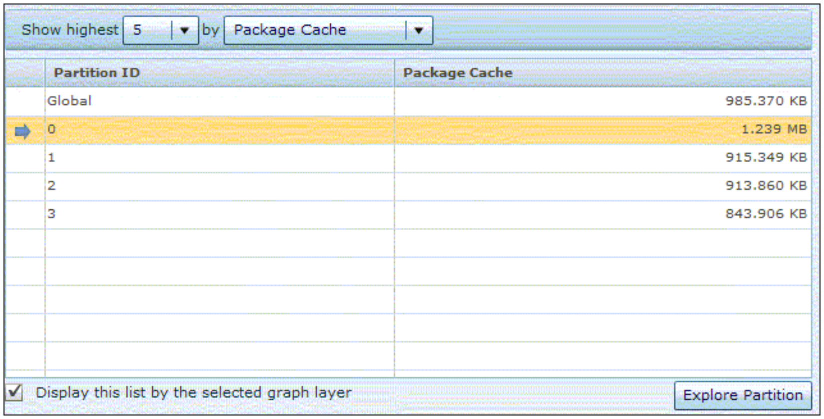


Health Overview for Partition 2

Memory Area	Configuration Parameter Name	Configuration Parameter Value	Current Size and Utilization	
Database memory limit	database_memory	645,552 pages		2.463 GB
Database heap - other	dbheap	2,397 pages		36.562 MB
Database heap - log buffer	logbufsz	256 pages		1 MB
Buffer pools	--			1.817 GB
Package cache	pckcachesz	100 pages		1 MB
Catalog cache	catalogcache_sz	300 pages		320 KB
Utility heap	util_heap_sz	26,322 pages		64 KB
Lock list	locklist	6,200 pages	0.429 %	26.625 MB
Shared sort heap	sheapthres_shr	13,915 pages	24.607 %	13.375 MB

Figure 6-31 Memory area details

For a partitioned system you can list the top partitions by the size of a memory area, view the memory area details for a single partition, and display the memory usage graph for a single partition. On the right side of the memory dashboard you find the top partition listing and partition selection. As an example we list the top partitions by package cache size shown in Figure 6-32 on page 242.



Partition ID	Package Cache
Global	985.370 KB
0	1.239 MB
1	915.349 KB
2	913.860 KB
3	843.906 KB

Figure 6-32 Top partitions by package cache size

The highest package cache size exists on partition 0. The reason for that is probably because partition 0 is the coordinator partitions and any query execution starts from there. Selecting a partition displays the memory area details such as size, hit ratio, and so on in the lower table on the dashboard. Clicking the Explore Partition button displays the memory usage graph for the selected partition. By default, The Global values displayed on the memory dashboard show the average values of all partitions.

6.3.3 Monitoring application memory usage

Memory used for the applications consist of application global memory and application private and shared memory.

DB2 uses the *APPL_MEMORY* database configuration parameter to control the maximum amount of application memory that is allocated by DB2 database agents to service application requests. This parameter is set to AUTOMATIC by default.

You can look at application global memory usage by selecting the **Application Global Memory** scope and you can look at application private and shared memory by selecting the **Application Private/Shared Memory** scope. The usage and navigation is the same as for the Instance and Database Global Memory scope therefore we do not go into more detail here. Just one screenshot showing the application global memory usage in Figure 6-33 on page 243 that shows nicely the different memory areas using different colors.

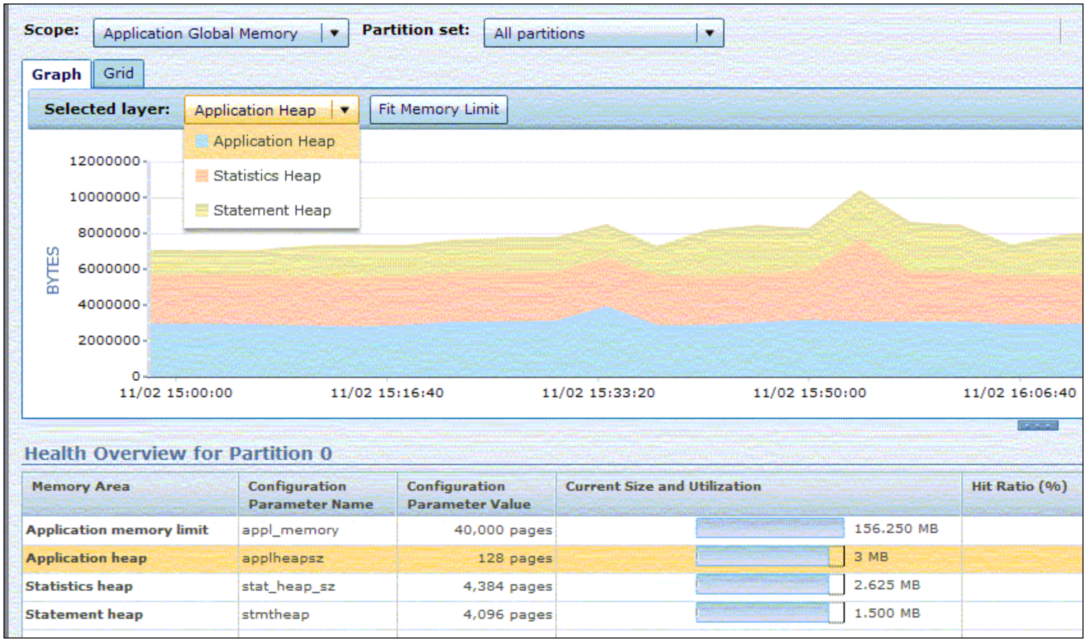


Figure 6-33 Application global memory areas

6.3.4 Detecting a too small memory area

The memory dashboard helps to detect if a memory area is too small. Monitoring the size of the memory areas over time is especially important if the self tuning memory manager (STMM) is turned off for the monitored database.

In the following example STMM is turned off on the partitioned database.

In Figure 6-34 on page 244, the Overview dashboard shows a low Package cache hit ratio of 39% while heavy statements are running, indicated by these metrics Rows read per fetched row, Maximum CPU time of running statements, and Maximum elapsed time of running statements.

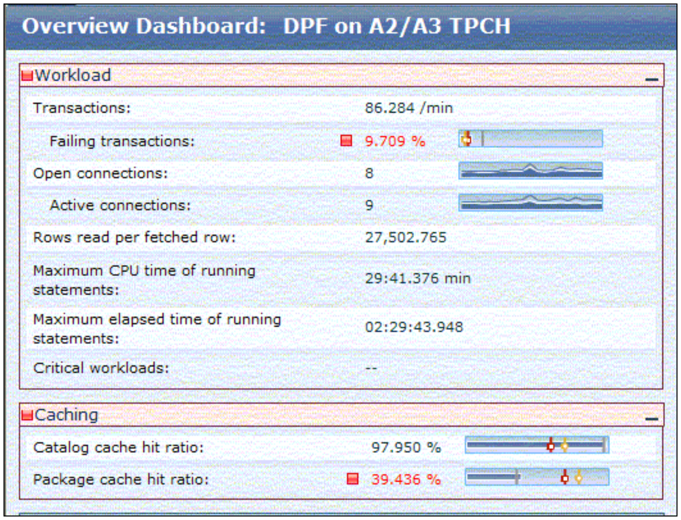


Figure 6-34 Part of Overview dashboard showing package cache hit ratio

Clicking on the bar chart next to Package cache hit ratio opens the graph shown in Figure 6-35 on page 245. This graph confirms that the package cache hit ratio is low over time. It also shows that there are differences among the partitions, the minimum hit ratio that occurs on one partition is lower than the average hit ratio over all partitions.

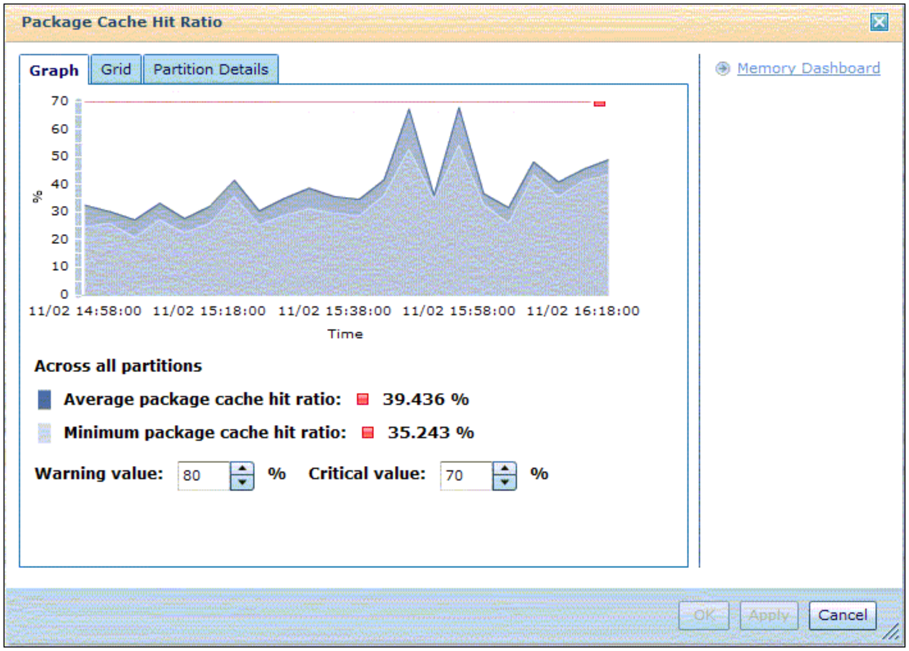
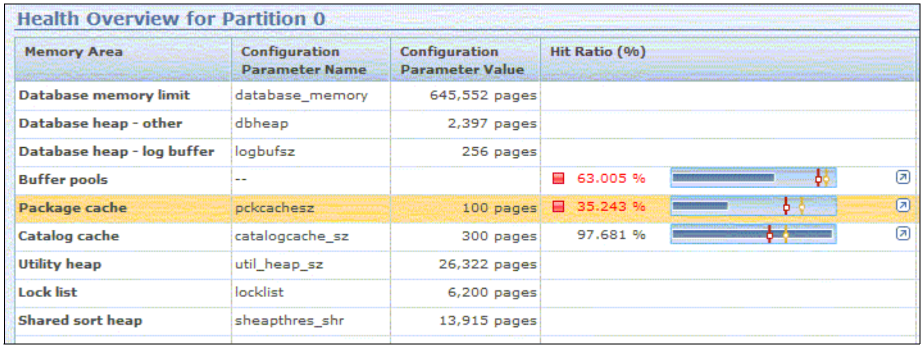


Figure 6-35 Package cache hit ratio graph.

To list the top partitions by package cache size, click **Memory Dashboard** from the graph and select the scope for Database Global Memory on the dashboard. Figure 6-32 on page 242 shows that the package cache size for partition 0 is around 1.2 MB and is around 900 KB for the other partitions. Select the coordinator partition 0 for he details as shown in Figure 6-36. The partition 0 has an average package cache hit ratio of 35% which is the lowest hit ratio of all partitions because partition 0 is the coordinator partition.


Health Overview for Partition 0

Memory Area	Configuration Parameter Name	Configuration Parameter Value	Hit Ratio (%)
Database memory limit	database_memory	645,552 pages	
Database heap - other	dbheap	2,397 pages	
Database heap - log buffer	logbufsz	256 pages	
Buffer pools	--		63.005 %
Package cache	pckcachesz	100 pages	35.243 %
Catalog cache	catalogcache_sz	300 pages	97.681 %
Utility heap	util_heap_sz	26,322 pages	
Lock list	locklist	6,200 pages	
Shared sort heap	sheapthres_shr	13,915 pages	

Figure 6-36 Package cache details of partition 0

Figure 6-37 shows the package cache hit ratio for partition 2 with 52%. The ratio is higher than on partition 0, but the number is not good.

Health Overview for Partition 2

Memory Area	Configuration Parameter Name	Configuration Parameter Value	Hit Ratio (%)
Database memory limit	database_memory	645,552 pages	
Database heap - other	dbheap	2,397 pages	
Database heap - log buffer	logbufsz	256 pages	
Buffer pools	--		52.622 %
Package cache	pckcachesz	100 pages	52.909 %
Catalog cache	catalogcache_sz	300 pages	99.348 %
Utility heap	util_heap_sz	26,322 pages	
Lock list	locklist	6,200 pages	
Shared sort heap	sheapthres_shr	13,915 pages	

Figure 6-37 Package cache details on partition 2

The database configuration parameter PCKCACHESZ shows only 100 pages, which is a very small size. We increase the size using the following command on a command line on the monitored system while our workload is still running:

```
db2 update db cfg using pckcachsz 1533
```

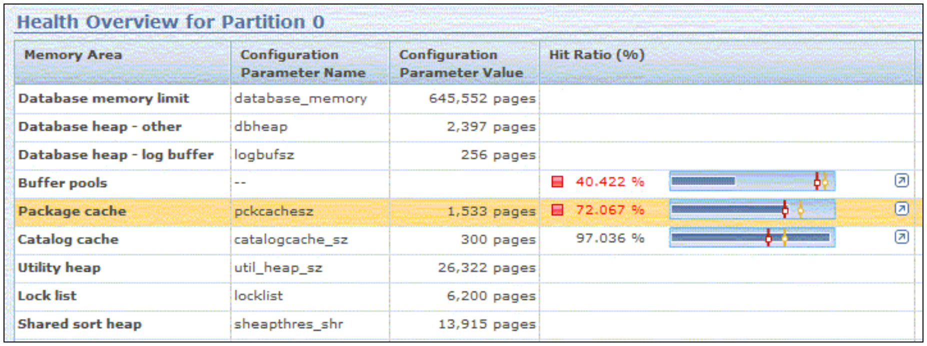
Some minutes later we check the actual sizes of the package cache on the Memory dashboard. On partition 0, the actual size increased by around 4 MB to 5 MB as shown in Figure 6-38.

Show highest 5 by Package Cache

Partition ID	Package Cache
Global	2.206 MB
0	5.037 MB
1	1.262 MB
3	1.262 MB
2	1.262 MB

Figure 6-38 Package cache size after increasing pckcachesz

Looking at the memory details for each partition confirms that the package cache hit ratio is much higher now on all partitions. It is about 72% on average on partition 0 as shown in Figure 6-39 on page 247. It still shows an alert, because the critical alert threshold is 70% and, at some timestamps during the monitoring interval the hit ratio was below 70%. Increasing the package cache size further would improve the hit ratio further.



Health Overview for Partition 0

Memory Area	Configuration Parameter Name	Configuration Parameter Value	Hit Ratio (%)
Database memory limit	database_memory	645,552 pages	
Database heap - other	dbheap	2,397 pages	
Database heap - log buffer	logbufsz	256 pages	
Buffer pools	--		40.422 %
Package cache	pckcachesz	1,533 pages	72.067 %
Catalog cache	catalogcache_sz	300 pages	97.036 %
Utility heap	util_heap_sz	26,322 pages	
Lock list	locklist	6,200 pages	
Shared sort heap	sheapthres_shr	13,915 pages	

Figure 6-39 Details on partition 0 after increasing pckcachesz

6.3.5 Checking configuration changes using reports

The database manager configuration and database configuration influence how the memory is used. In this section we describe how to use the *Database Manager Configuration report* and the *Database Configuration report* to check how your database manager and databases are configured and which parameters have changed over time.

To launch either Database Manager Configuration report or Database Configuration report, from **Task Manager** → **Reports**. On the reports launch page select the type of report and the reporting interval. A new browser window opens that displays the report. Both reports have similarities and offer you the ability to track changes that DB2 does when the memory related parameters were set to AUTOMATIC. By default, the tracking of changes to automatic parameters is off and you see only the changes that you did manually to the non-automatic parameters.

Using the Database Manager Configuration report

In Figure 6-40 on page 248 we show a partial report, which includes all parameters important for capacity management. Some of the parameters are displayed on the memory dashboard for the Instance scope. No changes were manually applied to the database manager configuration during the reporting interval of one day, therefore, you only see configuration parameters from one timestamp.

Capacity Management

Agents		Nov 3, 2010 12:10 AM
MAX_CONNECTIONS	[A] Maximum number of client connections	-1
MAX_COORDAGENTS	[A] Maximum number of coordinating agents	200
MAXCAGENTS	Maximum number of concurrent agents	0
MAXAGENTS	Maximum number of agents	0
NUM_POOLAGENTS	[A] Agent pool size	100
NUM_INITAGENTS	Initial number of agents in pool	0
AGENTPRI	Priority of agents	-1
MAXTOTFILOP	Maximum total of files open	0
Agent private memory		
AGENT_STACK_SZ	Agent stack size	1,024
PRIV_MEM_THRESH	Private memory threshold configuration parameter	0
QUERY_HEAP_SZ	Query heap size	1,000
SHEAPTHRES	Sort heap threshold	0
Agent/application communication memory		
RQRIOBLK	Client I/O block size	32,767
ASLHEAPSZ	Application support layer heap size	15
Database application remote interface (DARI) / Fenced processes		
KEEPFENCED	Keep fenced process	Yes
NUM_INIFENCED	Initial number of fenced processes	0
FENCED_POOL	[A] Maximum number of fenced processes	-1
Database manager instance memory		
MON_HEAP_SZ	[A] Database system monitor heap size	90
AUDIT_BUF_SZ	Audit buffer size	0
DIR_CACHE	Directory cache support	Yes
JAVA_HEAP_SZ	Maximum Java interpreter heap size	2,048
INSTANCE_MEMORY	[A] Instance memory	715,263

Figure 6-40 Capacity Management section from database manager report

Using the Database Configuration report

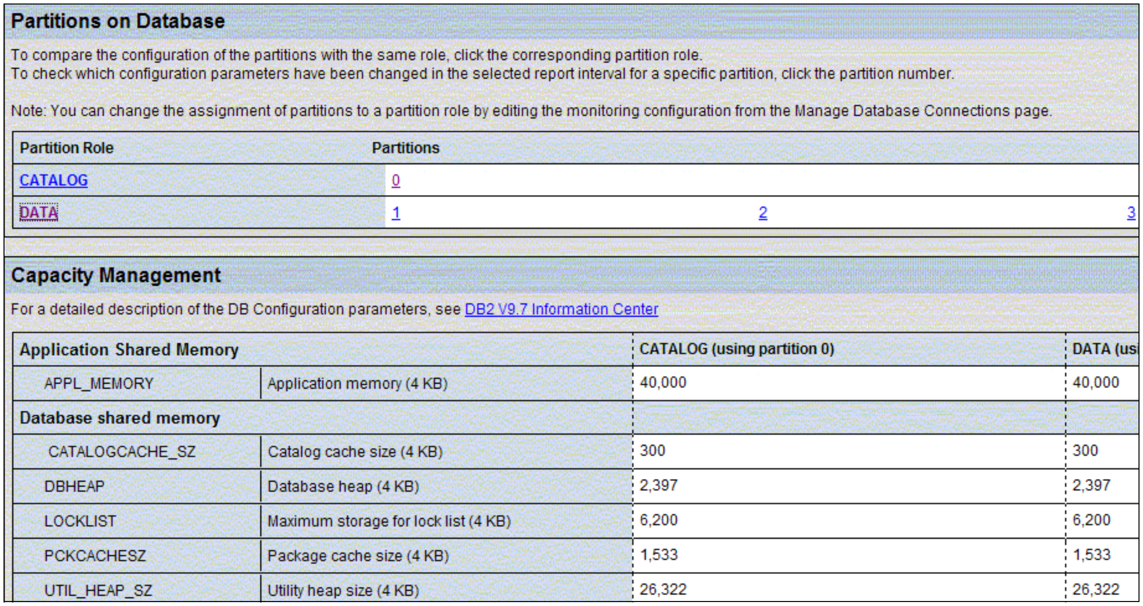
The Database Configuration report services two purposes:

- For a partitioned system, the report let you compare the database configuration of multiple partitions and highlights the differences.

  Figure 6-41 on page 249 shows a sample report with a few parameters on partition 0 and 1.

- For a single partition, the report shows and highlights the changes performed manually or automatically by DB2 during the reporting interval.

  In the example shown in Figure 6-42 on page 250 we see the manual change of the Package cache size that we did to improve the package cache hit ratio. Although STMM is turned off, we specify that we want to track changes of automatic parameters in order to see this manual change.



Partitions on Database

To compare the configuration of the partitions with the same role, click the corresponding partition role.
To check which configuration parameters have been changed in the selected report interval for a specific partition, click the partition number.

Note: You can change the assignment of partitions to a partition role by editing the monitoring configuration from the Manage Database Connections page.

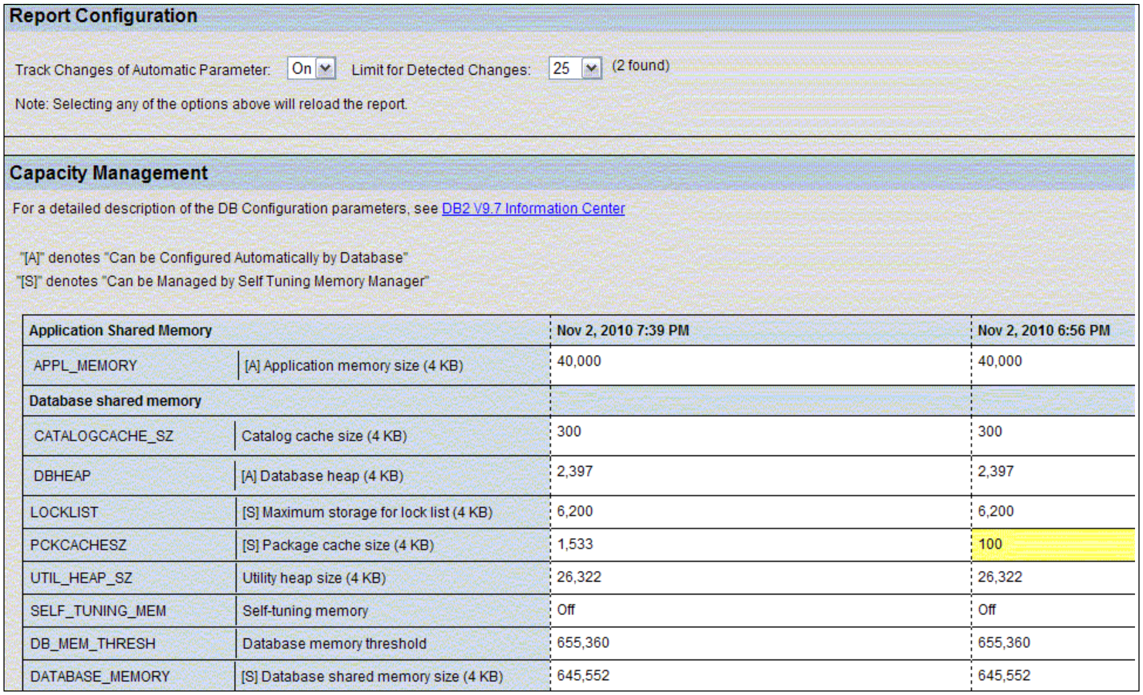
Partition Role	Partitions		
CATALOG	0		
DATA	1	2	3

Capacity Management

For a detailed description of the DB Configuration parameters, see DB2 V9.7 Information Center

Application Shared Memory		CATALOG (using partition 0)	DATA (usi
APPL_MEMORY	Application memory (4 KB)	40,000	40,000
Database shared memory			
CATALOGCACHE_SZ	Catalog cache size (4 KB)	300	300
DBHEAP	Database heap (4 KB)	2,397	2,397
LOCKLIST	Maximum storage for lock list (4 KB)	6,200	6,200
PCKCACHESZ	Package cache size (4 KB)	1,533	1,533
UTIL_HEAP_SZ	Utility heap size (4 KB)	26,322	26,322

Figure 6-41 Comparison of parameters on partition 0 and partition 1



Application Shared Memory		Nov 2, 2010 7:39 PM	Nov 2, 2010 6:56 PM
APPL_MEMORY	[A] Application memory size (4 KB)	40,000	40,000
Database shared memory			
CATALOGCACHE_SZ	Catalog cache size (4 KB)	300	300
DBHEAP	[A] Database heap (4 KB)	2,397	2,397
LOCKLIST	[S] Maximum storage for lock list (4 KB)	6,200	6,200
PCKCACHESZ	[S] Package cache size (4 KB)	1,533	100
UTIL_HEAP_SZ	Utility heap size (4 KB)	26,322	26,322
SELF_TUNING_MEM	Self-tuning memory	Off	Off
DB_MEM_THRESH	Database memory threshold	655,360	655,360
DATABASE_MEMORY	[S] Database shared memory size (4 KB)	645,552	645,552

Figure 6-42 Parameter changes on partition 0

7

Analyzing locking problems

Optim Performance Manager is good in identifying locking problems as the cause of a performance slowdown. Using either Inflight dashboards or Extended Insight dashboard or a combination of both let you easily find locking problems and the involved statements or applications.

This chapter provides two scenarios in that we identify and analyze locking problems:

- In the first scenario the system shows many deadlock alerts and we use the Locking dashboard to identify the cause of the deadlocks.
- In the second scenario we assume that the alerting for deadlock, lock, and timeout events is disabled in order to avoid the overhead of the lock event monitor on the monitored database. We show how you can identify locking problems easily even if the alerting is disabled.

7.1 Troubleshooting deadlock alerts

Assume that you are notified about a high number of deadlock alerts. The ways of notification include the following:

- You have configured e-mail notification and receive e-mails about deadlock alerts.
- You have configured notification through SNMP and see the Optim Performance Manager alerts in other products interfacing with SNMP.
- You are called by users who complain about bad response times and SQL errors indicating deadlocks or timeouts.
- You do your usual health check and see locking alerts on the Health Summary dashboard.

Open the Health Summary dashboard to verify the notifications. For the database `SAMPLE` you see a red critical alert indicator for the Locking category and the number of critical alerts is very high, shown in Figure 7-1.

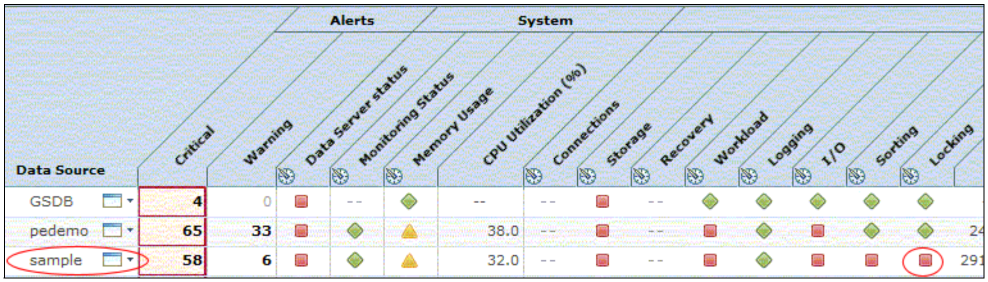


Figure 7-1 Health Summary showing alert indicators for databases

Click on the red alert indicator icon in the Locking category for the `SAMPLE` database. An overlay window opens listing all alerts that occurred during the selected monitoring time frame as shown in Figure 7-2 on page 253.

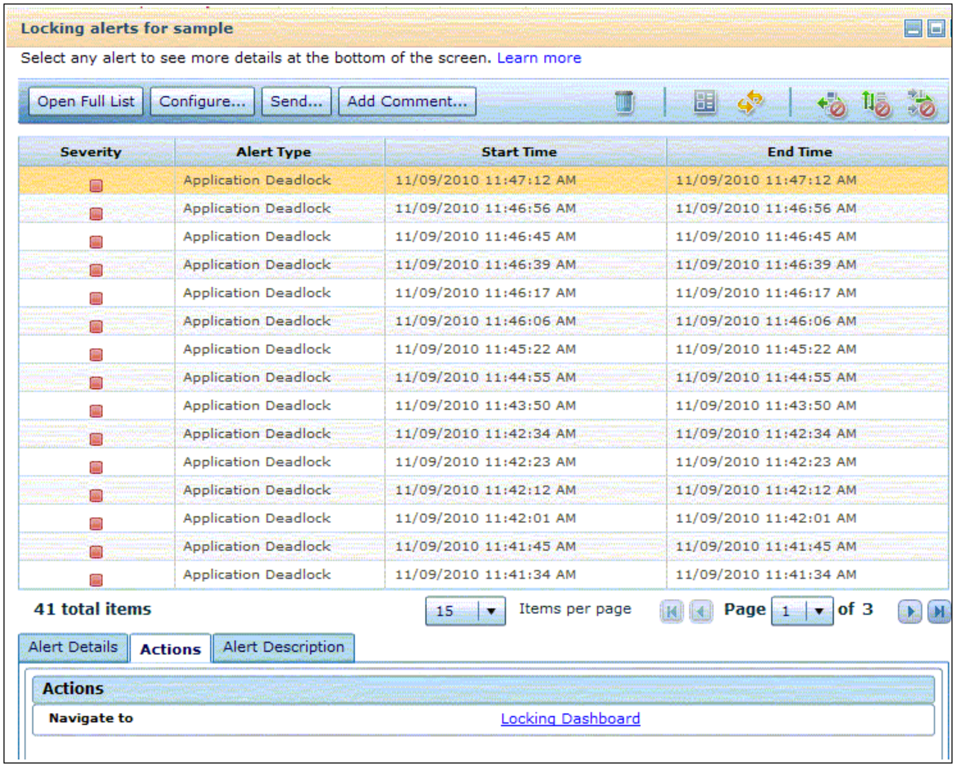


Figure 7-2 Deadlock alerts for SAMPLE database

You can analyze each single alert from here by selecting the Alert Details tab. Alternatively, you can open the Locking dashboard for further investigation. The Locking dashboard opens by clicking the link in the Actions tab. The Locking dashboard not just list all occurred deadlock events, it groups the deadlocks and other locking problems by connection attributes, for example, client application name or client user that causes the problems. The grouping by connection attributes is based on the client information fields that you can set in your application for each connection. Grouping by connection attributes results in so called workload clusters, each containing multiple connections with the same connection attribute values. This helps to distinguish applications having real locking problems from others.

In our example we see in Figure 7-3 on page 254 that we have workload clusters for the client users `db2user1` and `db2power` and workload clusters for the client applications `db2user1` and `db2power`. The client users `db2user1` and `db2power` use the client applications named `db2user1` and `db2power` which result in multiple database connections per client application and per client user. Both client applications get deadlocks which means that both users are impacted, `db2power` user with 28 deadlocks in the monitoring interval is a little bit more impacted than `db2user1` with 23 deadlocks. Clicking the `db2power` user shows in the lower table all deadlock events in that the `db2power` participates.

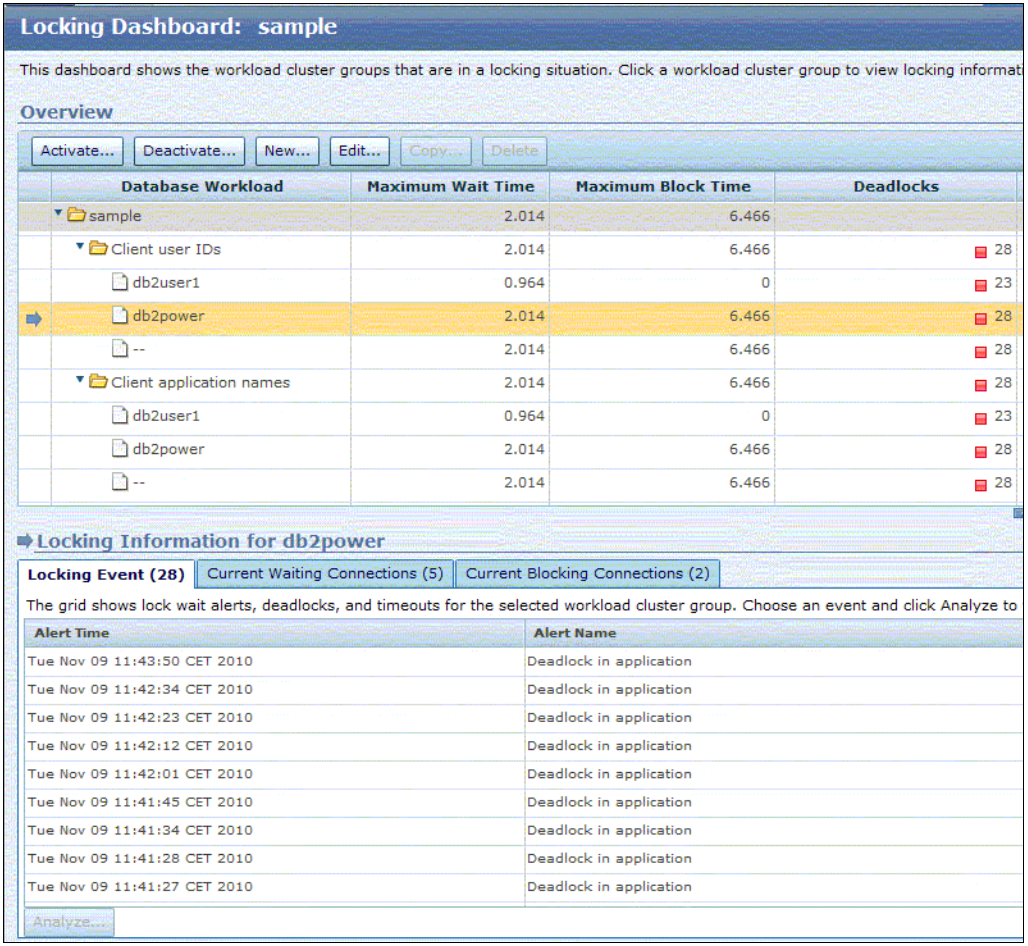
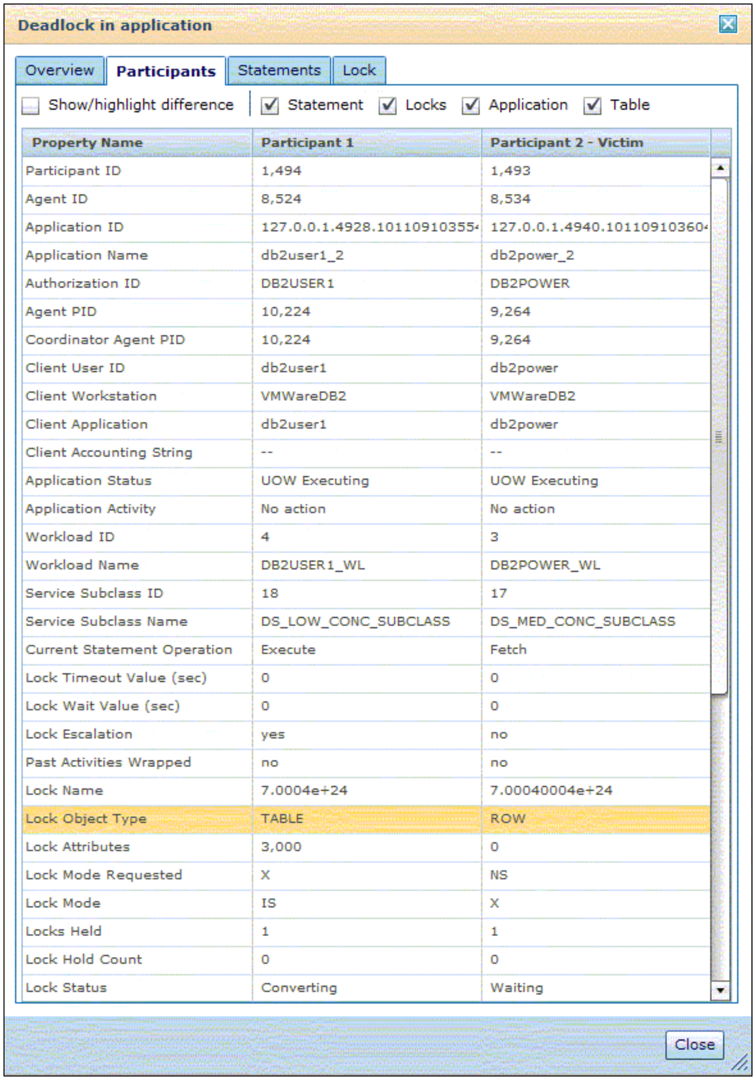


Figure 7-3 Locking dashboard showing deadlock events

Select one deadlock event in the lower table and click **Analyze** to get the deadlock details. An overlay window opens that shows the details about the involved participants and the executed SQL statements. The amount of details provided vary dependent on the detail level with that the lock or deadlock event monitor runs on the monitored database. You select the level of detail during configure monitoring.

In our example, every deadlock event shows similar details. We noticed that the DB2 table that participates most in the deadlocks is `DB2USER1.OPM_SAMPLE_TABLE`.

In most deadlocks, one participating connection holds a TABLE lock instead of single ROW locks. We illustrate this in Figure 7-4 which shows the details of the participating connections in the deadlock and in Figure 7-5 on page 256 which shows the SQL statements the participating connections execute.



Property Name	Participant 1	Participant 2 - Victim
Participant ID	1,494	1,493
Agent ID	8,524	8,534
Application ID	127.0.0.1.4928.10110910355	127.0.0.1.4940.10110910360
Application Name	db2user1_2	db2power_2
Authorization ID	DB2USER1	DB2POWER
Agent PID	10,224	9,264
Coordinator Agent PID	10,224	9,264
Client User ID	db2user1	db2power
Client Workstation	VMWareDB2	VMWareDB2
Client Application	db2user1	db2power
Client Accounting String	--	--
Application Status	UOW Executing	UOW Executing
Application Activity	No action	No action
Workload ID	4	3
Workload Name	DB2USER1_WL	DB2POWER_WL
Service Subclass ID	18	17
Service Subclass Name	DS_LOW_CONC_SUBCLASS	DS_MED_CONC_SUBCLASS
Current Statement Operation	Execute	Fetch
Lock Timeout Value (sec)	0	0
Lock Wait Value (sec)	0	0
Lock Escalation	yes	no
Past Activities Wrapped	no	no
Lock Name	7.0004e+24	7.00040004e+24
Lock Object Type	TABLE	ROW
Lock Attributes	3,000	0
Lock Mode Requested	X	NS
Lock Mode	IS	X
Locks Held	1	1
Lock Hold Count	0	0
Lock Status	Converting	Waiting

Figure 7-4 Deadlock participant details

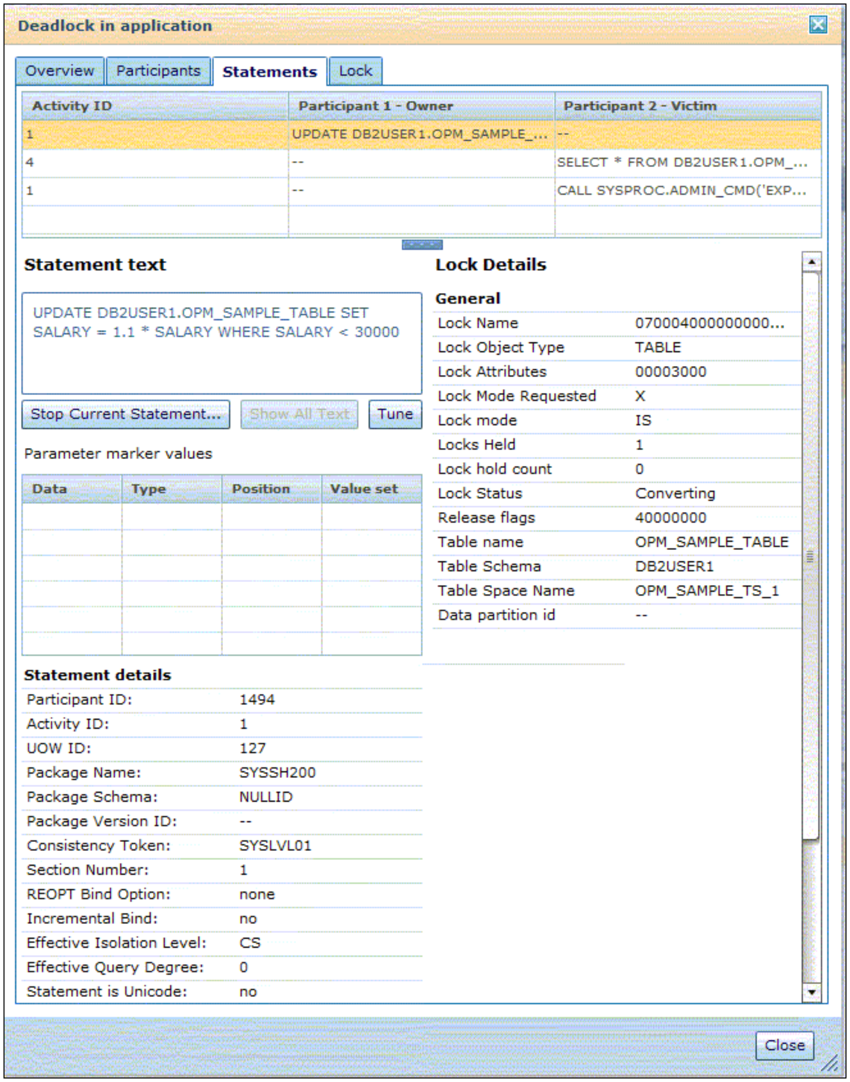


Figure 7-5 Deadlock statement details

From the TABLE locks that we see in each of the deadlocks we suspect that lock escalations occurred.

We confirm this suspicion in the next steps using other dashboards but first we want to explain what additional information you can gain from the Locking dashboard. Each deadlock causes lock wait times for the involved connections before the victim connection is rolled back. The columns *Maximum Wait Time* and *Maximum Block Time* show the maximum wait or block time of all the connections belonging to the selected workload cluster by the end timestamp of the monitoring interval. In the example in Figure 7-6, we select the `db2power` client user and select the **Current Waiting Connections** tab in the lower table. All connections of client user `db2power` are listed together with wait time. The highest wait time is 2.014 seconds. This information helps us to find the connection that is most affected by the wait time. Especially in lock wait situations that do not result in a deadlock, this is important in order to concentrate on troubleshooting the connections waiting longest.

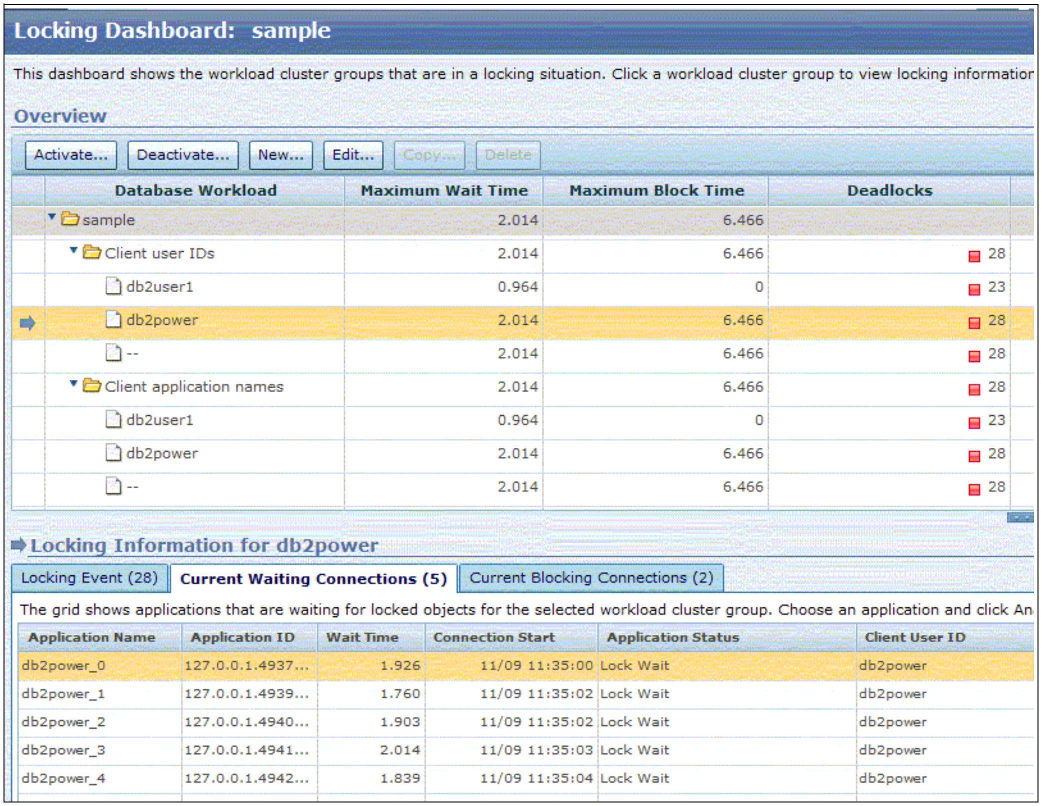


Figure 7-6 Waiting connections of user ID db2power

Selecting one connection and clicking **Analyze** gives you the details of the lock wait situation in that this connection is involved, including the complete tree of connections that wait and block each other. The details and the lock wait tree are calculated based on data collected at the last timestamp within the selecting monitoring interval on this dashboard. For a lock wait situation, the first connection listed in the tree is the top blocking connection. If the lock wait situation still exists, forcing this connection or canceling the running statement would resolve the locking situation to a great extent if not completely. In case of deadlocks as we have here, DB2 resolves the situation by rolling back a connection. In a deadlock situation you do not have one top blocking connection but have a group of connections blocking each other as a circle. Therefore, it is not possible to show a tree. Optim Performance Manager illustrates this circle by showing all connections involved in the deadlock in a tree and placing links next to the top connection and next to the leaf connection. Selecting these links will show the same connections again either as holding or waiting. This way you recognize that this lock tree illustrates a circle of waiting connections and represents a deadlock situation rather than a lock wait situation.

The Analyze Locking Situations panel in Figure 7-7 shows the lock tree after pressing the Analyze button. You see the *Show Waiting* and *Show Holding* links. We press **Show Waiting** and **Show Holding** which results in the lock tree as shown in Figure 7-8 on page 259. You recognize the never ending circle.

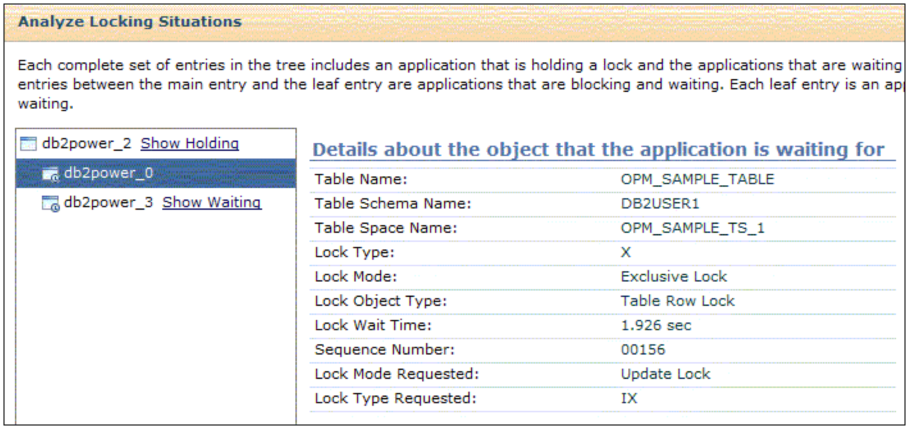


Figure 7-7 Initial deadlock tree

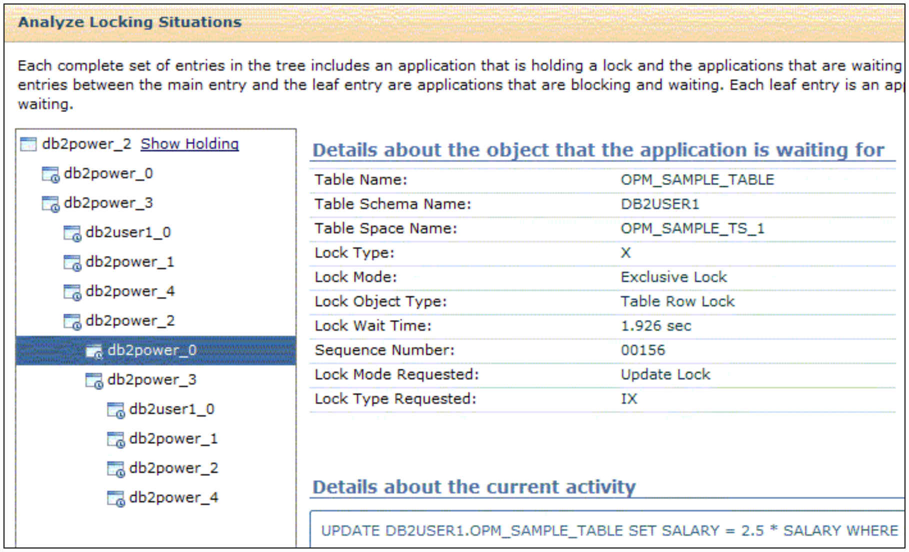


Figure 7-8 Expanded deadlock tree

7.1.1 Identifying lock escalations as the cause of the deadlocks

The Analyze Locking Situations panel displays details for each involved connection and we use that to verify that the connections have lock escalations. Lock escalation is the process of replacing row locks with table locks in order to reduce the number of locks in the list. Figure 7-9 on page 260 shows the connection details of the `db2power_2` application. In the *Details for all activities* section, you see that this connection has already had 16 lock escalations.

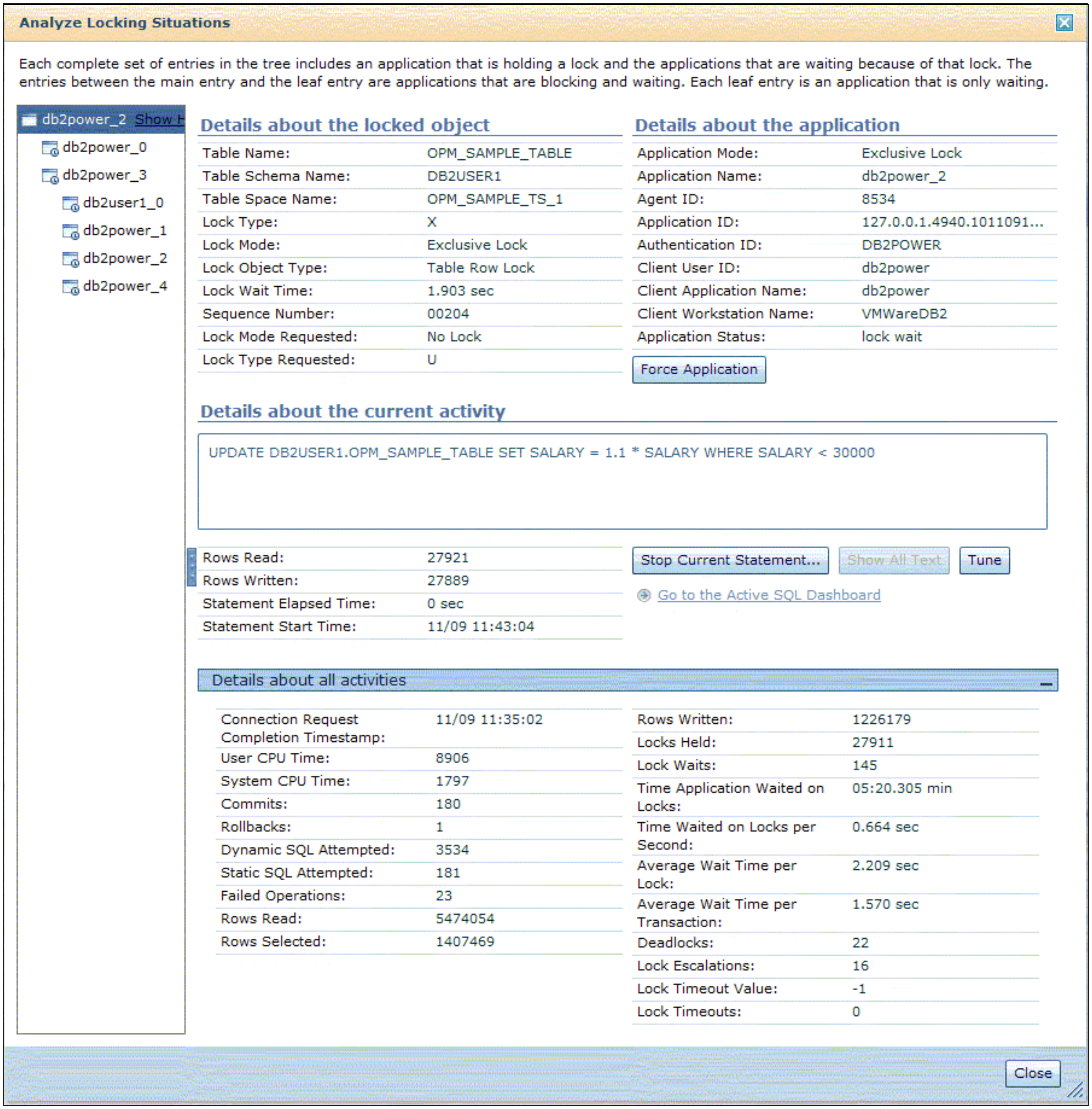


Figure 7-9 Application details and lock escalations for db2power_2

The Overview dashboard displays the total number of lock escalations for the database in the displayed monitoring time, which is 10 minutes for our example. Figure 7-10 on page 261 shows the Locking section of the Overview dashboard with 43 lock escalations.

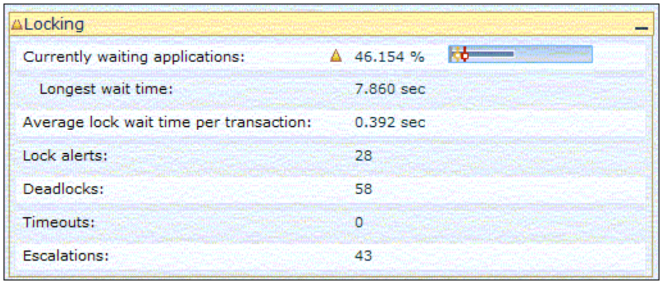


Figure 7-10 Lock escalations on the Overview dashboard

Further, we use the Memory dashboard, part of that shown in Figure 7-11 on page 262, to check the size and the utilization of the Lock list memory area. The Lock list memory area contains the locks held by all applications concurrently connected to the database. We select the **Lock List** layer from the drop down box above the graph and Optim Performance Manager highlights the lock list memory area in the graph and in the details table. From the graph you see that quite a high amount of memory used by the database is allocated for the lock list and from the details you see the utilization of the lock list at the end of the monitoring interval is 43,659%. Lock escalations occur either if the lock list memory area runs out of space or if a connection uses more space in the lock list memory area as defined in the MAXLOCKS database configuration parameter. Around 43% for the lock list memory is a high number but does not indicate an out of space problem.

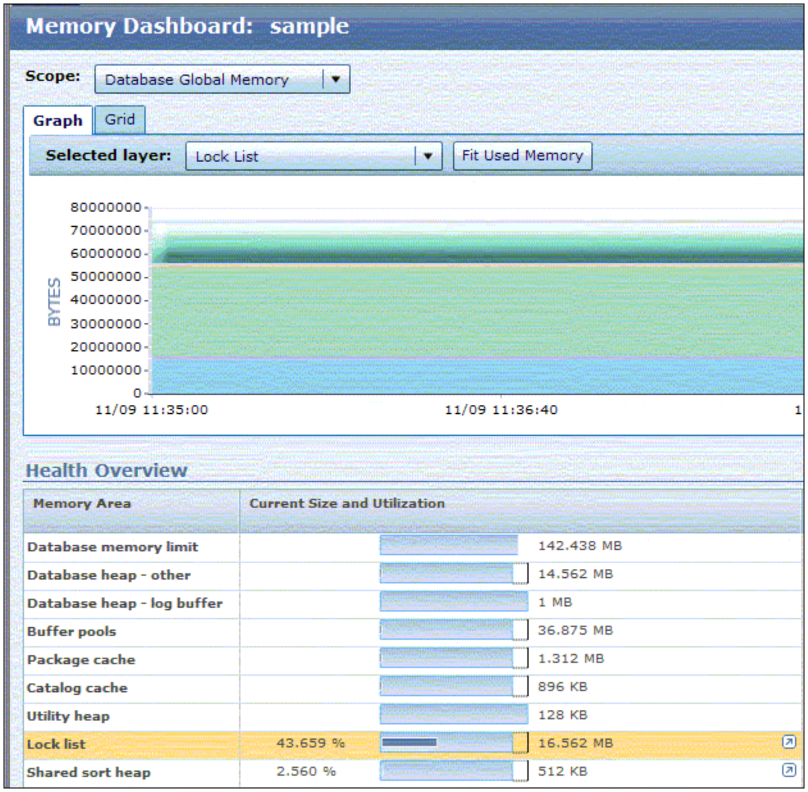


Figure 7-11 Lock list on Memory dashboard

By opening the graph for the lock list utilization in Figure 7-12 on page 263, we check the lock list utilization over the complete monitoring interval and not just at the end. Since we have an even utilization over the monitoring interval of around 43%, we confirm that we do not have an out of space problem as the cause for the lock escalations.

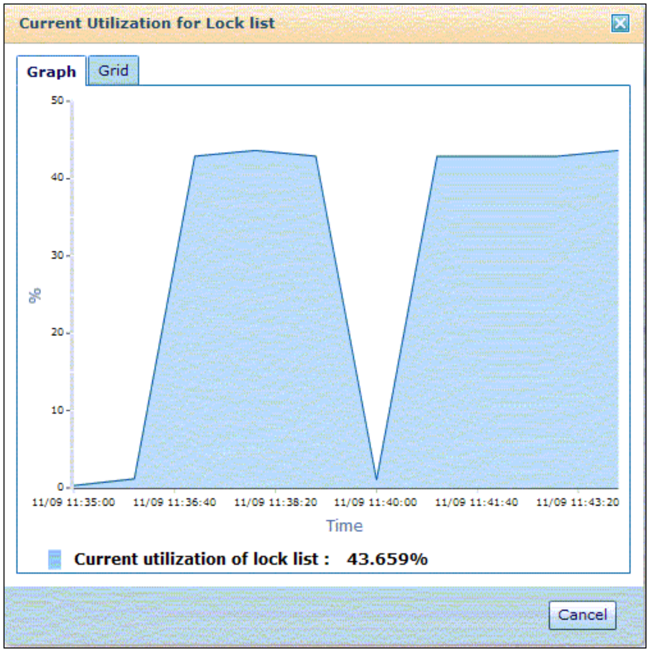


Figure 7-12 Current lock list utilization chart

Next, we check the MAXLOCKS parameter. In general, the lock escalations and the resulting concurrency problems such has deadlocks can be avoided by tuning the LOCKLIST and MAXLOCKS database configuration parameters if SELF_TUNING_MEM is set to OFF as in our example. The command **`db2 get db cfg for sample`** returns the following result for the parameters responsible for this situation:

```
Self tuning memory                      (SELF_TUNING_MEM) = OFF
Max storage for lock list (4KB)                (LOCKLIST) = 4096
Percent. of lock lists per application         (MAXLOCKS) = 22
```

From the Memory dashboard we have confirmed that the lock list memory area is big enough for the moment. However, MAXLOCKS is set to 22% only that means when the number of locks held by any one connection reaches this percentage of the total lock list size, lock escalation will occur for the locks held by that connection.

As a summary, the tuning action for this troubleshooting scenario is to increase the MAXLOCKS configuration parameter in order to avoid the lock escalations and the high number of deadlocks. If still too many lock escalations occur, increase LOCKLIST. The DB2 documentation provides guidelines about how to calculate adequate sizes for these parameters based on the number of connections to the database. See this link:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.config.doc/doc/r0000267.html

Alternatively, you can set STMM to ON to let DB2 adapt the LOCKLIST parameter dynamically.

7.2 Troubleshooting bad response times

Assume that you receive complains from the users of your applications about bad response times. You have the lock event monitor disabled on the monitored database thus you do not receive alert notification about lock wait, deadlock, or lock timeout events. In this situation, you can login to the Optim Performance Manager web console and open the Extended Insight dashboard to verify the response time degradation.

In Figure 7-13 on page 265, the Extended Insight dashboard shows an Average End-to-End Response Time of 9.5 secondes for the JCC applications workload cluster. This is indeed much longer than the expected and normal average response times of less than two seconds. In the graph you see the warning and critical threshold lines. For the JCC applications we specified a transaction response of two seconds as warning threshold and a transaction response time of four seconds as critical threshold. On the dashboard we display a monitoring interval of 30 minutes. During this monitoring interval, around 14% of all transactions breached the warning threshold and about 85% breached the critical threshold.

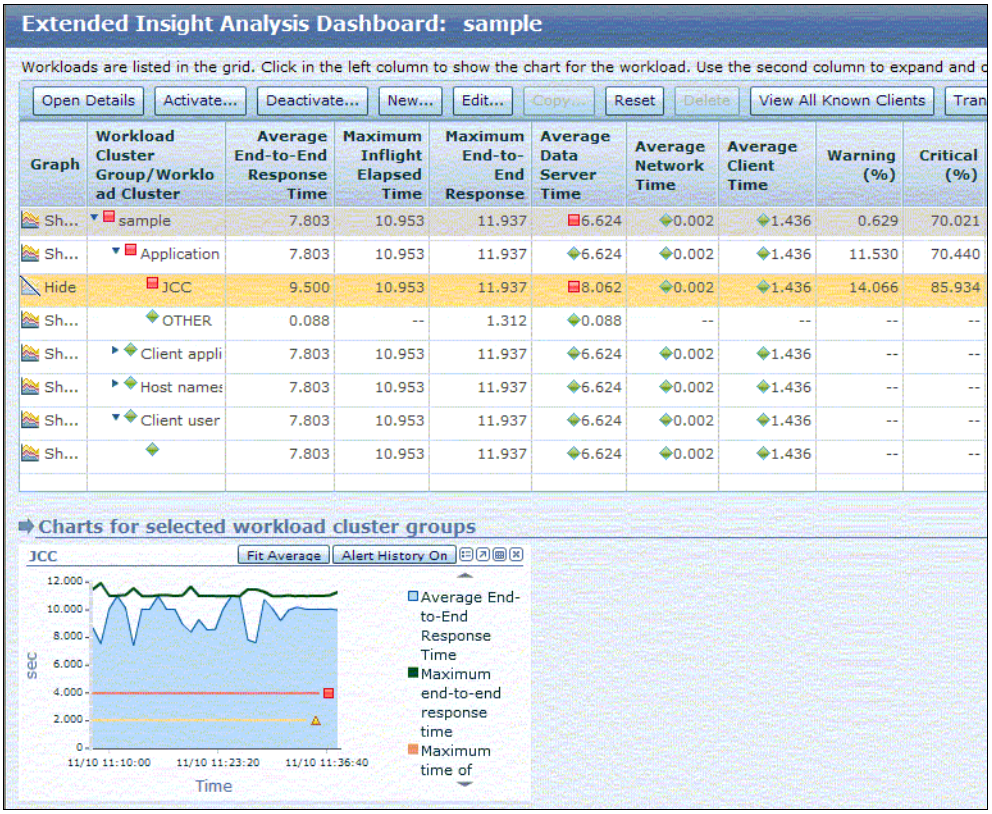


Figure 7-13 Extended Insight dashboard showing high response times for JCC

If you press the Details icon on the chart then an overlay panel opens showing you more details including the response time histogram, see Figure 7-14. This helps to understand the response time distribution among all transactions from that the average response time is calculated.

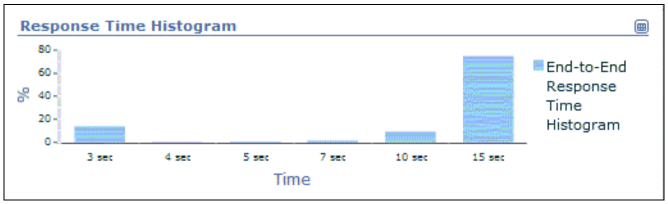


Figure 7-14 Response time histogram for JCC applications

In addition to the high response time values and the warning and critical percentages, you can determine from the Extended Insight dashboard where most of the response time was spent: On the data server, network, or on the client that runs the application. In our example an average of around eight seconds of each transaction is spent on the data server that identifies the cause of the bad response times residing from the data server. See column Average Data Server Time column in Figure 7-13 on page 265

Now we drill down to the details of the JCC applications to analyze why the data server response time is so high. On the Extended Insight dashboard, select the JCC workload cluster and double click. The Extended Insight details dashboard displays the colored response time distribution chart showing graphically in which time layers the time was spent, see Figure 7-15. In our example most of the time is spent for lock waits (the highlighted rose layer). Select the Average lock wait time layer to get more information about this time layer below the distribution chart. The Locking Problem chart shows a high number of *deadlocks*.

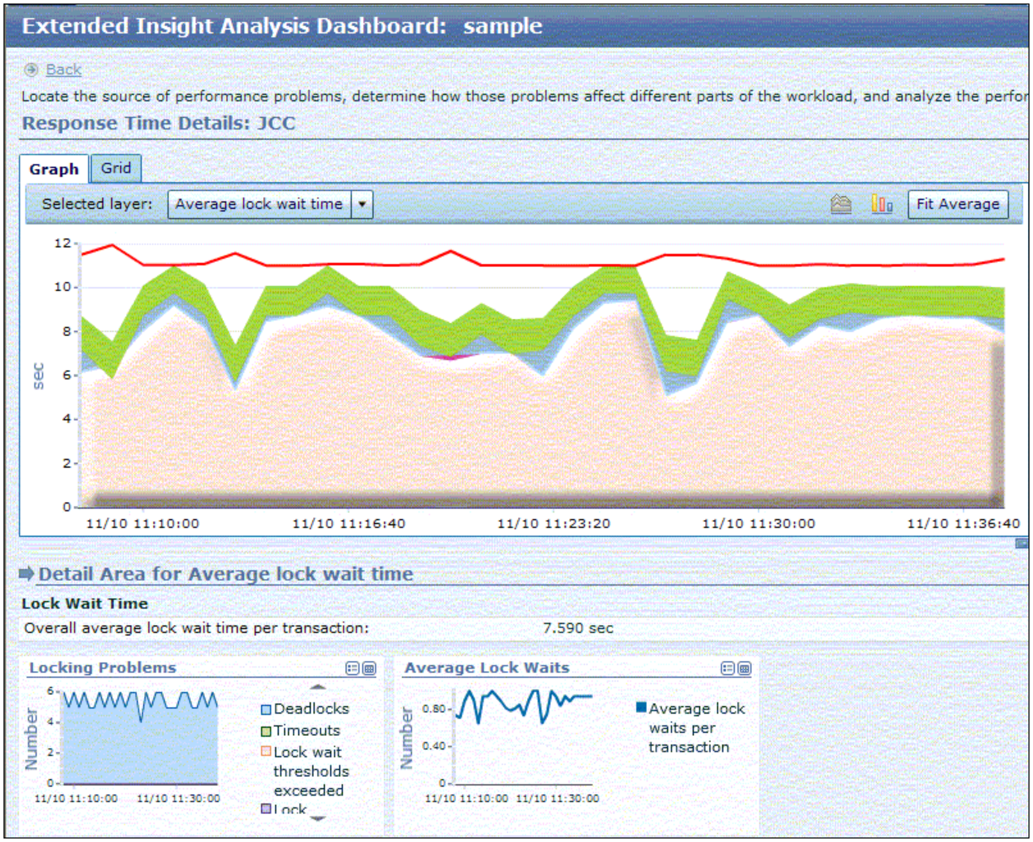


Figure 7-15 Response time distribution chart and lock wait time details for JCC application

Next to the response time distribution graph, the SQL statements are listed that the JCC applications execute during the monitoring interval. The top statements are UPDATE statements. We select the first one to get execution details. The statement execution details consist of two parts. One part shows the end-to-end execution details, including overall time distribution among data server, network, application, and the return code of the statement. The important information of this part is shown in Figure 7-16. We have a failure rate of 43% of this statement. The first SQL code is -911, which is the SQL code the victim application receives in case of a deadlock. Considering the high number of deadlocks we have seen in the Locking Problem chart in Figure 7-16, we assume that most failed statement executions receive SQL code -911.

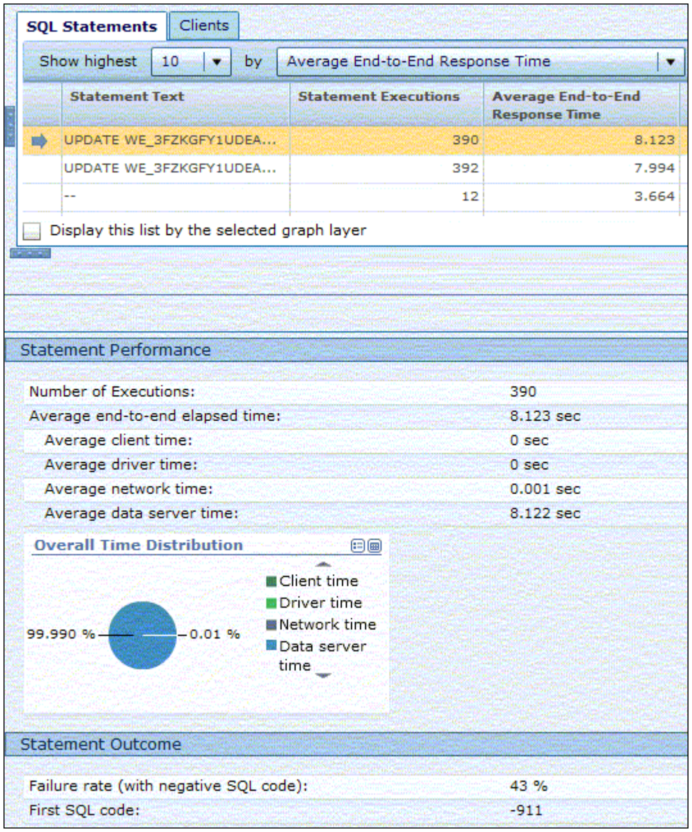


Figure 7-16 General statement execution information

The other part of statement execution details shows the data server execution details of the selected statement. This information is only available if the monitored database runs on DB2 V9.7 Fix Pack 1 or higher. This time the Overall Time Distribution chart in Figure 7-17 on page 268 shows the distribution of the statement execution time on the data server. It confirms that the statement spends most time for lock waits. Additionally, we see the reason for the high number of deadlocks. The statement uses isolation level repeatable read (RR).

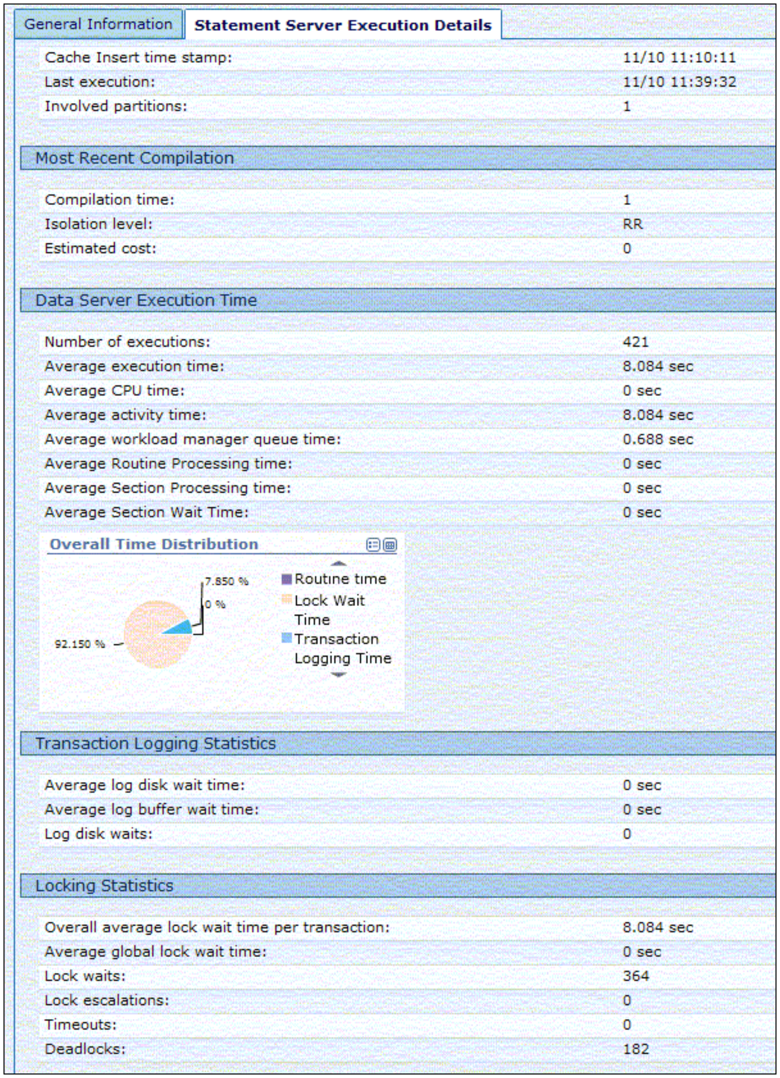


Figure 7-17 Data server statement execution information

The RR isolation level locks all the rows that an application references during a transaction. Every referenced row is locked, not just the rows that are retrieved. For example, if you scan 10 000 rows and apply predicates to them, locks are held on all 10 000 rows, even if, say, only 10 rows qualify. Because RR can acquire a considerable number of locks, this number might exceed limits specified by the LOCKLIST and MAXLOCKS database configuration parameters. To avoid lock escalation, the optimizer might elect to acquire a single table-level lock for an index scan, if it appears that lock escalation is likely. If you do not want table-level locking, use the read stability isolation level.

Let us shortly verify that we have no lock escalations but table locks. According to the statement execution information on the data server in Figure 7-17 on page 268, no lock escalation takes place. To verify the usage of table locks, we open the Locking dashboard that represents the deadlocks as locking situations. Remember that Deadlock alerts are disabled, therefore, we can not check the event details for table locks. We select one of the connections involved in the deadlocks and press **Analyze**. This opens the panel shown in Figure 7-18. The Lock Object Type field shows *Table Lock*.

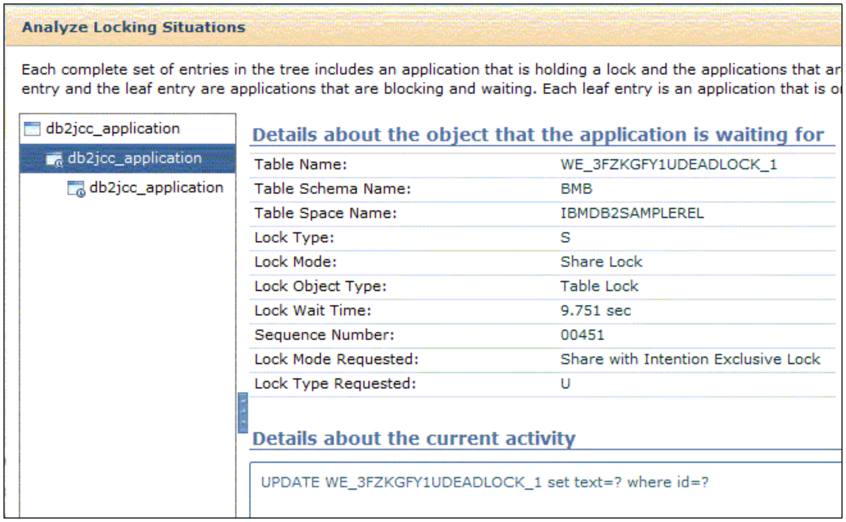


Figure 7-18 Lock details

To solve the high number of deadlocks in this example, use a lower isolation level for the statement executions.

8

Extended Insight analysis

Optim Performance Manager Extended Insight extends the performance monitoring capability from data server to the whole system. It enables users to monitor their production system end- to-end, including application client, application server, network, and data server.

Though the symptom of a performance issue often appears first in an application, the cause could be in anywhere of the entire system. For example, a user may feel that his application runs very slowly and the possible causes could be a bad application algorithm, an overloaded application server, bad network condition, or bad data server performance. From Extended Insight dashboard, you can see which application runs into a problem and drill down on the Extended Insight dashboard from panel to panel for details. Each panel provides the performance details of certain tier - application client, application server, network, or data server, for you to figure out the root causes. With the root causes on hand, you can then perform proper performance tuning including using the advice provided by Optim Query Tuner.

This chapter provides scenarios about using Optim Performance Manager Extended Insight toinvestigatethe cause of performance issues on the monitored system. Some scenarios involve tuning the performance issue using Optim Query Tuner and fixing the performance issue by applying the advice from Optim Query Tuner. We also explore using Extended Insight with WebSphere applications and dive deepr into workload clusters.

8.1 Application running slowly caused by index issue

IBM Optim Query Tuner (OQT) V2.2 Fix Pack 2 or later release is integrated well with Optim Performance Manager. When you use the Optim Performance Manager to identify the poor performed queries, you can pass the queries to Optim Query Tuner and run tuning advisor for tuning suggestions. The suggestions could include different optimization options for the data server, such as running statistics, creating indexes, and changing queries.

In this section, we describe how to use the Optim Performance Manager Extended Insight Analysis dashboards to analyze a poor performed application to identify the slow running query as well as how to obtain optimization advice from Optim Query Tuner.

The installation of Optim Query Tuner for DB2 for Linux, UNIX, and Windows affects two locations, a DB2 database server and a client system. To leverage Optim Query Tuner with Optim Performance Manger, on the monitored database, you must run an installation program to activate the product license and enable the Index Advisor stored procedure. On the client system where you run the browser to access Optim Performance Manager console, you must install the Optim Query Tuner client.

Once the Optim Query Tuner is set up, you can perform SQL statement tuning using Optim Query Tuner form the system where you access the Optim Performance Manager web console. Before tuning the SQL statements, you have to launch Optim Query Tuner first and make sure the data bridge is enabled (the icon shown in Figure 8-1 is selected).

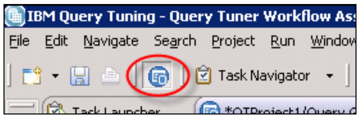


Figure 8-1 Click this button to enable data bridge

To learn more about Optim Query Tuner, refer to the Information Center by the following link:

`http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.qrytune.installconfig.doc/topics/installconfig.html`

8.1.1 Observing a performance issue

The Optim Performance Manager console Extended Insight Analysis Dashboard displays the application end-to-end response time and the breakdown details in both tabular and chart formats. The *end-to-end response time* means the overall response time of a transaction that includes the elapse time on application client, application server, network, and data server. It measures the time period between the application starts a database transaction on behalf of the users' request and ends the transaction by a commit or rollback. The time the application used to display the data on any user interface is not counted. For measuring the entire time used, use IBM Tivoli Composite Application Manager (ITCAM). We discussed the monitoring with ITCAM in Chapter 10, "Integration with Tivoli monitoring components" on page 323.

Setting thresholds for applications

On a monitored database, the applications are usually for various business functions and the response time requirements are different. By default, on the Extended Insight Analysis Dashboard, all the applications are in the same workload cluster group. You can categorize the applications by business characteristics and group them into workload cluster groups. A workload cluster group can have one or many applications, each of them is a workload cluster. Each workload cluster group can have a global response time threshold set or each workload cluster in the group can have an individual response time threshold set

A response time threshold consists of warning threshold and critical threshold. During the specified time duration, once the maximum response time of an application in the workload cluster group exceeds the warning threshold, a warning marked as a yellow triangle appears on the Extended Insight Analysis Dashboard. When the maximum response time exceeds the critical threshold, a critical alert marked as a red square appears.

In this scenario, we use the procedure described in 4.4, "Extended Insight dashboard" on page 178 to create a workload cluster group named `Checking Order Status`. The workload cluster group contains one application `orderstatus`. We defined four seconds for the warning threshold and seven seconds as the critical threshold for this workload cluster group as shown in Figure 8-2.

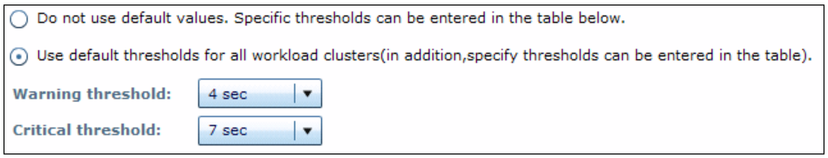


Figure 8-2 Set response time threshold for the workload cluster group

Monitoring the work cluster group

We started the `orderstatus` application and let it run for a while. Initially, this application runs with acceptable performance as shown in Figure 8-3. The maximum end-to-end response time is below warning threshold (the status marked with green diamond). The Warning (%) and Critical (%) show 0 for the one hour we were monitoring.

Graph	Workload Cluster Group/Workload Cluster	Average End-to-End Response Time	Maximum Inflight Elapsed Time	Maximum End-to-End Response Time	Average Data Server Time	Average Network Time	Average Client Time	Warning (%)	Critical (%)	Transactions (/min)
Sh...	irww51	0.881	6:18.725	7:08.483	0.856	0.012	0.012	--	--	271.705
Sh...	Client application name	0.881	6:18.725	7:08.483	0.856	0.012	0.012	--	--	271.705
Sh...	Checking Order Status	2.468	3.326	3.960	2.462	0.004	0.002	0	0	65.656
Sh...	orderstatus	2.468	3.326	3.960	2.462	0.004	0.002	0	0	65.656

Figure 8-3 Workload cluster group runs without performance issue

Later, we observe that the application performance slow down in the last one hour of the monitor time frame. 64% of the transactions of the `Checking Order Status` workload cluster group runs with response time beyond the critical threshold. See Figure 8-4. Note that you can configure the Optim Performance Manager to send alert messages by e-mail.

Graph	Workload Cluster Group/Workload Cluster	Average End-to-End Response Time	Maximum Inflight Elapsed Time	Maximum End-to-End Response Time	Average Data Server Time	Average Network Time	Average Client...	Warning (%)	Critical (%)
Sh...	irww51	32.149	26:29.251	27:08.483	32.129	0.010	0.011	--	--
Sh...	Client application names	32.149	26:29.251	27:08.483	32.129	0.010	0.011	--	--
Sh...	Checking Order Status	01:43.757	11:55.326	12:26.960	01:43.757	0	0.002	0	64
Sh...	orderstatus	01:43.757	11:55.326	12:26.960	01:43.757	0	0.002	0	64

Figure 8-4 Workload cluster group runs into a response issue

8.1.2 Figuring out the cause and tuning

To find out why 60% of transactions runs beyond the response time threshold, we click **Checking Order Status** workload cluster group shown in Figure 8-3 to open the detail page.

Figure 8-5 on page 275 shows the distribution of end-to-end response time. This chart shows how the response time varies in a specified period of time for each tier, application client, application server, network, and data server. The end-to-end response time growth can be caused by performance degradation of several tiers. The response time distribution chart clearly show the trend and distribution of each tier allowing you to identify the tiers with problem. From this graph, you also can tell the turn-point that the response time started growing up.

With the time the performance changed narrowed down, you can check if there is anything changed on the system during that time frame that might cause the respond time increased.

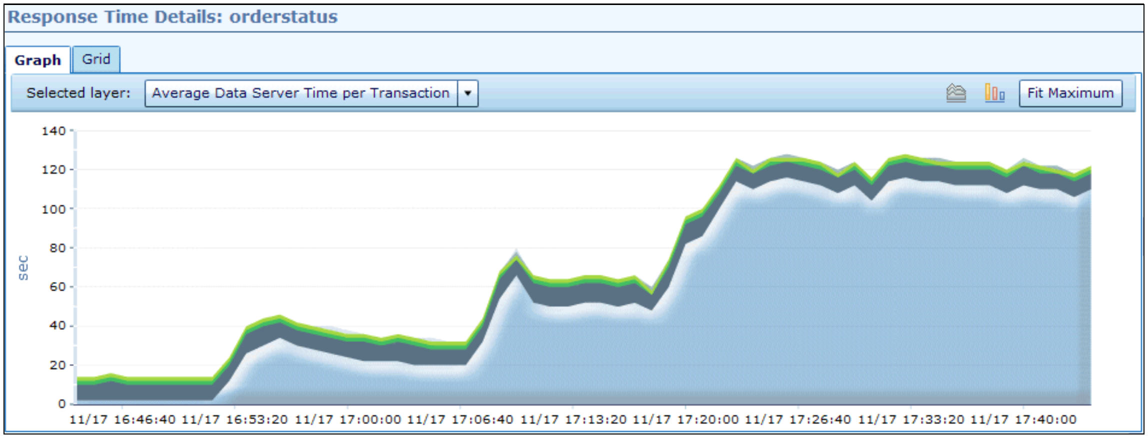
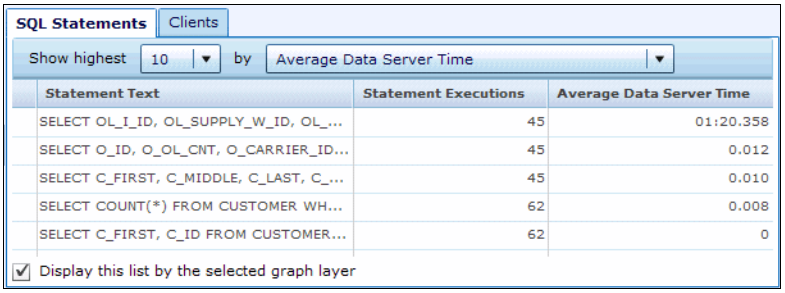


Figure 8-5 Response time distribution chart

In Figure 8-5, we see that the end-to-end response time started to increase at 16:53 and climbed up quickly in about 30 minutes to a high level. To learn what is the tier that contributes the most to the end-to-end response time, we click the tier in light blue at the bottom. The label in the Selected layer field shows that it is the *Average Data Server time per Transaction*. The long elapse time of data server led to the reported performance issue.

Next step is to check which SQL statements take the time. On the Extended Insight Analysis dashboard, click **Open Details**. On the right of the Extended Insight detail page, we choose to show the top 10 SQL statements which take longest “Average Data Server Time” as shown in Figure 8-6 on page 275. In this scenario, we have one SQL statement that takes a much longer elapse time on data server than the others.


SQL Statements | Clients

Show highest 10 by Average Data Server Time

Statement Text	Statement Executions	Average Data Server Time
SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_...	45	01:20.358
SELECT O_ID, O_OL_CNT, O_CARRIER_ID...	45	0.012
SELECT C_FIRST, C_MIDDLE, C_LAST, C_...	45	0.010
SELECT COUNT(*) FROM CUSTOMER WH...	62	0.008
SELECT C_FIRST, C_ID FROM CUSTOMER...	62	0

☑ Display this list by the selected graph layer

Figure 8-6 Top SQL statements which longest average data server time

Click the SQL statement in the Statement Text field and the Statement Performance chart shows. This chart tells how the execution time of this SQL statement is spent. The Overall Time Distribution pie chart shown in Figure 8-7 confirms that almost 100% of the end-to-end response time of our SQL is spent on the data server. We can proceed to tune this SQL on the data server.

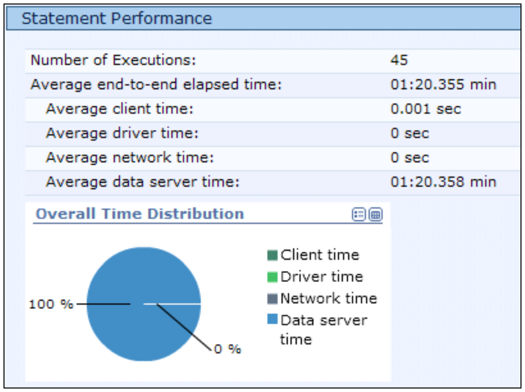


Figure 8-7 Response time distribution of an SQL statement

On the left side of the pie-chart shown in Figure 8-7, there is a text field showing the whole SQL statement. Because we have launched the Optim Query Tuner enabled the data bridge, we click **Tune** (Figure 8-8 on page 276) to pass this SQL to Optim Query Tune for tuning.

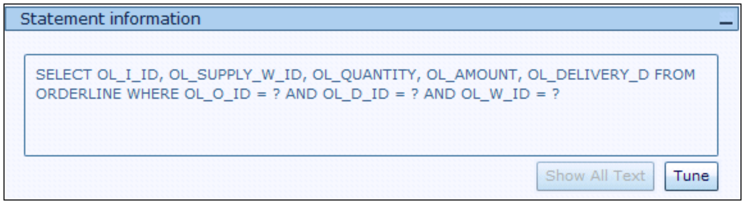


Figure 8-8 SQL statement

Figure 8-9 on page 277 shows the tuning panel of the Optim Query Tuner with our SQL statement to be tuned.

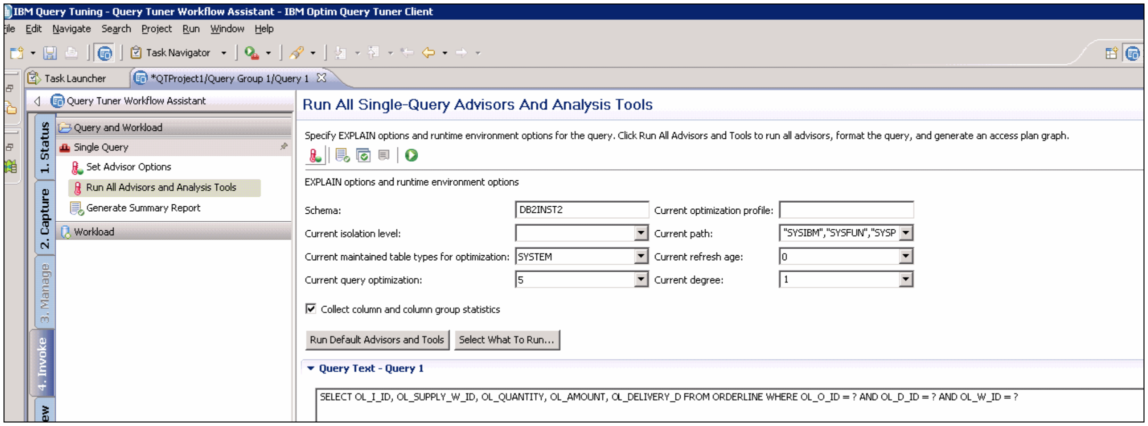


Figure 8-9 SQL statement passed into Optim Query Tuner and ready for tuning

Click **Select What to Run** (Figure 8-9 on page 277), the Select Query-Tuning Activities panel shows (Figure 8-10). You can select multiple tuning activities and run them together or you can run one activity at a time. In this scenario, we select all the tuning activities.

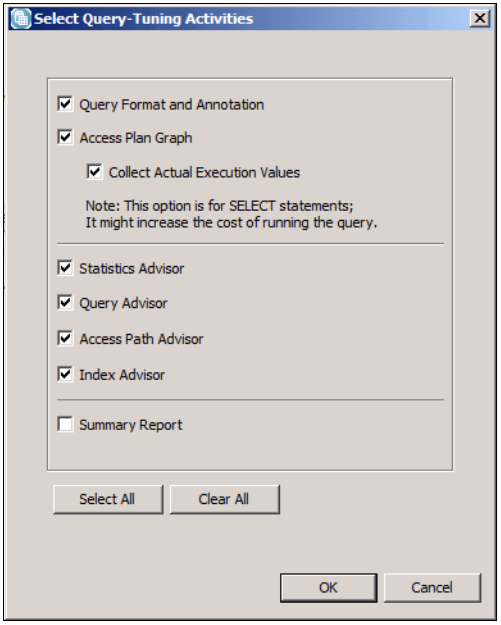


Figure 8-10 Select Query-Tuning Activities

At the end, Optim Query Tuner returns an index advice about creating indexes, as shown in Figure 8-11 on page 278.

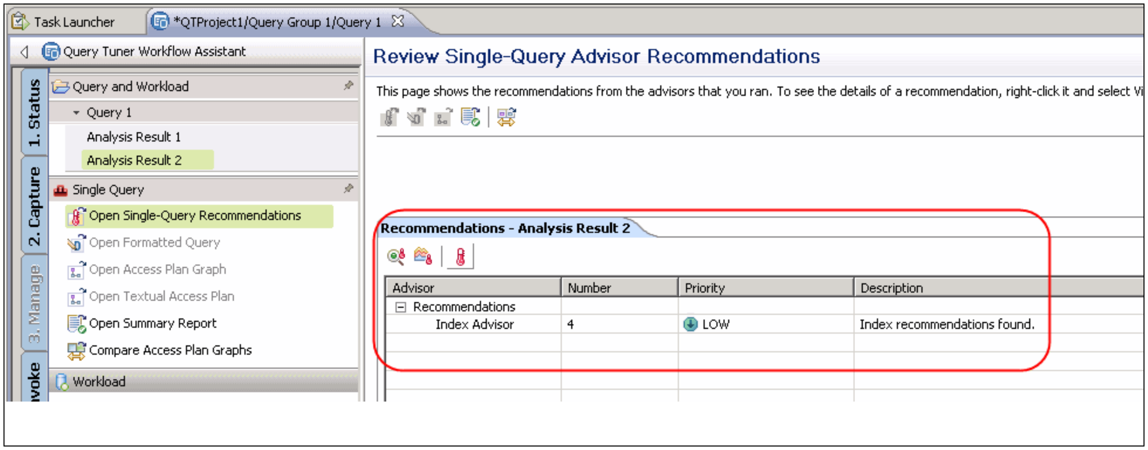


Figure 8-11 Optim Query Tuner advises to create index

We click the index advice to open the details panel, as show in Figure 8-12. As told by Optim Query Tuner, creating the advised index would improve the performance of this SQL by approximately 99%. Optim Query Tuner also gives an estimation on the disk space consumption of this new index.

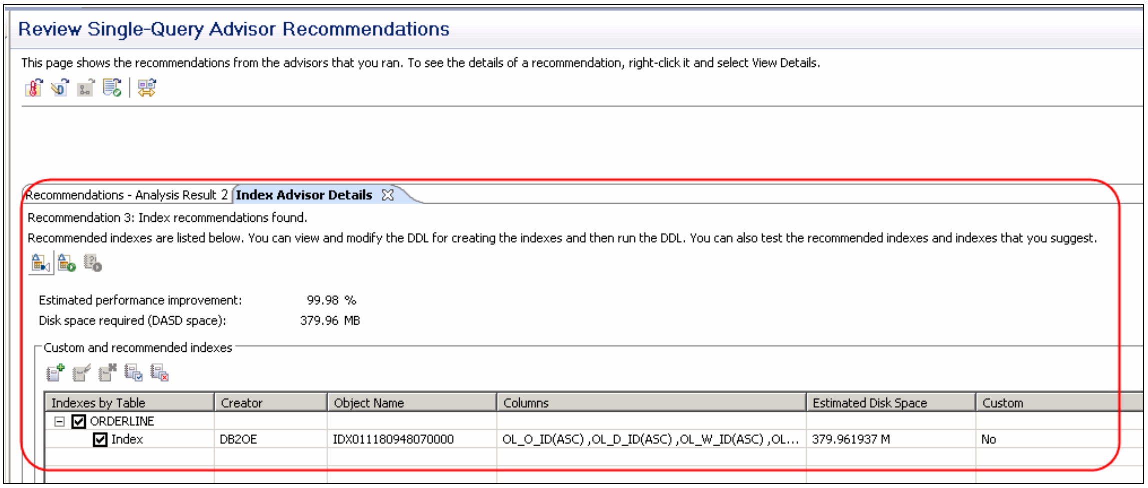


Figure 8-12 Index Advisor Details

We take this advice and create the advised index on our data server. Then we return to the Extended Insight panel. After several minutes, the average end-to-end response time drops below the warning threshold.

For the detail steps about turning SQL statement, refer to Information Center at http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.qrytune.sngqry.doc/topics/tsupertask_true.html

8.2 Diving into the application layer

We already know that Optim Performance Manager can monitor the DB2 database at a deep level. What about the application side? The database may be performing well, but the application is still having slow response times. If you use a monitor that only looks at the application, you may not get the full picture of the slowdown, just like if you only look at a database monitor, you may not determine where the problem is.... only where it *is not*.

8.2.1 Extended Insight analysis approach

Figure 8-13 depicts how the different Optim Performance Manager components can help monitor across the transaction landscape. We can suggest a general approach for how to examine the different layers using Extended Insight.

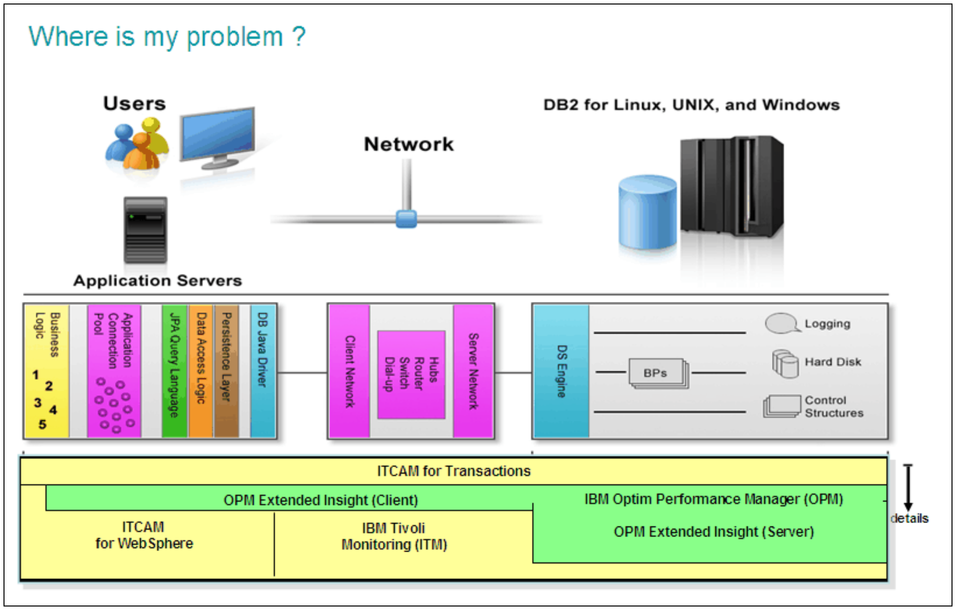


Figure 8-13 Investigating the transaction landscape with Extended Insight

A typical flow of problem analysis in Optim Performance Manager Extended insight is to look at the following metrics:

1. Check the overview page for alerts or outliers.
2. Check layers that contribute most to the response time.
3. Check the details about the top layer's top statement or top client.
4. Check for top statements in spending time within top layers or check top clients that execute the workload

Armed with information you learn by examining the layer metrics, you can more readily narrow down a performance issue to a specific statement or client.

8.2.2 WebSphere scenario description - Great Outdoors Company

In Chapter 4, "Getting to know Optim Performance Manager" on page 147, we learned about the kinds of metrics Optim Performance Manager Extended insight provides. Let us take a look at the case of a WebSphere application that uses DB2 database.

Extended insight can monitor WebSphere applications that use JDBC transactions with standard data source definitions. From the Extended insight perspective, WebSphere DB2 transactions can be considered as a special case of JDBC transactions.

Optim Performance Manager has awareness of WebSphere application server and will collect additional metrics related to the connection pool for the data source. Such metrics are crucial in telling the bigger story of enterprise application performance.

> **FAQ:** Can Optim Performance Manager Extended insight monitor application transactions running on some other kind of application server that is not WebSphere?
>
> In general, the answer is yes, if the application uses standard JDBC transactions. The application would be configured like any other JDBC application, not like a WebSphere application, and you would collect standard JDBC metrics. There are some additional metrics that Optim Performance Manager Extended insight collects specific to WebSphere.

We are using the shopping web site for the fictitious Great Outdoors Company. In our lab environment, the shopping application, GOSales, runs under the WebSphere Application Server that is on the `SD0D03L2` server. Our book environment is described in 3.1, "Lab environment" on page 60.

Suppose we receive a call from the help center saying some customers have complained they cannot complete their purchases because the web site is too slow. You are a DBA, what can you tell them about the web site being slow? We show an example of using Optim Performance Manager Extended Insight to provide analysis to the application and server support staff to help investigate the slowdown.

Extended Insight Dashboard

First, we open the Extended Insight dashboard, as in Figure 8-14 on page 281. What can we observe about the application?

We notice a few alerts have occurred. Some problem icons are shown for the Client layer, and the Data Server layer has a couple of warnings. (This scenario is not really about alerts, therefore, we do not discuss them further.)

The average end-to-end response time is 144 ms. We are familiar with this application, so we know that is higher than normal.

Let us drill down to the overall metrics for the entire database, so double-click on the GOSALES line.

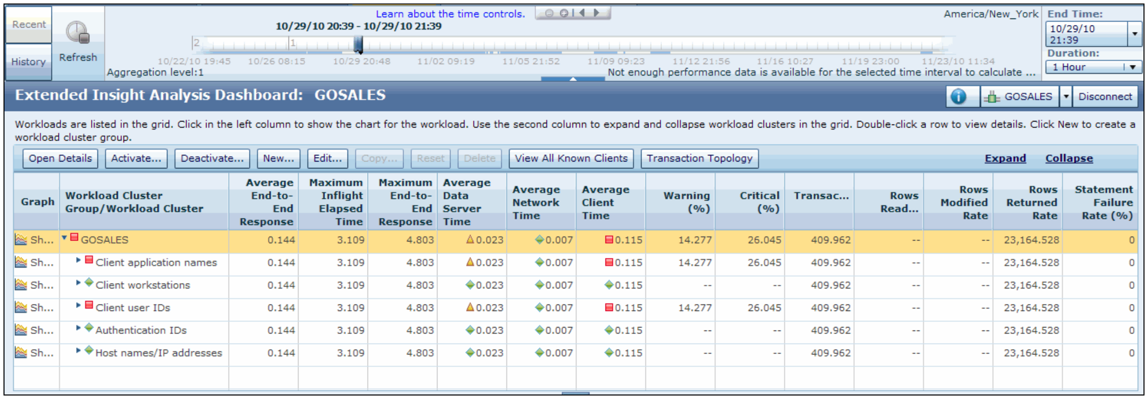


Figure 8-14 Extended Insight Dashboard for GOSales application

Extended Insight Analysis Detail Dashboard - Statements

When you first open the Extended Insight details dashboard, the main response time graph is shown, as in Figure 8-15 on page 282, with the maximum value plotted in red. We observe a steadily increasing maximum, rather than an occasional peak or a steady max. To get a better look at the average values, click **Fit Average**, which will hide the maximum line.

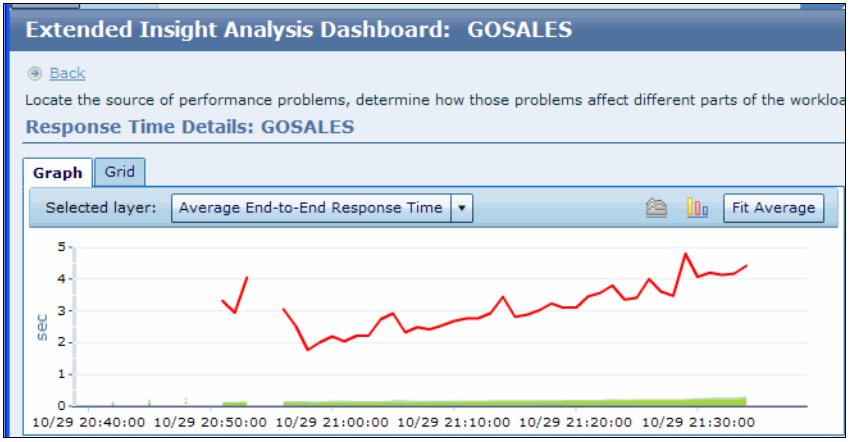


Figure 8-15 Extended Insight Response Time Details chart - with maximum

Figure 8-16 on page 283 shows the entire dashboard. Notice the response time chart is easier to read now without the red maximum plot line. Some charts are labeled in the figure so we can refer to them easily. What can we quickly observe from the information on this page?

- Steadily increasing overall average end-to-end response time (chart A).
- Steadily decreasing transaction throughput (chart B).
- The pie chart (chart C) shows the majority of time is spent in client (application) side:
  - Client time = 79.3%
  - Data server time = 15.8%
  - Network time = 4.8%
- We notice the statement failure rate is 0%, out of a total 23,798 statements executed in the hour. This is good to know, and we can rule out some kind of problem with failing statements.

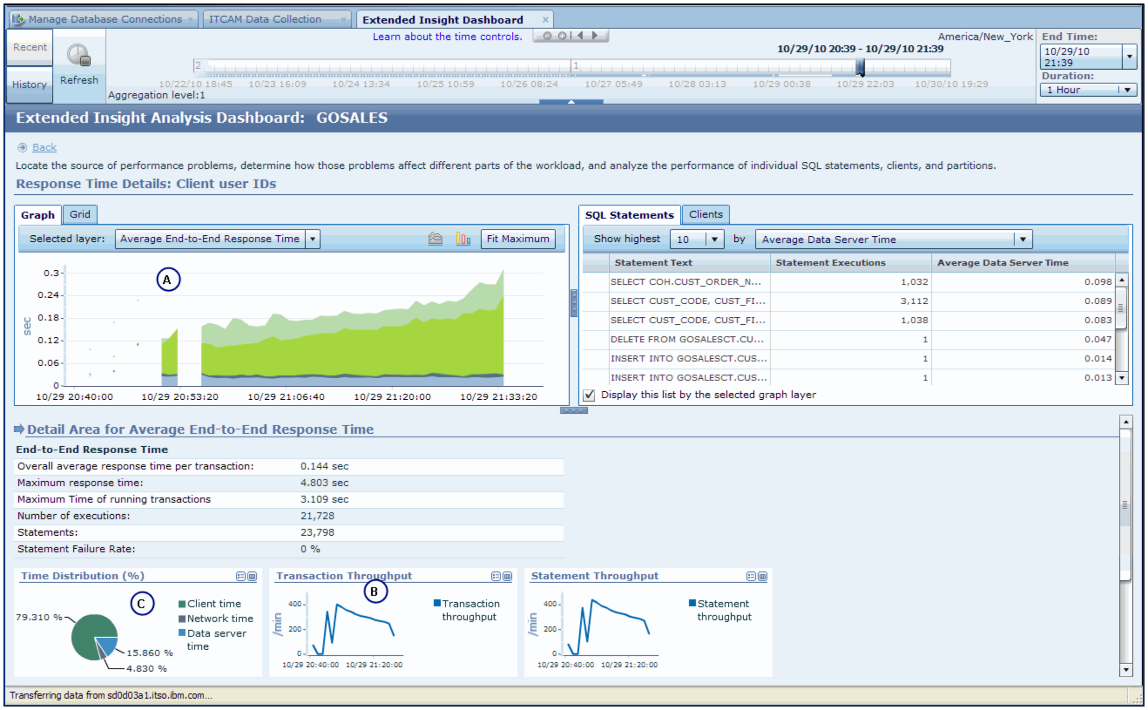


Figure 8-16 Extended Insight details for GOSales transactions

In the SQL Statements section, we see, by default, the 10 statements with the longest average data server time. The top three slowest statements are SELECTs with an average data server time in the 80-90 milliseconds range. We remember the warning alert on data server time, so we make a mental note of these statements, and may come back to look at them later. However, we still do not think these are major contributors to slow response, because we know the total data server time is only 15% of the total, and the actual data server time on the graph (it is the bottom-most area in medium blue) is flat.

Looking at just this information, we could form a hypothesis that the slowdown is not in the database. Though we think we can pass this issue off to the application support team, we do some quick diagnosis that might help the application support staff.

Extended Insight Analysis Dashboard - Clients

Optim Performance Manager shows a decrease in the transaction throughput and increase in response times. The database metrics appear to be otherwise satisfactory. Therefore, the problem may lie in the transaction landscape somewhere before the transaction arrives at DB2, which in this scenario means the WebSphere server. We look next at what kind of client metrics Optim Performance Manager can provide.

Click the **Clients** tab on the Extended insight dashboard, see Figure 8-17 on page 284. We again see the average response time over the selected hour is 144 ms. That is not so good for this application.

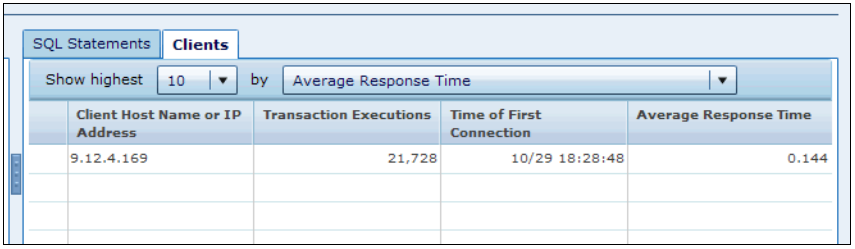


Figure 8-17 Extended Insight client tab

Select the client of interest (we only have one) to view more detail about it. See Figure 8-18 on page 285. The pie charts under the Client Performance section indicate the overall cumulative transaction distribution for the selected client, then a further breakdown within the client time. In this example in Figure 8-18 on page 285, we see 79.4% is spent in client time (the Overall Time Distribution pie chart). The Client Time Distribution pie chart on the right shows the further breakdown of client time into 70.8% spent in the application, a very large 28.7% of the time spent in waiting on the WebSphere connection pool, and a miniscule time less than 1% spent in the (JDBC) driver itself. Now we are suspicious of such a high percentage of connection pool waits.

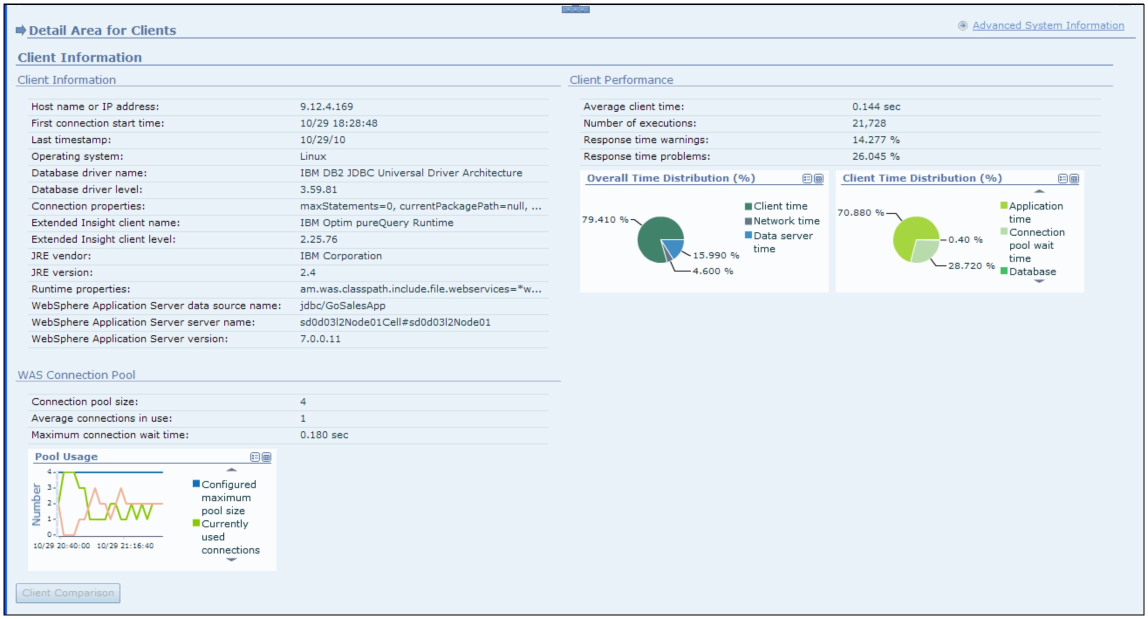


Figure 8-18 Extended Insight client details - WebSphere client

> **Reminder:** The Extended Insight metrics shown on these screens are all calculated within the scope of the duration selected on the time slider. The default is one hour. However, you can expand or contract that interval, or move back in history to view something from another day. Information about using the time slider is described in "Time slider and time controls" on page 155.

The *WAS Connection Pool* section on the lower left shows the configured connection pool size for this client is 4. That seems way too small for the GOSales application, and we should discuss this with the WebSphere administrator. The maximum wait time is shown to be 180ms. If we take another look at the Extended insight details, see Figure 8-19, this time we have selected the WebSphere Connection Wait Time layer in the response time chart. The Average wait time is shown to be 46ms, still pretty long.

Figure 8-19 Extended Insight details for WebSphere Connection Wait Time

We notice the highlighted section of the graph, however, does not show that the connection pool wait time is getting any longer over the hour, it is the application time that is increasing. We need to investigate this, it is either some problem dealing with the waits, or perhaps some secondary problem with the application itself. This all still points to a potential problem in the client side, because ideally we should not have any waiting for connections at all, and again the data server time is flat.

Investigating increase in application time

Let us look at the application layer more. We know the data server time is not increasing, but the time spent in the GOSales application is getting longer all the time. The pie chart in Figure 8-18 on page 285 showed over 70% of the client time was spent in the application.

Staying on the same screen, we return to the SQL Statements tab. You can change the Response Time chart view by selecting different layers from the drop down list, for example Figure 8-20 on page 287.

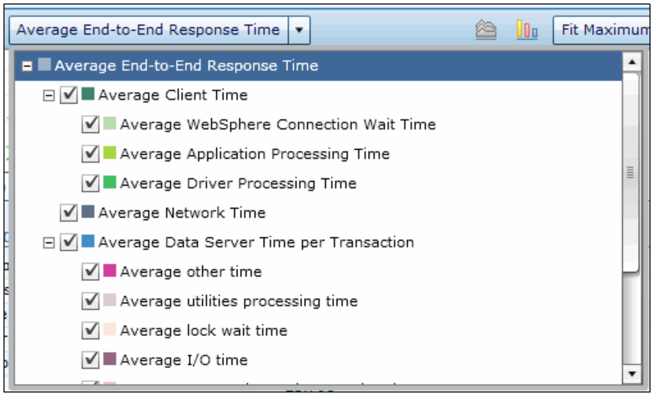


Figure 8-20 Extended Insight Response Time chart - list of layers

You can select or de-select different layers to highlight or hide them on the chart. In this case, we select the **Average Data Server Time per Transaction** layer on the response time chart to view some additional metrics, see Figure 8-21.

As before, we know the transaction throughput is decreasing but now we can see another chart showing Rows Returned is increasing. If the queries are returning more and more rows to the application, the application has to process more rows and thus take more time. DB2 seems to be handling the query well enough, but the application is not.

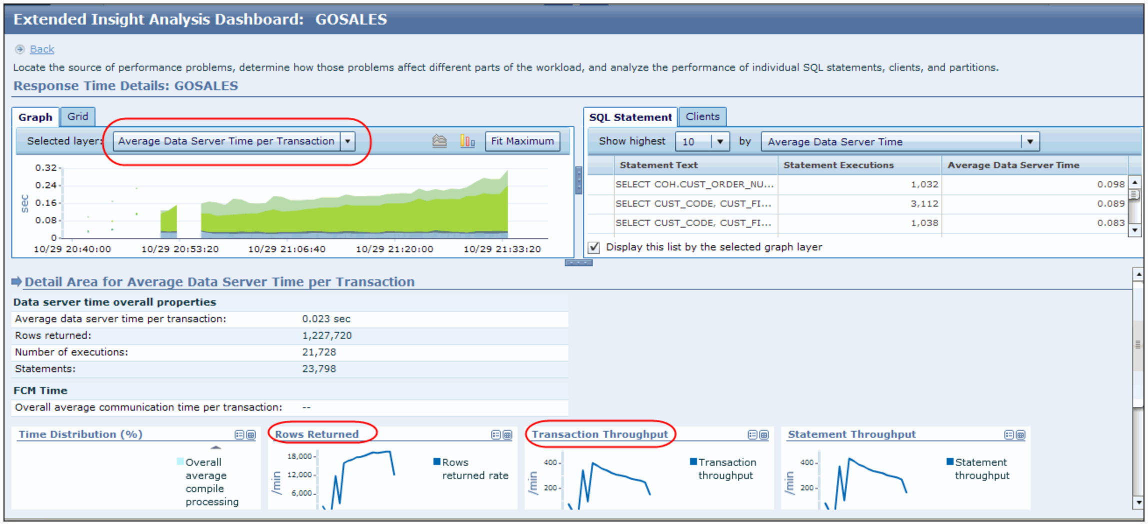


Figure 8-21 Extended Insight - Rows Returned

In fact, we examined our lab workload and discovered that it was doing repeated task of buying a product, then checking the order history. The order history transaction simply queries all the orders for that customer, returning them to the application for display on the page. If the same customer keeps buying more and more, of course the order history will grow. We recommend to the application owner that they should consider improving the order history transaction to include a time or volume range, so we are not retrieving so many rows from the database - for example, only retrieve orders from the last week, or only the last 10 orders.

We now have a lot more information to provide to the application support team - not only can we tell them the problem is not in DB2, we can tell them precisely which application server is showing a small connection pool size (4), which data source may be misconfigured (`jdbc/GoSalesApp`), and what version of WebSphere is running on the server (7.0.0.11). In addition, we have discovered a potential application design improvement in the order history processing.

What happened next?

For this example, in our lab we changed the connection pool size from 4 to 10, started some workload again, and we saw the connection pool wait time disappear. Notice the pie chart in Figure 8-22 on page 288, and the large connection wait time wedge is gone.

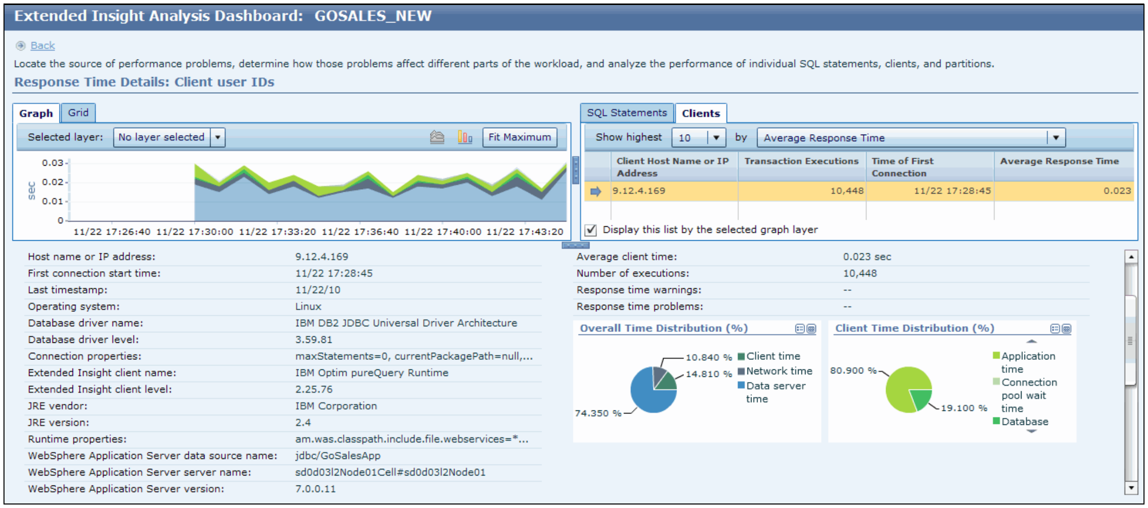


Figure 8-22 Extended Insight - after increasing connection pool size (1)

Figure 8-23 on page 289 is a picture of the lower section of same screen, where we see the new connection pool size is 10, and the line chart shows plenty of capacity. The average end-to-end response time is now about 23ms, compared to 144ms before.

The application has not yet been modified to improve order history processing.

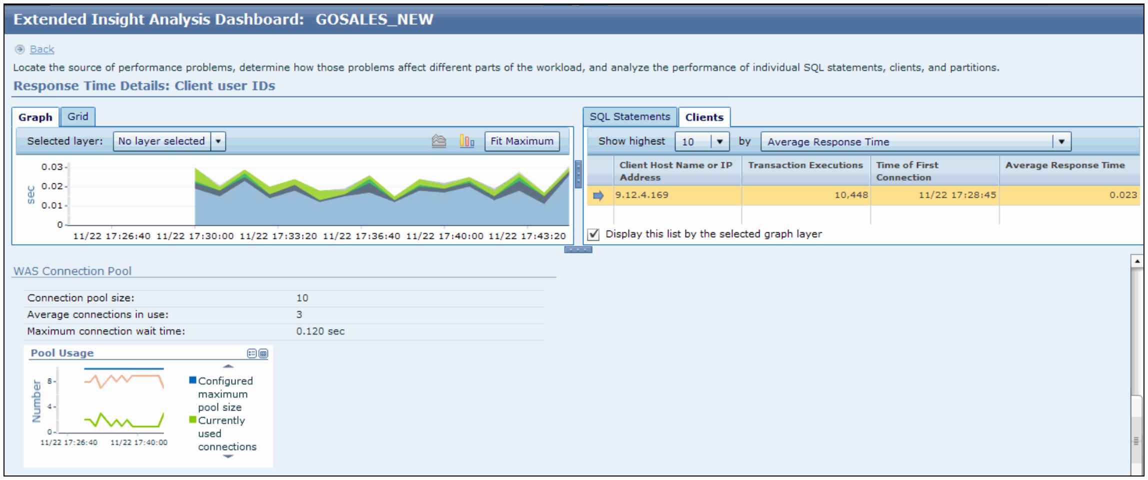


Figure 8-23 Extended Insight - after increasing connection pool size (2)

Working with multiple WebSphere clients

In our book lab, we only have one WebSphere server. If you run multiple application servers or clusters, you would see each of them on the Clients tab. They would be sorted and sortable by different metrics.

You must install and configure the Extended insight client software on each WebSphere Application Server you want to monitor.

Figure 8-24 on page 290 is an example of some metrics from an environment that has two WebSphere V7 application servers in a cluster, and a third WebSphere V6 server running standalone. All three servers run the same application against the same DB2 database. The application uses a single data source. This is just an example of what the client list would look like. This screenshot is not from our book lab environment.

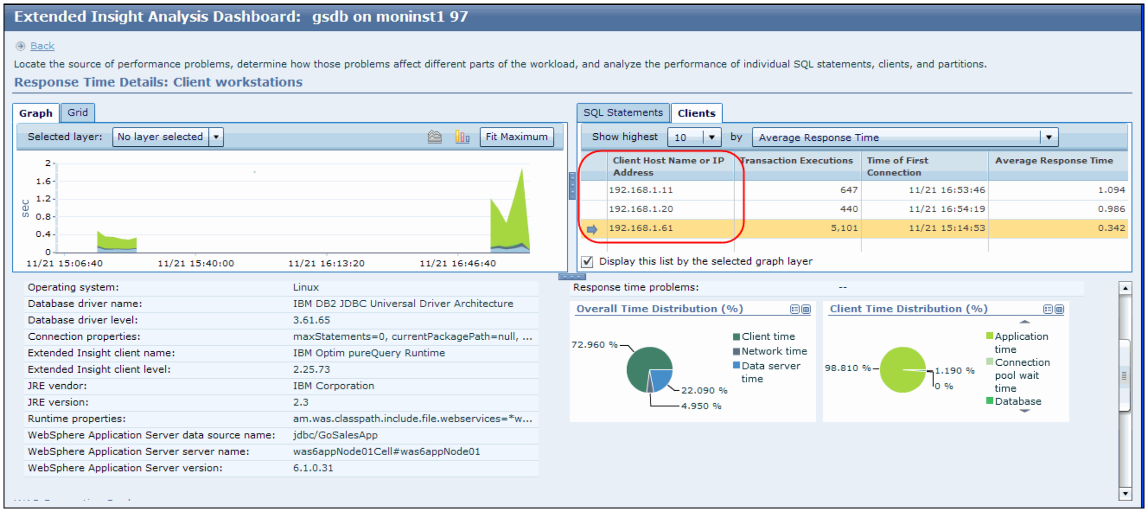
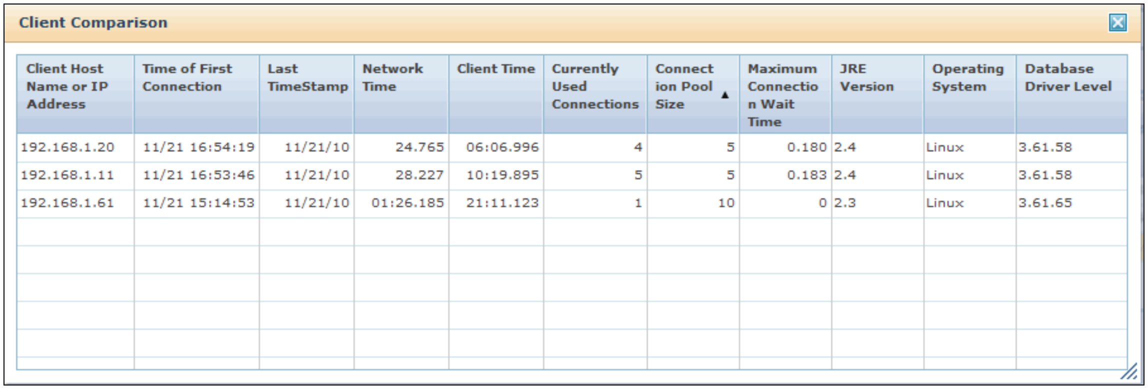


Figure 8-24 Extended Insight with multiple WebSphere clients

At the bottom of the client details section, there is a *Client Comparison* button. It is only enabled if there is more than one client. If you click it, you get a popup window with some information about all the clients, as in Figure 8-25 on page 290. The columns of the grid are sortable, which can be especially useful when you have a large number of clients. The view provides some key metrics to compare at a glance, across the different clients.


Client Comparison

Client Host Name or IP Address	Time of First Connection	Last TimeStamp	Network Time	Client Time	Currently Used Connections	Connection Pool Size	Maximum Connection Wait Time	JRE Version	Operating System	Database Driver Level
192.168.1.20	11/21 16:54:19	11/21/10	24.765	06:06.996	4	5	0.180	2.4	Linux	3.61.58
192.168.1.11	11/21 16:53:46	11/21/10	28.227	10:19.895	5	5	0.183	2.4	Linux	3.61.58
192.168.1.61	11/21 15:14:53	11/21/10	01:26.185	21:11.123	1	10	0	2.3	Linux	3.61.65

Figure 8-25 Extended Insight Client Comparison

8.2.3 Understanding workload clusters

In section 4.4.3, “Workload cluster groups and Workload clusters” on page 187, we introduced concepts about workload clusters. We explore that topic some more here, with specific focus on WebSphere applications.

Workload cluster groups, and their associated workload clusters, provide a way for you to look at your database and application transaction monitoring metrics from different perspectives. By using the JDBC connection properties on the connections you have configured for monitoring with Optim Performance Manager Extended Insight, you can increase the dimensions by which you can observe the performance data. We explore this now, in the context of the examples used in this section, which are specific to WebSphere. The general concepts apply to all connection types supported by Optim Performance Manager Extended insight, however, the specifics may vary slightly for the non-WebSphere JDBC and the CLI connections.

What are connection properties?

When you create a JDBC connection to DB2 database, the connection has certain attributes associated with it, such as the database name, the authentication ID, and the connection’s origin server. You can, optionally, specify more attributes by using client information properties, which DB2 and monitors such as Optim Performance Manager, can detect and use for other purposes. For example, DB2 Workload Manager can use connection attributes to define custom workload definitions. Optim Performance Manager uses these connection attributes as dimensions for workload clusters, which are used in the Extended Insight and Locking dashboards.

What are client information properties?

The IBM Data Server Driver for JDBC and SQLJ version 4.0 supports JDBC 4.0 client information properties, which you can use to provide extra information about the client to the server. This information can be used for accounting, workload management, or debugging. Extended client information is sent to the database server when the application performs an action that accesses the server, such as executing SQL.

You can read more about the client information properties in the DB2 Information Center. Search for “client information”, there are many references, this link is one example:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.apdv.java.doc/doc/t0052428.html

Why are connection properties interesting or useful?

When you are looking at performance data, it is important to be able to narrow down, or rule out, problem areas. Suppose you observe a downward trend in some application performance metric; if you can say with confidence the issue is related to a specific part of the application, or even a specific SQL statement, this is much more useful than just saying the application is getting worse, and only then getting out your performance analysis toolkit to begin investigation.

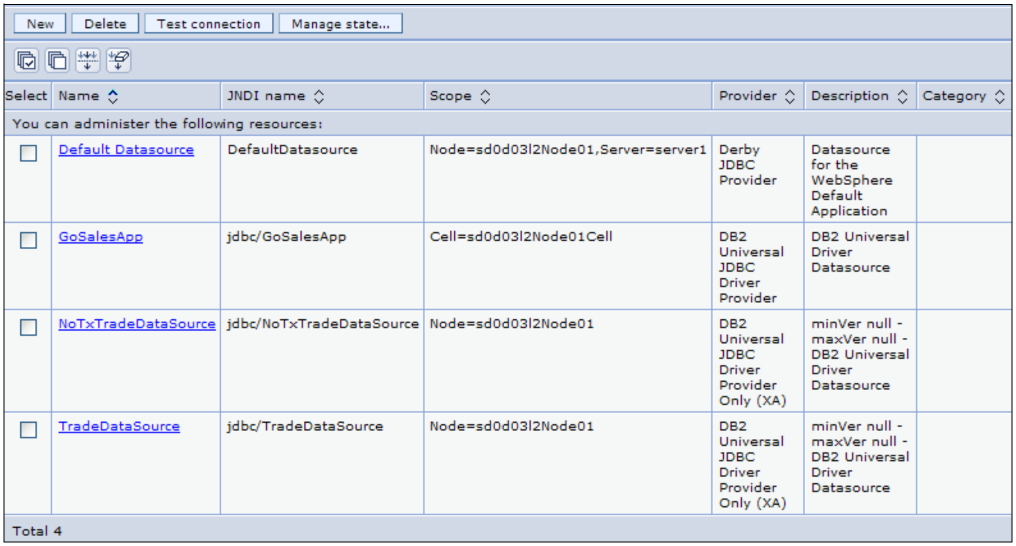
Optim Performance Manager Extended Insight can do the “slicing and dicing” of the performance metrics, across all the available connection property dimensions to give you the information you need to isolate the problem.

We’ll use two applications to highlight this point. In our book environment, we have two WebSphere test applications, DayTrader and GOSales.

Let us look at DayTrader first.

DayTrader scenario

The DayTrader application does not use client information properties, while GOSales makes extensive use of them. Another difference, as shown in Figure 8-26, which is a clip from the WebSphere administrative console view of JDBC resources, is that the DayTrader application that has two data sources, `NoTxTradeDataSource` and `TradeDataSource`, while GOSales has a single data source, `GoSalesApp`. Both DayTrader data sources point to the same database.


New Delete Test connection Manage state...

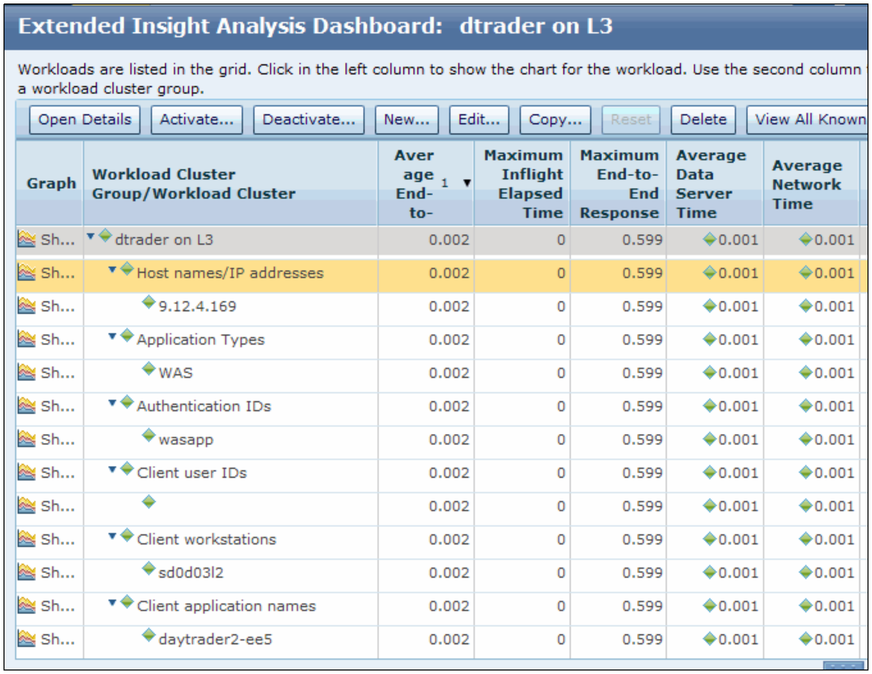
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☐	Default Datasource	DefaultDatasource	Node=sd0d03l2Node01,Server=server1	Derby JDBC Provider	Datasource for the WebSphere Default Application	
☐	GoSalesApp	jdbc/GoSalesApp	Cell=sd0d03l2Node01Cell	DB2 Universal JDBC Driver Provider	DB2 Universal Driver Datasource	
☐	NoTxTradeDataSource	jdbc/NoTxTradeDataSource	Node=sd0d03l2Node01	DB2 Universal JDBC Driver Provider Only (XA)	minVer null - maxVer null - DB2 Universal Driver Datasource	
☐	TradeDataSource	jdbc/TradeDataSource	Node=sd0d03l2Node01	DB2 Universal JDBC Driver Provider Only (XA)	minVer null - maxVer null - DB2 Universal Driver Datasource	

Total 4

Figure 8-26 WebSphere data sources for GOSales and DayTrader

From Extended Insight perspective, each data source is treated as a single client, therefore, they will appear as separate entities on the Clients dashboard.

Let us say, for argument, that both DayTrader data sources use the same authentication ID. Each will have identical connection properties, and because DayTrader does not use client information properties, all the metrics will end up grouped together in the Extended Insight dashboard. You would see something similar to Figure 8-27. While you can see different cluster groups, they all have only one cluster, and all the metrics are identical.



Graph	Workload Cluster Group/Workload Cluster	Average End-to-	Maximum Inflight Elapsed Time	Maximum End-to-End Response	Average Data Server Time	Average Network Time
Sh...	dtrader on L3	0.002	0	0.599	0.001	0.001
Sh...	Host names/IP addresses	0.002	0	0.599	0.001	0.001
Sh...	9.12.4.169	0.002	0	0.599	0.001	0.001
Sh...	Application Types	0.002	0	0.599	0.001	0.001
Sh...	WAS	0.002	0	0.599	0.001	0.001
Sh...	Authentication IDs	0.002	0	0.599	0.001	0.001
Sh...	wasapp	0.002	0	0.599	0.001	0.001
Sh...	Client user IDs	0.002	0	0.599	0.001	0.001
Sh...		0.002	0	0.599	0.001	0.001
Sh...	Client workstations	0.002	0	0.599	0.001	0.001
Sh...	sd0d03l2	0.002	0	0.599	0.001	0.001
Sh...	Client application names	0.002	0	0.599	0.001	0.001
Sh...	daytrader2-ee5	0.002	0	0.599	0.001	0.001

Figure 8-27 Extended Insight dashboard - DayTrader, single cluster

There is nothing inherently wrong with this grouping, but you would not be able to distinguish statement metrics *per data source*, because all the connection attributes are identical. Yes, you could see difference in client metrics, on the the Extended insight client details page because each data source is shown as separate client, but you could not narrow down to statements per data source.

Let us explore that a bit - we drill down on any cluster (remember, it does not matter which one, they are all the same) and we see some SQL statements are executed, as in Figure 8-28.

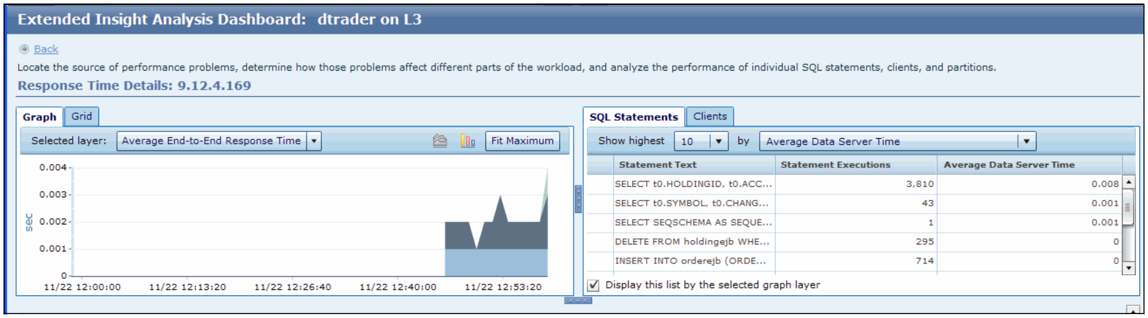


Figure 8-28 Extended Insight Response time details for DayTrader, single cluster

We have no way to know which statements are from which data source.

Open the **Clients** tab, as in Figure 8-29. Now we see the two clients, one has only four transactions, the other one nearly 50,000.

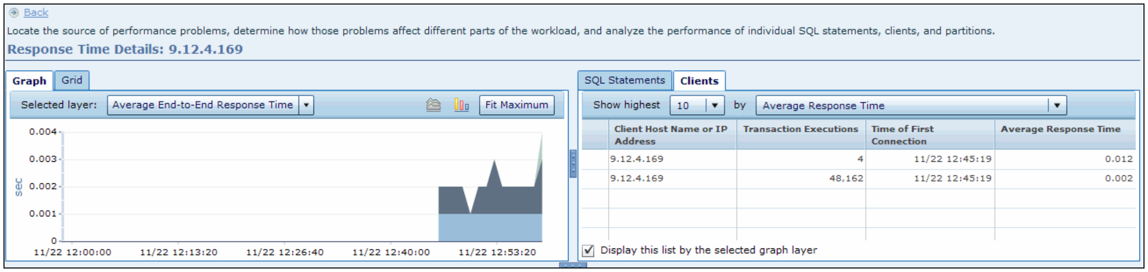


Figure 8-29 Extended Insight Clients tab for DayTrader, single cluster

Select the client with only four transactions, and view the client details on the lower part of the screen, as shown in Figure 8-30 on page 295. Now we can see the client metrics and know the data source with four transactions is the `NoTxTradeDataSource`. But we still do not know which statements it ran.

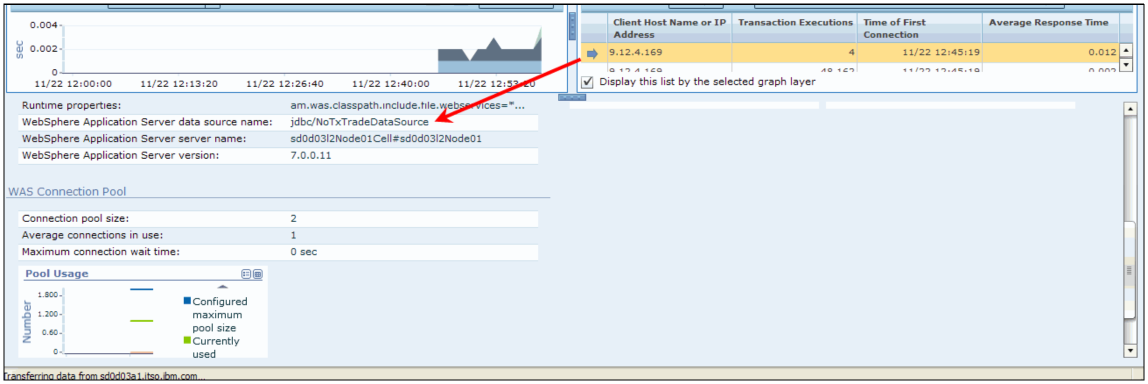


Figure 8-30 Extended Insight Client detail for NoTxTradeDataSource

How can you improve this state, if the application itself cannot change to use client information properties within its code?

One option is to use a different authentication ID for each data source, thus introducing a new dimension that Optim Performance Manager can group on. This is viable but does require work by your security team to add a new user ID, and there could be implications to the application if it depends on the authentication ID to qualify a table schema, for example.

Another option is to use the WebSphere capability to set default values for one or more of the four client information properties. This would require assistance from the WebSphere administrator, or who ever has privileges to change the application data source properties.

You may have noticed in Figure 8-27 on page 293 that the cluster value for the Client User IDs workload cluster group was blank. This is because neither WebSphere nor the DayTrader application set any value for that property. Suppose we keep the same authentication ID for both data sources, but we add a unique value for the `clientUser` property in WebSphere's data source definition. Here we set the `clientUser` property for `NoTxTradeDataSource` to `notxuser` in Figure 8-31. (This picture is from the WebSphere administration console.)

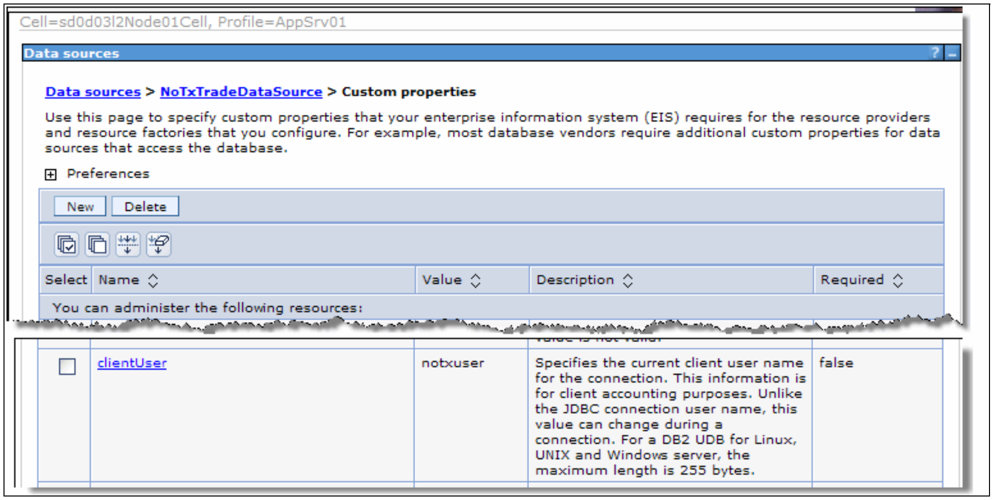


Figure 8-31 Set default clientUser value for NoTxTradeDataSource

For the `TradeDataSource` data source we set it to `itsoUser`, see Figure 8-32.

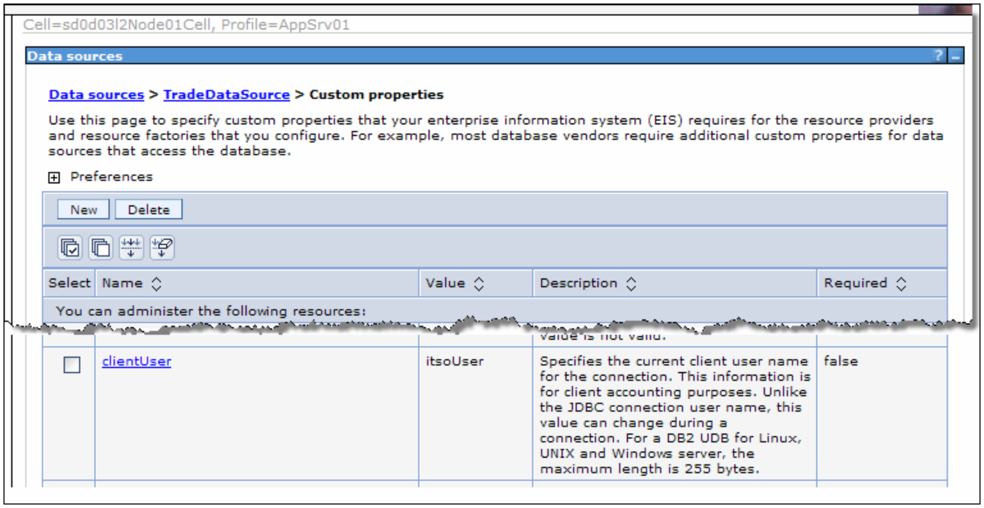
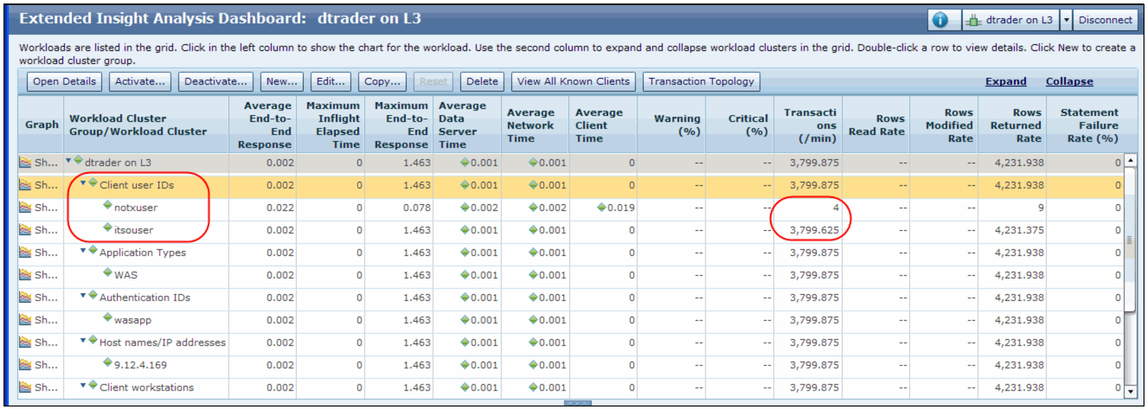


Figure 8-32 Set default clientUser value for TradeDataSource

> **Note:** You can access this WebSphere administrative console page in one of two ways:
>
> - **Resources → JDBC → Data sources → data_source**
> - **Resources → JDBC → JDBC providers → JDBC_provider → Data sources → data_source**

These data source changes are picked up at next WebSphere start. When a new connection is established, it will have the appropriate clientUser property value associated with it. Now our two data sources have uniqueness between them, manifested in the connection properties, which are monitored and collected by the Optim Performance Manager Extended Insight client.

We run some more workload through the DayTrader application, let us see how this changes the view of data on the Extended Insight dashboard. Observe in Figure 8-33 on page 297, now we see two workload clusters under Client User IDs workload cluster group - one for `itsouser` and one for `notxuser`. The properties now reflect the different data source from the application. We know this because we set the value in WebSphere, so when the application acquires a connection, it will always be set with the corresponding value on the client information property.



Graph	Workload Cluster Group/Workload Cluster	Average End-to-End Response	Maximum Inflight Elapsed Time	Maximum End-to-End Response	Average Data Server Time	Average Network Time	Average Client Time	Warning (%)	Critical (%)	Transactions (/min)	Rows Read Rate	Rows Modified Rate	Rows Returned Rate	Statement Failure Rate (%)
Sh...	dtrader on L3	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	Client user IDs	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	notxuser	0.022	0	0.078	0.002	0.002	0.019	--	--	4	--	--	9	0
Sh...	itsouser	0.002	0	1.463	0.001	0.001	0	--	--	3,799.625	--	--	4,231.375	0
Sh...	Application Types	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	WAS	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	Authentication IDs	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	wasapp	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	Host names/IP addresses	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	9.12.4.169	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0
Sh...	Client workstations	0.002	0	1.463	0.001	0.001	0	--	--	3,799.875	--	--	4,231.938	0

Figure 8-33 Extended Insight dashboard - DayTrader, two clusters for Client User IDs

How is this helpful to us? Let us drill down on the Client User IDs workload cluster group first, by double-clicking on the row. We see the same kind of statement information we saw before as in Figure 8-28 on page 294 where we cannot discern which statements came from which data source. The *Clients* tab still show both data sources as separate clients. The summary information is useful for the bigger picture, but now we can drill down on each individual workload cluster - the client user ID values in this case - to see more detail.

Back on the Extended Insight overview screen, double-click the `notxuser` workload cluster, to view its details, as seen in Figure 8-34 on page 298. Now we can see exactly which statements were executed under that data source. There are only three statements for this data source, which is used only for keeping track of some sequence counters for the application, and the statements are not executed very often. (Note also we switched the graph to a bar chart, to make the single data point easier to see.)

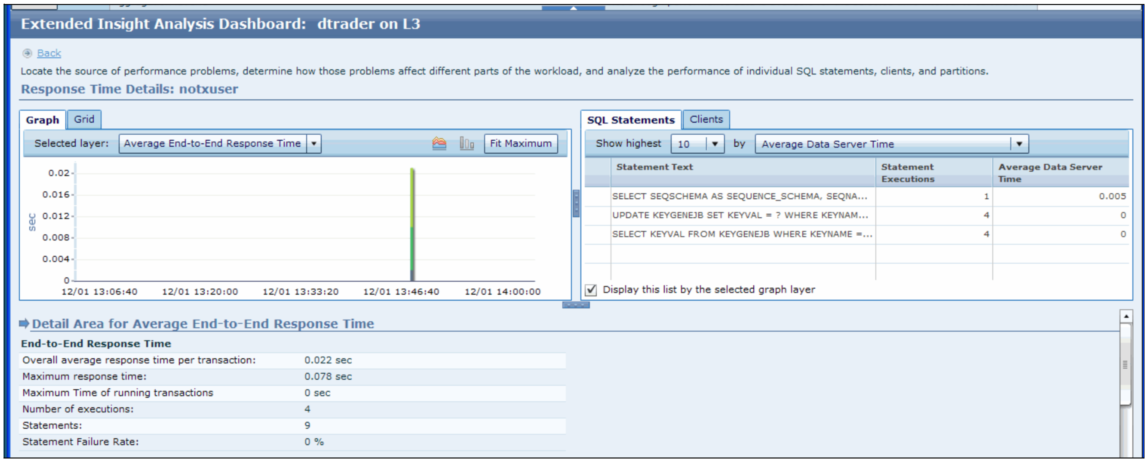


Figure 8-34 Extended Insight details for notxuser Workload Cluster

If we look at the *Clients* tab for this cluster, what do you think we will see? We can see only one client now, because we have filtered the view to look at only metrics for that single data source - shown in Figure 8-35.

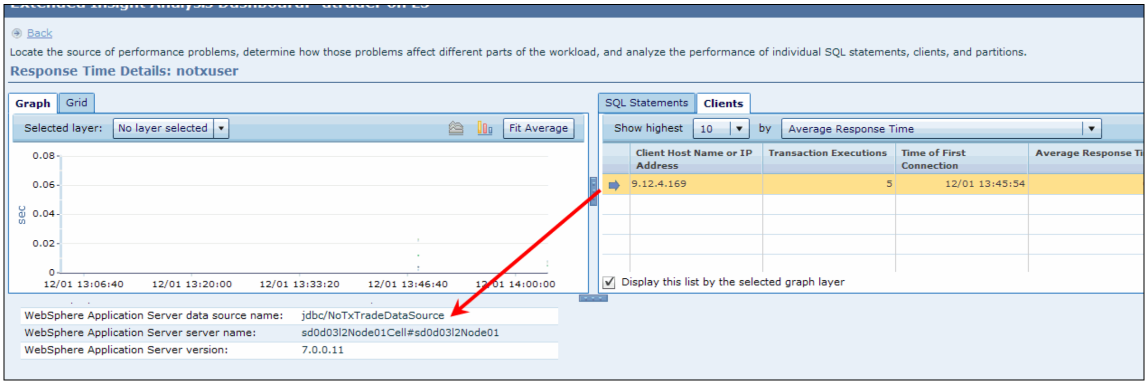


Figure 8-35 Extended Insight client detail for notxuser

Summary for DayTrader scenario

In this scenario, we have shown how you can set a default client information property in WebSphere data source, and have that value carried through the connection and all transactions that occur with that data source. Optim Performance Manager Extended Insight stores that information as another dimension by which you can view the metrics in the Extended Insight dashboards.

For DayTrader scenario, even though the application does not use client information properties, we were still able to control how the workload clusters were grouped. No application changes were required.

In the example just discussed, we only showed using one of the client information properties. You can use any or all of the four properties to add more dimensions to Extended Insight.

Great Outdoors Company scenario

As we mentioned in “Why are connection properties interesting or useful?” on page 292, we have two different sample applications in our environment. The Great Outdoors Company’s GOSales application makes extensive use of client information properties within the application. Let us take a look at that application next.

For the GOSales data sources defined in WebSphere, no default client information properties are set. Remember also the GOSales application uses only one data source.

When we run some workload for GOSales application, the Extended Insight dashboard shows a lot of workload clusters, because the application sets and changes the connection properties frequently. See for example in Figure 8-36 on page 300. Observe there are several workload clusters under each workload cluster group. We are using the default workload cluster groups - we have not yet created our own. If we look at workload cluster group `Client application names`, for example, we see three clusters:

- `customer log in`
- `product viewing`
- `order entry`

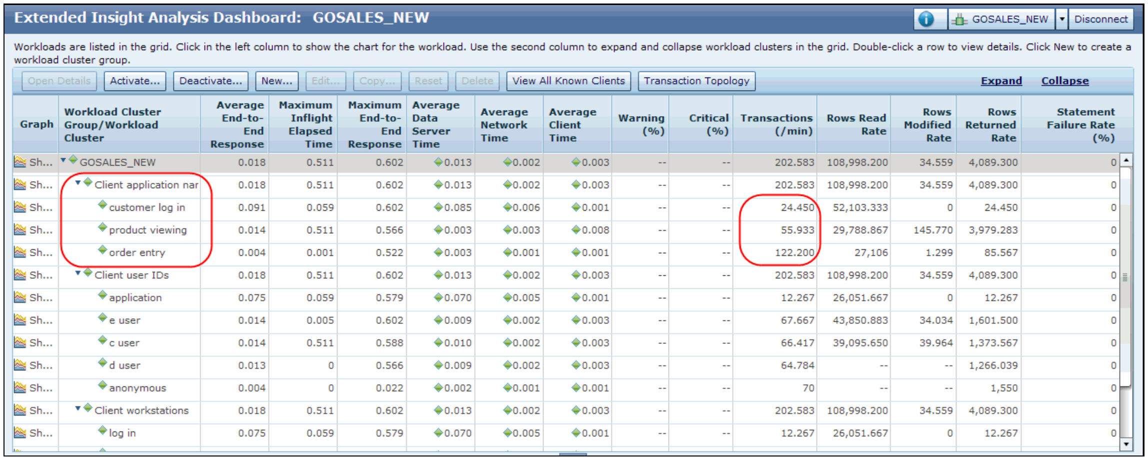


Figure 8-36 Extended Insight dashboard - GOSales with multiple workload clusters

For this grouping, we look at the Transactions per Minute column, to see which cluster has the most activity. The `order entry` cluster shows the highest transaction rate of the three clusters. If we were investigating some alerts, we might want to focus on the heavy-hitter cluster first - `order entry` in this case. Or, we might want to focus on just the cluster with longest end-to-end transaction time - in our example this is the `customer log in` cluster. That is the strong point of Extended Insight - you *can* view the metrics in many different ways.

As with the DayTrader scenario, when you drill down into any workload cluster, you see the metrics associated only with that cluster. So, if we want to see what statements were executed for the `order entry` cluster, we just double-click on it to view the details. In Figure 8-37, it looks like there are a lot of INSERT statements (no big surprise for an order entry transaction).

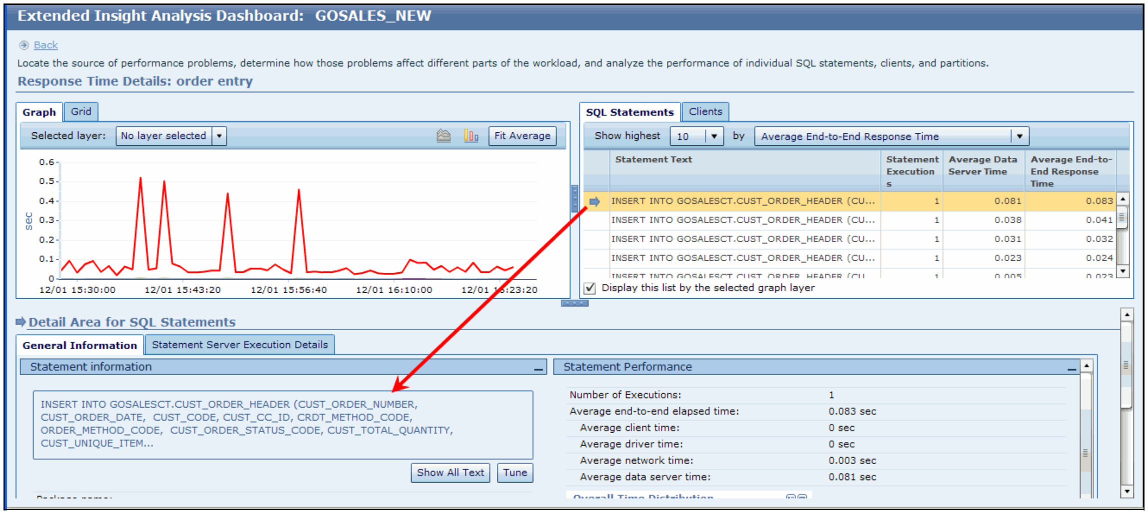


Figure 8-37 Extended Insight details for GOSales order entry workload cluster

To see the full text of the SQL statement, click **Show All Text**. In Figure 8-38 on page 301, we see the statement has a literal in the text, as well as some parameter markers. We might want to talk to developers about that too, knowing that another parameter marker might be needed. We could also launch the Optim Query Tuner from here, but we do not go into that with this scenario. You can see an example of the Optim Performance Manager-Optim Query Tuner integration in 8.1, “Application running slowly caused by index issue” on page 272.

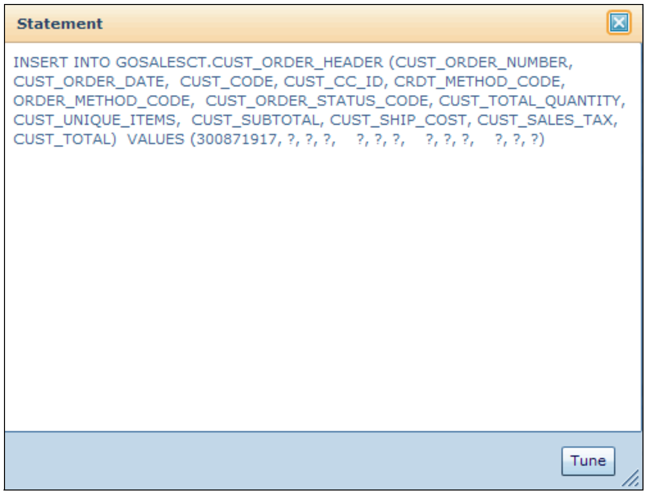


Figure 8-38 GOSales order entry - show all text

Statement Server Execution Details tab

While not directly related to this scenario, it is a good time to mention the additional metrics you can collect with Extended Insight. Up to now, we have been concerned with metrics collected on the Extended Insight client side - in this case WebSphere application server. If your DB2 data server runs at least version 9.7 Fix Pack 1, you can also collect metrics about statements and transactions at the data server side. This is configurable in the Extended Insight section of the Configure Monitoring dialogs, see Chapter 3, “Installing and configuring Optim Performance Manager” on page 59. We do not go into detail about it here but let us take a quick look.

in Figure 8-39 on page 302, we selected the **Statement Server Execution Details** tab, to see some more metrics. The data here is only available if you have configured its collection, otherwise, the fields will be empty.

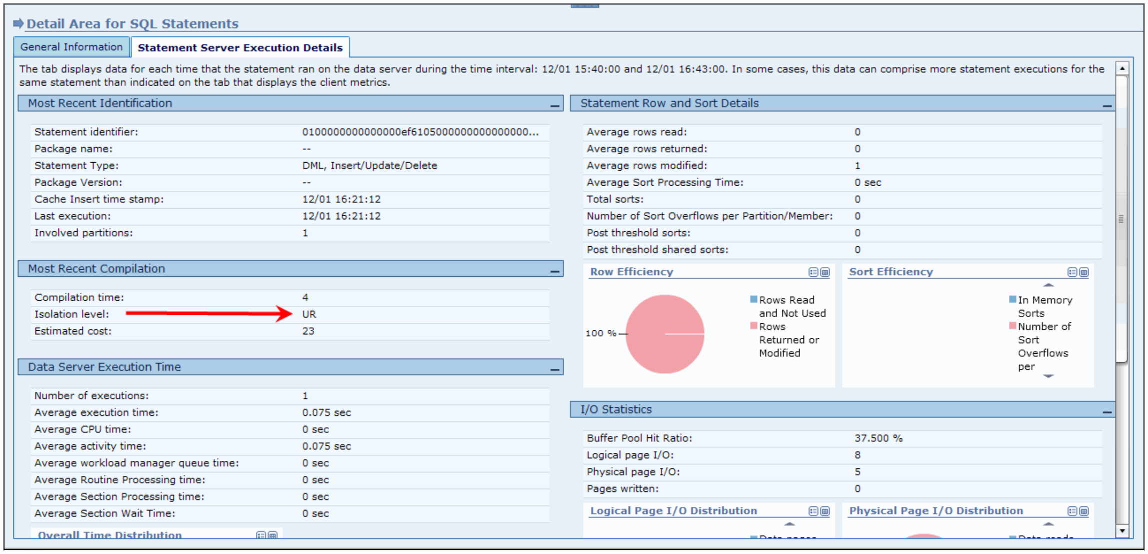


Figure 8-39 Extended Insight - GOSales Statement Server Execution Details

A single INSERT statement, in this case, may not show anything especially interesting, but you can still see the type of information that would be collected if you enable the server-side statement metrics. One metric many people find useful is the isolation level, in this case it uses UR (uncommitted read). The server-side statement metrics are collected using package cache monitor functions, and only if your monitored DB2 is version 9.7 Fix Pack 1 or higher.

Even more metrics can be collected if you also enable the server-side transaction metrics. We did not enable it for this example. The server-side transaction metrics are collected using the DB2 Unit of Work event monitor, and only if your monitored DB2 is version 9.7 Fix Pack 1 or higher.

To enable or disable the server-side collections, see the Configuration dialog for Extended Insight, as shown in Figure 8-40 on page 303.

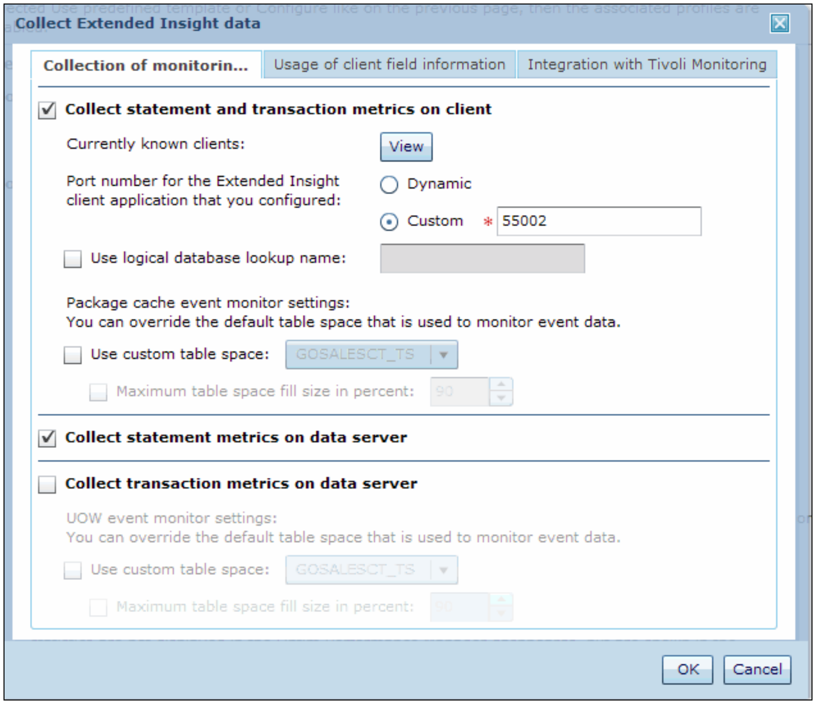


Figure 8-40 Extended Insight configuration

Customizing Workload Cluster Groups

In the GOSales scenario, we saw how the metrics for the three client application names were grouped as separate workload clusters, within the Workload Cluster Group for Client application names. What if we wanted to see metrics for `order entry` and `product viewing` grouped together, and only isolate the `customer log in` metrics as a separate cluster. Is this possible?

Yes, you can create your own workload cluster groups using a variety of grouping and filtering. Let us create a group with that example, and add another dimension as well, only include metrics from client users “`c user`” and “`e user`”.

In section 4.4.3, “Workload cluster groups and Workload clusters” on page 187, we showed how to create a new workload cluster group. In this section we show this again but with different focus on the clustering and filtering aspects, rather than the thresholds.

On the Extended Insight overview dashboard, click **New...** to open the Workload Cluster Group dialog. We name the new group `Test group`, see Figure 8-41 on page 304.

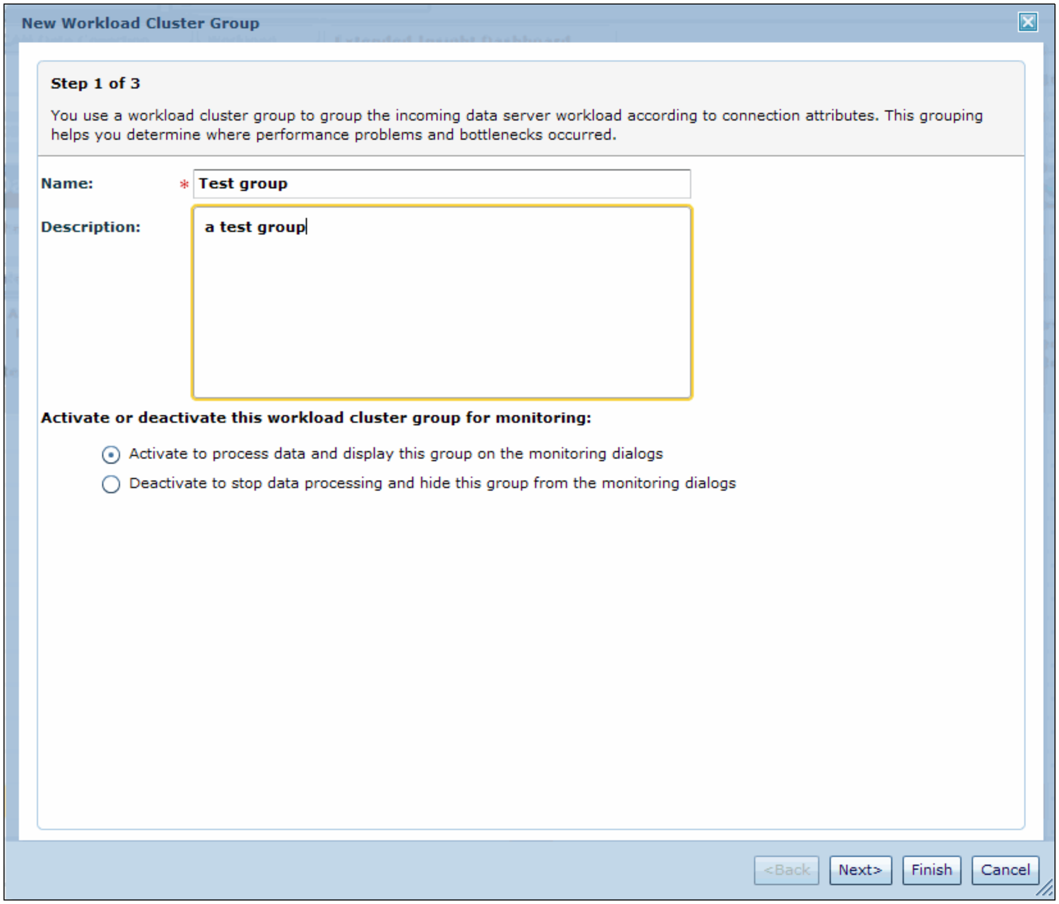


Figure 8-41 Create new Workload Cluster Group - step 1

We want to cluster on the `Client application name` attribute, so we select that checkbox. But we also want to exclude some values, so we also select the Filter checkbox. See Figure 8-42 on page 305.

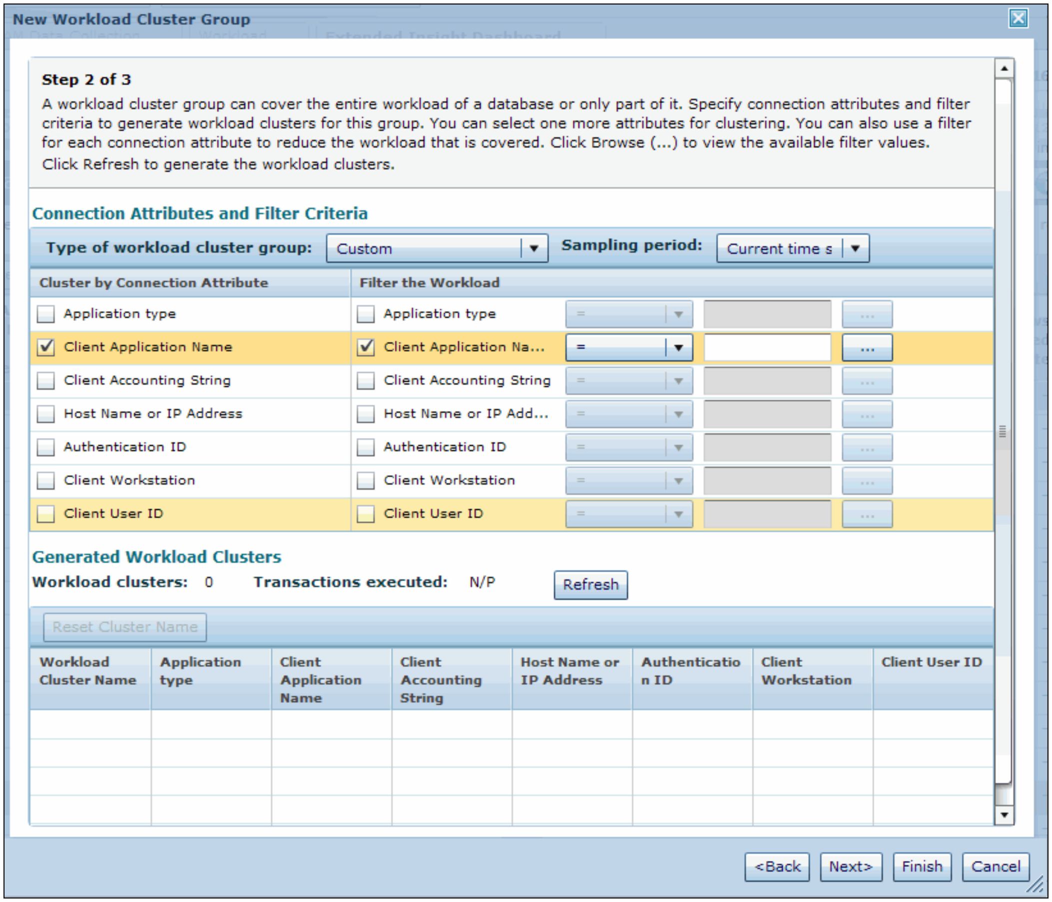


Figure 8-42 Create new Workload Cluster Group - step 2- before filtering

Click the button marked “**...**” to open the filter dialog. As shown in Figure 8-43 on page 306, un-check the `customer log in` attribute since we do not want to see it in our cluster group.

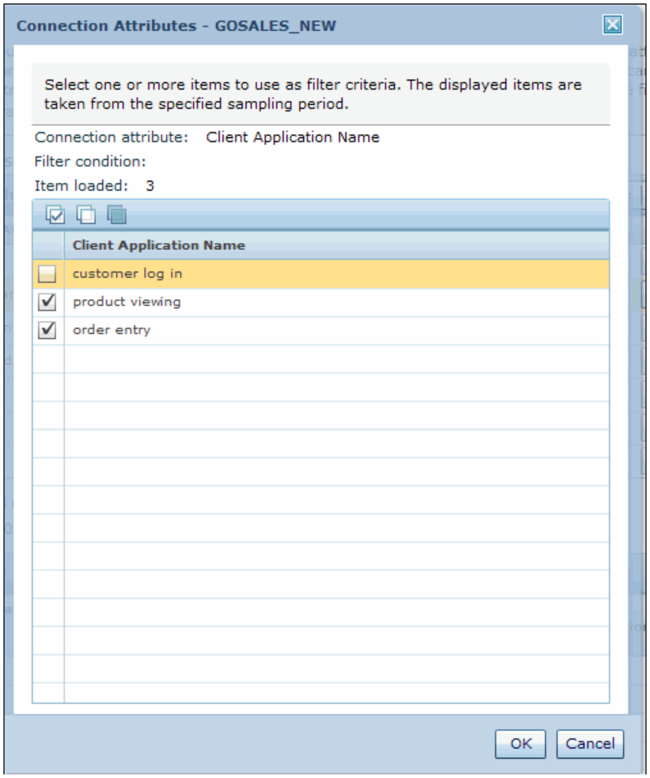


Figure 8-43 New Workload Cluster Group - filter selection

Click **OK** to accept the filter list. Now back on the main dialog, see in Figure 8-44, the filter is displayed, showing the client application name will only be shown for the application names in the list.

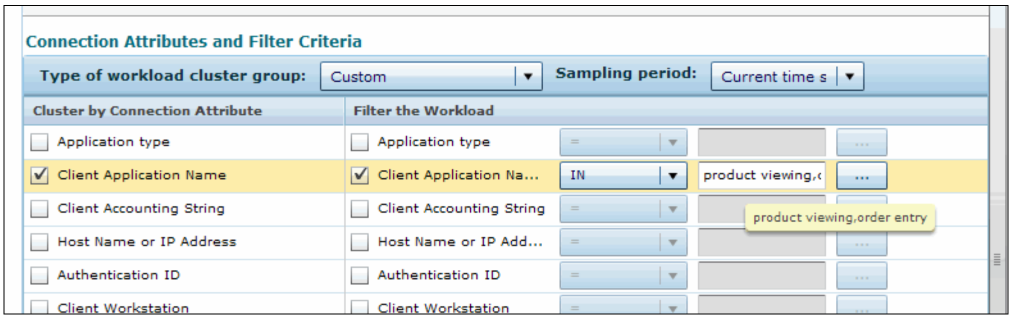


Figure 8-44 New Workload Cluster Group - cluster and filter on Client Application Name

We also want to create an additional filter for only metrics for two users. This is an important point about workload cluster groups - you can *cluster* and/or you can *filter*. For this case we only want to filter.

Select **Filter** for client User ID, but leave the "cluster by" checkbox unchecked. Click the filter button ("**...**") and select only the desired users (Figure 8-45), then click **OK**.

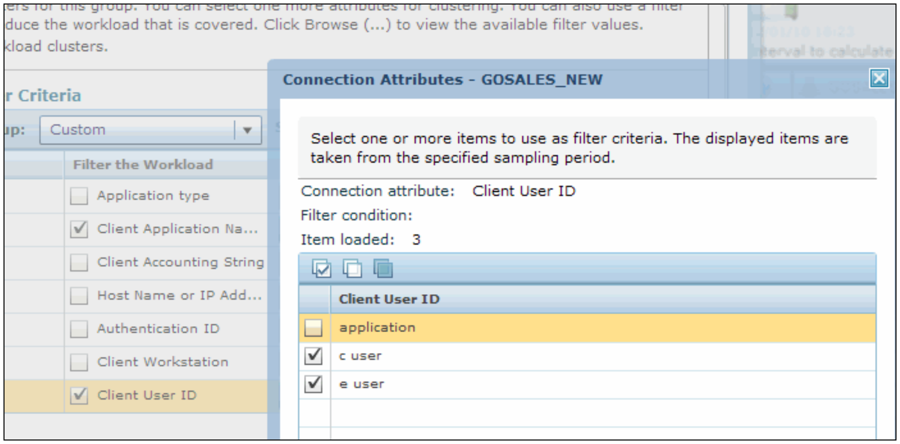


Figure 8-45 New Workload Cluster Group - filter on Client User ID

Now back on the Step2 page of the Workload Cluster Group dialog, click **Refresh** to view the list of clusters you would see for this set of cluster/filter options. In Figure 8-46 on page 308, now we see the results of the combination of clustering and filtering. We did not choose to *cluster* on the client user ID attribute, so we do not get a separate row for that dimension. The metrics shown for the new Workload Cluster Group, however, will include only transactions from the two client user IDs we selected as the filter. In this case, we have two clusters.

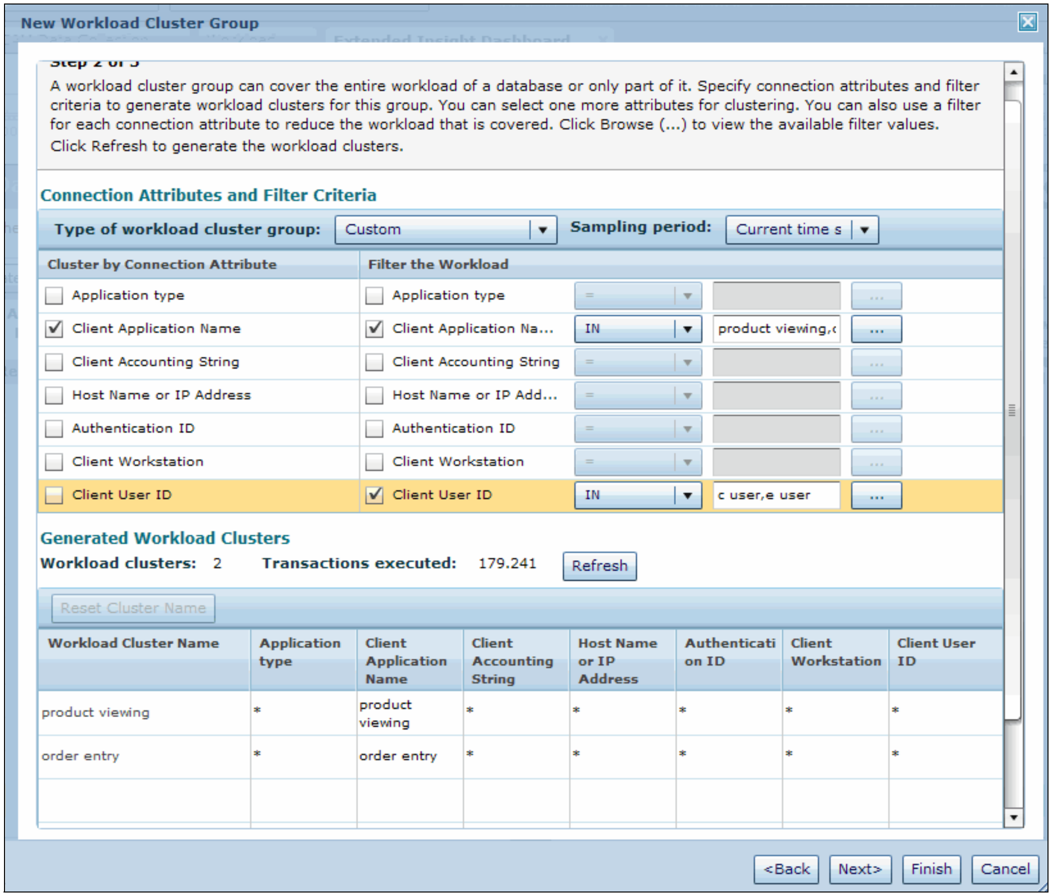


Figure 8-46 New Workload Cluster Group - clustered and filtered

We will not set any alert thresholds at this time, so click **Finish** to save the new workload cluster group.

If we had chosen also to cluster, as well as filter, on the client user ID attribute, we would end up with four clusters - one for each combination of attributes. An example of this is shown in Figure 8-47 on page 309.

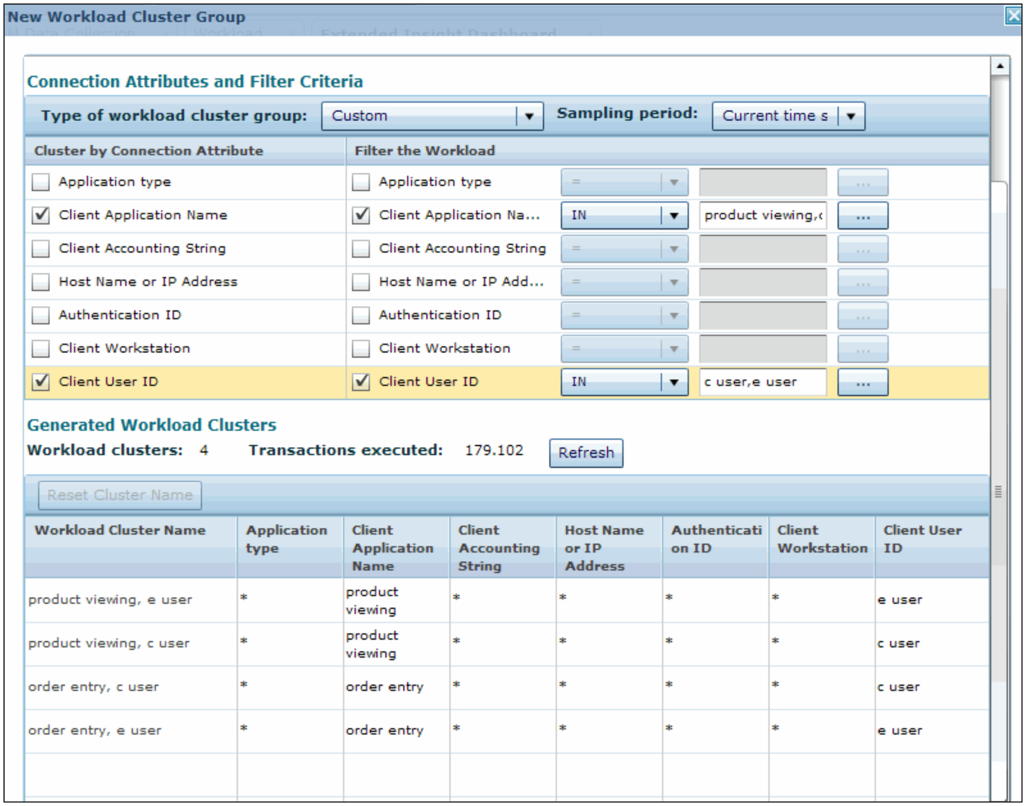


Figure 8-47 New Workload Cluster Group - clustered and filtered on both attributes

You might want to experiment with clustering and filtering with your own data, it is quite interesting and easy to see metrics in different combinations.

> **Important:** The workload cluster groups and clusters operate on data that is already collected. You could look at current or history data, decide you want to view some other dimensions, create a new cluster group and view it. You can create, modify, delete, or just disable different workload cluster groups at will. However, it is best to leave the default groups alone. There is no cost to creating new ones.

Now that we saved the new workload cluster group, how does it appear on the Extended Insight dashboard? See Figure 8-48 on page 310.

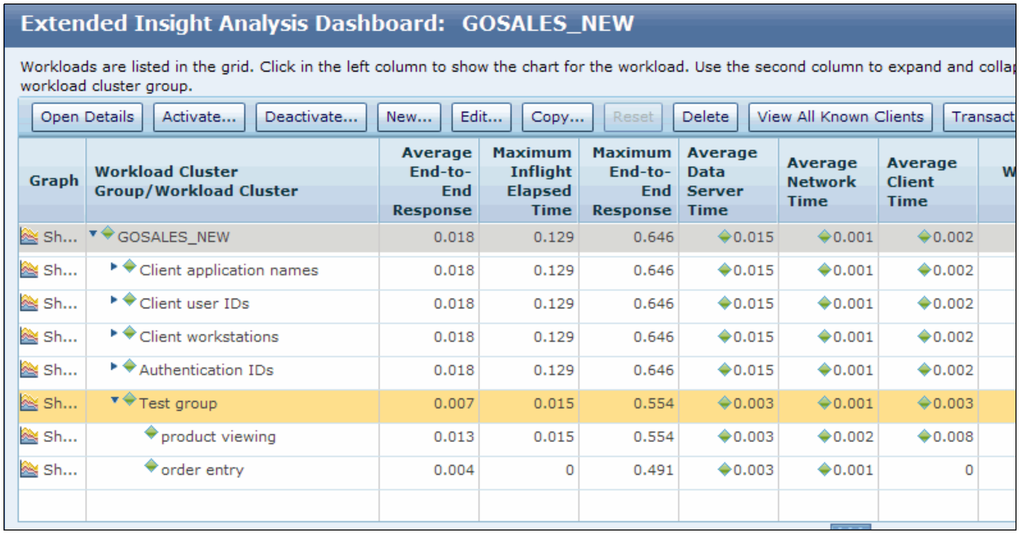


Figure 8-48 Extended Insight dashboard with new Test group

We notice that the `Test group` cluster group shows the two applications we wanted, but we do not have any indication that it is also filtered on user ID. Can we improve this? *Yes*. When you create or modify a workload cluster, you can also change the default name. Let us do that now. Select `Test group`, and click **Edit...** to open the Workload Cluster Group dialog for that group.

Click **Next>** to go to Step2 page. Click **Refresh** to cause the lower sample section to populate with data. If you have active workload running, you will see the current values displayed here. You can also switch to different time span by using the *Sampling period* drop down list. See Figure 8-49 on page 311. Click on the cell for the cluster name, and it will move into edit mode. Type any value you want for the cluster name. Here we added some text to indicate a filter. Each individual cluster can have its own custom name, or you can keep the default.

Click **Finish** to save these changes and return to the Extended Insight dashboard.

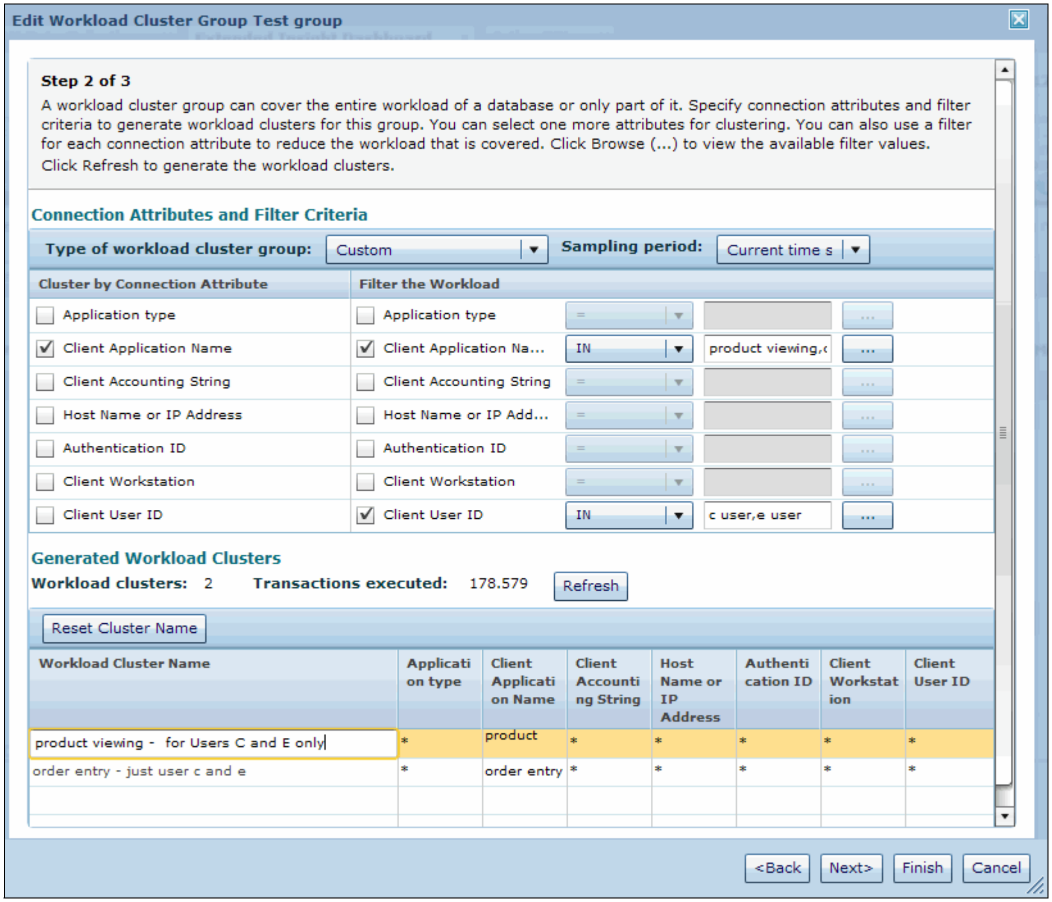


Figure 8-49 Modify Workload Cluster names

Now the Extended Insight overview dashboard looks like the example in Figure 8-50 on page 312, clearly identifying the two clusters as having a filter. Here we changed the individual cluster names, but we could have changed the Workload Cluster Group name as well.

.

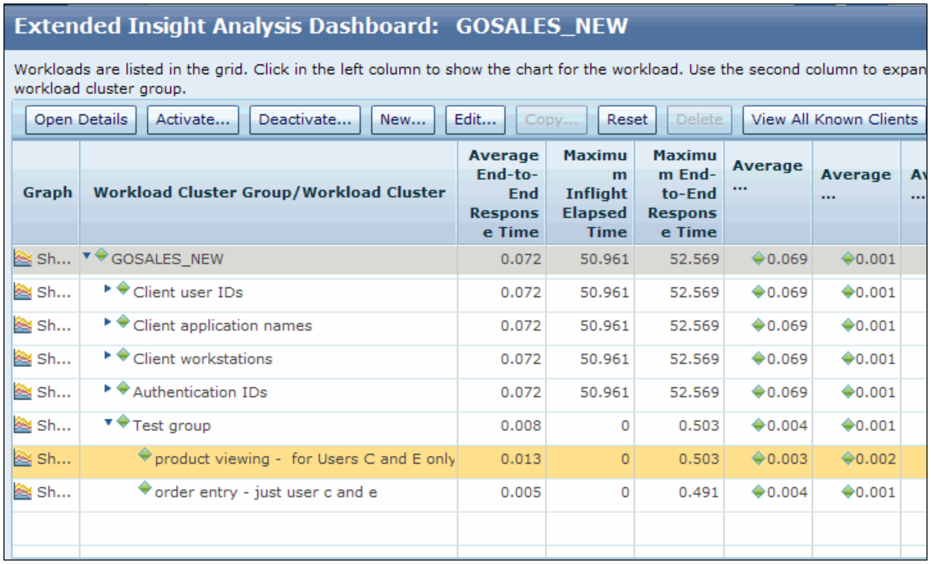


Figure 8-50 Extended Insight dashboard with new Test group custom cluster names

> **Recommendation:** There are no visual indicators that a filter is in effect, therefore we recommend using custom names if you use filtering without clustering, so that you don't forget about the filter later.

Now we have a custom group that we can monitor easily, and we have added some additional filtering to further narrow down its contents. If you have applications that use client information attributes, you will find the capabilities of Workload Cluster Groups quite powerful.

Summary for GOSales scenario

In the scenario for the GOSales application, we have seen how Optim Performance Manager Extended Insight uses the connection properties to allow different views of the application transaction metrics. If your application developers can add such properties to the `getConnnection` requests, you will have greater insight into application behavior.

We also learned about customizing Workload Cluster Groups to create the best views for your shop.

9

Troubleshooting failing transactions alert

If you receive an alert about a high percentage of failing transactions then you want to know which transactions failed and why. In this chapter we show you how to troubleshoot this alert in Optim Performance Manager.

To calculate the percentage of failing transactions Optim Performance Manager uses the number of rollbacks and commits occurred in the database. Each rollback is counted as a failed transaction, each commit is counted as a successful transaction.

Optim Performance Manager considers both internal rollbacks and explicit rollbacks.

An internal rollback occurs when any of the following cannot complete successfully:

- A reorganization.
- An import.
- A bind or precompile.
- An application ends as a result of a deadlock situation or a lock timeout situation.

- An application ends without executing an explicit commit or rollback statement (on Windows).

An explicit rollback is a `ROLLBACK` statement executed by an application. A well-designed application issues a `ROLLBACK` statement after an SQL statement execution failed.

In Optim Performance Manager you can perform the following tasks to troubleshoot failing transaction alerts:

- Identify failed statements.
- Check for deadlocks or lock timeouts and analyze them.
- Determine connections with a high number of failed statements or rollbacks.
- Check utility executions.

If you use Extended Insight, you can easily drill down to the failed statements that cause a transaction to fail, but even if you do not use Extended Insight, Optim Performance provides features that let you identify failed statements.

In the following sections we show you the following tasks in more detail:

- Identify failed statements using the Extended Insight dashboard.
- Determine connections with a high number of failed statements or rollbacks using Performance Expert Client. In addition, we use SQL and activity tracing capabilities provided by Performance Expert Client to find failed statements. You can use this combination of tasks if you do not have Extended insight.

9.1 Identify failed statements using Extended Insight dashboard

Let us start with failing transaction alerts that you receive from Optim Performance Manager for one of your monitored databases. In the Health Summary dashboard, list the alerts by clicking the red alert icon in the Workload category. An overlay opens which show all alerts belonging to the Workload category that occurred in the displayed monitoring period. In this example we choose a monitoring period of three hours. A number of failing transaction alerts occurred as shown in Figure 9-1 on page 315. The selected alert shows a failing transaction percentage of 9%.

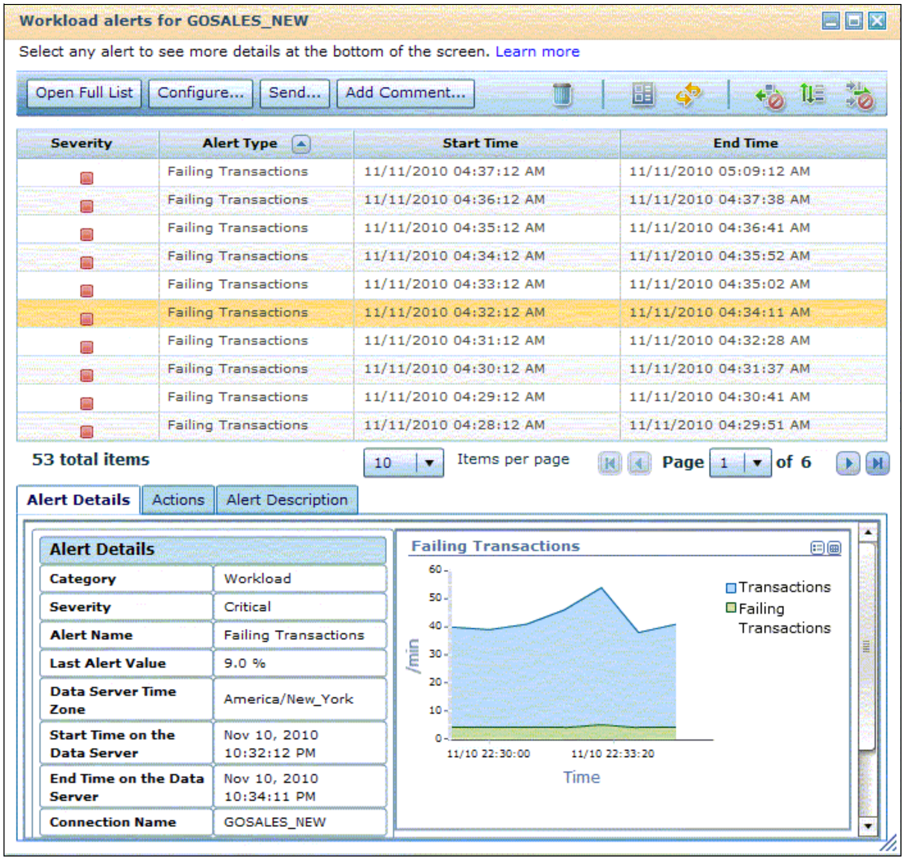


Figure 9-1 Failing transaction alerts on Health Summary

Open the Overview dashboard to do a quick check about whether, during the monitoring period, deadlocks or lock timeouts occurred or whether utilities executed that could have failed. In our example the Overview dashboard does not show any deadlocks, lock timeouts, or utility executions. Therefore, most likely the failing transaction are caused by failed statement executions.

Before we continue to identify the failed statements, let us point out one interesting thing we notice on the Overview dashboard shown in Figure 9-2 on page 316. The number of failing transactions is only 0.826%, however, it shows a red alert icon although the critical threshold is set to 5%.

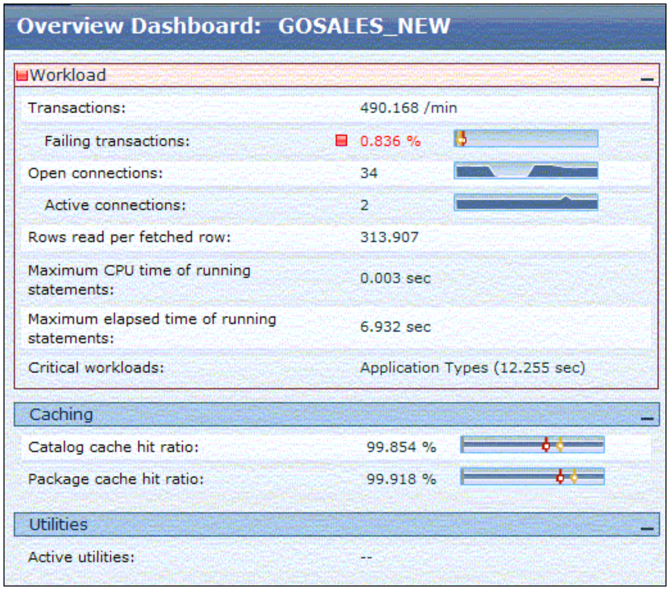


Figure 9-2 Failing transactions percentage on Overview dashboard

The explanation of this low failing transaction rate marked with a critical alert icon is as follows. Optim Performance Manager displays the average percentage of failing transactions during the monitoring period. Our monitoring period for this example is three hours. During this period, there are times where the percentage of failing transactions is higher than the critical threshold and there are times where the percentage is lower. This results in an average value of only 0.826%, but the alert is marked as red since there were critical alerts occurred during the monitoring period. Opening the failing transactions graph confirms that as shown in Figure 9-3 on page 317.

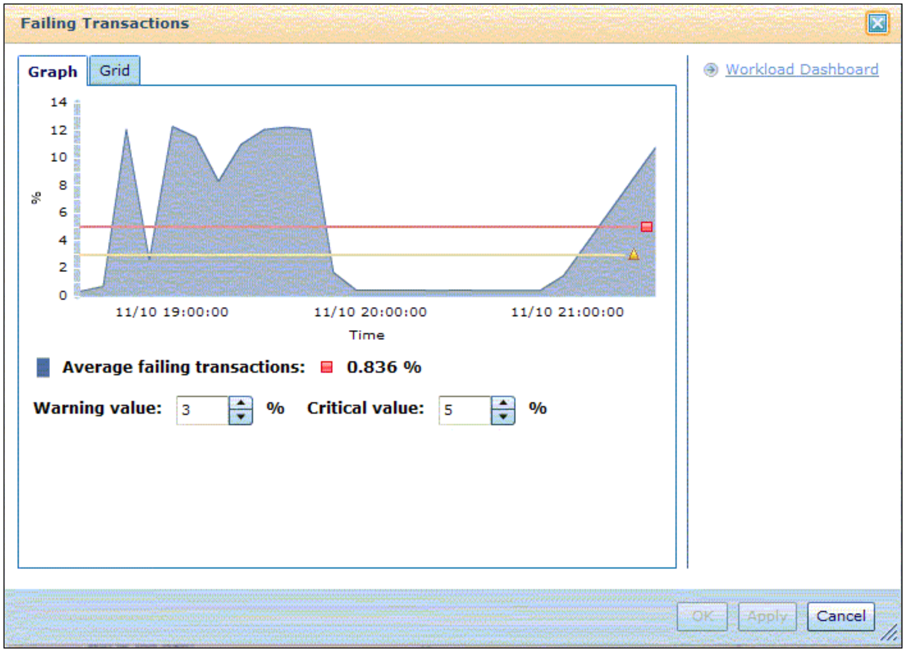


Figure 9-3 Failing transactions graph on Overview dashboard

If you click on the red icon next to the failing transactions percentage value on the Overview dashboard in Figure 9-2 on page 316, an overlay opens showing the last alert that occurred in the monitoring period. See Figure 9-4.

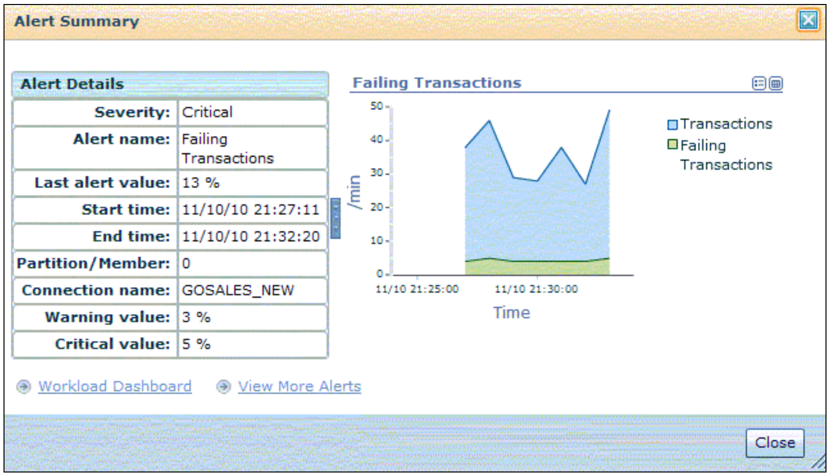


Figure 9-4 Alert details opened from Overview dashboard

Now, open the Extended Insight dashboard to identify the failed statements for the same monitoring period, a part of the dashboard is shown in Figure 9-5. Extended Insight dashboard displays the failed statement rate in percentage (%) for all workload clusters and workload cluster groups of applications that use Extended Insight client. For applications that do not use Extended Insight client, no statement failure rate can be calculated. Therefore, the average statement failure rate can be lower than the failing transactions rate.

Workload Cluster Group/Workload Cluster	Average End-to-End Response	Statement Failure Rate 1 ▼ (%)
GOSALES_NEW	0.434	0.043
Application Types	0.434	0.043
Client user IDs	0.434	0.043
anonymous	0.675	5
e user	0.478	0
c user	0.472	0
a user	0.433	0
h user	0.433	0
yong hua zeng	0.425	0
g user	0.423	0
f user	0.421	0

Figure 9-5 Part of Extended Insight dashboard

We see that user `anonymous` generates a statement failure rate of 5%. Drill down to the executed statements of this user by double clicking. We use the drop down boxes to list the top statements by *Failed Statement Executions (%)* as shown in Figure 9-6 on page 319. The statement that failed with 100% is a `DELETE` statement. It was executed 40 times in the monitoring period. The negative SQL code of the first execution is -4228. We assume that all 40 executions failed with the same SQL code. You can verify that by shortening the monitoring period on the Extended Insight dashboard.

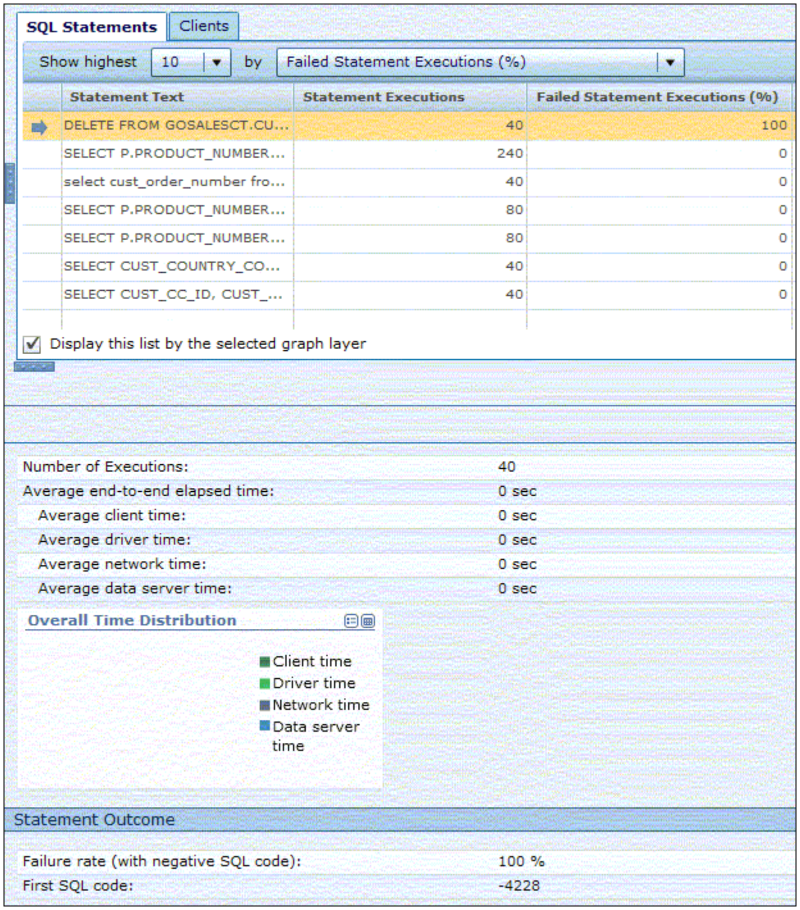


Figure 9-6 Failed statements

9.2 Identify failed statements using Performance Expert Client

Assume that you receive failing transactions alerts similar to what we described in the previous 9.1, "Identify failed statements using Extended Insight dashboard" on page 315, however, you do not have Extended Insight set up for this database. Using the Overview dashboard you have excluded deadlocks, lock timeouts, or utility executions as the primary reason for the failing transactions. What now?

First, look at the connections to determine the number of failed statements, rollbacks, deadlocks, and lock timeout. In the Optim Performance Manager web console, two features are available to look at connections:

- Current application connections dashboard to monitor connections connected now
- Connection report to monitor connections connected at a specified timestamp

Both features list the connections including execution metrics in tabular format. At the time of book writing, they do not provide the ability to customize the displayed execution metrics in the table. Therefore, we can not use these features to list the connections with the most failed statements, rollbacks, deadlocks, or lock timeouts in a single view.

Performance Expert Client provides us this ability. We open the *Application Summary* window and customize the displayed columns to obtain the top connections for the number of failed statements and rollbacks. See the top connections in Figure 9-7. From the list we see that the first connection executes a lot of failed statements whereas the next four connections seem to be involved in repeated deadlock situations.

Database Name	Application Name	Application Status	Rollbacks	Failed Operations	Deadlocks Detected	Application ID
SAMPLE	db2jcc_application	UOW waiting	2,406	4,812	0	127.0.0.1.2806.101111163133
SAMPLE	db2jcc_application	lock wait	252	207	207	127.0.0.1.2913.101111163355
SAMPLE	db2jcc_application	UOW waiting	252	0	0	127.0.0.1.2911.101111163353
SAMPLE	db2jcc_application	lock wait	250	207	207	127.0.0.1.2909.101111163350
SAMPLE	db2jcc_application	UOW waiting	250	0	0	127.0.0.1.2907.101111163347
SAMPLE	db2wlmd	connect comple...	0	0	0	*LOCAL.DB2.101111155149
SAMPLE	db2jcc_application	UOW waiting	0	1	0	127.0.0.1.2591.101111162602

Figure 9-7 Application Summary in Performance Expert Client

If the first connection is still connected and refreshing of the Application Summary display shows that the number of failed operations and rollbacks increase, then you can use tracing features in Performance Expert client to trace the entire SQL activity of this connection. If Application Summary displays historical data and the connection is already closed, then you are done at this point. There is no way to get the failed statements this connection tried to execute.

For the connections involved in deadlock situations, it is possible for historical data to detect the statements involved in the deadlocks by looking at deadlock event details. You can do this from Performance Expert client or from the Optim Performance Manager web console.

Let us go back to the SQL and activity tracing features that Performance Expert client provide. You can start activity or SQL traces for the complete database workload or filtered. Prerequisite to use them is that the *Performance Warehouse* monitoring profile is enabled. When you start a trace for a specified time frame, the statement event monitor (For SQL Activity Traces) or activity event monitor (For WLM activity traces) is created on the monitored database. It is dropped when the specified time frame is over. Perform a trace only for a few minutes since these event monitors create higher overhead on the monitored database than others. The collected trace data is displayed in a browser window that opens after the tracing time frame is over.

For example, you can start an SQL trace for the first connection in Figure 9-7 on page 320 by right clicking and selecting the option to create a SQL activity summary report. Specify the time frame to run, for example one minute, and wait for the report to come up. This report showed us that all statements of this connection failed with SQL code -206.

10

Integration with Tivoli monitoring components

In this chapter, we explore the integration of Optim Performance Manager with several Tivoli monitoring products. We describe installation, configuration, usage, and troubleshooting for Optim Performance Manager specifics only. Discussion of implementing Tivoli monitoring in your enterprise is beyond the scope of this book.

10.1 How does Optim Performance Manager integrate with Tivoli?

Optim Performance Manager Extended Edition integrates the deep database performance insight of Optim with the broad enterprise-wide insights provided by IBM Tivoli monitoring products. This powerful combination extends transaction response-time monitoring from the database to the complete end-to-end transaction path.

Database application environments can be complex, often including several middleware components through which transactions can flow, including Web servers, application servers, message servers, transaction servers, and database servers.

The IBM Tivoli Composite Application Manager (ITCAM) for Transactions product can keep a watch over the entire end-to-end transaction path that touches many of these components. When ITCAM for Transactions detects a transaction execution problem, it can isolate the problem to individual components in the end-to-end transaction path. It can then provide a launch point for deep-dive investigation into the components. For any transaction problems in the DB2 database component, ITCAM for Transactions can launch the Extended Insight dashboard in Optim Performance Manager Extended Edition in the context of the problematic database transactions. This capability enables you to use the deep database insights provided by Optim to further isolate the problem and drive it swiftly to resolution. Furthermore, Tivoli monitoring provides deeper, more extensive operating system, network, and storage information that you can access from within the system dashboard of Optim Performance Manager.

In Figure 10-1 on page 325, we show the components involved in the integration between Optim Performance Manager and ITCAM. Notice that WebSphere Application Server is part of the integration - this is because we depend on the ITCAM WebSphere monitoring agent for the integration. The integration of performance metrics between Optim Performance Manager and ITCAM is only possible for WebSphere JDBC applications.

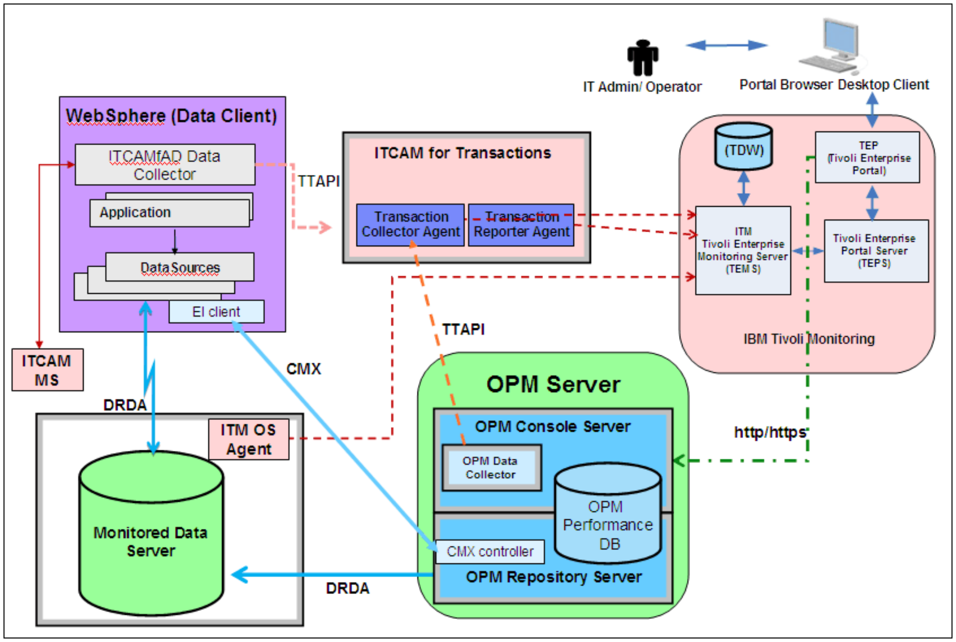


Figure 10-1 Optim Performance Manager-ITCAM integration monitoring architecture

Packaging and licensing

The capability of integrating with ITCAM is part of the Optim Performance Manager Extended Insight feature. However, purchasing Optim Performance Manager Extended Insight does not entitle you to any Tivoli software. The pre-requisite Tivoli products must be purchased and implemented separately.

Installation

When you activate the Extended Insight on your Optim Performance Manager server, you have also enabled the Optim Performance Manager server for the ITCAM integration. There are no additional installation steps on the Optim Performance Manager server side.

On the Tivoli side, you must install the Optim Performance Manager Plug-in for TEP onto the machine where your Tivoli Enterprise Portal Server (TEPS) runs. The Optim Performance Manager workspace and links are added to the TEPS to extend the ITCAM Transaction Collector views, allowing for the launch of Optim Performance Manager inside the Tivoli Enterprise Portal (TEP).

10.2 Implementing Optim Performance Manager integration with Tivoli ITCAM

The integration between Optim Performance Manager and ITCAM is dependent on multiple software components being in working order. You should proceed step by step when implementing the components - it will make your life much easier than trying to set up everything at once and then try to find the mistake if things are not working properly. We recommend the sequence in 10.2.1, "Pre-requisites to Optim Performance Manager-ITCAM integration" on page 327 for implementing the integration between Optim Performance Manager and ITCAM.

> **Reminder:** Optim Performance Manager-ITCAM integration can only be used for applications running in WebSphere and using JDBC transactions with the IBM JDBC driver.

Summary of the integration steps

The Optim Performance Manager-ITCAM integration procedure includes seven steps of two major tasks:

- Verify the pre-requisites products:
  a. Base Optim Performance Manager is working.
  b. Optim Performance Manager Extended Insight is working.
  c. Tivoli Monitoring (ITM) is working.
  d. Tivoli ITCAM components are working.

  We provide the details about how to verify if these products are working in 10.2.1, "Pre-requisites to Optim Performance Manager-ITCAM integration".
- Enable the integration:

  The enabling details are provided in the following sections:

  e. Enable the Optim Performance Manager Data Collector on Optim Performance Manager: See "Enable in Optim Performance Manager" on page 332.
  f. Enable the Optim Performance Manager on ITCAM WebSphere agent: See "Enable at ITCAM WebSphere agent side" on page 340.
  g. Install the Optim Performance Manager TEP plug-in on the TEPS: See "Install the Optim Performance Manager TEP workspace" on page 343.

Roles required to perform the setup tasks

You should understand the types of activities required to fully deploy the Optim Performance Manager-ITCAM solution. Consult with your enterprise support staff before deploying the solutions, many of the tasks also require root level access. Listed below are some of the roles that could be required for a full implementation, we discuss more in later sections:

- Optim Performance Manager administrator: To configure ITCAM collection in Optim Performance Manager.
- WebSphere administrator: In case that you want or have to adjust any settings for Optim Performance Manager Extended Insight.
- ITCAM for Transactions administrator: To provide ITCAM host and port information to Optim Performance Manager administrator.
- ITCAM for WebSphere administrator: To modify an ITCAM properties file to enable Optim Performance Manager integration.
- TEPS administrator: To install the Optim Performance Manager plug-in.

10.2.1 Pre-requisites to Optim Performance Manager-ITCAM integration

Before you try to integrate the Optim Performance Manager and ITCAM, you should have a working environment for both Optim Performance Manager and ITCAM. The numbered steps here correspond to the "Summary of the integration steps" on page 326.

1. Install and configure the base Optim Performance Manager components.

   Why: Establish the base product is functioning properly in your environment.

   This step is already described earlier in this book.

   a. See 3.2, "Installing and running Optim Performance Manager" on page 65 for information about installation.

   b. See 3.3, "Configuring Optim Performance Manager" on page 91 for information about configuration.

   c. *Verify*: You can verify that some data is being collected by Optim Performance Manager, by viewing one of the inflight dashboards, such as Workload dashboard, as shown in Figure 10-2. If the screen is empty, then you should investigate why Optim Performance Manager data is not being shown.

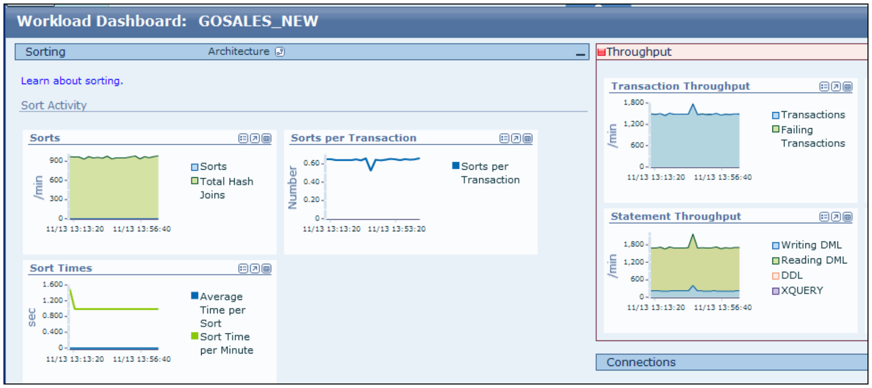


Figure 10-2 Optim Performance Manager - Workload dashboard

2. Activate, install, and configure the Extended Insight components on your Optim Performance Manager server and your WebSphere Application Server.

   Why: Establish the Extended Insight communication is working properly.

   a. See 3.2.2, “Activating the Optim Performance Manager license” on page 80 for information on Extended Insight activation.

   b. See 3.4, “Installing and Configuring Extended Insight Client” on page 129 for information on installation and configuration of the Extended Insight client software.

   c. See 3.4, “Installing and Configuring Extended Insight Client” on page 129 for information on enabling and configuring Extended Insight monitoring for your database.

   d. See 3.4, “Installing and Configuring Extended Insight Client” on page 129 for WebSphere Application Server version customization details.

   e. *Verify*: After running your WebSphere application for some time, you should see data on the Extended Insight dashboard, as in Figure 10-3. It is not important in this example what the data is, only that there is something there. You may have to wait several minutes before all the data is properly flowing and presented on the screen the first time. If the screen is still empty after 15 minutes, you should investigate before proceeding.

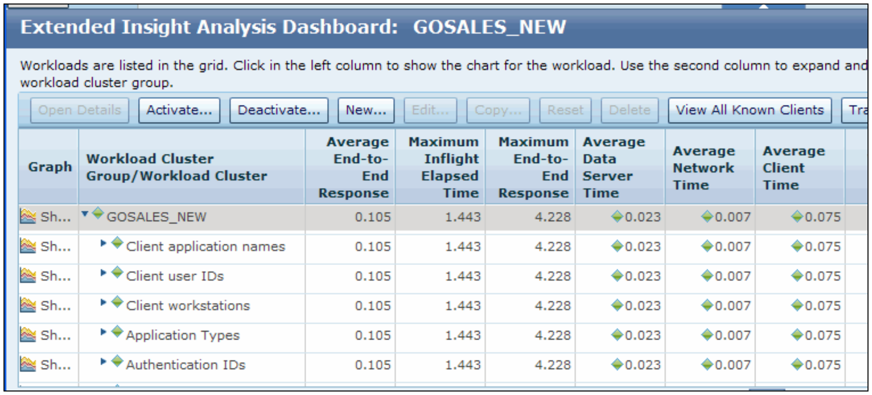


Figure 10-3 Active Extended Insight dashboard

3. Install and configure your IBM Tivoli Monitoring (ITM) infrastructure.

   Optim Performance Manager assumes a working ITM environment exists, and does not provide how-to information. Implementing Tivoli monitoring is not a trivial task, so we strongly recommend that you work with your IBM Tivoli technical representatives to ensure that you have the proper monitoring design for your shop. A working Tivoli monitoring environment is a pre-requisite for the integration with Optim Performance Manager.

   The Information Center for Tivoli Monitoring has extensive documentation about implementation:

   `http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.2fp2/welcome.htm`

   In-depth discussion of Tivoli implementation is beyond the scope of this book.

   In the lab environment for this book, we use the ITM OS Agents for our AIX and Linux servers. We also have a simple configuration with most of the components (TEMS and TEPS) on one machine. This is not a typical enterprise production topology, but works fine for the needs of the book. Types of monitoring topologies are discussed at length in the Tivoli monitoring documentation.

   a. *Verify*: In Figure 10-4, we show an example TEP OS workspace for our lab Linux DB2 server, `SDOD03L3`. Seeing this information indicates that the Tivoli infrastructure is operational.

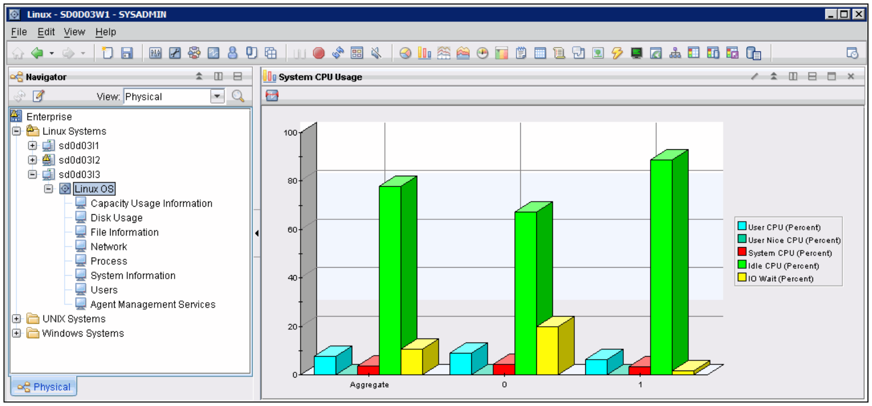


Figure 10-4 TEP OS Agent workspace

4. Install and configure ITCAM for Transactions and ITCAM for Application Diagnostics base products.

   Optim Performance Manager assumes a working ITCAM environment exists, and does not provide how-to information in its documentation. Implementing ITCAM is not a trivial task, so we strongly recommend that you work with your IBM Tivoli technical representatives to ensure you have the proper monitoring design for your shop. A working Tivoli ITCAM monitoring environment is a pre-requisite for the integration with Optim Performance Manager.

   The Information Center for Tivoli Composite Application Monitoring has extensive documentation about implementation:

   `http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcamfad.doc_7101/ic-homepage.html`

   In-depth discussion of Tivoli implementation is beyond the scope of this book.

   In the lab environment for this book, we use a single Transaction Collector and Transaction Reporter, which are also installed on the same machine as the TEMS and TEPS. This may not be typical enterprise production topology, but works fine for the needs of the book. ITCAM topologies are discussed at length in the ITCAM documentation.

   a. *Verify*: Figure 10-5 on page 331 shows the TEP workspace for ITCAM Transaction Aggregate Topology view for the JDBC transactions. This is the typical view of an enterprise application transaction flow. Remember, we have not yet integrated Optim Performance Manager into this picture.

> **Note:** You must specifically enable TTAPI for JDBC in order to see the JDBC node as shown in Figure 10-5. By default, an ITCAM Agent for WebSphere Applications Data Collector will not monitor JDBC transactions when it is set to MOD Level 1 by the Managing Server. If you need JDBC tracking on MOD L1, you must enable it. ITCAM configuration is beyond the scope of this book, for more information on enabling the TTAPI for JDBC transactions see
>
> `http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcamt.doc_7.2.0.2/tt/dita/cam_ad/enable_disable_jdbcl1.html`

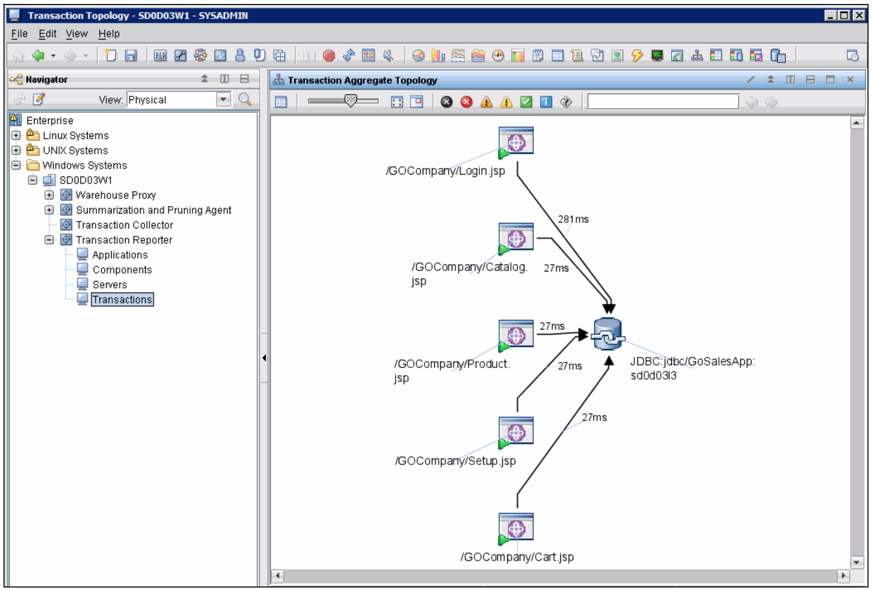


Figure 10-5 TEP ITCAM Transaction Aggregate Topology view

Checkpoint: At this point if you have established a functioning Optim Performance Manager with Extended Insight, and a working Tivoli monitoring infrastructure with ITM and ITCAM, as described in steps 1-4 above, you are ready to enable the integration of Optim Performance Manager with Tivoli.

10.2.2 Enabling the integration

The enabling items could be performed in any order, however, we recommend the order described here - it will be easier to verify your progress at each step. If we continue the step numbering from the previous section, the steps for enabling the integration are as follows:

5. Enable the Optim Performance Manager Data Collector on Optim Performance Manager.
6. Enable the Optim Performance Manager on ITCAM WebSphere agent.
7. Install the Optim Performance Manager TEP plug-in on the TEPS.

Who perform this task

The person with role of Optim Performance Manager Administrator can perform this task, but will need some specific information from the ITCAM administrator. Someone with privileges for, and knowledge of, the TEP is also required to perform the verify step.

Enable in Optim Performance Manager

This step is to enable Optim Performance Manager data collector. In this step, you tell Optim Performance Manager where to send the Extended Insight data, defining the location of the ITCAM for Transactions Transaction Collector, and how often to send the data.

There are two methods to enable the Optim Performance Manager data collector:

- Using Optim Performance Manager configuration dialog: This method is only available in Optim Performance Manager V4.1.0.1,
- Using the ITCAM Data Collection dialog of Optim performance Manager: This method is available in both Optim Performance Manager V4.1 and V4.1.0.1.

Configure data collection using Optim Performance Manager configuration dialog

Perform these steps to configure data collection:

1. On the Optim Performance Manager Manage Database Connections page, Select **Configure Monitoring...** for the database you want to configure, and go to Step 2 of the configuration dialog.
2. Select **Collect Extended Insight data** to modify the existing Extended Insight configuration (remember we have assumed here, and recommend, that your Extended Insight is already working), see Figure 10-6.

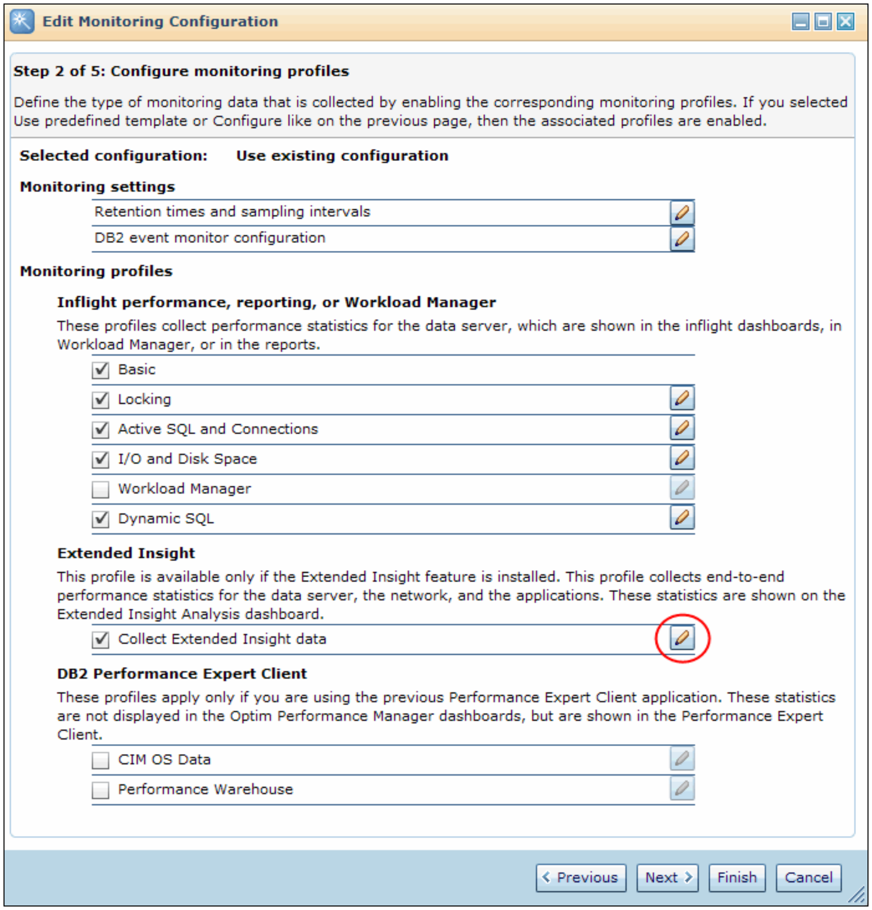


Figure 10-6 Extended Insight configuration

3. On the Extended Insight configuration page, select the **Integration with Tivoli Monitoring** tab as shown in Figure 10-7 on page 334.

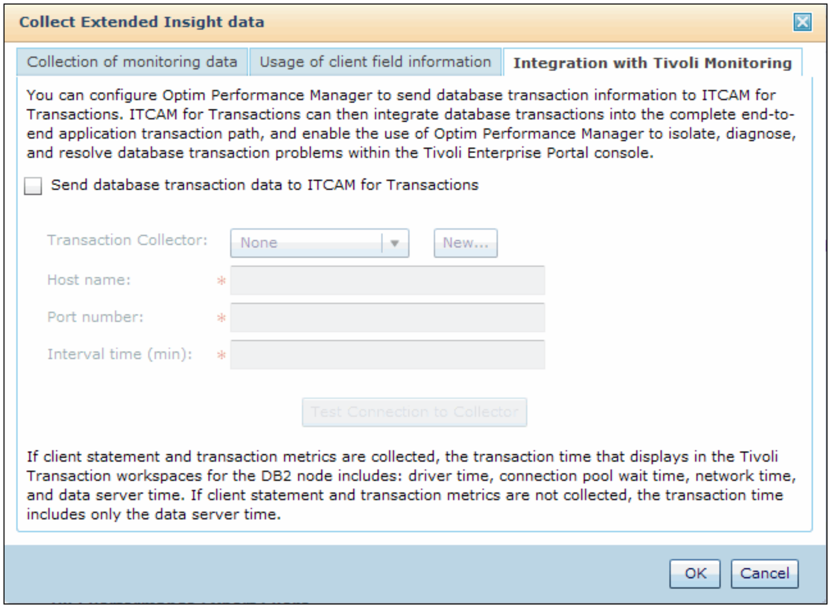


Figure 10-7 Optim Performance Manager-ITCAM configuration page - initial view

4. Select **Send database transaction data to ITCAM for Transactions**, which will enable the "Transaction Collector" field on the screen.

   a. The "Transaction Collector" drop down list will show any collectors that already exist. You can reuse an existing one, or you can define a new one. In this example we are adding a new collector.

   b. Click **New**, to enable the other fields.

   c. Enter the host name (or IP address) and port number of your ITCAM for Transactions transaction collector machine. You must have or obtain this from your Tivoli administrator. The default port for the transaction collector component is `5455`, so Optim Performance Manager pre-fills that field. If your shop uses a different port, you will have to enter that value here. The host name is not required to be fully qualified name, but it must be reachable in the DNS.

      In this example, we use our lab environment transaction collector host `SDOD03W1` and the default port, `5455`.

   d. The "Interval time" value tells Optim Performance Manager how often to retrieve, aggregate and send data to ITCAM. Optim Performance Manager uses the data collected by the Extended Insight client and stored in the Optim Performance Manager repository database tables as the source of data it sends to ITCAM. The Optim Performance Manager transaction data collector process will wake up at the frequency you specify here, and aggregate the Extended Insight data that has arrived over the last interval, then send it to ITCAM. Optim Performance Manager uses the data source connection attributes to identify the workload groups - similar to what you see on the Extended Insight dashboards in the workload cluster groups.

The default interval is five minutes, however in our ITSO lab environment, we found using eight minute interval provided the better results with our small test workload.

> **Note:** The 5-minute default interval in Optim Performance Manager corresponds to the default ITCAM for Transactions aggregation interval. You may need to coordinate the intervals between Optim Performance Manager and ITCAM for Transactions. For more information on ITCAM for Transactions configuration, see
>
> `http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcamt.doc_7.2.0.2/tt/dita/reference/kto_tools_datacoll_coll.html`

See Figure 10-8 on page 335 for the final definition of the new collector we added.

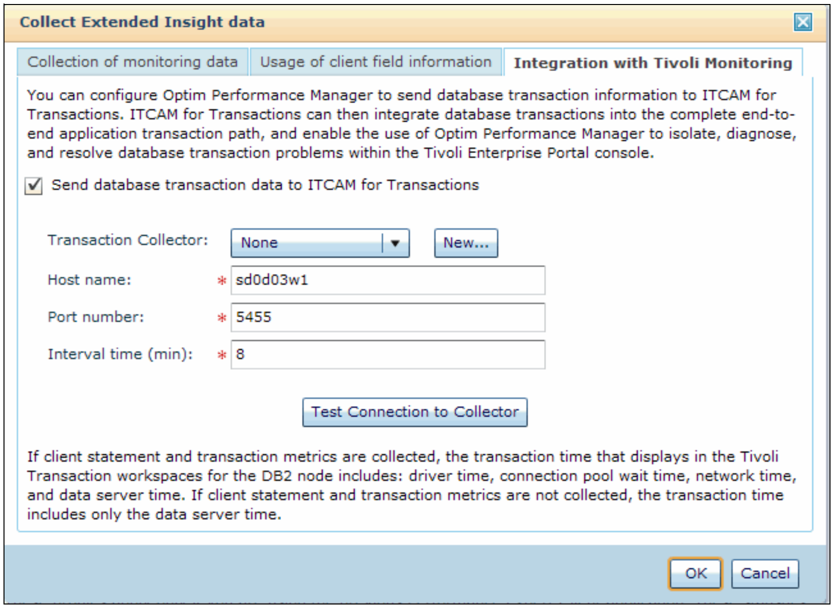


Figure 10-8 Add a new Optim Performance Manager transaction data collector

e. You can use the "Test Connection to Collector" button to perform a simple ping to the host and port you specified. See Figure 10-9 for an example of a successful ping.

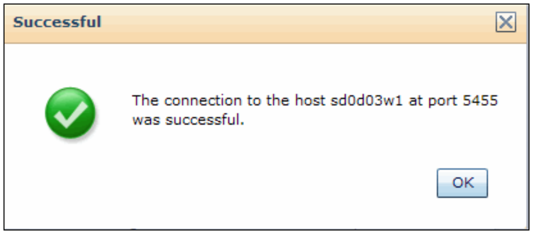


Figure 10-9 Successful test to ITCAM Transaction Collector

> **Note:** Currently, this only pings to the host and port and verifies it is listening - it does not confirm it is indeed an ITCAM Transaction Collector listening. You should verify the ITCAM host and port data with your Tivoli administrator to be positive of these values.

f. Click **OK** to accept the new collector and dismiss the Extended Insight configuration window.

g. Click **Finish** to save the configuration.

> **IMPORTANT:** The collector you added is not saved until you save the entire configuration.

Checkpoint: Now you have added a collector. If your WebSphere workload is still running and your collector is configured correctly, Optim Performance Manager will start to send the Extended Insight data to ITCAM. You should wait for at least two cycles of the interval period before checking for data on the TEP Transaction Reporter workspace.

5. When Optim Performance Manager data is successfully transferred to the ITCAM Transaction Collector, a new transaction node type will appear in the Transaction Reporter workspaces in TEP. The node is named `DB2_LUW`, and has additional strings after it, describing the connection attributes. The connection attributes are discussed later in this chapter when we look at an integration scenario, see 10.3, “Usage scenario” on page 351.

6. *Verify*: As shown in Figure 10-10 on page 337, some `DB2_LUW` nodes are present but there are no arrows connecting the JDBC node to them. This is called “stitching”, which is missing here. There are a few more steps required to have full integration with ITCAM. However, if you see this much, it means you are on the right track.

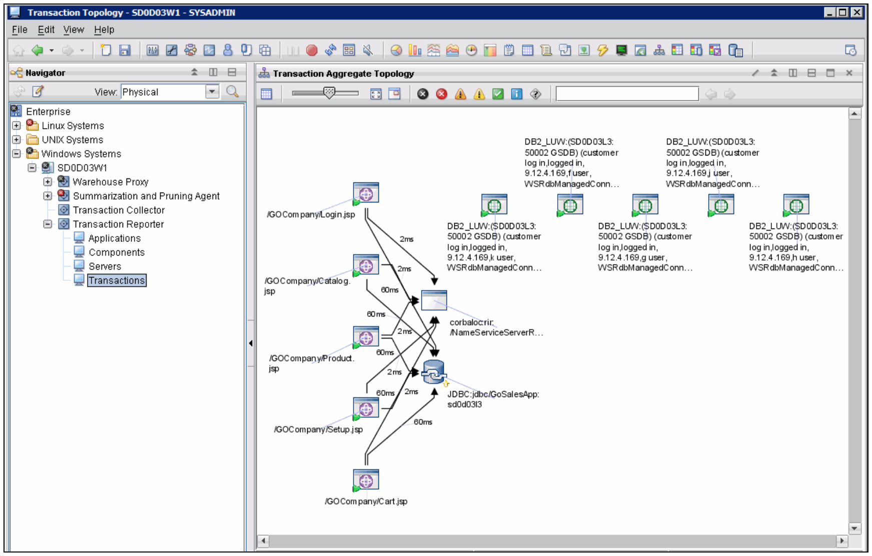


Figure 10-10 TEP Transaction Reporter workspace with unstitched JDBC and `DB2_LUW` nodes

This is the end of the configuration steps required on the Optim Performance Manager server side.

Configure using the ITCAM Data Collection dialog of Optim Performance Manager (alternate)

The ITCAM Data Collection is another dialog where you can add, update, and remove data collectors, and this dialog was the only option in Optim Performance Manager V4.1. While the new option of configuring the ITCAM data collection during the Optim Performance Manager Extended Insight configuration was added with Optim Performance Manager V4.1 Fix Pack 1, the ITCAM Data Collection dialog was retained for several reasons:

- Ability to modify existing collectors.

  The Optim Performance Manager configuration dialog only allows adding a new collector and/or associating your database connection with an existing collector. If you want to make any changes to the collector parameters, you must use the dialog described here.

- Ability to configure multiple databases at one time.

  Suppose you have 25 database connections you want to configure for Optim Performance Manager-ITCAM integration. The new method would require you to edit each connection's configuration separately and enable the data collection. If you use the ITCAM Data Collection dialog, you can enable them all at one time.

- The ITCAM Data Collection dialog is a view grouped by the data collector itself, rather than the connection-level view. This makes it easier to manage if you have many databases or many ITCAM Transaction collectors.
- The ITCAM Data Collection dialog also now provides a status check, which is described later in this section.

We describe here using the alternate method to configure another database to use the same collector we defined before.

1. Click the **Task Manager** from the Overview dashboard.
2. Select **ITCAM Data Collection** under the Setup heading, see Figure 10-11.

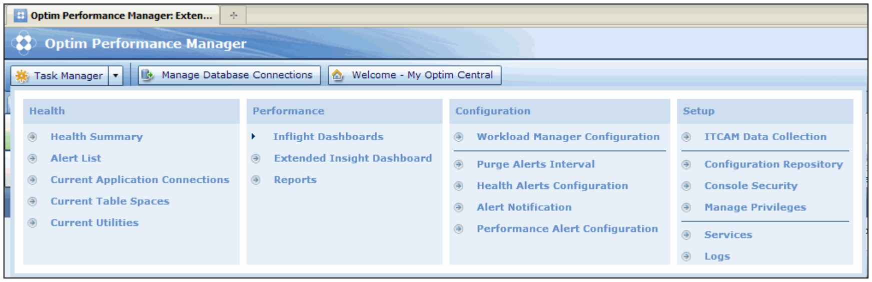


Figure 10-11 Task Manager menu - ITCAM Data Collection

3. On the ITCAM Data Collection screen (Figure 10-12 on page 338), we see the one data collector we added for `SD0D03W1`.

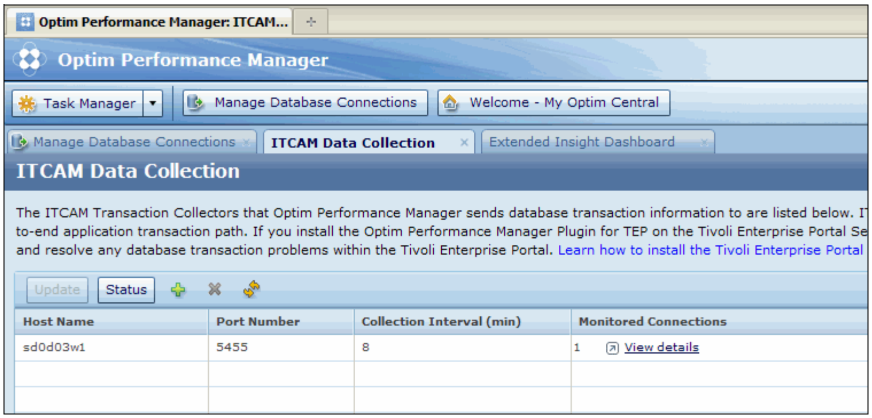


Figure 10-12 ITCAM Data Collection

In this example, we want to use this same collector for another database - `DTRADER`. Open the Extended Insight configuration for `DTRADER` database and enable monitoring there, but we will use the alternate method this time.

4. Select the collector, and click **Update**.
5. You could change any of the collector parameters here, and these changes would apply to any database using that collector. In this case, we are only adding another database. Select the **Monitored connections** drop down list, and choose the `DTRADER` database connection, as shown in Figure 10-13.

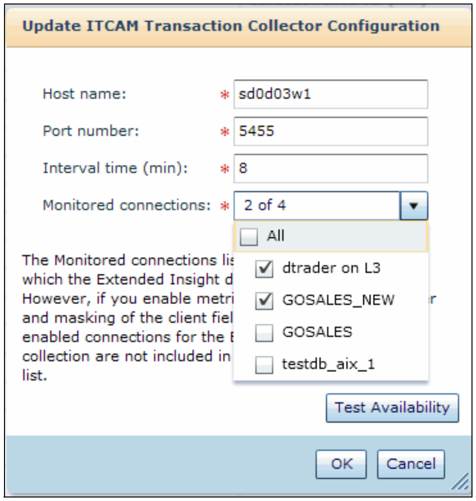


Figure 10-13 Update existing ITCAM collector

6. Click **OK** to save your changes.
7. On the ITCAM Data Collection screen, now you can see there are 2 databases associated with the one collector. Figure 10-14 shows the View Details window, indicating both the `GSDB` and `DTRADER` are configured.

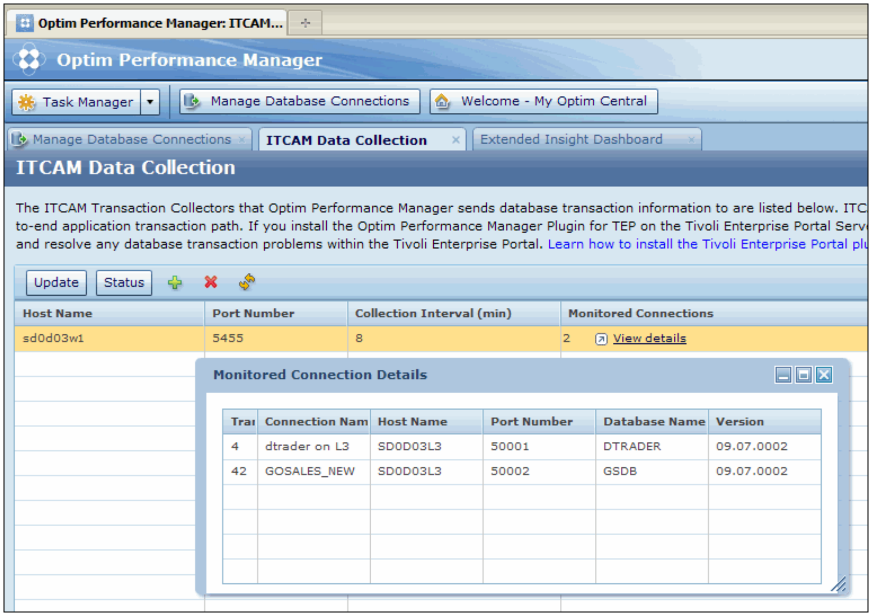


Figure 10-14 Updated ITCAM collector with two monitored connections

If your shop uses multiple ITCAM transaction collectors, you should collaborate with your ITCAM administrator to understand the appropriate transaction collector to specify. A database connection monitored by Optim Performance Manager can be associated with one and only one ITCAM transaction collector.

Enable at ITCAM WebSphere agent side

You must also enable a property in the ITCAM for Application Diagnostics WebSphere agent property file. You may see references to the Tivoli product code for this agent as "*yn*".

Who perform this task

The property file must be modified by someone with privileges to modify the file - an ITCAM administrator for example. It may vary at your shop. Generally the file is owned by root user, so you should plan accordingly.

Enabling procedure

If IBM Optim Performance Manager is installed, you can enable TTAPI integration between ITCAM for Application Diagnostics, Transaction Tracking, and Optim Performance Manager.

Optim Performance Manager provides detailed information about DB2 JDBC calls. If integration is enabled, you can “drill down” from transactions displayed in Transaction Tracking workspaces to the Optim Performance Manager console and dashboard to view deep database diagnostics information and detailed SQL statement performance data.

The Information Center for ITCAM for Transactions has a good description of how to enable this integration.

http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcamt.doc_7.2.0.2/tt/dita/cam_ad/enabling_optim_integration.html

To customize ITCAM to handle the incoming `DB2_LUW` data from Optim Performance Manager, perform these steps:

1. Modify the property file

   To enable Optim Performance Manager integration, modify `DCHOME/runtime/platform.node.server/custom/toolkit_custom.properties`

   Set the following property in this file:

   ```
   com.ibm.tivoli.itcam.dc.ttapi.jdbc.opm.enabled=true
   ```

   If any monitored J2EE application changes the JDBC connection client attributes during an active session, also set the following property:

   ```
   com.ibm.tivoli.itcam.dc.ttapi.jdbc.opm.clientinfo.reset=true
   ```

   Example 10-1 shows a snippet from the property file in our book lab environment. The file name in this server is:

   ```
   /opt/IBM/ITM/lx8266/yn/wasdc/7.1.0.1.2/runtime/was70.sd0d0312Node01.server1
   /custom/toolkit_custom.properties
   ```

   Example 10-1 ITCAM toolkit_custom.properties file

   ```
   ###################################################################
   # This property enables Data Collector and Optim Performance Manager
   # integration through TTAPI.
   #
   # Uncomment this property to enable DC and OPM integration through TTAPI
   ###################################################################
   com.ibm.tivoli.itcam.dc.ttapi.jdbc.opm.enabled=true


   ### add this property to enable clientinfo changing in app
   com.ibm.tivoli.itcam.dc.ttapi.jdbc.opm.clientinfo.reset=true
   ```

2. Restart your WebSphere application server instance.

3. *Verify*: Run your WebSphere workload again, as in the verify step described in step 5 on page 339, and again after waiting several intervals, you should see arrows stitching the JDBC and `DB2_LUW` nodes together. An example is shown in Figure 10-15 on page 342.

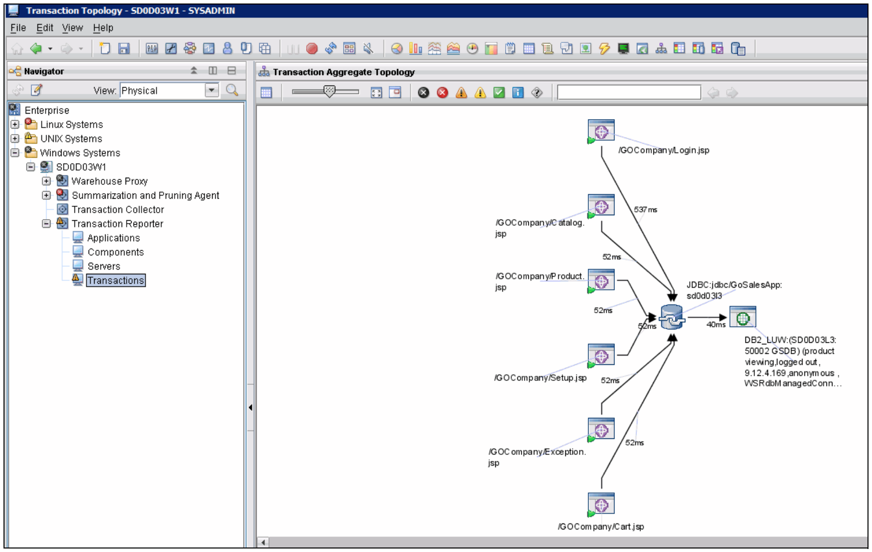


Figure 10-15 TEP Transaction Reporter workspace with stitched JDBC and DB2_LUW nodes

At this point of the configuration, you have the end-to-end transaction flow visible in TEP, but you do not yet have capability to drill down into Optim Performance Manager from the transaction in TEP. There is one more installation piece that is required to fully enable the ITCAM integration, and is covered in section “Install the Optim Performance Manager TEP workspace” on page 343.

To verify whether the TEP workspace is configured yet (maybe your administrator already installed the workspace), you can right-click one of the `DB2_LUW` transaction objects in TEP, select the **Link To...** menu item, then check if there is “Database Diagnostics” menu item listed. In the picture shown in Figure 10-16 on page 343, there is no such link so we know the Optim Performance Manager TEP plug-in is not yet installed.

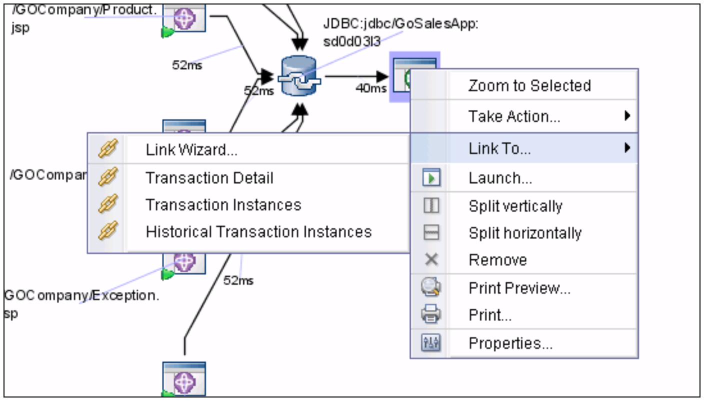


Figure 10-16 Optim Performance Manager TEP workspace not available yet

Checkpoint: If you have the correct stitching visible in the TEP workspace, then you are on the right track. If stitching does not appear from JDBC to `DB2_LUW` nodes after some waiting, then you should begin investigation. This may require assistance from ITCAM administrators who know how the Transaction Collector and Reporter are configured.

Install the Optim Performance Manager TEP workspace

Use the Optim Performance Manager plug-in for TEP installation wizard to create a workspace and several dynamic links in IBM Tivoli Monitoring (ITM) under Transaction Reporter application support. This workspace is required to integrate Optim Performance Manager in your TEP Console.

Who performs this task

You need someone who has root or administrator privileges to the Tivoli Enterprise Portal Server (TEPS) machine. The TEPS must be restarted after the install process.

Installation procedure

The Optim Performance Manager TEP plug-in installer program is on a separate CD, or is a separate package you download with Optim Performance Manager Extended Edition. You can read more detail at the Information Center page:

`http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.tep.installconfig.doc/tep_install.html`

Install Optim Performance Manager application support into TEP workspace, using a GUI:

1. On the TEPS server, launch the installer.

In our lab environment, the TEPS machine runs on Windows server `SDOD03W1`, and the installer program is `install.bat`.

2. When prompted, confirm the directory where the IBM Tivoli Monitoring (ITM) is installed. In our case, it is the `C:\IBM\ITM` directory. See Example 10-2. The install script (.BAT or .sh) will automatically stop and start the TEPS server as needed, so be aware that this will restart TEPS.

Example 10-2 Optim Performance Manager TEP Plug-In installation dialog - start

```
Specify the directory in which to install TEP plug-in.

     Default installation directory:  C:\ibm\ITM

Enter an absolute path, or press Enter to accept the default:
The TEP plug-in is installing.
The TEP Server service is stopping.
The TEP Server service was stopped.
```

3. After the TEPS is stopped, the install script launches the GUI install, as shown in Figure 10-17. Click **Next** to continue.

Figure 10-17 Optim Performance Manager TEP Plug-in installer GUI

4. Confirm the ITM installation path, and the installer location, as shown in Figure 10-18 on page 345, and click **Next**.

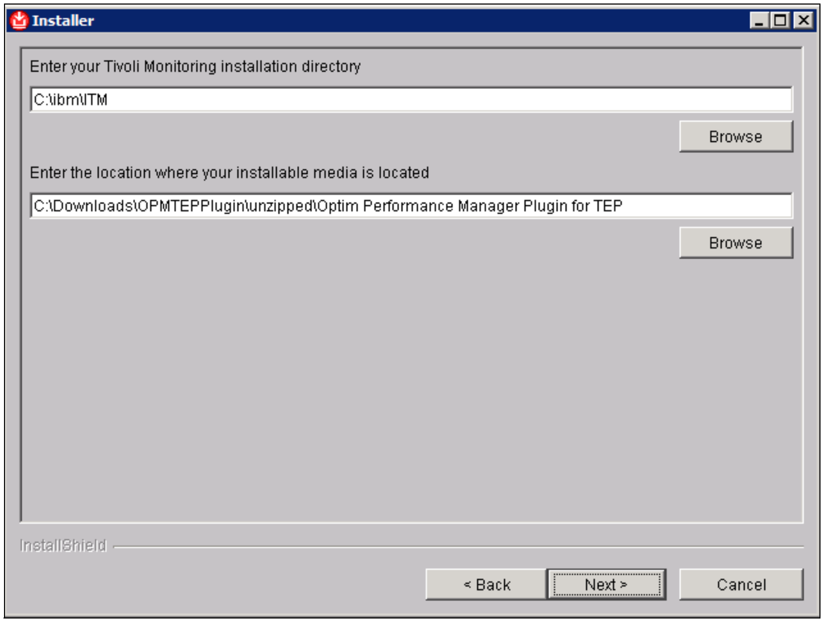


Figure 10-18 TEP Plug-in installer - confirm paths

5. On this screen you only have one choice, TEPS, and it must be selected, see Figure 10-19 on page 346. Click **Next** to continue.

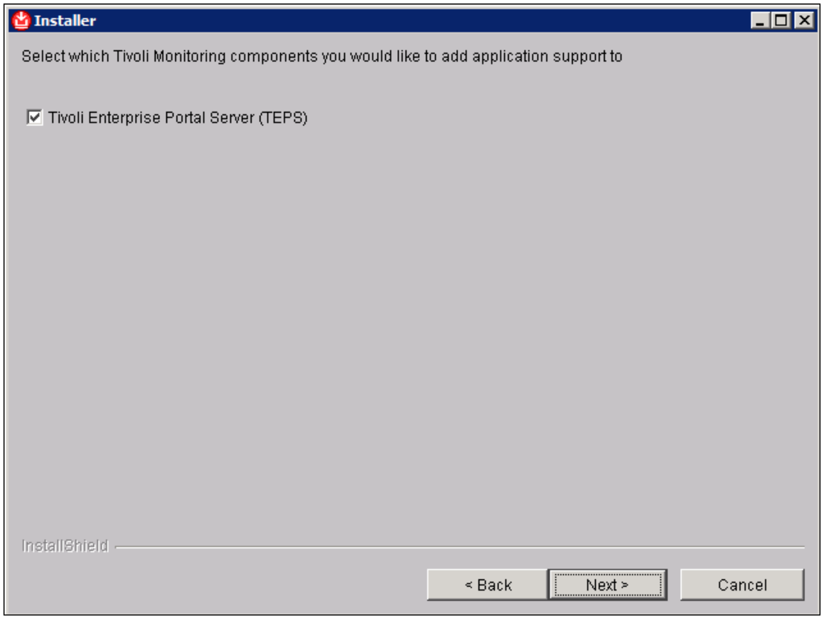


Figure 10-19 TEP Plug-in installer - select TEPS

6. On this screen, check the box for TEPS and click **Next**, as shown in Figure 10-20

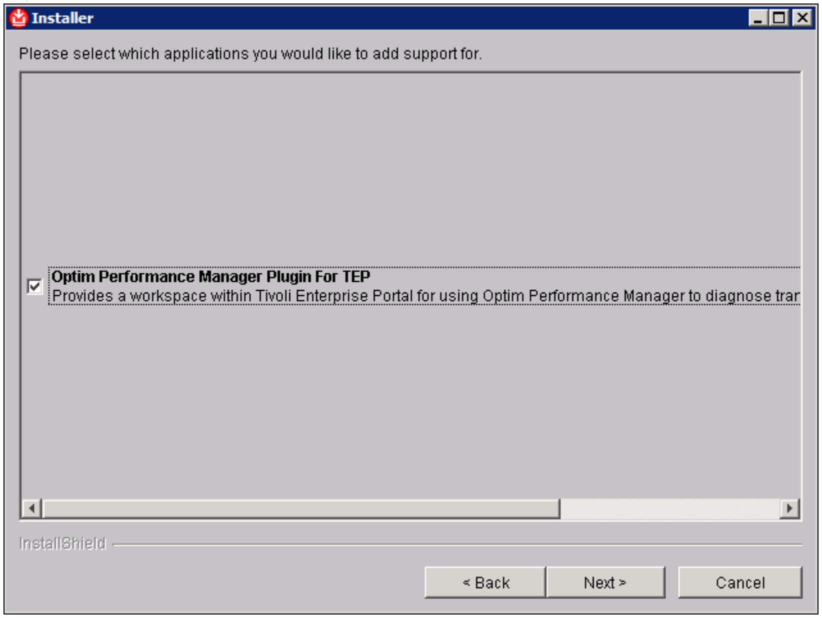


Figure 10-20 TEP Plug-in installer -select application support for Optim Performance Manager

7. Confirm your choices one more time, and click **Next** to initiate the install, as shown in Figure 10-21 on page 348.

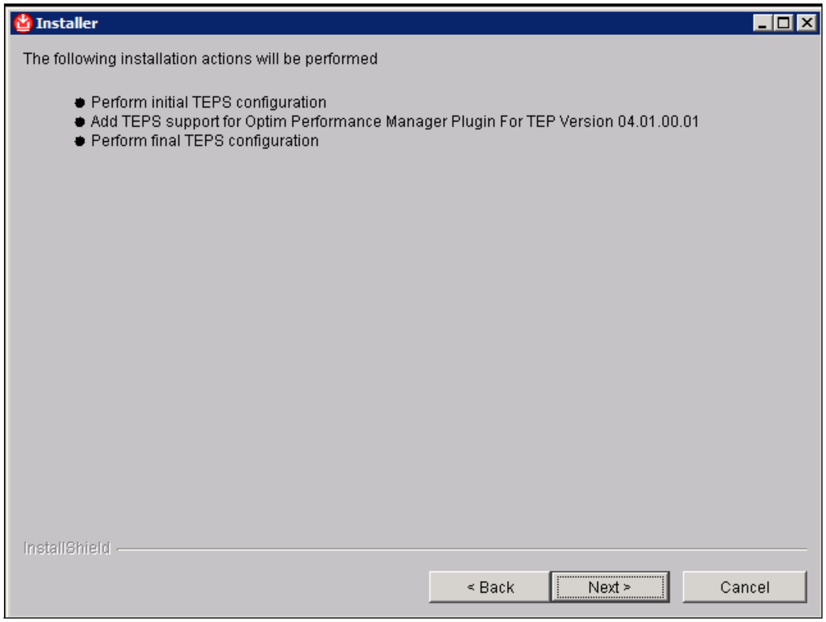


Figure 10-21 TEP Plug-in installer - confirm choices

8. You can watch the progress screen, and when install is complete, there is a message, as shown in Figure 10-22 on page 349. The install was successful. Click **Finish** to return back to the install.bat program.

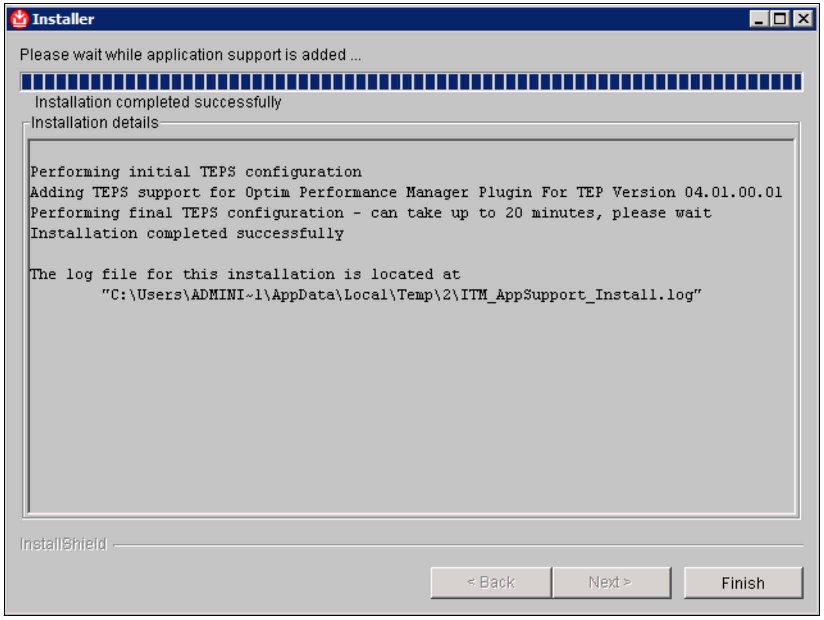


Figure 10-22 TEP Plug-in installer - successful completion

9. The original installer script will restart the TEPS process, as we see in the bold text of Example 10-3. Press any key to dismiss the console window.

Example 10-3 Optim Performance Manager TEP Plug-In installation dialog - finish

```
Specify the directory in which to install TEP plug-in.

     Default installation directory:  C:\ibm\ITM

Enter an absolute path, or press Enter to accept the default:
The TEP plug-in is installing.
The TEP Server service is stopping.
The TEP Server service was stopped.
The TEP Server service is starting.
The TEP Server service was started.
The TEP plug-in was installed.
Press any key to continue . . .
```

10. *Verify*: Now your Optim Performance Manager workspace has been installed and the TEPS restarted. You can verify the workspace by checking the same link as in Figure 10-16 on page 343, but this time there should be a “Database Diagnostics” choice on the menu, as shown in Figure 10-23 on page 350.

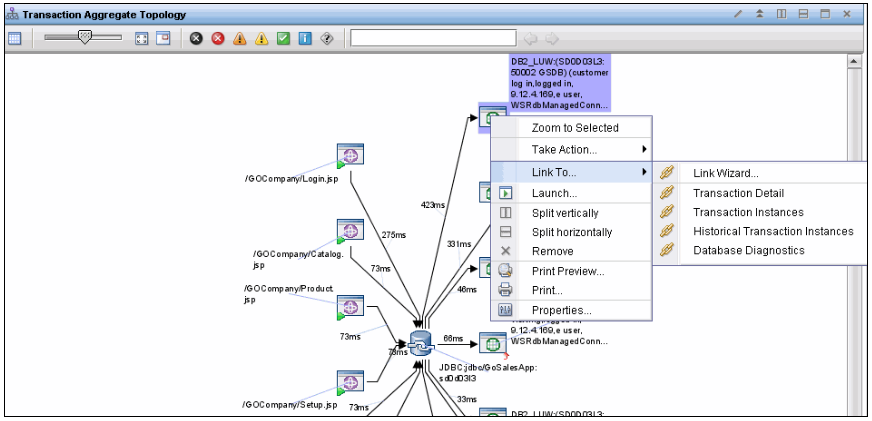


Figure 10-23 Optim Performance Manager TEP workspace is present - Database Diagnostics

Silent install Optim Performance Manager application support into TEP workspace, (alternate)

If you do not have a GUI available on your TEPS machine, or if you prefer not to use a GUI, you can install the Optim Performance Manager TEP Plug-in using a silent install. A sample response file is included in the install image, which you can update and pass to the install script.

We do not show an example here, but the process is described in the Optim Performance Manager Information Center:

http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.tep.installconfig.doc/tep_install.html

Verify Optim Performance Manager product presence on TEPS (optional)

Your TEPS administrator might want to know the two-character product code for Optim Performance Manager, that is “O9” (the letter O, not zero.) The TEPS administrator could verify the Optim Performance Manager application support by using the Tivoli command **kincinfo** (Windows) or **cinfo** (UNIX).

For example, we can see the Optim Performance Manager information in the output as shown in Example 10-4.

Example 10-4 TEPS - kincinfo command output

```
C:\Users\Administrator>kincinfo -t o9

*********** Tuesday, November 16, 2010 4:15:19 PM ***********
```

```
User       : Administrator        Group     : NA
Host Name  : SDODO3W1             Installer : Ver: 062202000
CandleHome : C:\ibm\ITM
Installitm : C:\ibm\ITM\InstallITM
*************************************************************
...Product Inventory


PC  APPLICATION SUPPORT DESC                      PLAT   APP VER      BUILD         INSTALL DATE

09  TEPS Support for Optim Performance Manager Pl  WINNT  04.01.00.01  200705161329  NOVALUE
09  TEPB Support for Optim Performance Manager Pl  WINNT  04.01.00.01  200705161329  NOVALUE
```

10.3 Usage scenario

In the earlier sections of this chapter, we showed some isolated screenshots of Optim Performance Manager and TEP screens in varying states of integration. We explore a simple scenario in this section.

Suppose that a system operator is looking at the TEP console and notices some slow transactions in the GOSales application. Can they quickly narrow down the problem area and open a ticket to the right group?

10.3.1 Initial analysis in Tivoli Enterprise Portal

Start this scenario by looking at the top-level view in the Tivoli Enterprise Portal (TEP). In Figure 10-24 on page 352, we observe a situation has been raised about slow transactions for the GOSales application. There are many ways to use the TEP to investigate problems, and this book is intended to focus on Optim Performance Manager, so we just dive right to the relevant areas of TEP for this scenario.

The person using the TEP might be a general system operator, or it could be anyone who has privileges to use the software.

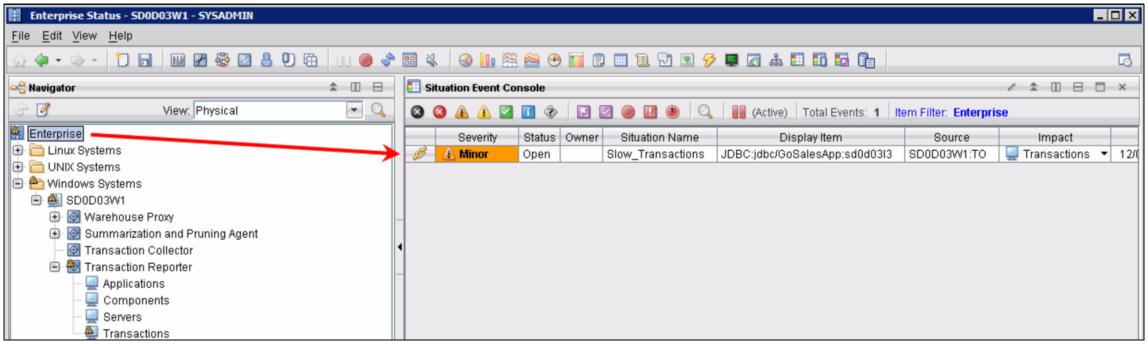


Figure 10-24 TEP showing slow transaction situation for GOSales

We know that for the GOSales application, we can use the Transaction Reporter workspaces in TEP. These belong to the ITCAM for Transactions product, which is what Optim Performance Manager integrates with.

Using Transaction Reporter workspaces

Navigate to the **Transaction Reporter** → **Transactions** workspace on the TEP navigator, on the left side of the screen. See Figure 10-25 on page 353.

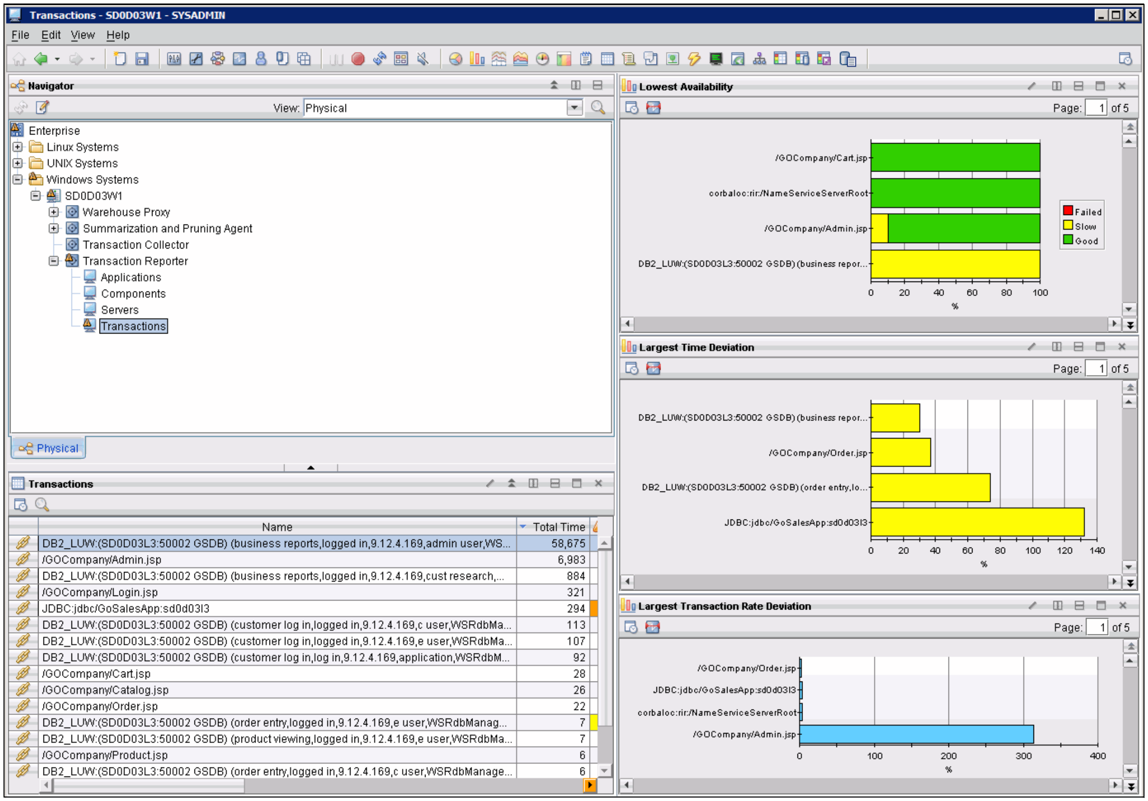


Figure 10-25 TEP Transaction Reporter Transactions workspace

The TEP screen is made up of different views. For example, if we zoom in on the workspace above, in Figure 10-26, we show just the Lowest Availability view from the top-right part of the screen. It shows 100% of the `DB2_LUW` transactions for "business reports" and "admin user" are slow (yellow).

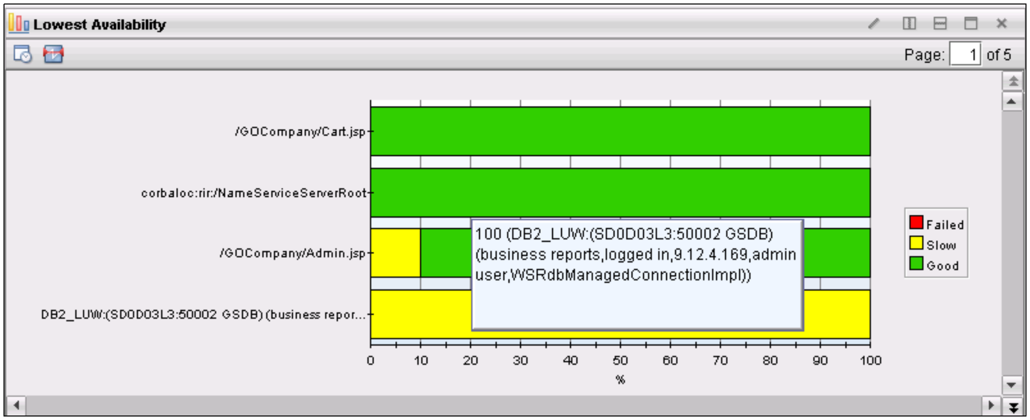


Figure 10-26 TEP Transactions workspace, Lowest Availability view

> **Note:** For information on understanding the Transaction Reporter workspaces, consult the User's Guide section in the Information Center for ITCAM for Transactions:
>
> `http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcamt.doc_7.2.0.2/tt/dita/concept/kto_ws_oview.html`

In the lower-left area of the screen, shown in Figure 10-27, is the Transactions table view, which we have sorted by the *Total Time* column. Clearly the first row belongs to `DB2_LUW` and is by far the slowest transaction. The next slowest is the `/GOCompany/Admin.jsp`, which sounds like it might be related to someone named "admin user" who is doing these slow business reports we saw above.

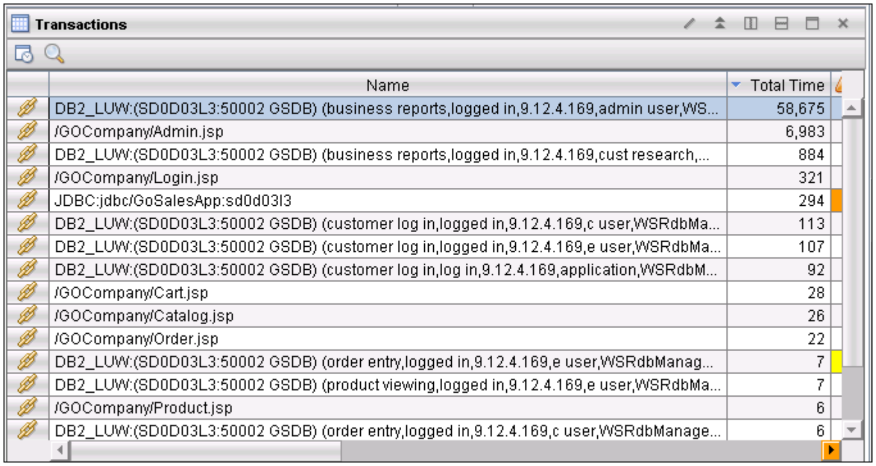


Figure 10-27 TEP Transactions workspace, Transactions view

The JSP data does not come from Optim Performance Manager - it comes from the ITCAM WebSphere agent. Notice there are several JSPs listed, as well as a JDBC transaction, for our `GOSalesApp` data source. What else can the TEP show us about these transactions?

Select the small link icon on the JDBC transaction row, and choose the **Transaction Topology** menu item. This displays a different workspace, as shown in Figure 10-28 on page 356. The ITCAM for Transactions workspaces can show the transaction data in several formats - such as the graphical flow layout (topology view) we see in this diagram, or in table views and other charts. In this topology diagram, we see an aggregated view of all the JSPs that reference the JDBC data source, and all the `DB2_LUW` transactions on the other side of those JSPs.

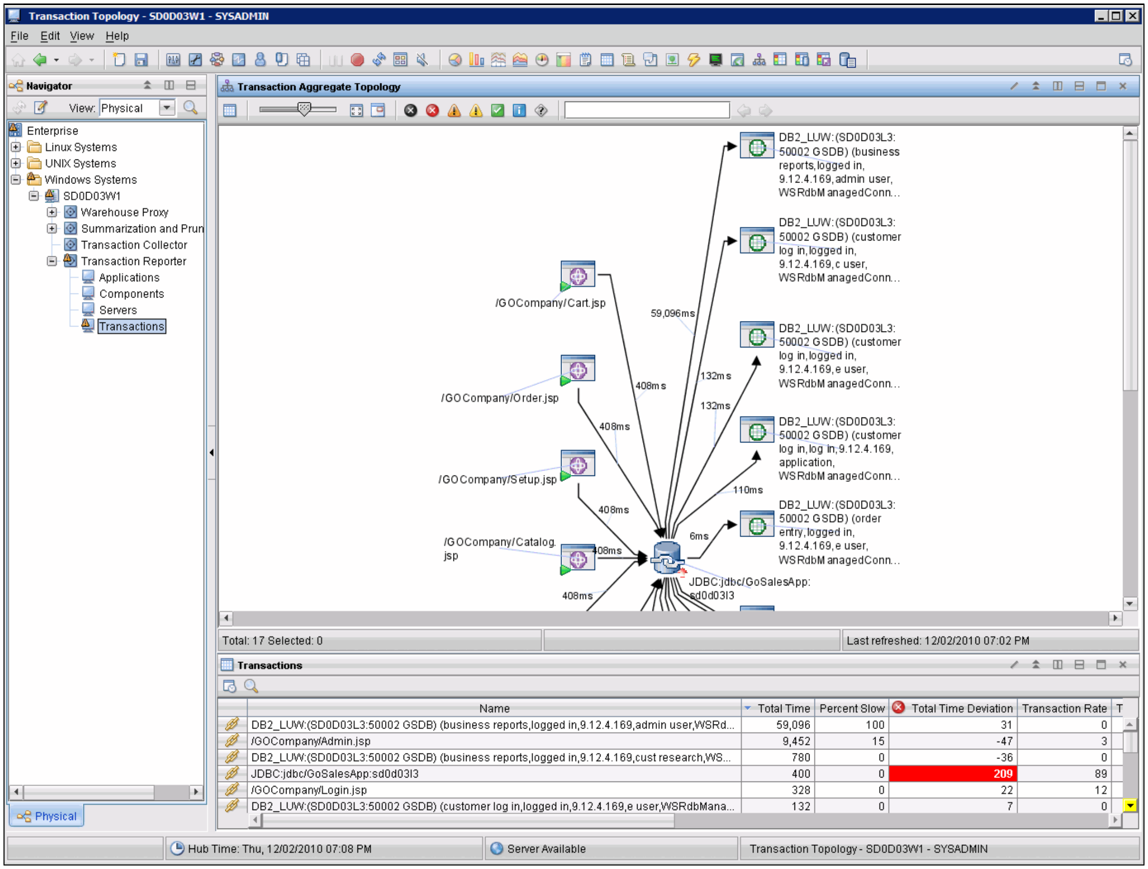


Figure 10-28 Transaction Aggregate Topology workspace

Notice the arrow pointing from the JDBC node to the top-most `DB2_LUW` node shows a number `56,096 ms`. This corresponds to the “total time” value you can see in the table view below the topology view, and indicates the time collected by Optim Performance Manager Extended Insight, which was later sent to the Transaction Collector. (See “Configure data collection using Optim Performance Manager configuration dialog” on page 332, where we set up that communication.)

At this point of the analysis, using ITCAM, we hypothesize that the slowdown of the transactions is occurring on the data server side, and that the slowest transaction appears to be something with an admin business report.

Connection attributes and "stitching"

How did the transactions get the information about "`business reports`" and "`admin user`"? These are parts of the connection attributes, which we discussed in 8.2.3, "Understanding workload clusters" on page 291, and are central to the integration of Optim Performance Manager and ITCAM.

The Tivoli ITCAM for Application Diagnostics WebSphere agent collects performance metrics about transactions and sends those metrics to the ITCAM for Transactions Transaction Collector. Optim Performance Manager Extended Insight also collects performance metrics about the same DB2 JDBC transactions, and sends them to the ITCAM for Transactions Transaction Collector.

It is the job of the ITCAM Transaction Reporter component to figure out how to tie those metrics together to form the transaction flow across the different components. This process is called "stitching". The stitching is made by using the connection attributes, which we discussed in 8.2.3, "Understanding workload clusters" on page 291.

10.3.2 Optim Performance Manager inside TEP

The power of the ITCAM family of products is being able to look at high-level and/or aggregate views, but then to drill down to the domain expert for analysis of specific areas, such as WebSphere or DB2. What the Optim Performance Manager integration provides now is just that ability, using Optim Performance Manager as the domain expert on DB2 performance.

Of course you could just open a new browser and look at Optim Performance Manager, but you lose the context of the particular transaction topology you were just looking at in TEP. Let us see how to launch to Optim Performance Manager in context for those bad `business reports` transactions.

Right click on the row in the table view (or on the `DB2_LUW` node in the topology view) to bring up the context menu, as in Figure 10-29 on page 358. Select **Link to...** , then **Database Diagnostics**. (You could also use the link icon at the left of the table row, to bring up the link menu.)

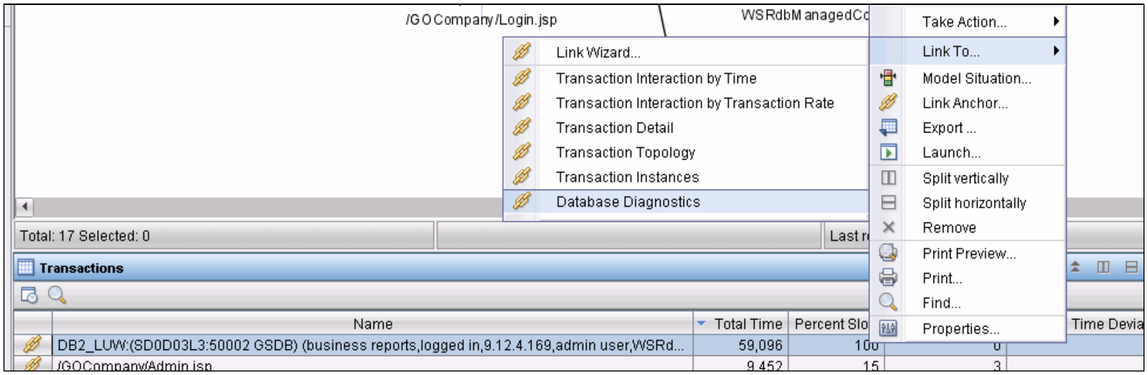


Figure 10-29 Link menu for `DB2_LUW` node

Selecting **Database Diagnostics** opens a new workspace in TEP. The primary view is a browser, and it is pointed to the Optim Performance Manager server associated with the DB2 database you clicked from in TEP.

> **Note:** Your default browser must have Flash player installed. If it is not detected, you will get a prompt to install it.

You will be prompted to log in to Optim Performance Manager, if this is the first launch of your session. Enter your credentials just like in standalone Optim Performance Manager, as shown in Figure 10-30 on page 359.

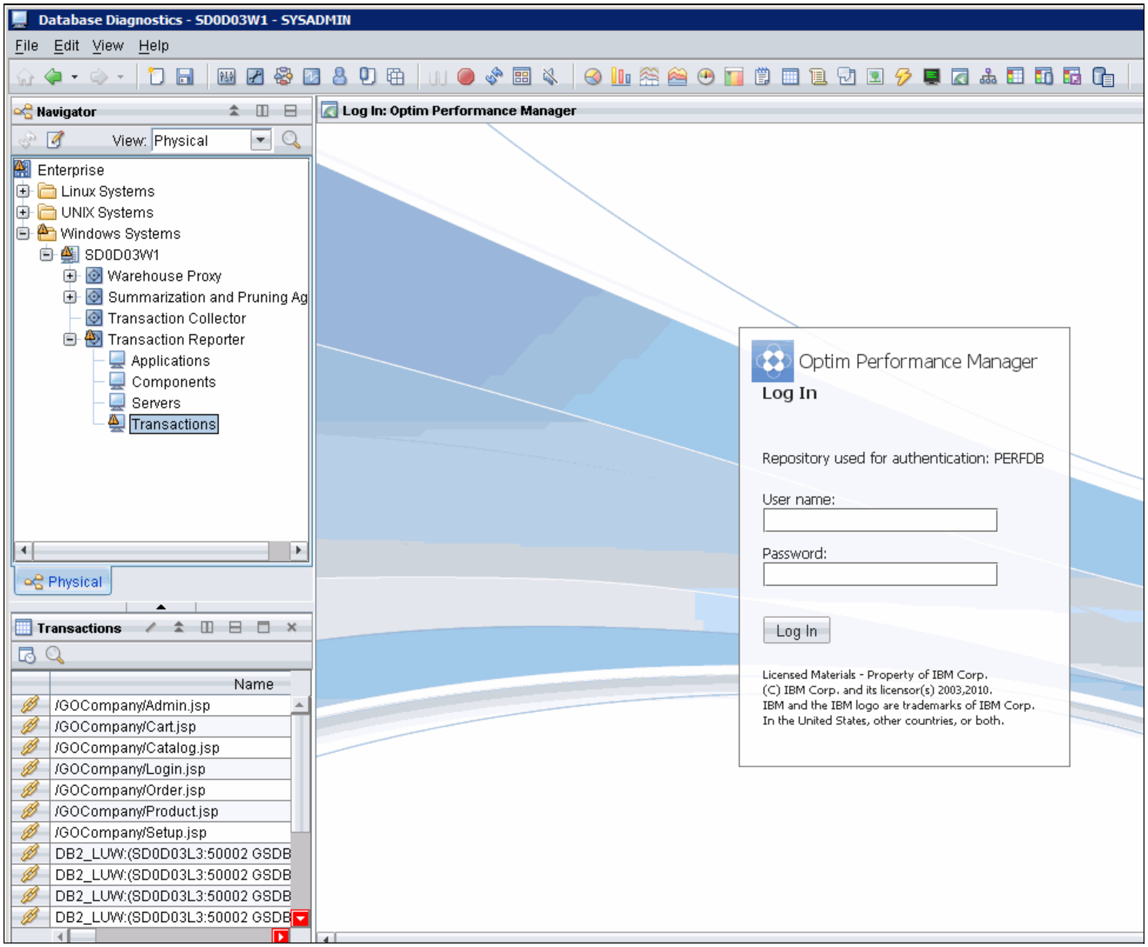


Figure 10-30 TEP Optim Performance Manager Workspace- Log in

After you log in, the information about the transaction you clicked on in TEP is carried through to the Extended Insight dashboard. A special ad-hoc custom workload cluster is created, filtered on just the connection properties for that particular transaction. The launching of Optim Performance Manager from TEP is referred to as “launch in context”, because the browser is automatically positioned to the Extended Insight details dashboard for the specific transaction you selected in TEP. It knows the context because of the connection attributes.

For example, we know from the TEP the connection attributes are for Client Application name “`business reports`” and Client user ID “`admin user`”, along with some other attributes. See Figure 10-31 on page 360. You do not have to create special workload clusters with these attributes - Optim Performance Manager will do this on the fly, when it launches from TEP.

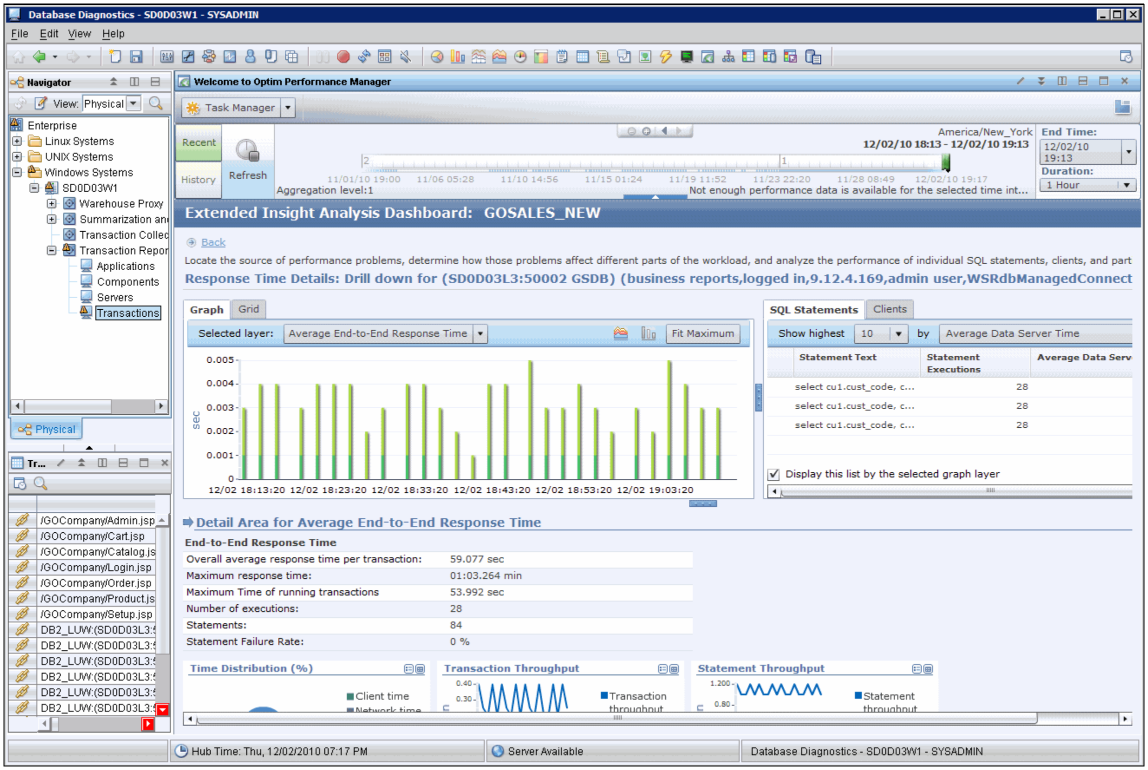


Figure 10-31 Optim Performance Manager Extended Insight details for in-context transaction

We changed the graph to a bar chart view, Figure 10-32 on page 361, because these transactions do not run very often so the line chart is not continuos. It is easier to see this data in a bar chart. Also we set the view to show only the average values, and exclude the maximum line.

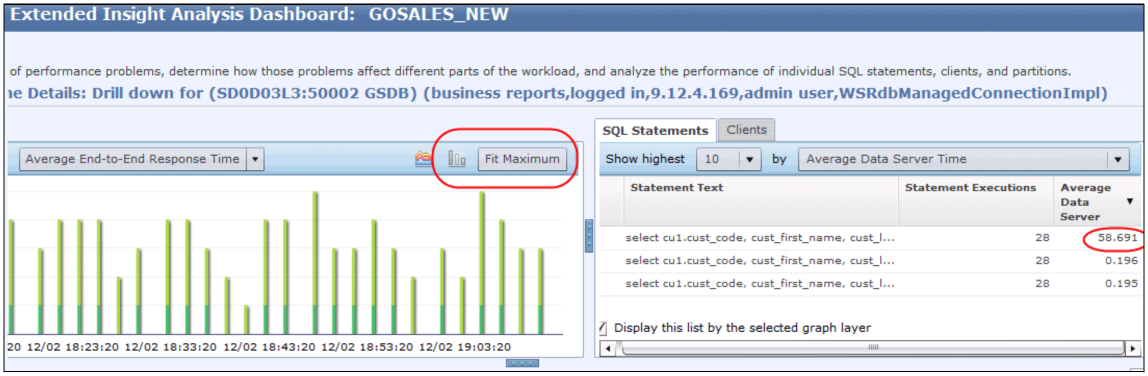


Figure 10-32 Extended Insight - statements for "business reports" and "admin user"

In the SQL Statements table, there are only three statements executed over the last hour, for this set of connection attributes (remember we launched here from TEP, for a specific transaction.) One of those statements shows an average data server time of nearly 59 seconds. The other two statements are much faster.

Select the bad statement, to view some details about it, as in Figure 10-33 on page 361. This view does not tell us a lot we did not already see previously, so let us click on the **Statement Server Execution Details** tab.

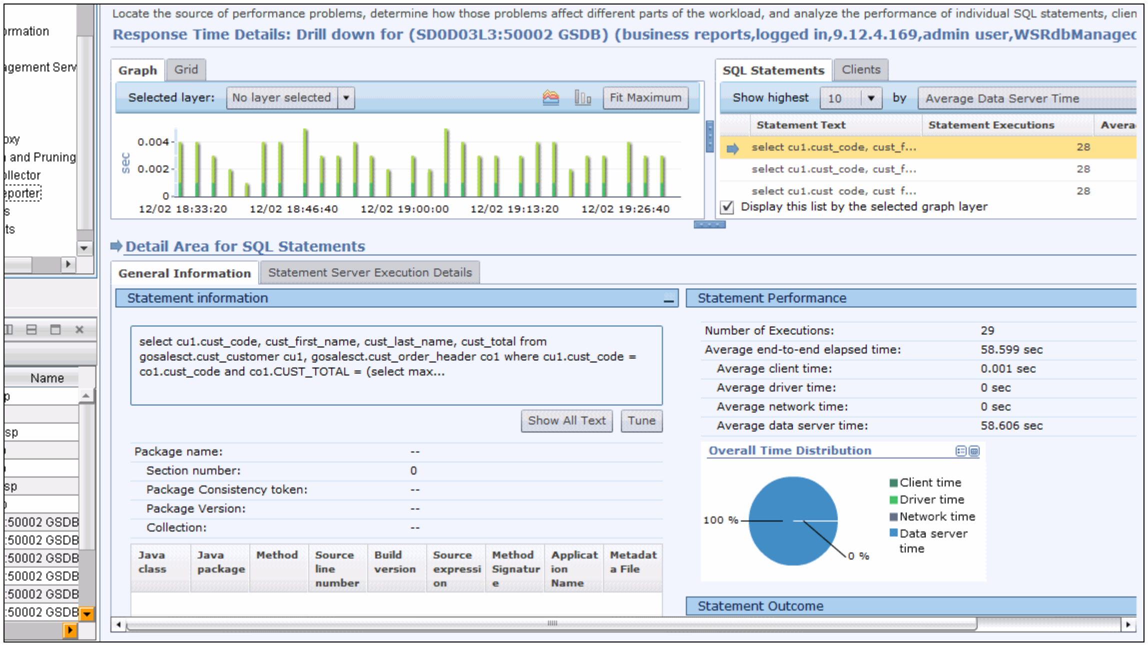


Figure 10-33 Extended Insight - General Information tab for slow statement

Statement details such as those shown in Figure 10-34, are collected only when the Extended Insight configuration has enabled the “Collect statement metrics on data server”. This configuration option is described in 3.3, “Configuring Optim Performance Manager” on page 91. If you do not have this enabled, you may still see some of these metrics on the **Active SQL** dashboard, which you can read about in 4.3.7, “Active SQL dashboard” on page 175.

The metrics `Average rows read` and `Average rows returned` show a over 76 million rows read for one selected row, which is very bad. This statement is an excellent candidate for tuning.

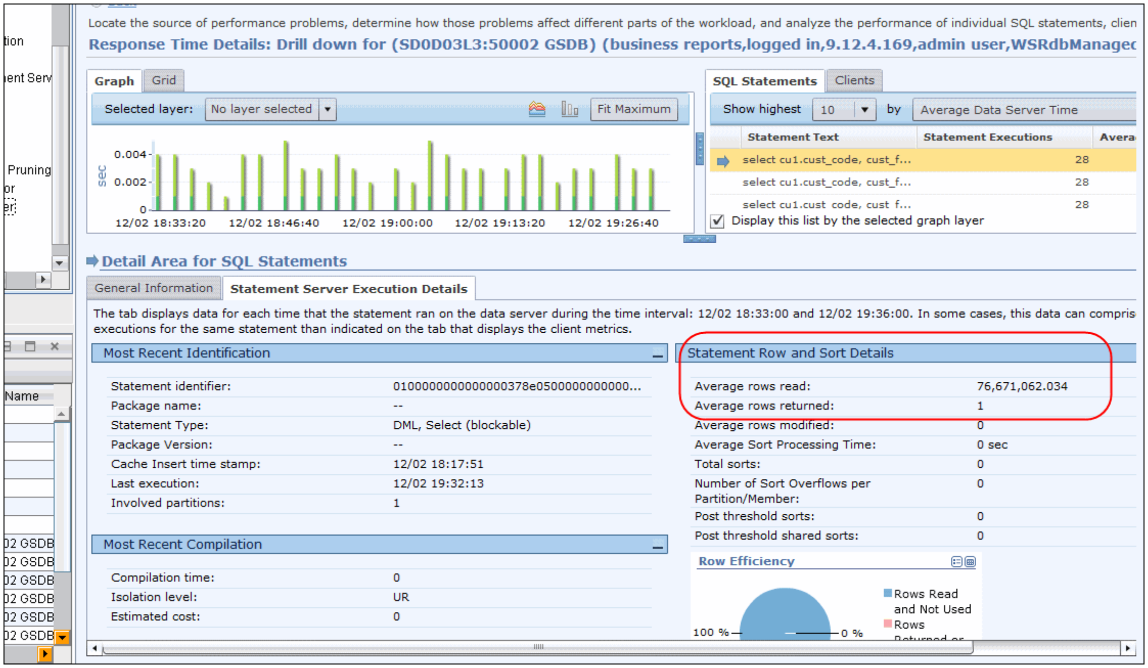


Figure 10-34 Extended Insight - Statement Server Execution Details tab for slow statement

We are quite sure now that this statement is the likely cause of the `GOSales` slowdown, but let us look at a one more screen.

Using the typical Extended Insight analysis process, take a look at the Clients tab, just to be thorough. In Figure 10-35 on page 363, the pie chart shows that 99.9% of the overall transaction time takes place in the data server. This is consistent with our previous analysis.

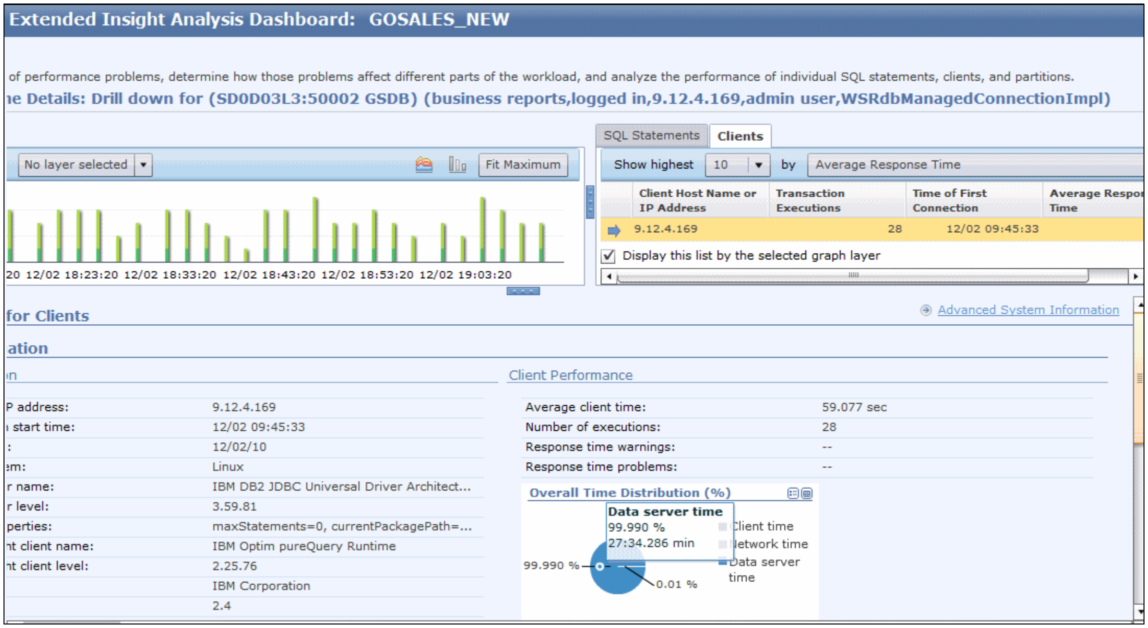


Figure 10-35 Extended Insight - Clients tab

What have we learned so far? The TEP console showed `GOSales` application with slow transactions. The ITCAM showed admin.jsp and one DB2 transaction for `business reports` accounted for the bulk of the slowness. Then in Optim Performance Manager, one statement looks like the main contributor. At this point, the operator could open a ticket with this information, routed to the application DBA. The DBA could also use Optim Performance Manager, either inside TEP or stand-alone, to investigate that statement. For example the DBA could launch to the Optim Query Tuner.

This analysis was done completely within the scope of the Extended Insight dashboards. Of course there are many more metrics available on other Optim Performance Manager dashboards that might help in the analysis as well.

10.3.3 More launch-in-context from Optim Performance Manager

We already saw how Optim Performance Manager is launched from TEP in context of the transaction.

Another area of integration between Optim Performance Manager and Tivoli monitoring is the ability to launch to the TEP workspace in the context of the data server or the client WebSphere Application Server. This integration requires that you are monitoring the data server or WebSphere server with the Tivoli monitoring OS Agent.

Launching to TEP workspaces from Optim Performance Manager

When you are looking at the Clients tab as in Figure 10-36, notice the link under the clients list, “Advanced System Information”. If you are using Optim Performance Manager stand-alone browser, the link only displays an information popup. If you are using Optim Performance Manager inside TEP, however, there is TEP awareness, so the link is “hot” and it launches back to the TEP system workspace for the client you have selected. In our example, our WebSphere server is on `9.12.4.169`, or `SD0D03L2`.

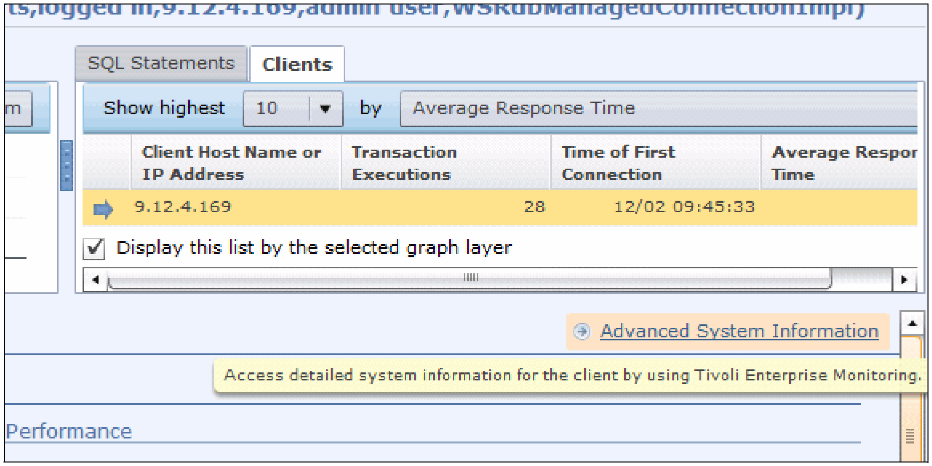


Figure 10-36 Extended Insight - Advanced System Information

Click the **Advanced System Information** link. A message appears, as in Figure 10-37 on page 365, reminding you that you are going back to TEP, and that certain conditions must exist for the link to work. You can check the box to hide the message in future.

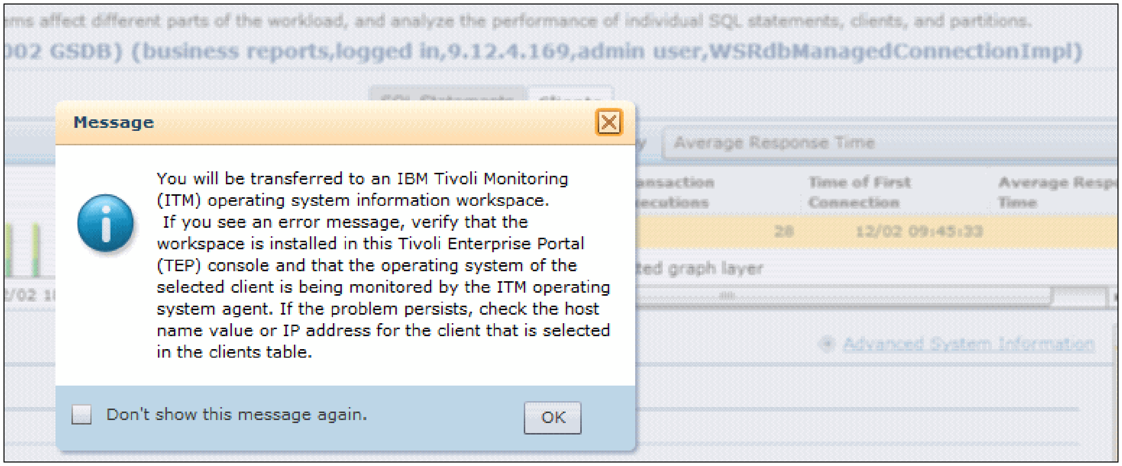


Figure 10-37 Optim Performance Manager launches to TEP, warning message

Click **OK** to continue back to TEP. You now change from the Optim Performance Manager workspace to the system workspace for the Optim Performance Manager client you selected, the WebSphere server in our example, see Figure 10-38 on page 366. In our example, the slow transactions were found to be slow on the data server side. If we had found the slowdown was on the client side, we might find it useful to use the domain expert for the operating system - the Tivoli monitoring agent in this case - to see if there were any system issues on that client.

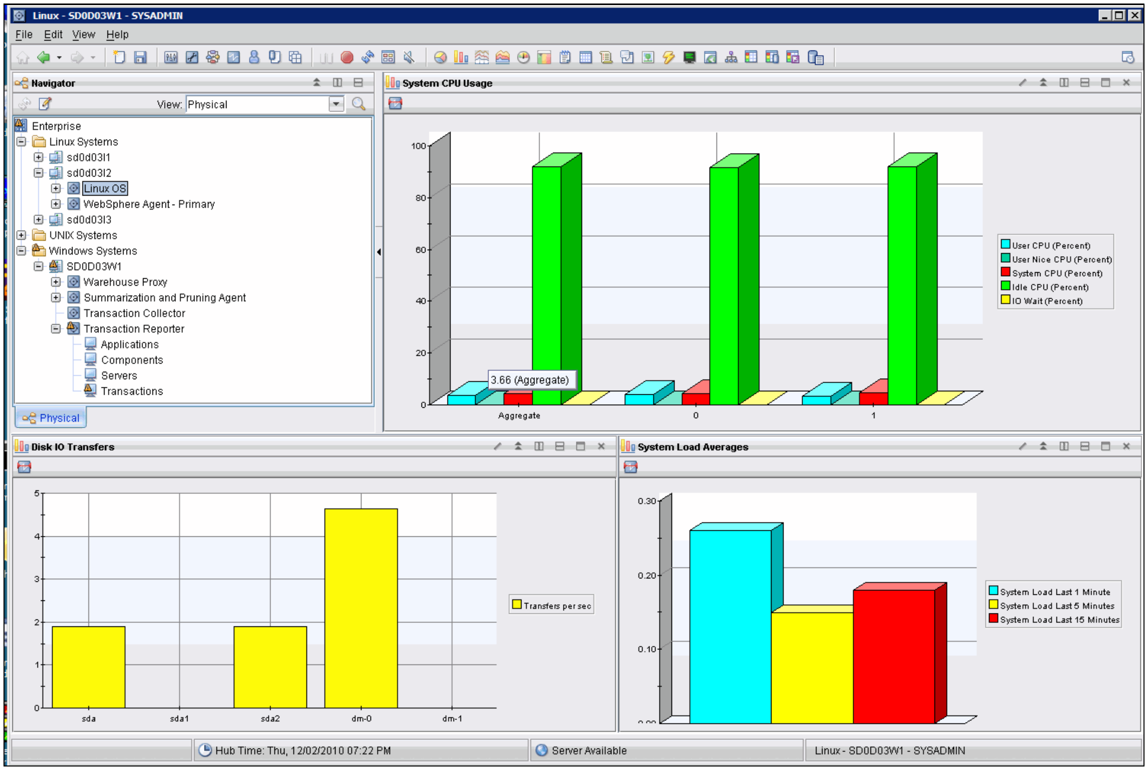


Figure 10-38 TEP workspace for WebSphere server SD0D03L2

Optim Performance Manager can also launch to the system workspace for the data server. Again, our example showed a slow statement, but suppose we suspected an overall system slowdown instead. We mentioned earlier the Optim Performance Manager inflight dashboards are available, so let us open the System Dashboard : on the Optim Performance Manager view, select **Task Manager** → **Inflight Dashboards** → **System**. The resulting dashboard is shown in Figure 10-39 on page 367.

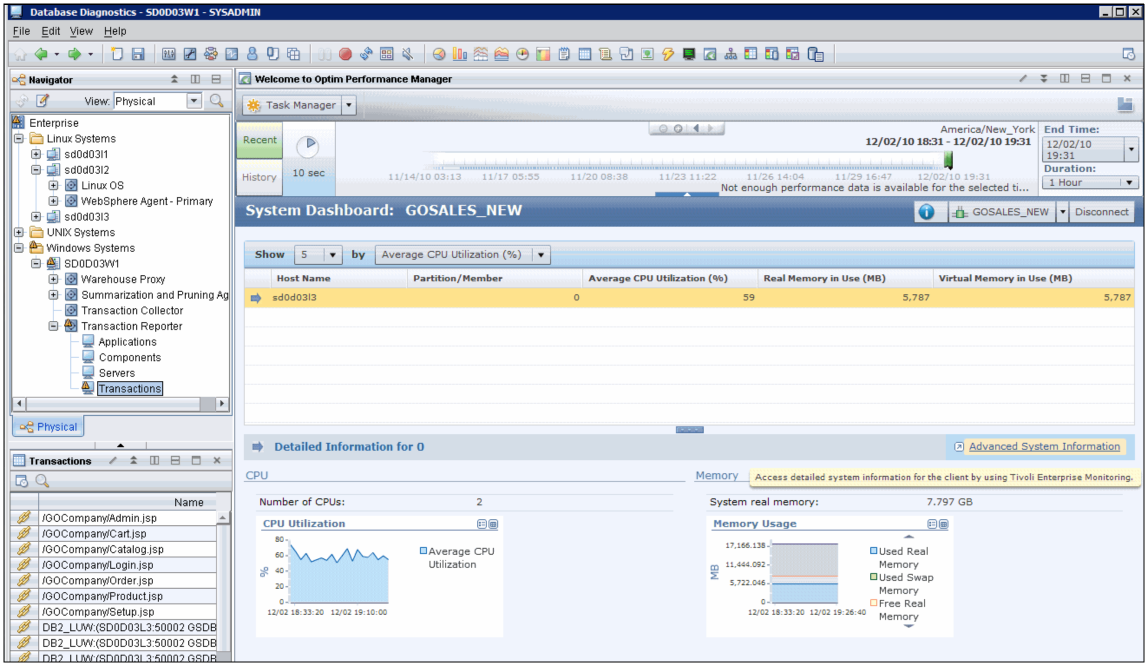


Figure 10-39 Optim Performance Manager - System dashboard inside TEP workspace

We do not see any unusual values here, however, we could still launch back to TEP's workspace for the data server. This dashboard has the same *Advanced System Information* link as we saw on the Extended Insight Clients page, and it works the same way.

Click the **Advanced System Information** link, and click **OK** on the popup message, then you see a screen similar to Figure 10-40 on page 368.

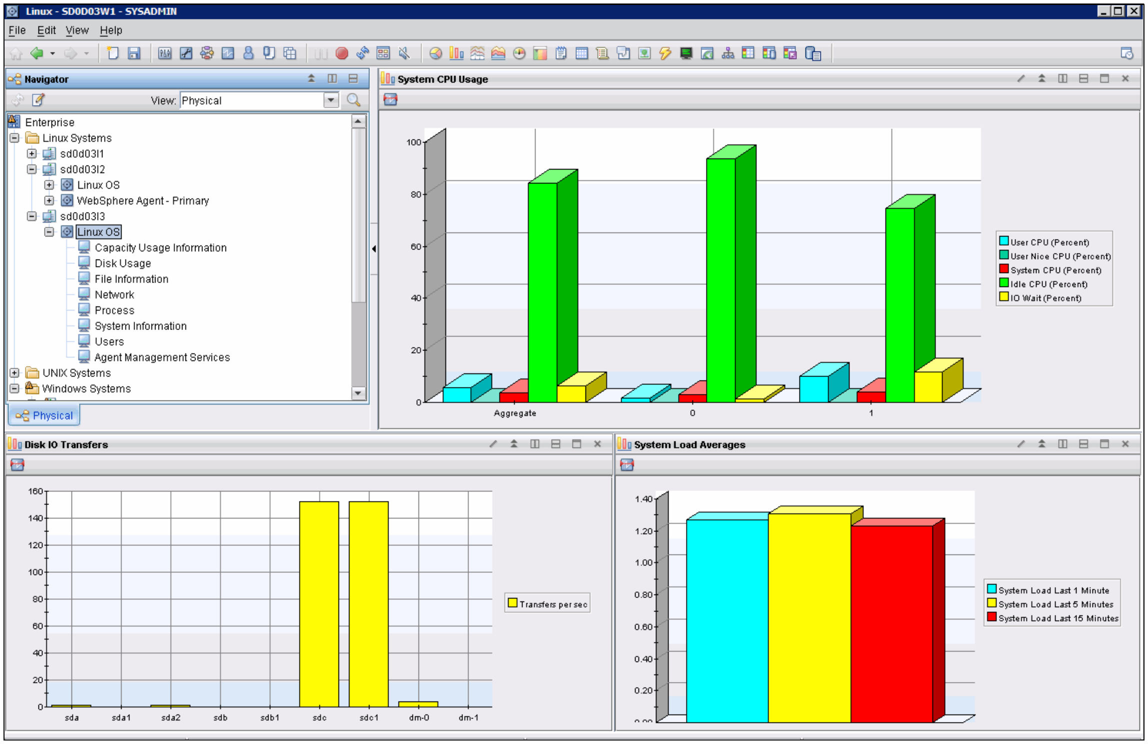


Figure 10-40 TEP workspace for DB2 server SD0D03L3

We do not see any evidence that system issues are at the root of our transaction slowdown, just as expected.

Some differences from stand-alone browser

When viewing Optim Performance Manager inside the TEP browser, there are a few differences from how it looks in a standalone browser. Some of the Task Manager menu options are not available, such as reports or connection management. You can view all the Inflight dashboards, however, as well as the alerts and health summary pages. You can also look at the ITCAM configuration. A sample Task Manager menu is shown in Figure 10-41 on page 369.

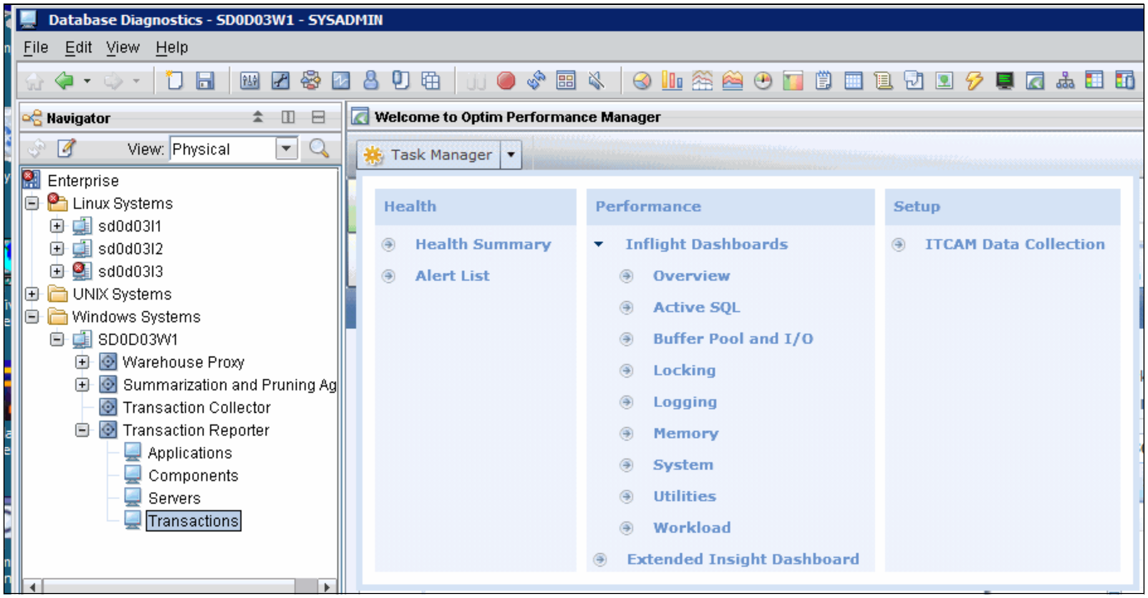


Figure 10-41 Optim Performance Manager Task Manager menu - inside TEP browser

Optim Performance Manager also has a way to quickly launch back to the ITCAM workspace. When you first come into Optim Performance Manager from TEP, you land on the Extended Insight DETAILS page for a specific transaction. If you want to look at other workload clusters, for example you must navigate back to the Extended Insight overview page. Click the **Back** link as shown in Figure 10-42.

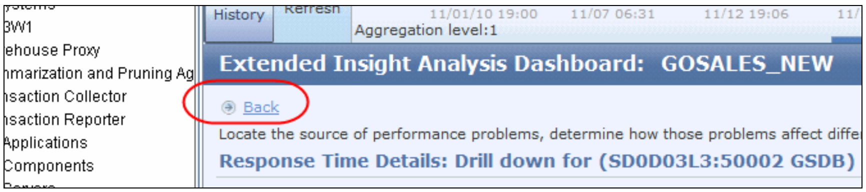


Figure 10-42 Extended Insight Details - navigate back to overview

On the Extended Insight overview page, there is a Transaction Topology button, see Figure 10-43 on page 370. This button is only functional when you are using Optim Performance Manager inside the TEP workspace. Otherwise, it just puts up an informational popup.

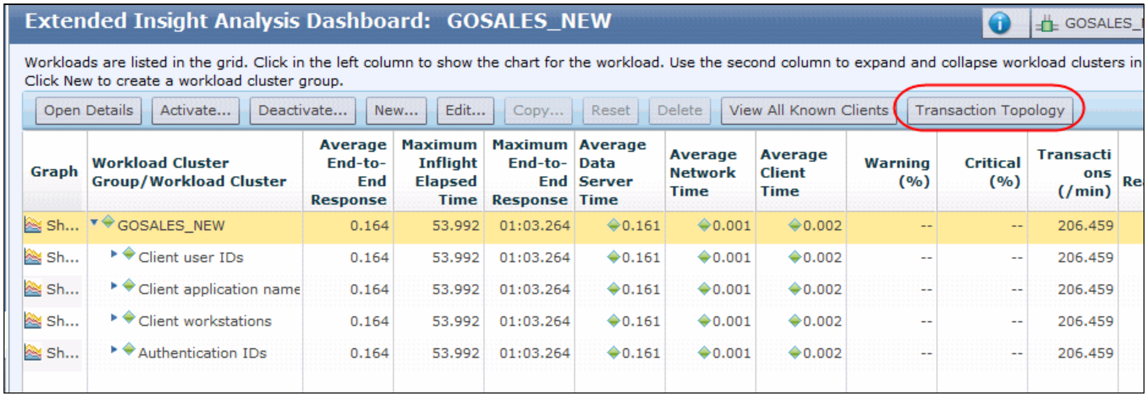


Figure 10-43 Extended Insight - Transaction Topology button

When you click the button, you will launch back to the ITCAM Transaction Reporter Transaction Topology workspace as shown in Figure 10-44 on page 371.

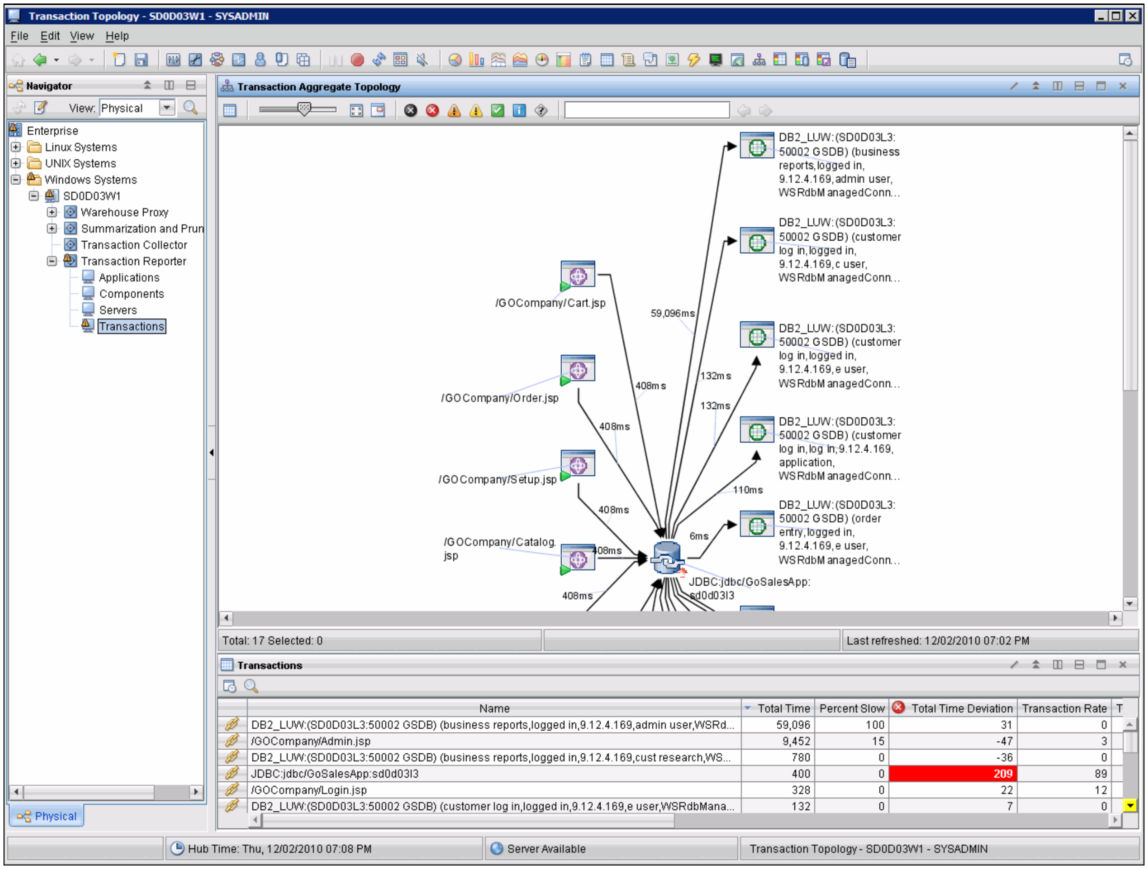


Figure 10-44 Transaction Aggregate Topology workspace

11

Workload Manager configuration tool

The Workload Manager configuration tool is a graphical user interface that you can use to configure and monitor DB2 Workload Manager V9.5 or higher. It is automatically installed with Optim Performance Manager. The Workload Manager configuration tool offers two solutions to set up DB2 Workload Manager in order to control and manage database workload:

- Concurrency method

  Concurrency solution provides following capabilities:

  – Implement business priorities by applying concurrency controls.
  – Dedicate shares of system resources to database work.
  – Manage the database activities of applications, users, groups, and more.
  – Prioritize work by categories and by estimated costs.
  – Control and stabilize response times.
  – Implement policies for disruptive work that over-consumes resources.
  – Refine policies by limiting diverse classes of work.
  – Analyze live monitoring data to report resource consumption, estimate system capacity, track adherence to response times, troubleshoot performance issues, and validate controls.

- Priority aging method

  Priority aging solution provides following capabilities:

  – Downgrade the priority of work as required automatically.
  – Set the initial priority of work by cost or category.
  – Adjust the cost limits of expensive activities.
  – Customize the runtime limits of database work.

In this chapter we document setup and deployment of a concurrency method using the Workload Manager (WLM) configuration tool.

11.1 Setting up concurrency method

The concurrency method manages resources for the database server by enforcing concurrency limits on certain types of work. The concurrency method runs urgent work without concurrency limits or restraints. The concurrency method allocates most of the concurrency tickets to ordinary work, and a lesser number of concurrency tickets to batch jobs to limit the impact on more urgent work. When the database server is busy, the concurrency method queues batch jobs so that they must wait for an opportunity run.

The concurrency method also provides control of service superclasses, response time objectives and monitoring, and support for workload management thresholds.

11.1.1 Configuring Optim Performance Manager to collect WLM statistics

Before you can use the monitoring graphs and reports of Workload Manager, you need to configure Optim Performance Manager to monitor the WLM statistics.

On the Optim Performance Manager web console, click **Manage Database Connection**, select the database for which you want to collect Workload Manager metrics and click **Configure Monitoring**. See Figure 11-1 on page 375.

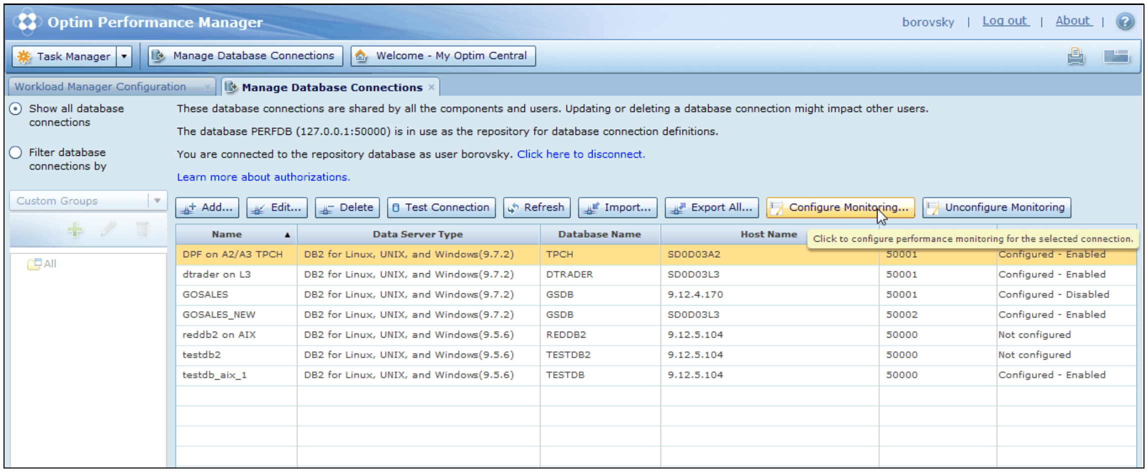


Figure 11-1 Configuring WLM monitoring for selected database

Step through the configure monitoring wizard and enable collection of WLM monitoring information as shown in Figure 11-2.

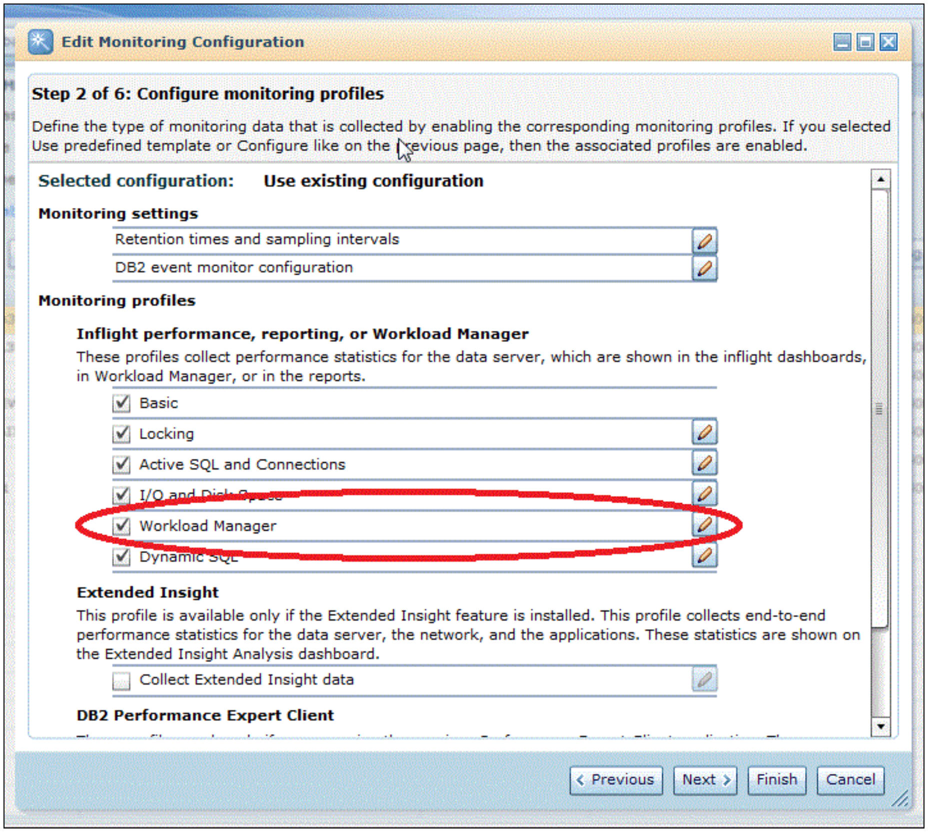


Figure 11-2 Collecting the WLM monitoring information

11.1.2 Workload Manager template configuration

After enabling the collection of WLM monitoring information for the selected database, you can proceed to Workload Manager configuration by performing the following steps:

1. From the Optim Performance Manager web console, click **Task Manager** and select **Workload Manager Configuration**. See Figure 11-3 on page 377.

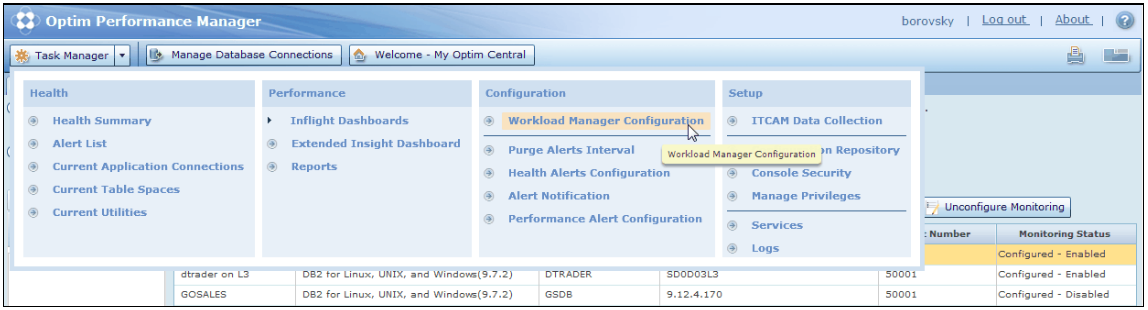


Figure 11-3 Invoking Workload Manager Configuration

2. Select the database that you want to configure and click **Connect** to connect to the database. This will take you to the WLM configuration wizard (Figure 11-4 on page 377).

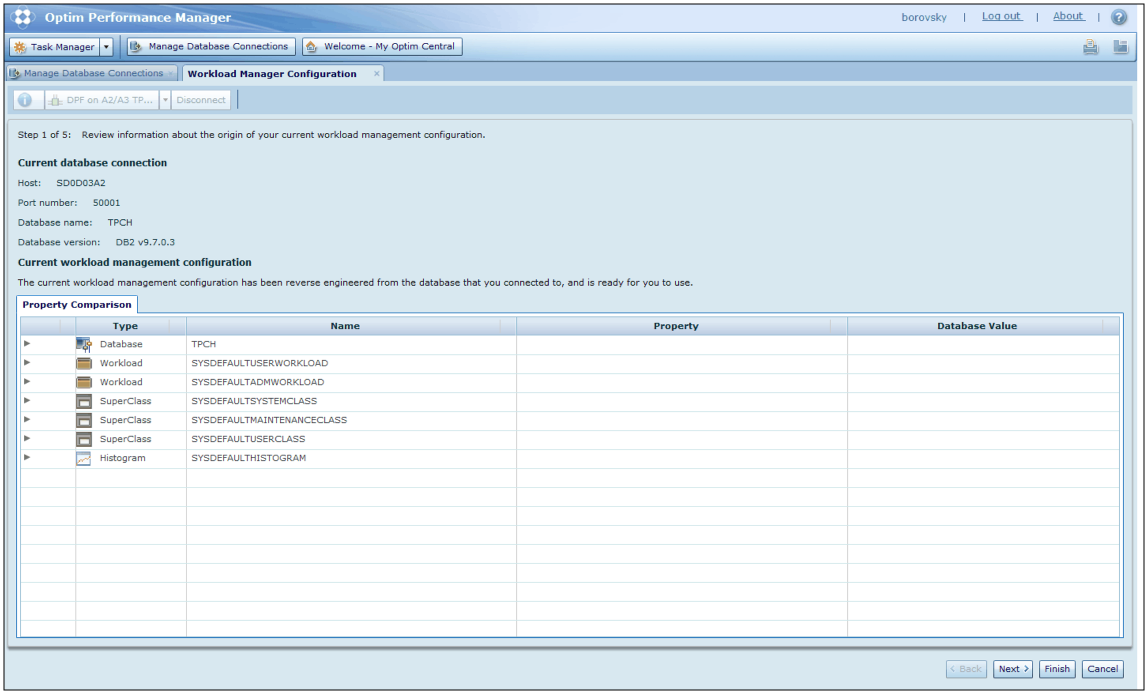


Figure 11-4 WLM configuration wizard - Step1

In this page the wizard shows you the existing WLM configuration in the database that you are connected to. If this is initial WLM configuration, there are no user defined WLM objects and you see only the default workloads and service classes.

3. On Step 2 (Figure 11-5), the wizard provides a choice of two WLM configurations: concurrency method or priority aging method. The default is the concurrency method. For our example, we accept this default.

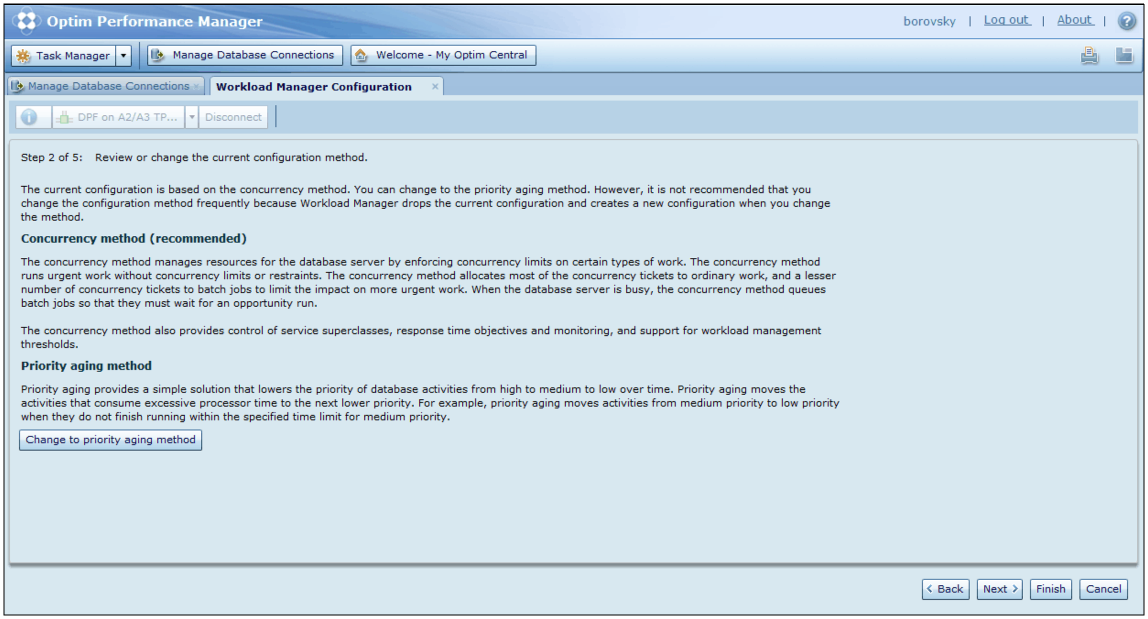


Figure 11-5 WLM configuration wizard - Step 2

4. WLM configuration wizard constructs a template WLM configuration, which contains a service super class, several subclasses, and related WLM infrastructure. Step 3 of the wizard (Figure 11-6 on page 379) displays the current WLM database configuration as well as the new template WLM configuration. It also highlights the differences between these two configurations.

> **Note:** No changes are made to the database until you choose to deploy this template WLM configuration.

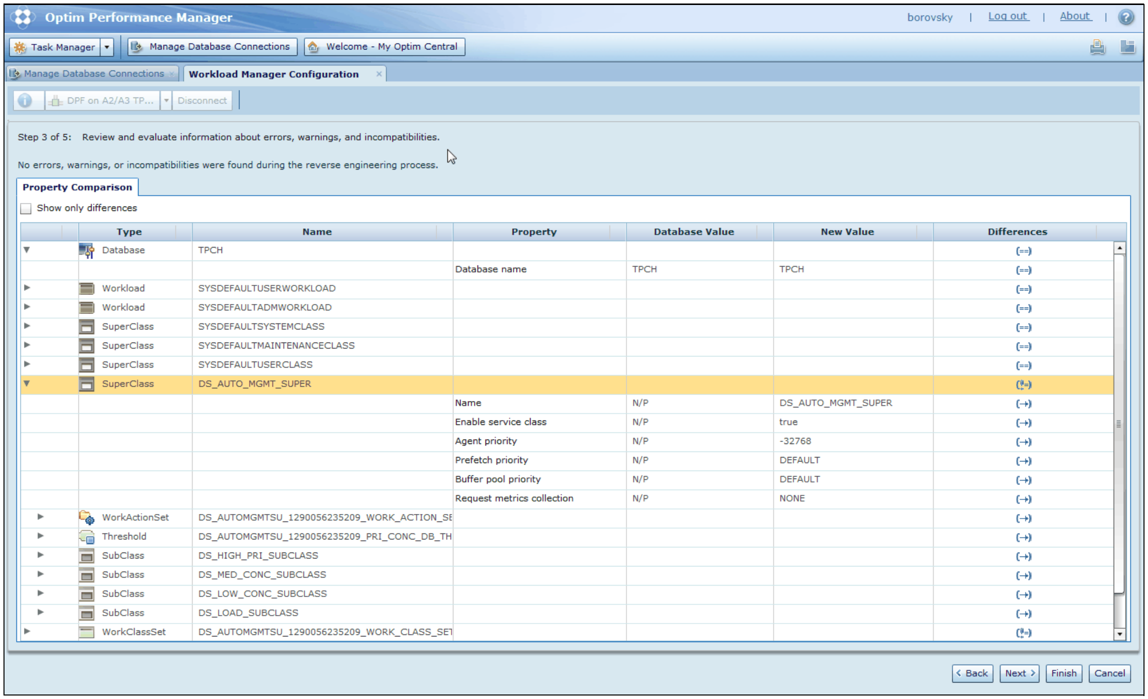


Figure 11-6 WLM configuration - Step 3

5. Step 4 of the wizard (Figure 11-7 on page 379) shows a few key points about your WLM configuration. It calls attention to issues such as disabling access to the database for some of your workloads.

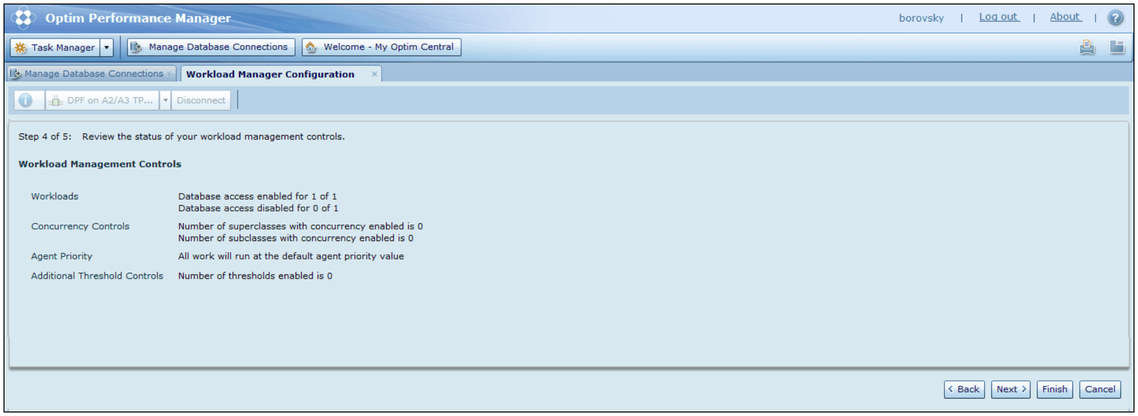


Figure 11-7 WLM configuration - Step 4

6. The final step of the wizard (Figure 11-8 on page 380) shows a summary of the state of your WLM configuration. It also checks for several common setup problems that will prevent WLM monitoring from working. If the wizard detects any such problems, it will tell you what is wrong and offer help on how to correct it.

   Click **Finish** to complete the wizard.

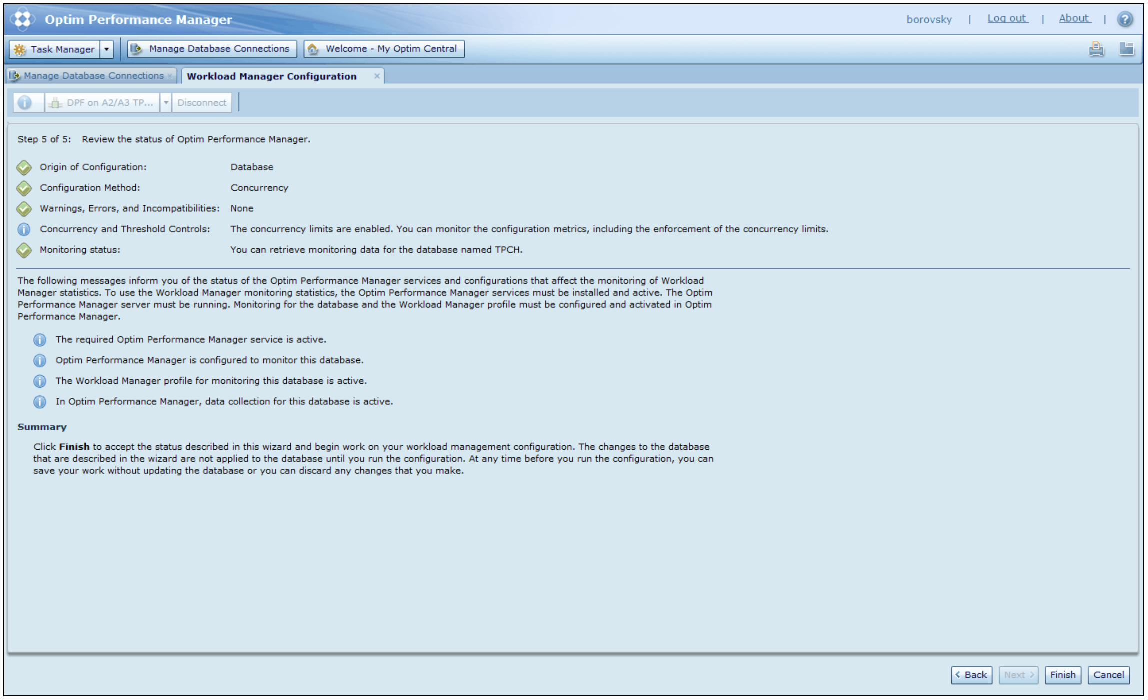


Figure 11-8 WLM configuration - Step 5

7. Workload Manager tool takes you to the main configuration and monitoring panel (Figure 11-9 on page 381), where you can further customize your WLM configuration as well as deploy it to the database.

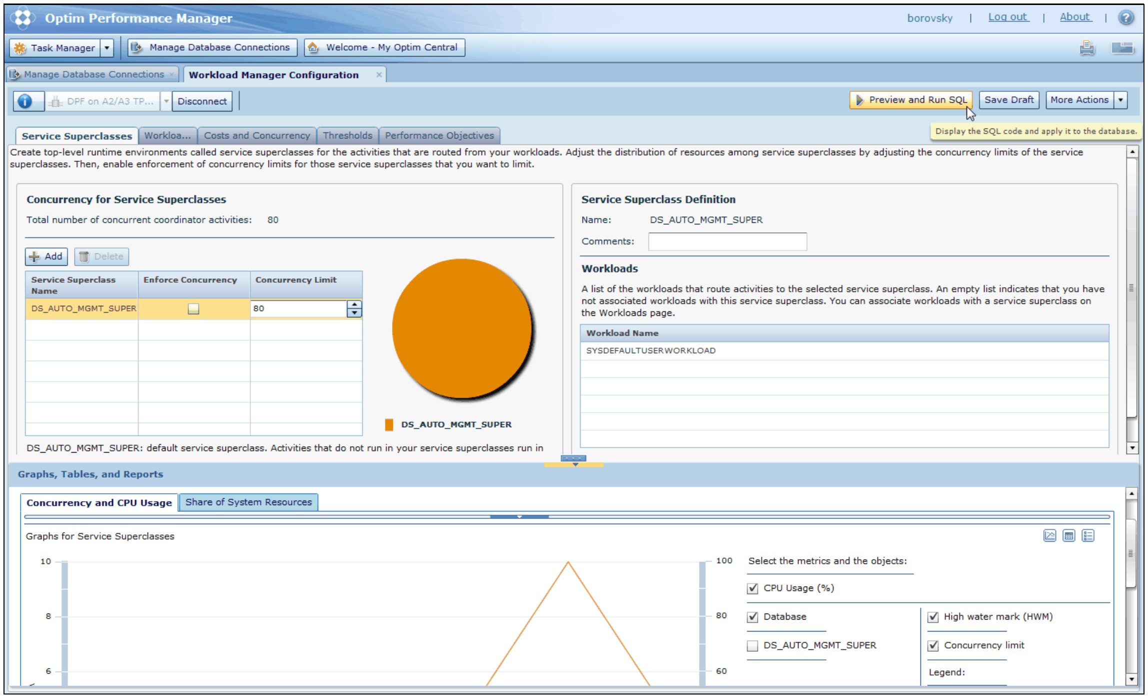


Figure 11-9 WLM configuration and monitoring panel

8. To deploy WLM configuration to the database, click **Preview and Run SQL**. This will generate and display the DDL necessary to deploy the WLM configuration to the database. See Figure 11-10 on page 382. You can choose to let the WLM tool to run it for you, or copy and paste the generated DDL to inspect it and run it yourself.

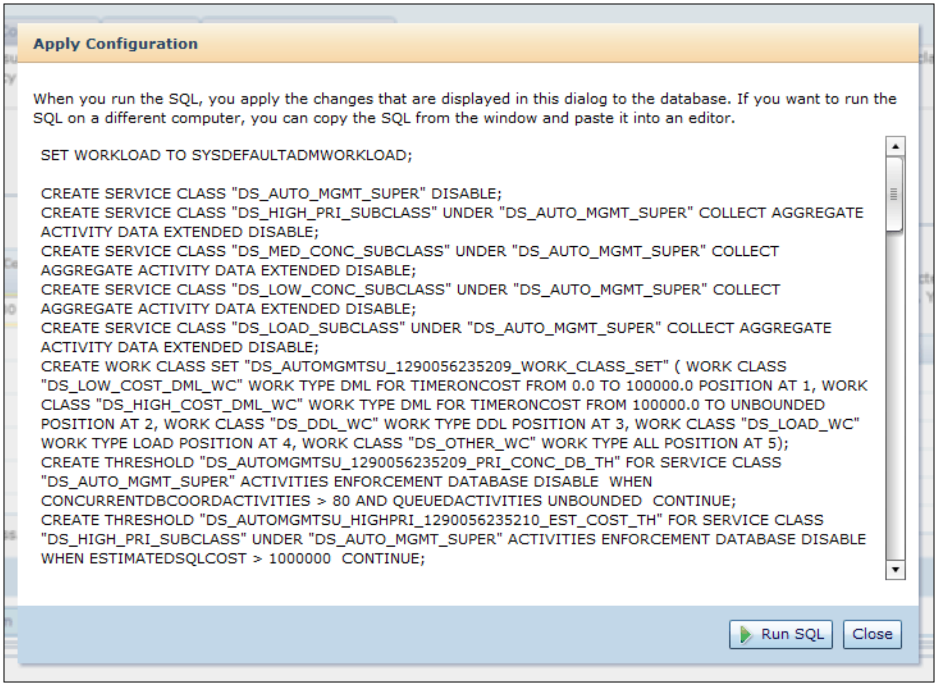


Figure 11-10 Deploy WLM configuration to the database

> **Note:** This initial configuration is intentionally structured for monitoring only. That is, all of the thresholds are disabled and all of the service classes are configured to use the default agent priority. Deploying it will not impact the performance of your database. Rather it will begin categorizing and monitoring work.

11.1.3 Workload Manager objects in concurrency solution

When you configure a concurrency solution, you work with the following set of WLM database objects. Workload Manager provides default objects in the template configuration, and you can also create your own objects.

Service superclasses

A service superclass is a top-level runtime environment that you create to represent a business entity. You can distribute the total concurrency of the database among the runtime environments by defining a concurrency limit for each service superclass. `DS_AUTO_MGMT_SUPER` is the name of the default service superclass.

Workloads

A workload is a category that you create to identify database activities by connection attributes that represent groups of users, important applications, IP addresses, and more. SYSDEFAULTUSERWORKLOAD is the name of the default workload.

Service subclasses

A service subclass is a second-level runtime environment that you create in a service superclass. You can distribute the concurrency of the service superclass among the service subclasses of the service superclass. The system automatically creates the following default service subclasses in every service superclass: DS_HIGH_PRI_SUBCLASS, DS_MED_CONC_SUBCLASS, DS_LOW_CONC_SUBCLASS, and DS_LOAD_SUBCLASS.

Thresholds

A threshold is an additional control that you can apply to the work that runs in the service subclasses. You can configure thresholds that enforce time limits, limit the number of rows read and returned, and control temporary table space usage. You can configure thresholds to stop database activities from running if they exceed a threshold limit, and to monitor database activities that exceed a threshold limit.

11.2 Customizing concurrency method

After deployment of WLM template configuration, you can use the Workload Manager configuration tool to further customize WLM objects - service superclasses, workloads, service subclasses, and thresholds.

11.2.1 Customizing service superclasses

On the Service Superclasses page of Workload Manager configuration tool, you create the top-level runtime environments that can represent applications, lines of business, departments, divisions, and other important business entities.

When you add or modify a service superclass, you work with the following objects and attributes:

- *Default service superclass*

  The system creates the default service superclass (DS_AUTO_MGMT_SUPER) and the default service subclasses in it. Initially, the default workload (SYSDEFAULTUSERWORKLOAD) routes activities to the default service superclass.

Initially, the activities of the default workload run in the default service subclasses according to the concurrency or priority setting of the default workload.

- *Total number of concurrent coordinator activities*

  The total number of concurrent coordinator activities is the total concurrency limit of the database. This total is a system-defined, read-only number that you can adjust by adding service superclasses and adjusting the concurrency limits of the service superclasses. By adding service superclasses and adjusting the concurrency limit of each service superclass, you divide the total concurrency limit among all the service superclasses, including the default service superclass. When you adjust the concurrency limits of the service superclasses, the system automatically adjusts the total concurrency of the database to be equal to the total of the concurrency limits of all the service superclasses.

- *Enable enforcement of a concurrency limit*

  Initially, concurrency limits of service superclasses are disabled. You can enable enforcement of concurrency limits for individual service superclasses.

  To do baseline monitoring before you configure Workload Manager, you can run the default configuration with the concurrency limits of all service superclasses disabled. When you are ready to monitor the effects of applying concurrency limits to your service superclasses and service subclasses, enable the concurrency limits of service superclasses as necessary.

- *Workloads*

  A list of the workloads that route activities to the selected service superclass. You can associate workloads with a service superclass on the Workloads page.

In our example we set the concurrency limit of the DS_AUTO_MGMT_SUPER service superclass to 20, and leave the enforcement of concurrency disabled, as shown in Figure 11-11 on page 385.

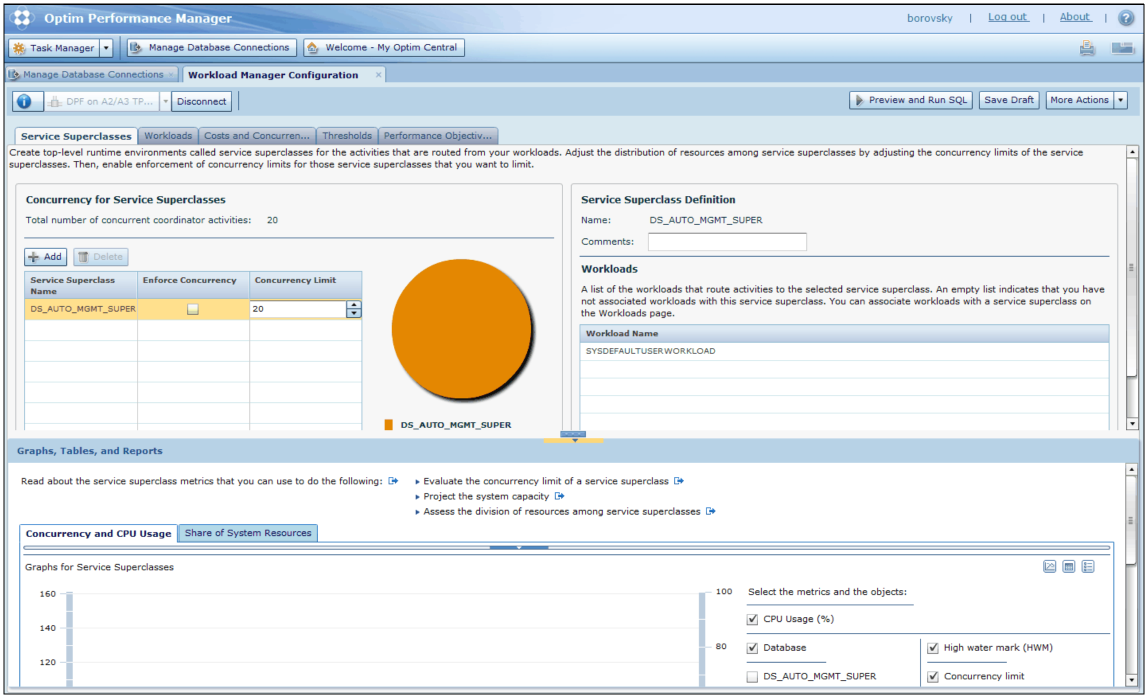


Figure 11-11 Service superclasses page

11.2.2 Customizing workloads

On the Workloads page of Workload Manager configuration tool, you can group similar sources of database activities into categories called workloads. Workloads represent who or what is connecting to the database and submitting requests.

When you add or modify a workload, you work with the following objects and attributes:

- *Default workload*

  When the database is created during the installation, the installation also creates the default workload (`SYSDEFAULTUSERWORKLOAD`). When database connections do not match the connection attributes of your workloads, the system assigns the connections to the default workload. You cannot delete the default workload.

  The concurrency of the activities that are assigned to the default workload is determined by the estimated SQL cost of the activities. The activities that are assigned to the default workload initially run in the default service superclass

(DS_AUTO_MGMT_SUPER). You can run the activities in one of your service superclasses by changing the related service superclass of the default workload.

- *Workload evaluation*

  The database server assigns a request to the first workload in the workload evaluation table that has the same connection attributes as the request. For example, the database server selects a matching workload in the second row of the table over a matching workload in the third row.

  The workload evaluation table displays all the workloads in the current evaluation order. When you create a new workload, the workload is automatically positioned after all the user-defined workloads in the table, but before the default workload, which is always in the last position. You can change the evaluation order of your workloads by changing the value in the Evaluation position field or by moving the workload up or down in the table.

- *Related service superclass*

  You specify a top-level runtime environment for the activities of a workload by selecting the related service superclass. By default, the activities that are assigned to a new workload run in the default service superclass (DS_AUTO_MGMT_SUPER) until you change the related service superclass of the workload.

- *Enable database access*

  The system can assign activities to a workload only when the database access to the workload is enabled.

- *Connection attributes*

  When you create a workload, you define the connection attributes of the workload to identify the activities that are assigned to the workload. For a new workload, you must define at least one value for one connection attribute. Otherwise, the system cannot assign activities to the workload.

  You can define values for the following connection attributes:

  – Application name

    The name of the application that the data server recognizes and that is running on the client.

  – User ID

    The authorization ID of the user who connects to the database. This ID is set in the SYSTEM_USER special register.

  – Session user ID

    The authorization ID of the current session of the application. This ID is set in the SESSION_USER special register.

- Group ID

  The groups to which the current session user belongs.
- Role ID

  The roles that are granted to the current session user.
- Client user ID

  The client user ID from the client information in the `CURRENT CLIENT_USERID` (or `CLIENT USERID`) special register.
- Client application name

  The application name from the client information in the `CURRENT CLIENT_APPLNAME` (or `CLIENT APPLNAME`) special register.
- Client workstation

  The workstation name from the client information in the CURRENT `CLIENT_WRKSTNNAME` (or `CLIENT WRKSTNNAME`) special register.
- Client accounting string

  The accounting string from the client information in the CURRENT `CLIENT_ACCTNG` (or `CLIENT ACCTNG`) special register.
- IP address (DB2 Version 9.7 and higher)

  The address that the client uses to communicate with the database server. TCP/IP is the only supported protocol for the address. The address must be an IP Version 4 address, an IP Version 6 address, or a secure domain name.

- *Concurrency/priority*

  To specify how concurrency or priority is determined for the activities of a workload, select an option that corresponds to one of the following types of work:

  - Urgent work

    Specify the `DS_HIGH_PRI_SUBCLASS` option for urgent work. Use this option sparingly for high-priority work and rush jobs that start to run immediately, run to completion at the high priority, and never wait in a queue. The activities of high-priority workloads run without a concurrency limit.
  - Ordinary work

    Specify the `DS_MED_CONC_SUBCLASS` option for everyday work and queries. The system distributes a majority of the concurrency tickets to the `DS_MED_CONC_SUBCLASS` service subclass.
  - Batch jobs

Specify the DS_LOW_CONC_SUBCLASS option for work that takes a long time to run, work that can disrupt urgent work, and work that can acceptably wait in a queue. The system distributes fewer concurrency tickets to the DS_LOW_CONC_SUBCLASS service subclass than it does to the DS_MED_CONC_SUBCLASS service subclass.

– Jobs with variable SQL cost

Specify the *Estimated SQL cost* option for the cost of activities to determine the concurrency limit at which they run. When the concurrency of activities is determined by cost, the activities are routed to the default service superclass or to your service superclasses. Depending on the costs of the activities, the activities run in the DS_MED_CONC_SUBCLASS, the DS_LOW_CONC_SUBCLASS, or a performance objective subclass. Any LOAD type activities of the workloads that determine concurrency by cost automatically run in the DS_LOAD_SUBCLASS.

– Jobs with performance objectives

Performance objectives will be discussed later in this chapter.

For help with defining the connection attributes of a workload, you can open the View Current Activities report (Figure 11-12 on page 389) from the Workloads page and see the values that are currently in use.

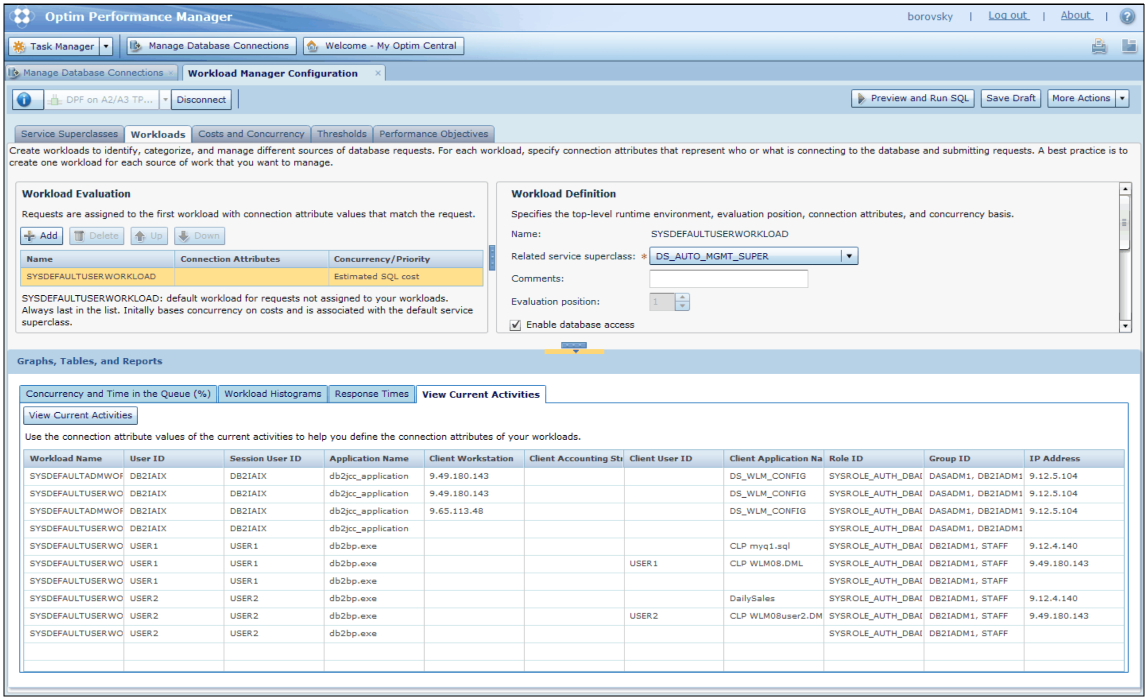


Figure 11-12 View current activities report

In our example, based on the current activities report, we create two additional workloads, USER1_WL and USER2_WL. They are based on the value of the User ID connection attribute, such that USER1_WL will contain database work from USER1 and USER2_WL will contain database work from USER2.

To add a workload, go to the Workloads page click **+Add** and specify the name of the workload. See Figure 11-13.

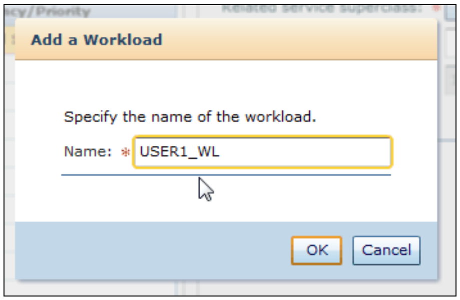


Figure 11-13 Add workload panel

In the connection attributes section of the Workloads page, select the User ID as a connection attribute and click **Add**, as shown in Figure 11-14.

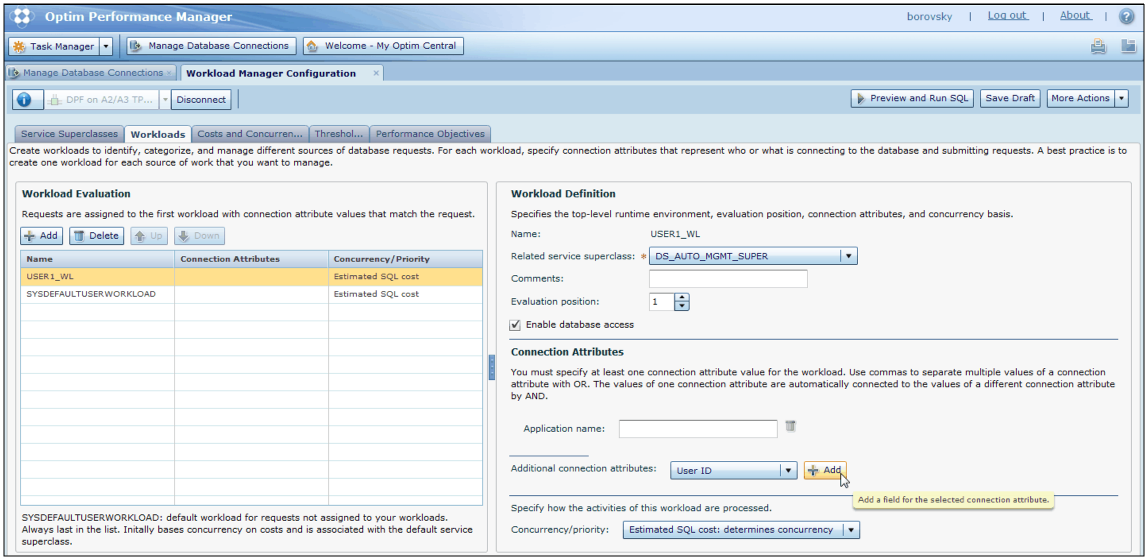


Figure 11-14 Add connection attribute

Specify the value of the User ID attribute, `USER1` in our example, as well as the Concurrency/Priority selection. We use **Estimated SQL cost: determines concurrency** value, which means that workloads will be routed into service subclasses based on the SQL cost value of individual statements. See Figure 11-15 on page 391. To apply changes press **Enter** key.

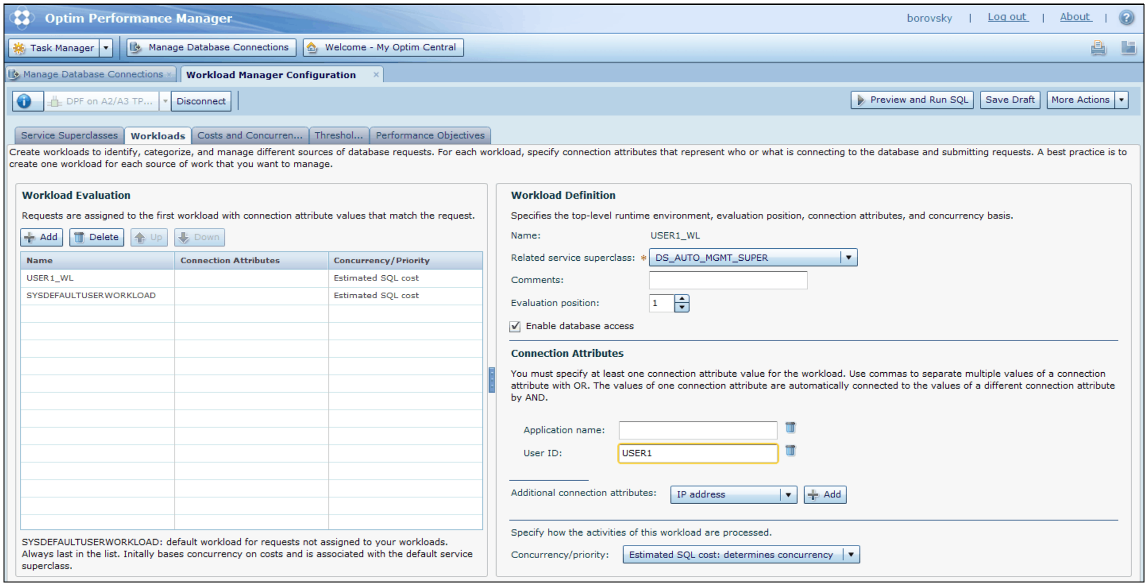


Figure 11-15 Specify the value of the connection attribute

Similarly, add `USER2_WL`, and specify `USER2` as a value of the User ID connection attribute, as shown in Figure 11-16 on page 391.

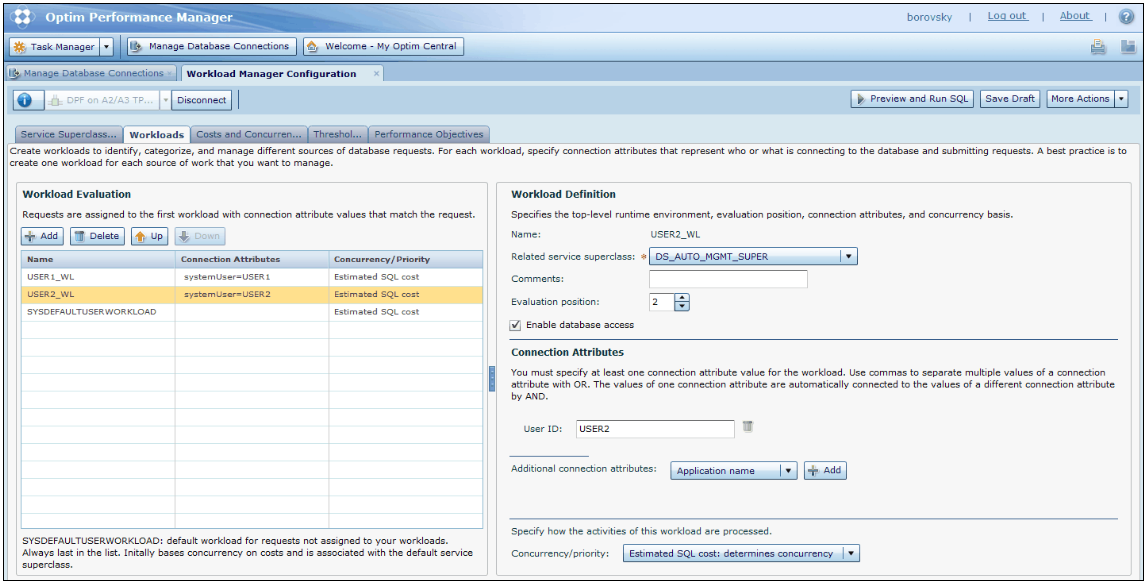


Figure 11-16 Workloads page with custom defined workloads

11.2.3 Customizing service subclasses - costs and concurrency

On the Costs and Concurrency page of Workload Manager configuration tool, you can create service subclasses, which are the second-level runtime environments that you create to distribute the concurrency of your service superclasses.

The system creates the default service subclasses in every service superclass. The system also creates special service subclasses for the activities that run according to target response times.

Figure 11-17 on page 393 shows examples of a how the system routes activities to the service subclasses where they run depending on the priority of the workload, the concurrency of the workload, or the activity type, such as LOAD activities.

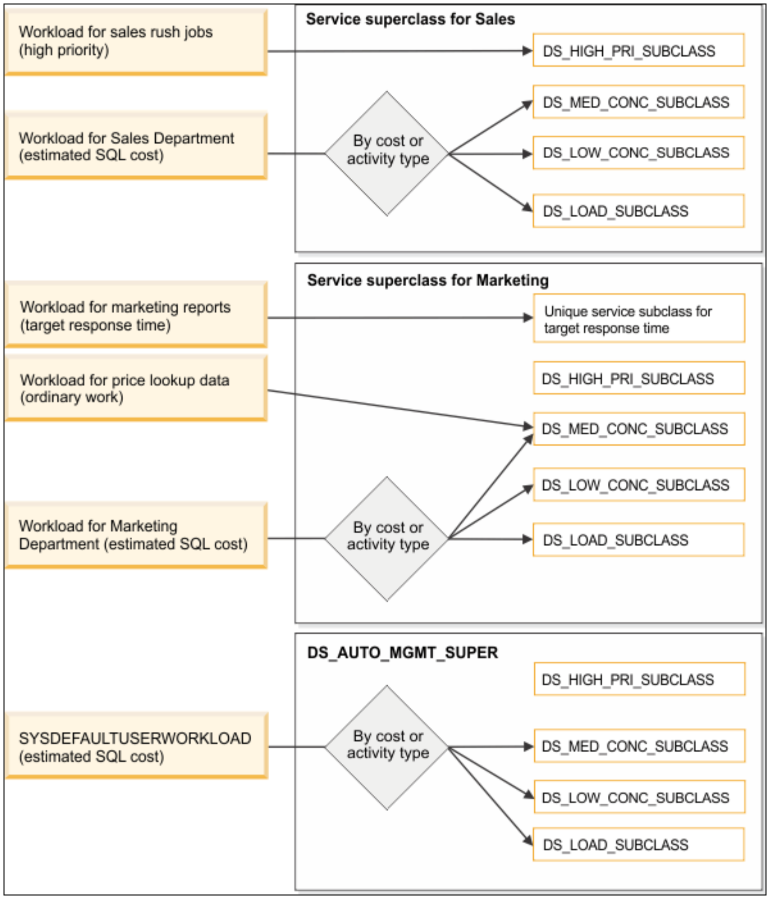


Figure 11-17 Sample diagram of how system routes activities to service subclasses

When you add or modify a service subclass, you work with the following objects and attributes:

- *Default service subclasses*

  The system automatically creates the following default service subclasses in every service superclass. The concurrency limit of a service superclass is distributed among the medium concurrency default service subclass (`DS_MED_CONC_SUBCLASS`), the low concurrency default service subclass (`DS_LOW_CONC_SUBCLASS`), the default service subclass for load activities (`DS_LOAD_SUBCLASS`), and any service subclasses that you create. You cannot delete the default service subclasses.

  For each service superclass, the set of default service subclasses consists of the following:

- DS_HIGH_PRI_SUBCLASS

  Only the activities of the high-priority workloads run in this service subclass. The system routes high-priority activities directly from the workload to this service subclass. The system does not apply a concurrency limit to the activities that run in this service subclass. The agent priority of this service subclass is set to the default value.

- DS_MED_CONC_SUBCLASS

  The system routes ordinary work, which includes most of the database activities, directly from the workload to this service subclass, where it runs. For the activities of workloads that base concurrency on the estimated SQL cost, the system routes only the low-cost activities to this service subclass, where they run. The system applies a medium concurrency limit and the default agent priority value to this service subclass.

- DS_LOW_CONC_SUBCLASS

  This service subclass limits the impact of long-running activities, disruptive activities, and batch jobs. The system routes batch jobs and low-priority work directly to this service subclass, where the activities run. For the activities of workloads that base concurrency on the estimated SQL cost, the system routes only the high-cost activities to this service subclass, where they run. The system applies a low concurrency limit and the default agent priority value to this service subclass.

- DS_LOAD_SUBCLASS

  All the LOAD activities from the workloads that process activities by the estimated SQL cost automatically run in this service subclass. The system applies a low concurrency limit and the default agent priority value to this service subclass.

- *Service subclasses for performance objectives*

  When you specify a performance objective for the activities of a workload, you create an additional, unique service subclass in which the activities run. We will discuss this later in the chapter

- *Limits for the service subclasses of a service superclass*

  The table of service subclasses displays the concurrency and priority limits that apply to the activities that run in each service subclass of the selected service superclass. When you add a new service subclass, the system automatically redistributes the concurrency of the parent service superclass among the child service subclasses.

  - Minimum and maximum cost in timerons

When the concurrency of the activities that run in the service subclass is determined by the estimated SQL cost, you can define the maximum and minimum costs in timerons.

– Agent Priority

  The Agent Priority property helps to determine the priority at which activities run. The agent priority value specifies the priority of all the agents that work in the service subclass relative to the priority of all the other DB2 agents. UNIX and Linux values range from -20 to 20, where a negative value denotes a higher relative priority than a positive value. Windows values range from -6 to 6, where a negative value denotes a lower relative priority than a positive value.

– Prefetch priority

  The Prefetch Priority property controls the priority with which the agents in the service superclass submit prefetch requests. Prefetchers empty the priority queues in order from high to low. The default value for a service superclass is the medium prefetch queue.

> **Note:** A service subclass inherits the prefetch priority from the parent service superclass when the parent value is set to the default value, which is MEDIUM. Select the high prefetch queue infrequently to allow requests with a lower priority to be submitted.

– Buffer pool priority

  The Buffer Pool Priority property influences the proportion of pages in the buffer pool that can be occupied by activities in the service subclass, which can improve the throughput and performance of activities in that service subclass.

– Concurrency limit

  The system maintains the concurrency limits of the child service subclasses to equal the concurrency limit of the parent service superclass. You can adjust the concurrency limits of the child service subclasses within the concurrency limit of the parent service superclass. To adjust the amount of concurrency that is allocated to the child service subclasses, you must adjust the concurrency limit of the parent service superclass.

– Performance objective

  We discuss performance objectives later in this chapter.

In our example, we change the maximum SQL cost limit for the `DS_MED_CONC_SUBCLASS` to 10000 as well as increase its agent priority value to -10. We apply concurrency limit of 14 for this service subclass.

`DS_LOW_CONC_SUBCLASS` will receive workload with the SQL cost higher then 10000. We apply concurrency limit of 4 for this service subclass. See Figure 11-18.

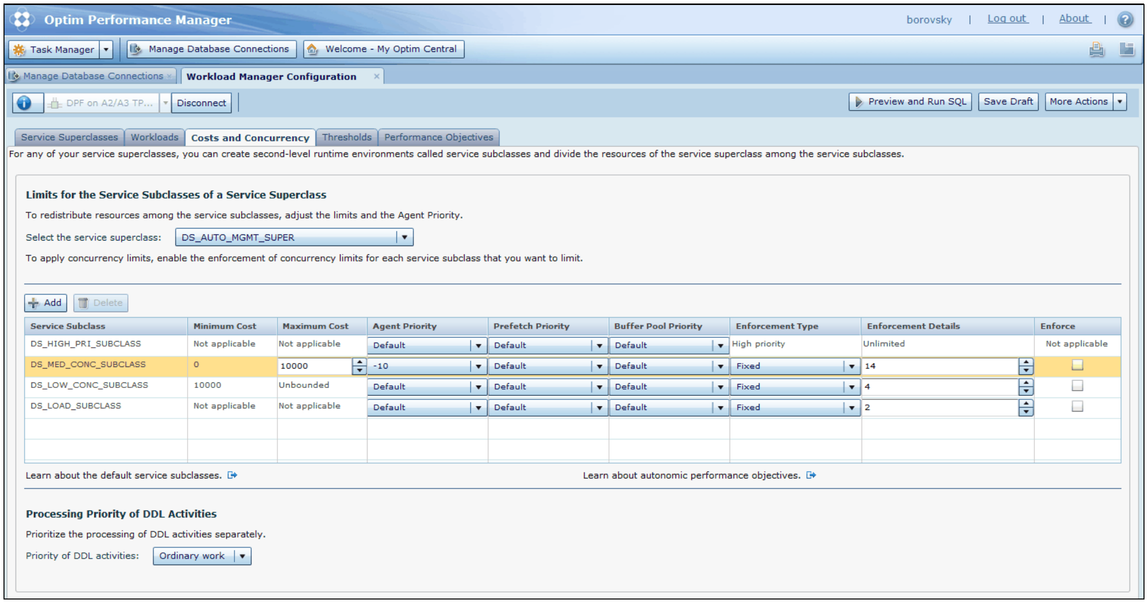


Figure 11-18 Costs and concurrency page

11.2.4 Customizing thresholds

On the Thresholds page of Workload Manager configuration tool, you can apply controls to your service subclasses in addition to the concurrency and priority controls.

When you configure a threshold for a service subclass, you work with the following objects and attributes:

- *Threshold limit*

  The threshold limit is the enforcement point at which any actions that you enable begin to occur. You can enable either or both of the following actions:

  – Stop activities that exceed the limit

  When the threshold limit is reached, this action stops any additional activities from running and affecting the system.

  – Monitor activities that exceed the limit

  When the threshold limit is reached, this action monitors the statistics of any additional activities.

- *Enable threshold*

  The system enforces threshold limits and applies the stop and monitor actions only when a threshold is enabled.

- *Threshold types*

  In addition to the system-defined threshold that enforces concurrency, Workload Manager supports the following threshold types for service subclasses:

  – Activity total time (`ACTIVITYTOTALTIME`)

    This threshold specifies the maximum amount of time that the data server can spend processing an activity. The threshold is enforced on the coordinator and nested activities of the database. The total time that is measured by the threshold includes the time that is spent in a queue.

  – Estimated SQL cost (`ESTIMATEDSQLCOST`)

    This threshold specifies the maximum estimated cost in timerons for DML activities that are issued at the coordinator partition. This threshold is enforced on the database.

  – CPU time (`CPUTIME`)

    (DB2 V9.7.x only) This threshold specifies the maximum amount of combined user and system processor time that an activity can use on a particular database partition while the activity is running. This threshold is enforced on a database partition. This threshold tracks DML and CALL activities.

  – SQL rows returned (`SQLROWSRETURNED`)

    This threshold specifies the maximum number of rows that the database server can return to the client. This threshold is enforced on the database.

  – SQL rows read (`SQLROWSREAD`)

    (DB2 9.7.x only) This threshold specifies the maximum number of rows that a DML activity can read on a database partition, and is enforced on the database partition. This threshold controls the maximum number of rows that are read during query evaluation.

  – SQL temp space (`SQLTEMPSPACE`)

    This threshold specifies the maximum amount of system temporary table space that a DML activity can consume at any database partition. This threshold is enforced on the database partition.

For our example we accept default settings of thresholds as shown in Figure 11-19.

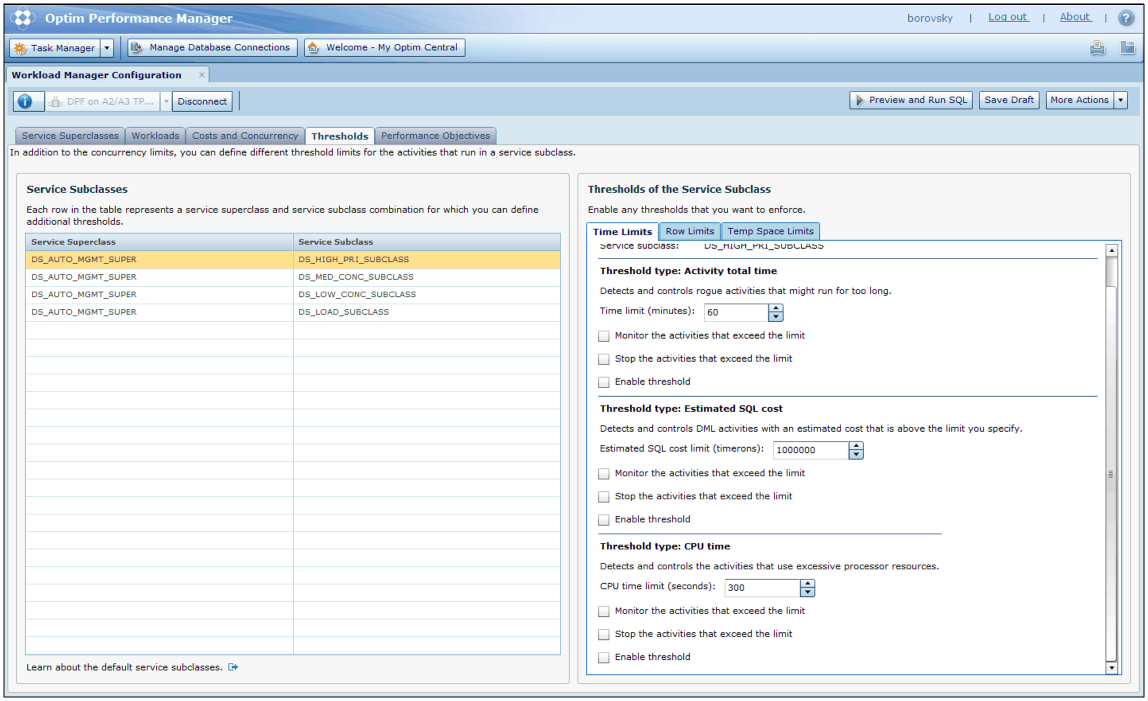


Figure 11-19 Thresholds page

11.2.5 Deploying customized WLM configuration

After having completed the customization of WLM objects in Workload Manager configuration tool, you can deploy these changes into database by clicking **Preview and Run SQL**. You can review the list of DDL statements, which modify WLM objects based on the previous customization. See Figure 11-20 on page 399. Click **Run SQL** to deploy these changes to database.

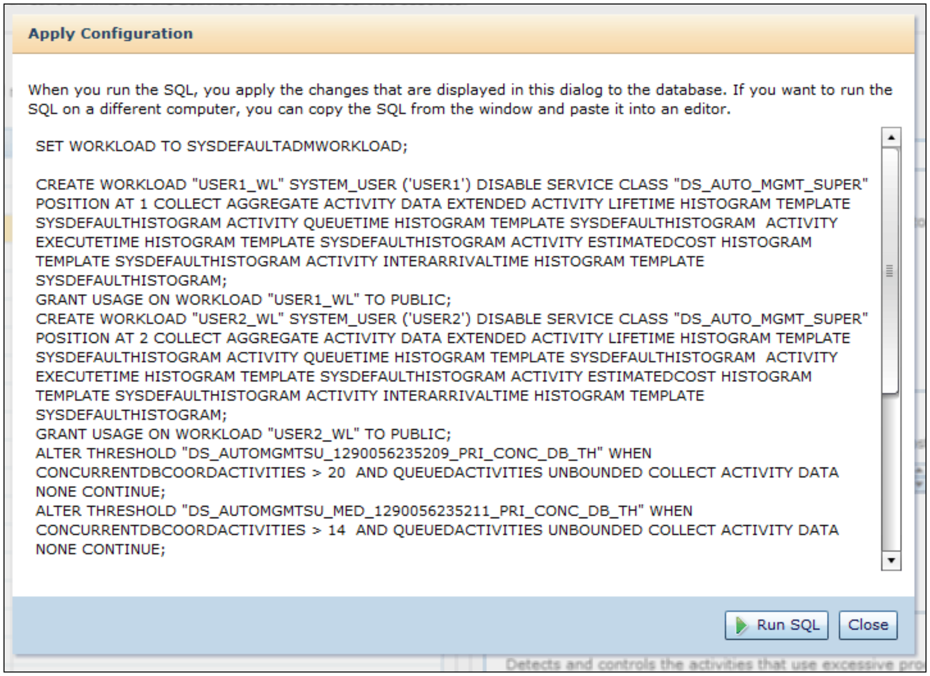


Figure 11-20 Deploy WLM configuration changes

11.3 Analyzing the monitoring statistics of a concurrency solution

When you create a configuration by using the concurrency method, you can plot graphs of the Workload Manager monitoring statistics. You can also see tables of the graph data and reports that help you analyze the performance of your service superclasses and service subclasses.

11.3.1 Analyzing the monitoring statistics of service superclasses

From the Graphs, Tables, and Reports area of the Service Superclasses page of Workload Manager configuration tool, you can plot graphs and review tables of the related monitoring metrics. You can use this information to:

- Project the system capacity
- Evaluate the concurrency limit of a service superclass

You can also see the Share of System Resources report and review the resource usage of each service superclass.

Project the system capacity

You can use the concurrency high water marks of the database and the CPU usage percentage to project the system capacity.

The concurrency high water marks of the database are plotted over time so that each point represents the high water mark for a specific collection interval. The CPU use percentage is the CPU usage of the system.

To plot concurrency high water marks and CPU usage graphs, follow these steps:

1. From the Graphs, Tables, and Reports area of the Service superclasses page, select the **Concurrency and CPU Usage** tab.
2. Define the time period that you want to evaluate.
3. Select **Database**.
4. Plot the concurrency high water marks of the database by selecting **High water mark**.
5. Plot the system CPU usage percent by selecting **CPU usage**.
6. Compare the two graphs.

Figure 11-21 on page 401 shows the concurrency high water mark and CPU usage graphs our scenario. Highest concurrency high water mark has a value of 29 and the associated CPU usage value is 25%. It could be reasonably predicted, that the system could handle a workload with the concurrency high water mark of up to 100.

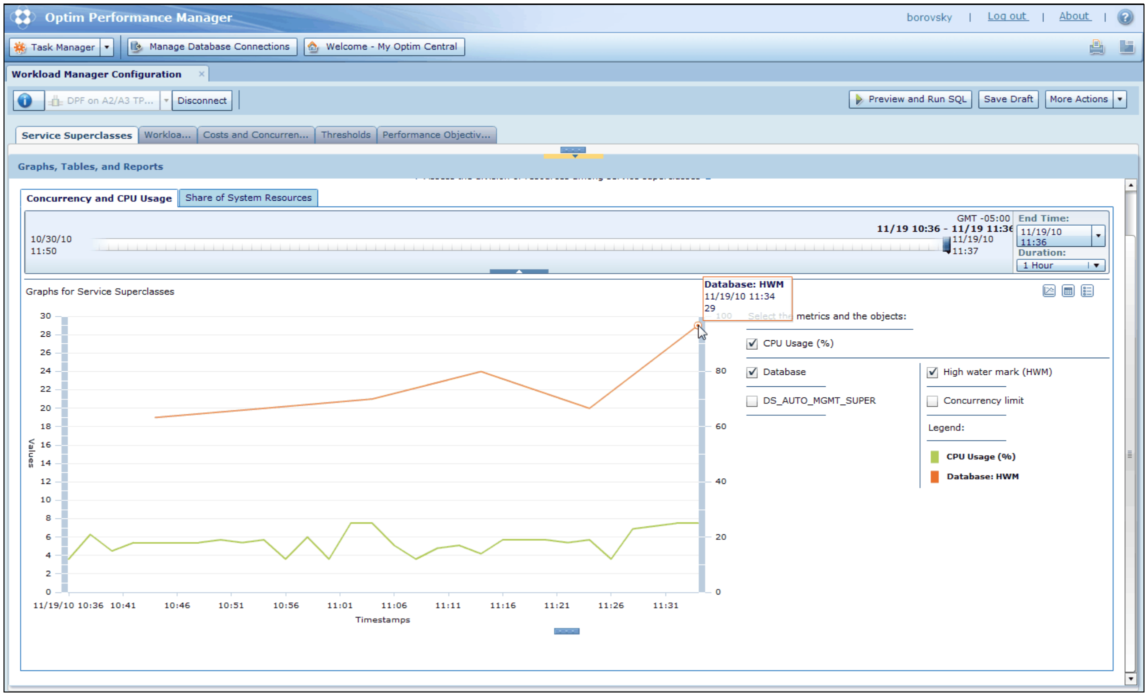


Figure 11-21 Concurrency high water mark and CPU usage graph

Evaluate the concurrency limit of a service superclass

You can use the concurrency high water marks and the concurrency limit metrics to evaluate whether the concurrency limit of a service superclass is appropriate.

The concurrency high water marks are plotted over time so that each point represents the high water mark for a specific collection interval. The concurrency limit specifies the maximum number of concurrent coordinator activities that the service superclass can run.

To plot concurrency high water marks and CPU usage graphs, follow these steps:

1. From the Graphs, Tables, and Reports area of the Service Superclasses page, select the **Concurrency and CPU Usage** tab.
2. Define the time period that you want to evaluate.
3. Select the check box next to the service superclass that you want to analyze.
4. Plot the concurrency high water marks by selecting **High water mark**.
5. Plot the concurrency limit by selecting **Concurrency limit**.
6. Compare the two graphs.

Figure 11-22 shows the concurrency high water mark and concurrency limit graphs for our example. Concurrency limit is set to 20 and concurrency high water mark averages at around 18.

The concurrency limit is appropriate if the concurrency high water mark graph is close to the concurrency limit graph throughout the specified time period.

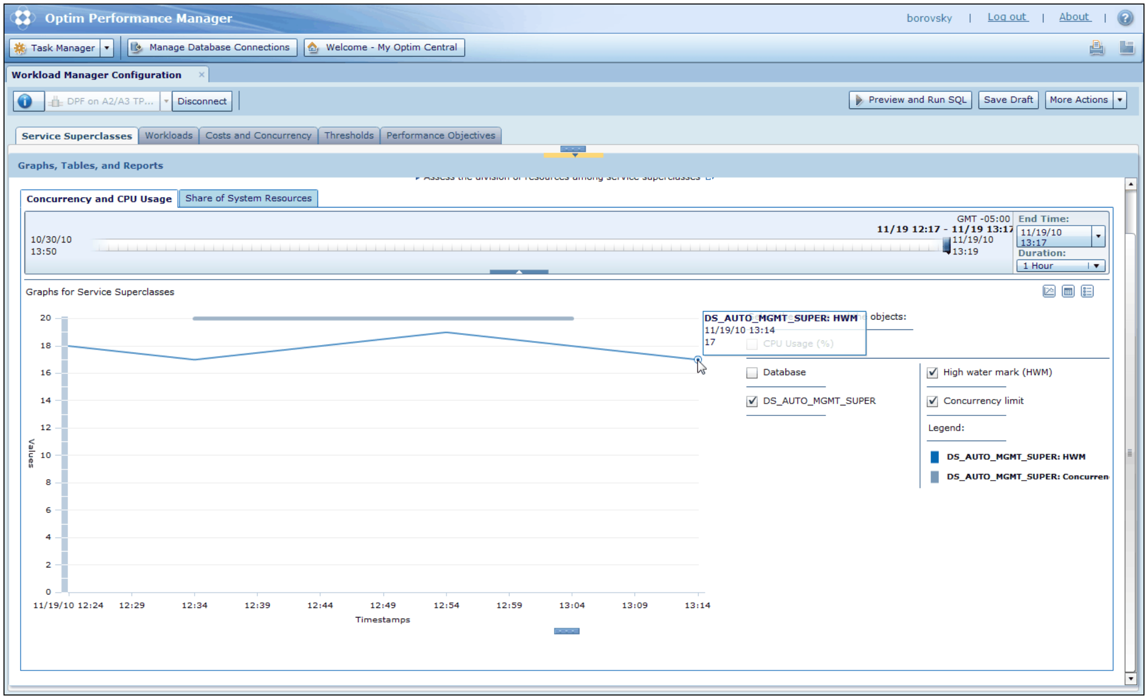


Figure 11-22 Concurrency high water mark and concurrency limit graph

11.3.2 Analyzing the monitoring statistics of service subclasses

From the Graphs, Tables, and Reports area of the Costs and Concurrency page of Workload Manager configuration tool, you can plot graphs of the monitoring metrics, create and review histograms, and see the response time statistics of your service subclasses.

This information could be used to:

- Evaluate the distribution of concurrency among service subclasses
- Stabilize the response times of a service subclass

Evaluating the distribution of concurrency among service subclasses

You can use the concurrency high water mark, the concurrency limit, and the time in the queue metrics to analyze and improve the distribution of concurrency.

Normally, you distribute the concurrency limit of a service superclass among the service subclass that runs ordinary work, the service subclass that runs batch jobs, and any additional service subclasses that you create for the service superclass. You expect that batch jobs often wait in the queue and that ordinary work starts to run immediately and spends little or no time in the queue.

You can find any service subclasses that are not using all the concurrency that is distributed to them. When you learn which service subclasses are under-utilizing the concurrency allotment, you can go to the Costs and Concurrency page and change the concurrency limits of those service subclasses.

To plot concurrency high water mark, concurrency limit and time in queue graph for selected service subclasses, follow these steps:

1. From the Graphs, Tables, and Reports area of the Costs and Concurrency page of Workload Manager configuration tool, select the **Concurrency and Time in the Queue Percentage** tab.
2. Define the time period that you want to observe.
3. Select the check box next to the service subclass that you want to analyze.
4. Plot a graph of the concurrency high water marks by selecting **High water mark check**.
5. Plot a graph of the concurrency limit by selecting **Concurrency limit**.
6. Compare the graph of the high water marks to the graph of the concurrency limit over the specified time period.

Figure 11-23 on page 404 shows the concurrency and time in queue graph for service subclass from our example. High water mark for `DS_MED_CONC_SUBCLASS` never reaches concurrency limit. However, `DS_LOW_CONC_SUBCLASS` high water mark is constantly above the concurrency limit for this service subclass.

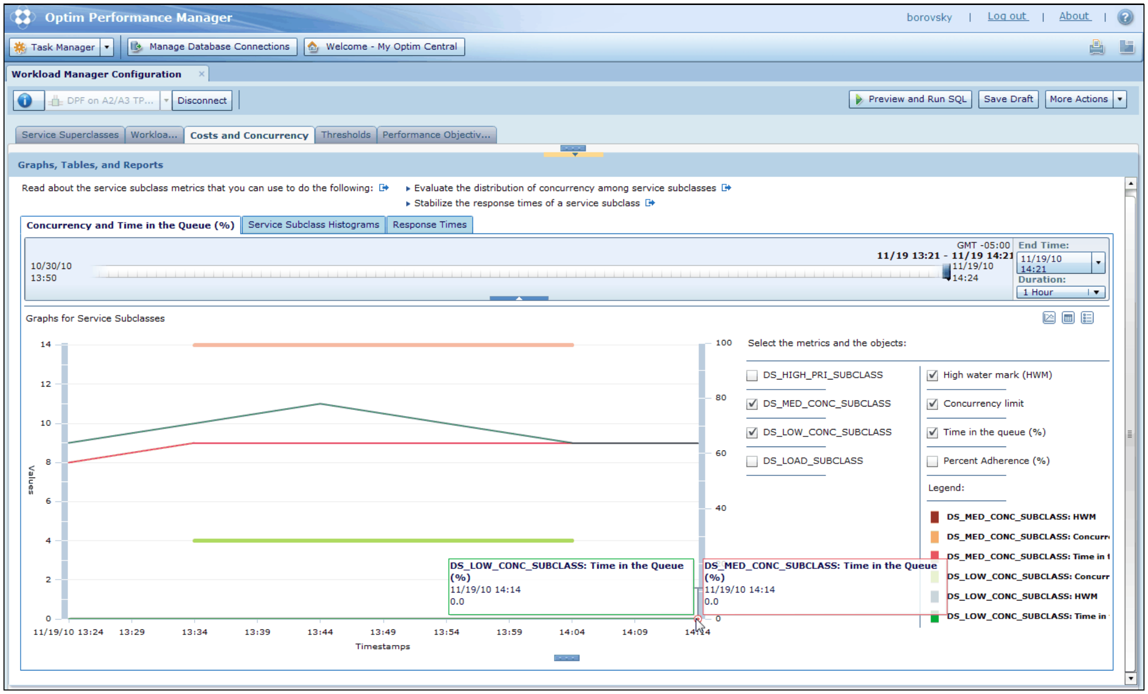


Figure 11-23 Concurrency and time in queue graph for selected service subclass

This concurrency overrun in case of `DS_LOW_CONC_SUBCLASS` is due to the fact, that concurrency limits have not been enforced for this subclass. To enforce these limits, go to the Cost and concurrency page and in the service subclass grid, click **Enforce** for the `DS_LOW_CONC_SUBCLASS` service subclass. Then click **Preview** and **Run SQL** to apply this change to the database. See Figure 11-24 on page 405.

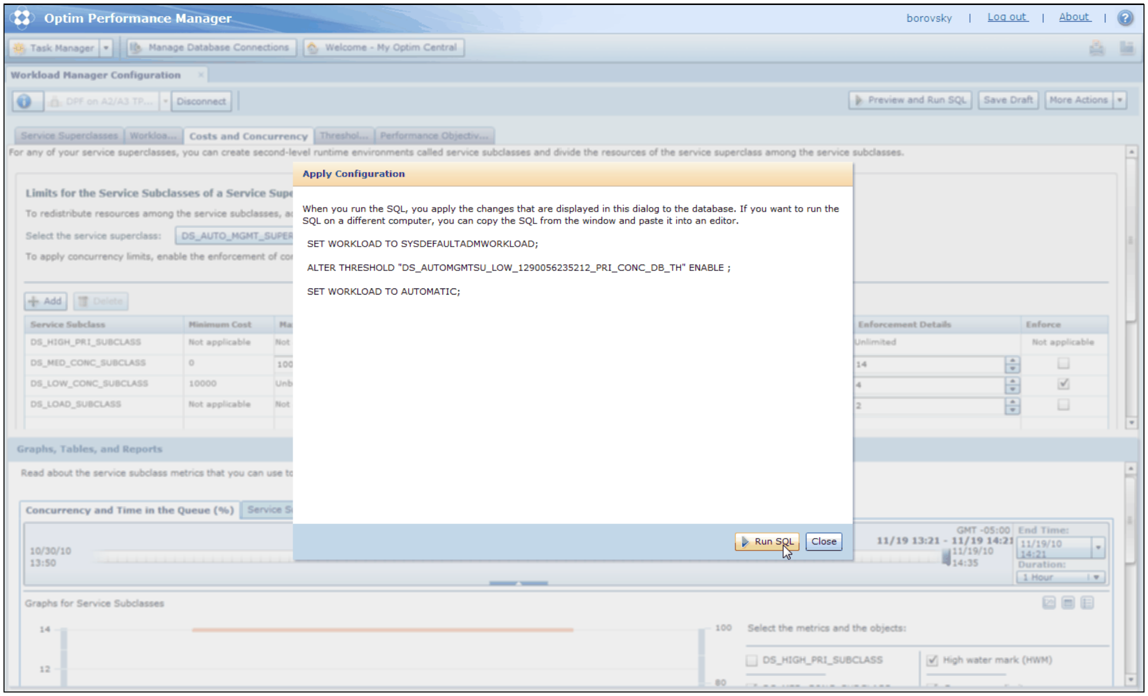


Figure 11-24 Concurrency limit enforcement for the selected service subclass

After enforcing concurrency limits on your service subclasses, continue to monitor them from the Cost and concurrency page.

If the concurrency high water mark of a service subclass rarely or never reaches the concurrency limit, you can go to the Costs and Concurrency page and reduce the concurrency limit.

If a service subclass frequently reaches the concurrency limit, determine whether only a few activities are queued or a pervasive concurrency issue exists, and make the required adjustments. Plot a graph of the percentage of time the activities of the service subclass spend in the queue over the specified time period by selecting **Time in the queue**.

If the activities of the service subclass spend an excessive amount of time in the queue, go to the Costs and Concurrency page and reduce the concurrency limit of the service subclass.

Stabilizing the response time of a service subclass

You can use the activity total time and the activity queue time histograms of the Cost and concurrency page to evaluate and stabilize erratic response times that result from low concurrency limits.

When erratic response times result from low concurrency limits, you can increase the concurrency limits of your service subclasses to improve the response times of the activities that run in the service subclasses.

To verify, that the response times are erratic, go to the Graphs, Tables, and Reports area of the Costs and Concurrency page, select the **Service Subclass Histograms** tab. Review the activity total time histogram of the service subclasses. Wide variation among the response times might be an indication of erratic response times.

Activity total time histogram for service subclass, collects and distributes total activity lifetime data into discrete ranges called *bins*. For each bin it displays the number of activities which completed within the defined range.

Figure 11-25 on page 407 shows a DS_MED_CONC_SUBCLASS activity total time histogram.

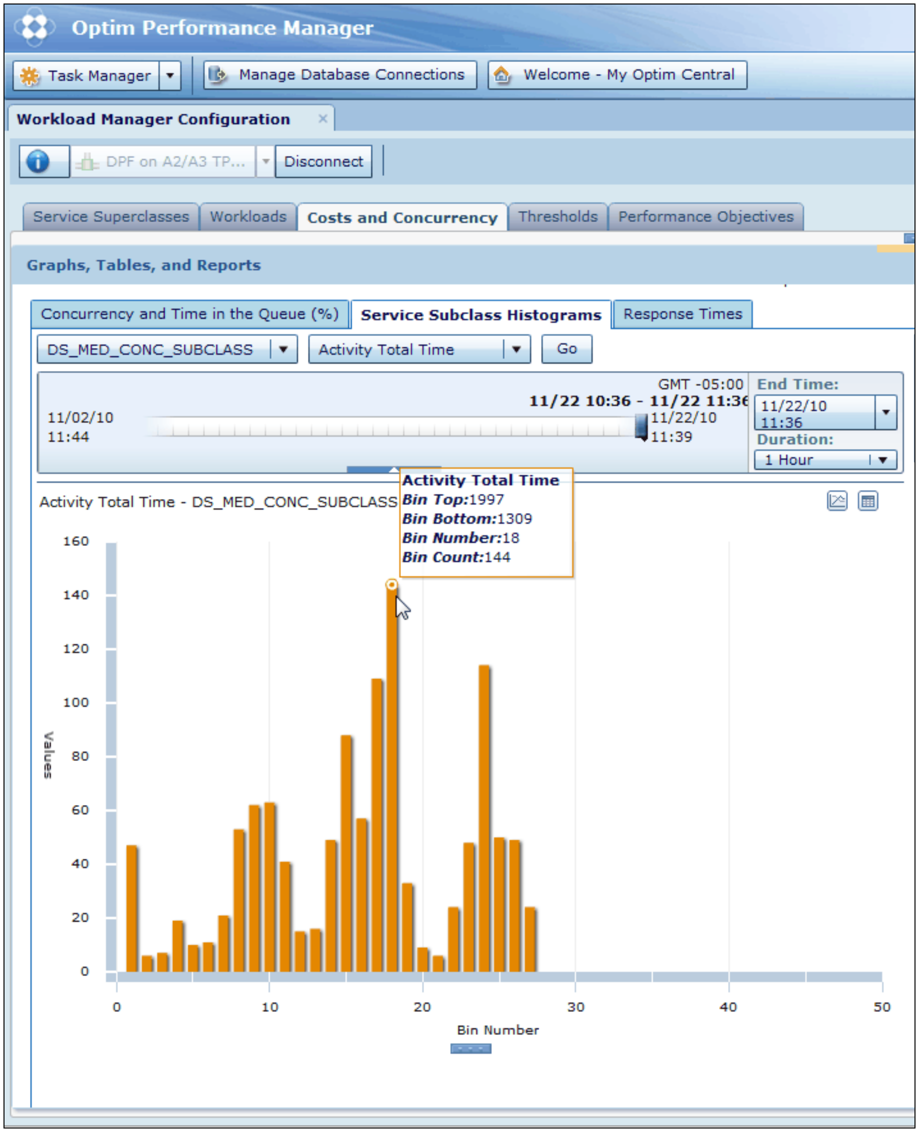


Figure 11-25 DS_MED_CONC_SUBCLASS activity total time histogram

In our example, the activity total time histogram for `DS_MED_CONC_SUBCLASS` shows that most activities (144) completed within the range of 1309ms to 1997ms.

To verify that you can stabilize the response times by increasing the concurrency limit of the service subclass, review the activity queue time histogram of the service subclass.

Look for a large number of nonzero values as evidence that the concurrency limit is too low. If the concurrency limit is too low, go to the Costs and Concurrency page and increase the limit.

Figure 11-26 shows a comparison of activity queue time histograms for the DS_MED_CONC_SUBCLASS and DS_LOW_CONC_SUBCLASS service subclasses. The DS_MED_CONC_SUBCLASS histogram shows that all activities of this service subclass spent 0 to 1ms in queue, which means that this service subclass has no activity queueing. The DS_LOW_CONC_SUBCLASS histogram shows that there were activities in this service subclass that spent various amount of time in queue. However, overall number of queued activities in DS_LOW_CONC_SUBCLASS does not represent a substantial number, so there is no need to change concurrency limit for this service subclass.

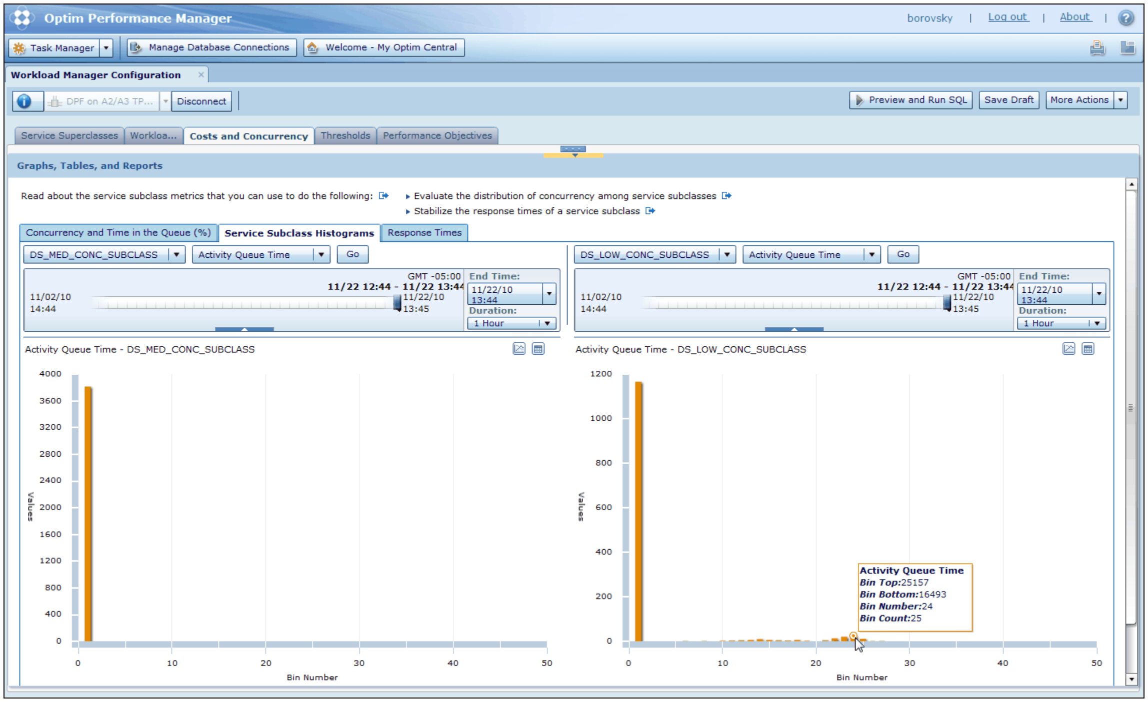


Figure 11-26 Activity queue time histogram

11.3.3 Analyzing the monitoring statistics of workloads

From the Graphs, Tables and Reports section of Workloads page you can analyze monitoring statistics of workloads. It is very similar to analyzing monitoring statistics of service subclasses. Collected data however is based on workloads rather than service subclasses.

You can use the *Concurrency and Time in queue* percentage tab to plot the graph of workload occurrences high water marks for the defined workloads as well as time queue percentage for each workload.

Figure 11-27 on page 409 shows a Workloads Concurrency and Time in queue percentage graph.

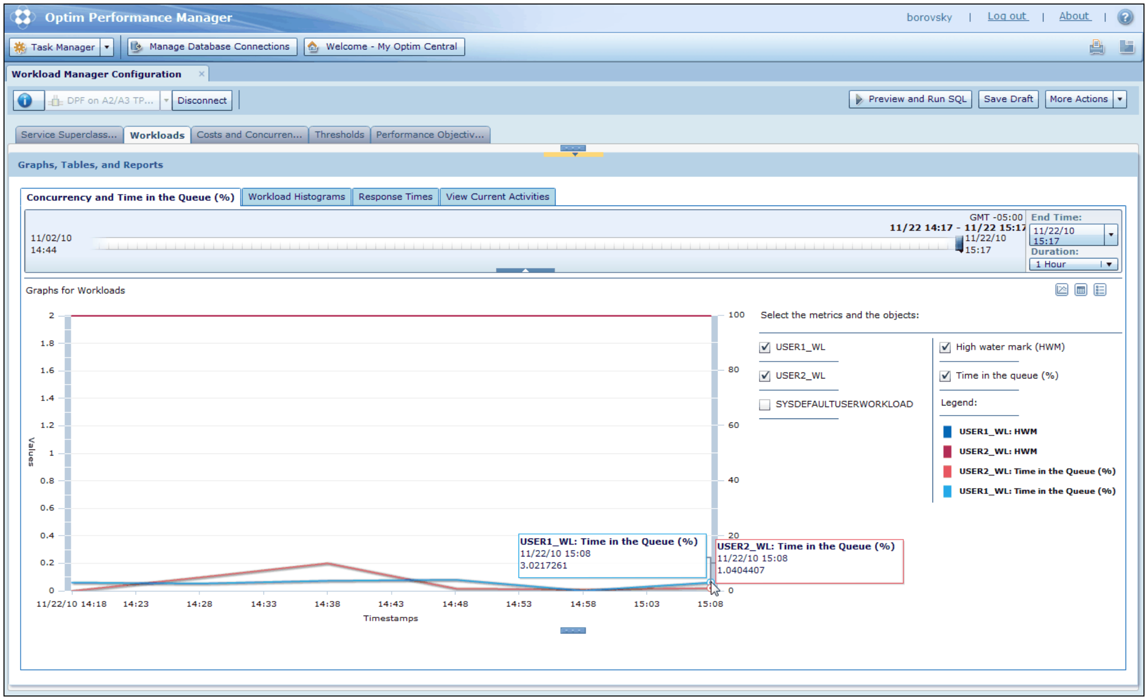


Figure 11-27 Workloads Concurrency and Time in queue percentage graph

Use the *Workload histograms* tab to plot activity total time and activity queue time histograms for defined workloads. Figure 11-28 on page 410 shows a Workloads activity total time histogram.

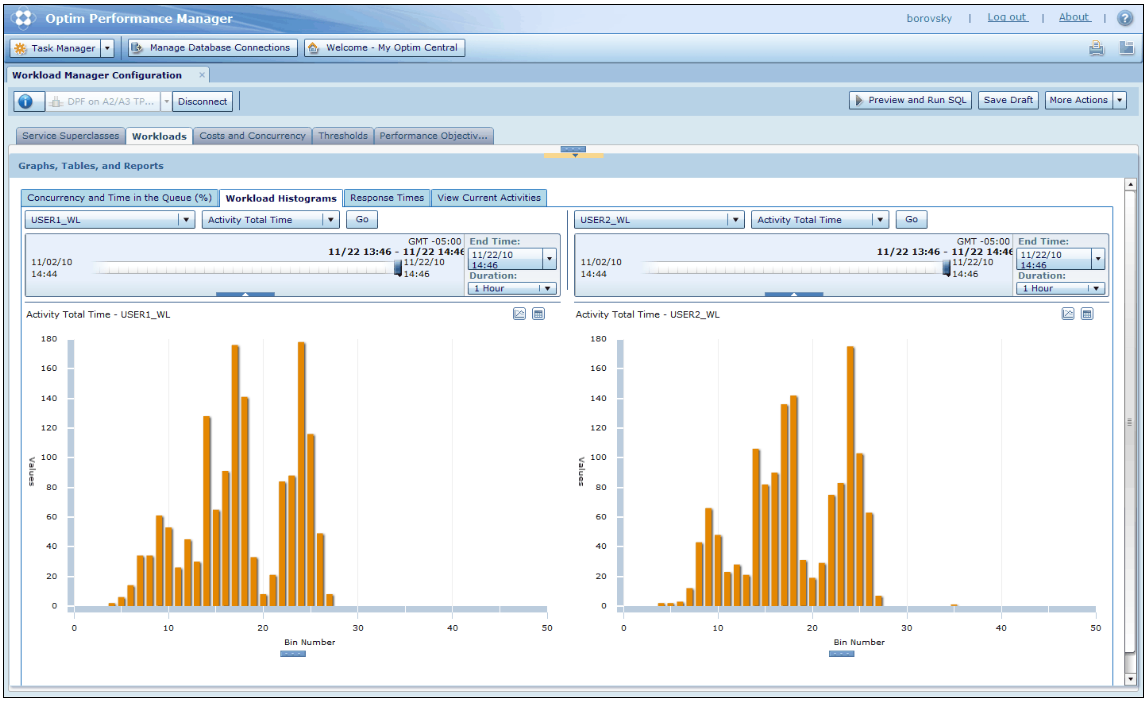


Figure 11-28 Workloads activity total time histogram

11.4 Autonomic performance objectives for workloads

You can configure Optim Performance Manager to continually adjust the concurrency settings of service subclasses so that your workloads meet their performance objectives.

When you specify a performance objective for the activities of a workload, you create an additional unique service subclass in which the activities run.

A performance objective is defined by a target response time and the percentage of activities that must adhere to the target response time for the performance objective to be met. If you enable autonomic performance objectives, Optim Performance Manager will continually adjust the concurrency limit of the workloads associated with the service subclass to ensure that activities meet the performance objective.

You can create a performance objective when you create a new workload, or you can change a fixed service subclass or a discretionary service subclass to a performance objective.

11.4.1 Configuring autonomic performance objectives for workloads

Before you begin to configure autonomic performance objectives for workloads, go to the Service Superclasses page and ensure that the concurrency limit for the service superclass is set to an appropriate value. If the concurrency limit is too low, the autonomic performance objective service might overly restrict the concurrency of discretionary subclasses of the service superclass. Therefore, a low concurrency limit can degrade the performance of activities that run in discretionary service subclasses. If the concurrency limit is too high, activities that run in a subclass of the service superclass can use more system resources than you want.

In our example we have created `PRICE_LOOKUP_WL` workload, which contains workload activities based on the value of Client Application Name set to `PRICE_LOOKUP`.

Performance objective workload

To create a performance objective for this workload, go to the Workloads page and select the workload from the Workload Evaluation list. In the Concurrency/priority field, select **New performance objective**, as shown in Figure 11-29.

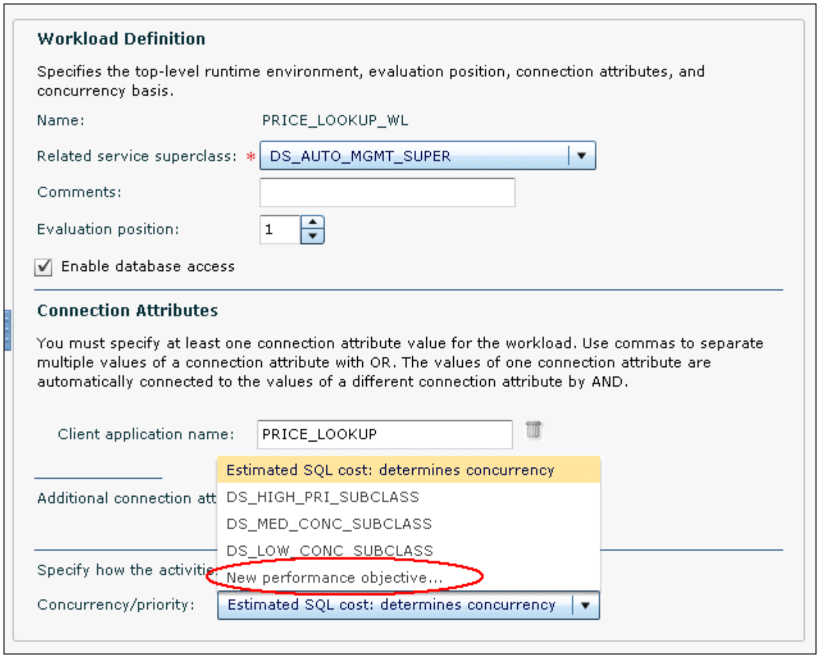


Figure 11-29 Create performance objective workload

On the Add a Service Subclass panel (Figure 11-30) define the new performance objective by specifying:

- Name of the service subclass, which will execute performance objective workload
- Target time in milliseconds
- Percent adherence

If you specify a value in the *Maximum estimated SQL cost* field, the service subclass of the performance objective is inserted into the cost hierarchy for workloads that are configured to determine concurrency by estimated SQL cost.

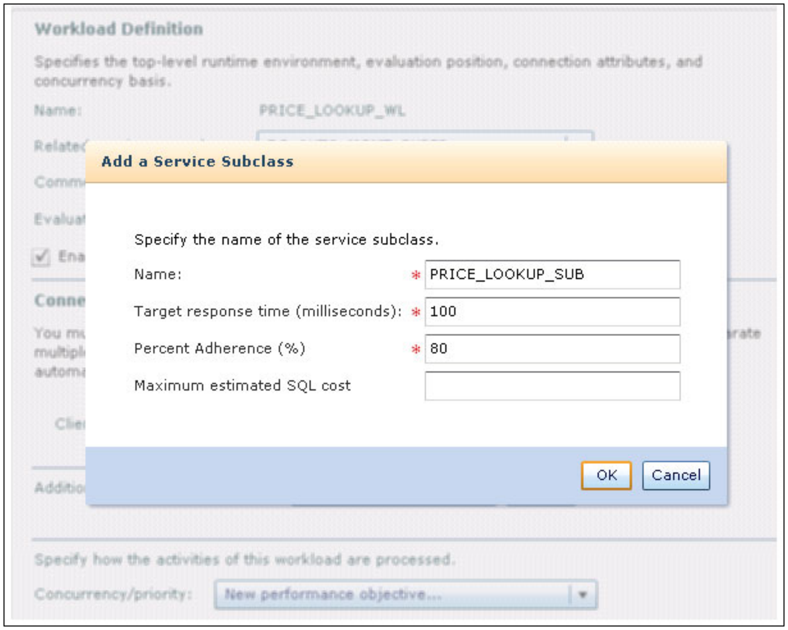


Figure 11-30 Performance objective service subclass settings

Deploy these configuration changes to database by clicking **Preview and Run SQL**.

Performance objective service subclass

A side effect of specifying a performance objective in the workloads tab is the creation of a new service subclass where `PRICE_LOOKUP_WL` activities will run. In the costs and concurrency tab, you can see the new service subclass as shown in Figure 11-31 on page 413.

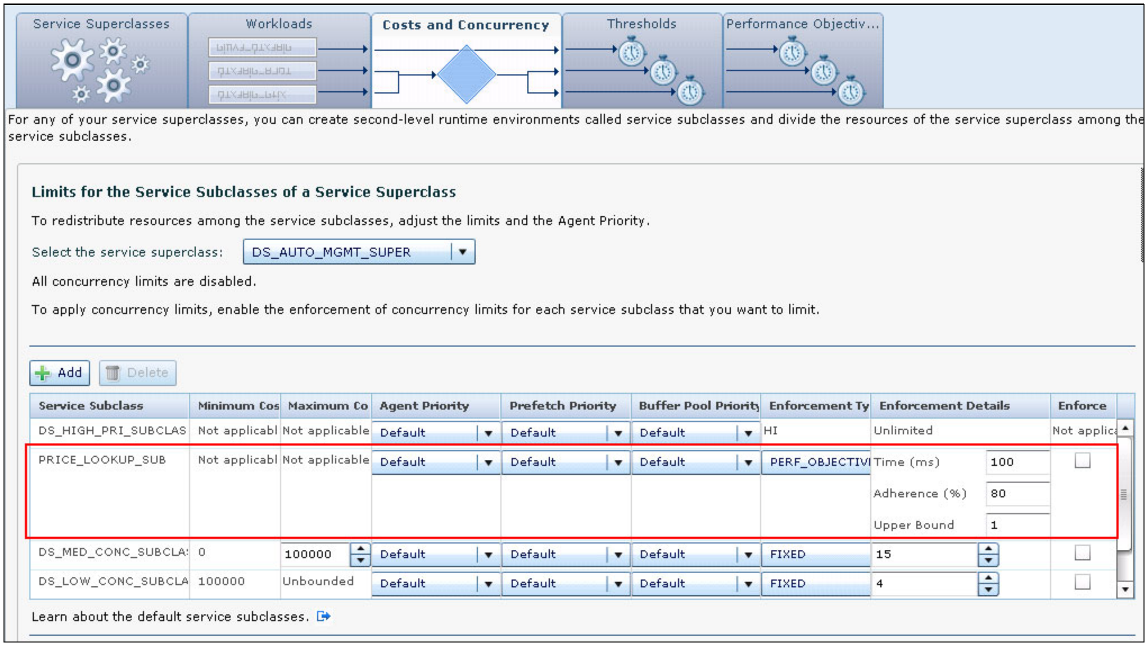


Figure 11-31 New performance objective service subclass

On the Costs and Concurrency page, you can also specify an upper bound for the concurrency of the service subclass that corresponds with your performance objective. The system prevents the maximum concurrency of the service subclass from exceeding the value that you specify as the upper bound.

Discretionary service subclass

Performance objectives are enforced by taking resources away from some service subclasses. You indicate which service subclasses are available for this purpose by marking them as discretionary. There must be at least one service subclass marked as discretionary before you can enforce a performance objective.

In this example, the existing service subclass `DS_MED_CONC_SUBCLASS` is designated as a discretionary service subclass. See Figure 11-32 on page 414.

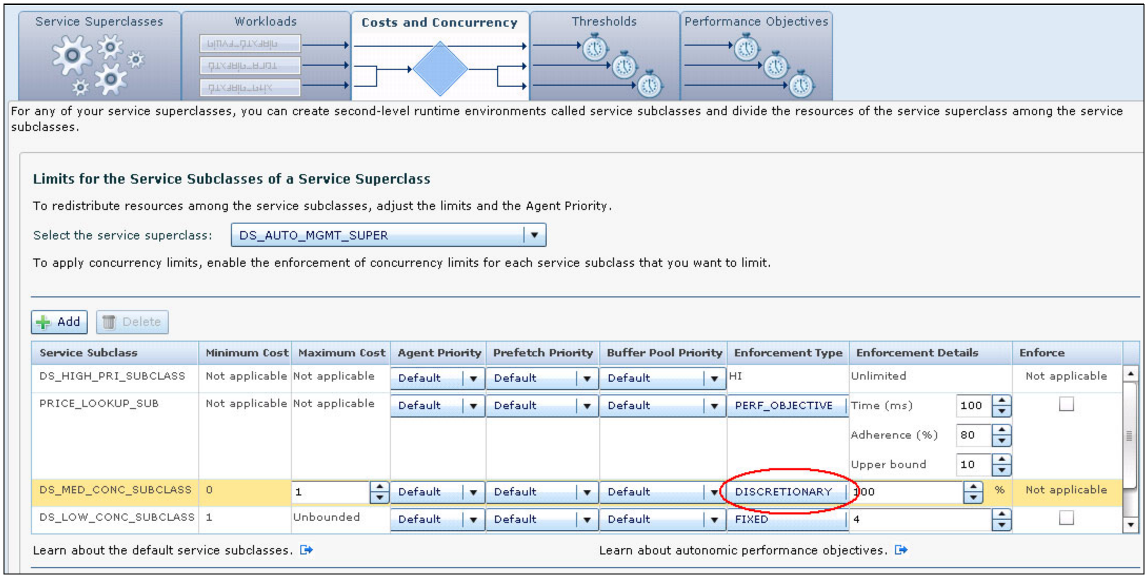
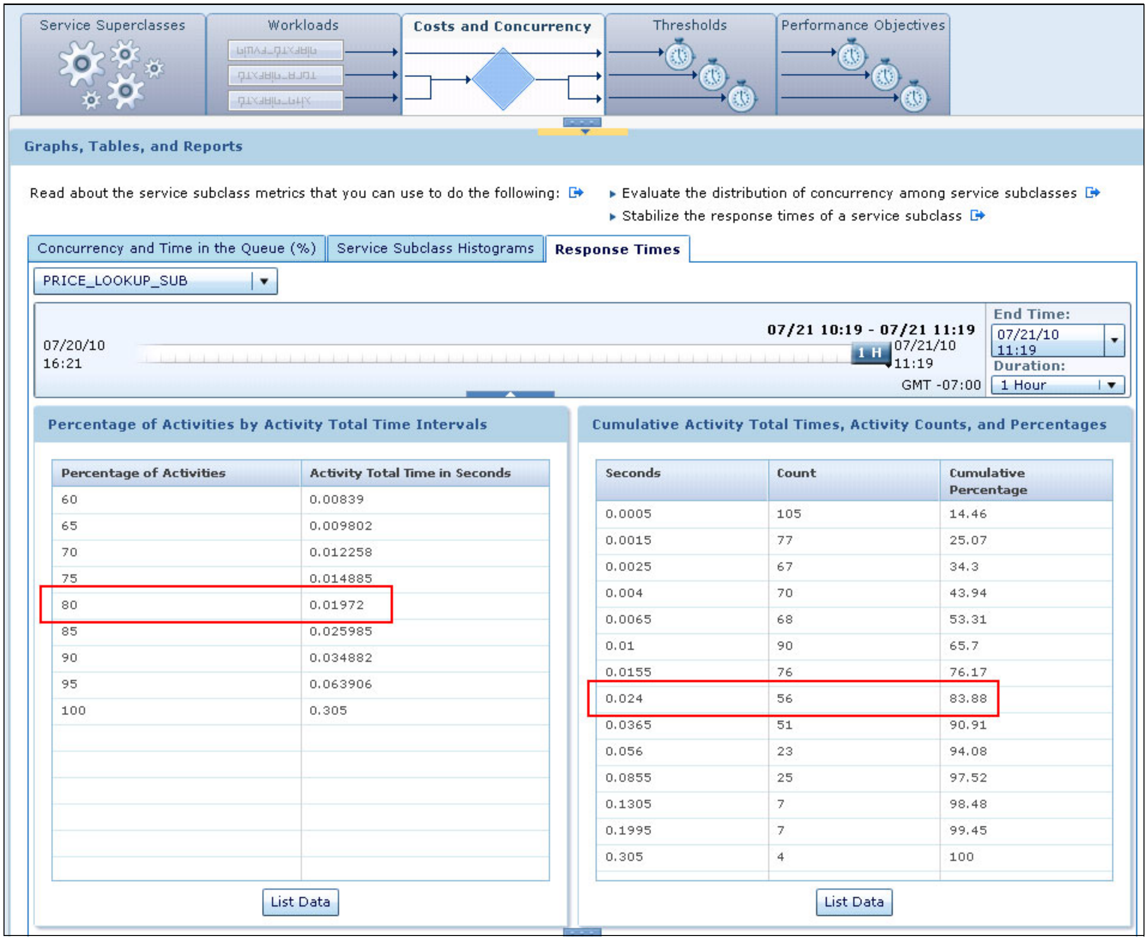


Figure 11-32 Discretionary service subclass definition

After you defined performance objective infrastructure, use the Response Times report in the Graphs, Tables, and Reports section of the Costs and Concurrency page to monitor the activities in the service subclass to verify that the expected activities are running and to determine an appropriate target response time. Then, if necessary, change the target response time setting of the performance objective on the Costs and Concurrency page.

Figure 11-33 on page 415 shows a performance objective service subclass response times report. In this example, 80% of the activities completed in less than 20 ms. So a reasonable goal might be 18 ms, this is 10% better than the observed performance when there are no concurrency limits on the discretionary subclasses.



Percentage of Activities	Activity Total Time in Seconds
60	0.00839
65	0.009802
70	0.012258
75	0.014885
80	0.01972
85	0.025985
90	0.034882
95	0.063906
100	0.305

Seconds	Count	Cumulative Percentage
0.0005	105	14.46
0.0015	77	25.07
0.0025	67	34.3
0.004	70	43.94
0.0065	68	53.31
0.01	90	65.7
0.0155	76	76.17
0.024	56	83.88
0.0365	51	90.91
0.056	23	94.08
0.0855	25	97.52
0.1305	7	98.48
0.1995	7	99.45
0.305	4	100

Figure 11-33 Performance objective service subclass response times

11.4.2 Deploying autonomic performance objective

To enforce the performance objective for the service subclass, go to the Cost and Concurrency page and select **Enforce**.

On the Performance Objectives page (Figure 11-34), ensure that autonomic performance objectives are enabled. When autonomic performance objectives are enabled, the concurrency settings of all service subclasses with enforced performance objectives are continually adjusted.

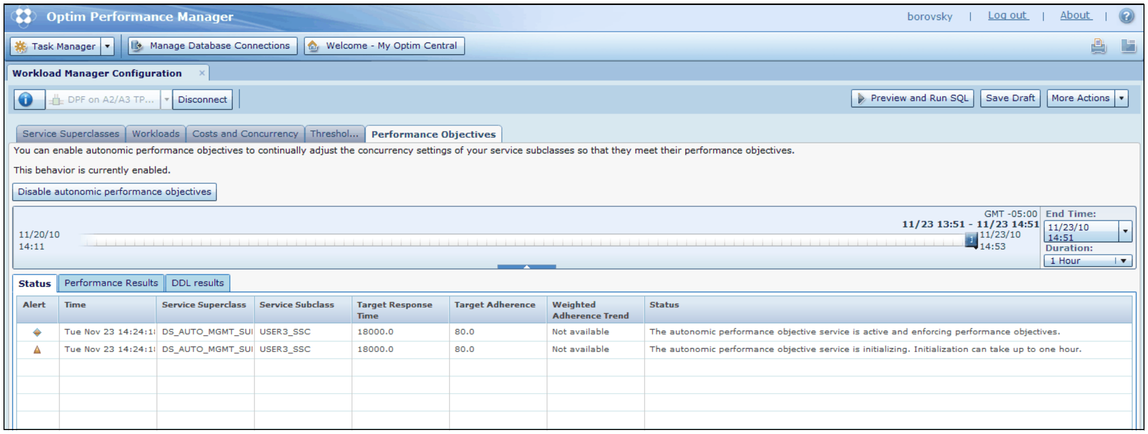


Figure 11-34 Performance objective page

Click **Preview and Run SQL** to deploy your configuration. If you are prompted to save the performance objective or to enable autonomic performance objectives, click **Yes**.

11.4.3 Monitoring autonomic performance objective

When you enable autonomic performance objectives for the first time for a workload, the autonomic performance objective service initializes by collecting monitoring data. Initialization can take more than an hour.

After the autonomic performance objective service initializes, Optim Performance Manager continually adjusts the concurrency settings of the service subclass to meet the performance objective of the workload. You can monitor these adjustments on the Graphs, Tables and Reports section of the Costs and Concurrency page.

Figure 11-35 shows how Optim Performance Manager automatically increased concurrency limit for performance objective service subclass. This adjustment also increased the Percent adherence value for this service subclass.

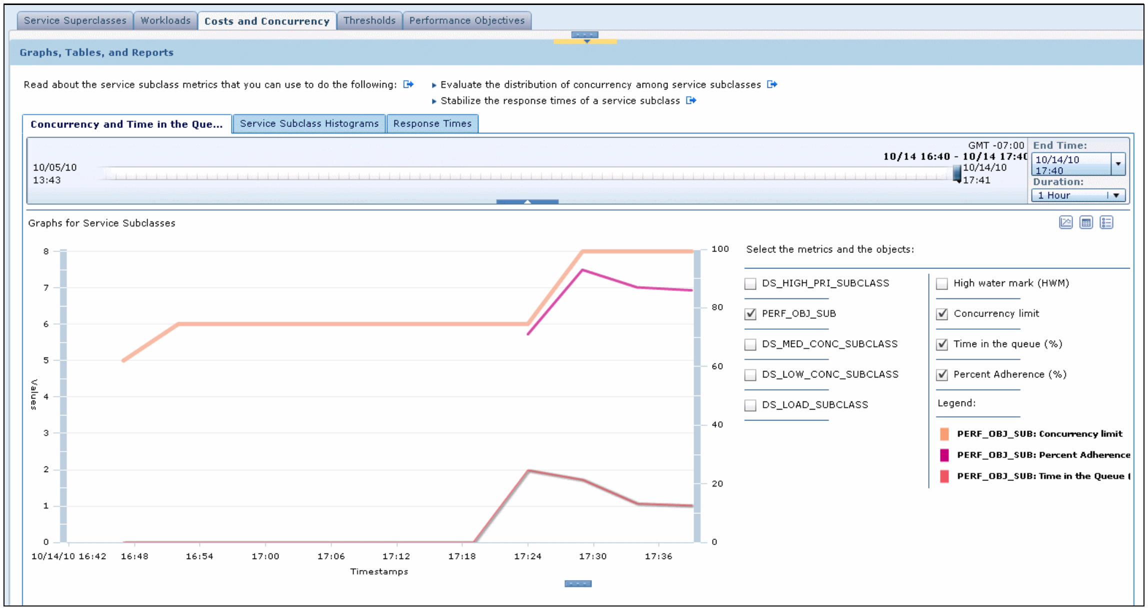


Figure 11-35 Concurrency and Time in queue graph for the performance objective service subclass

You can also use this tab to show the impact that the automatic increase or decrease of the concurrency limit for the performance objective service subclass has on discretionary service subclasses. Figure 11-36 on page 418 shows an example of this behavior.

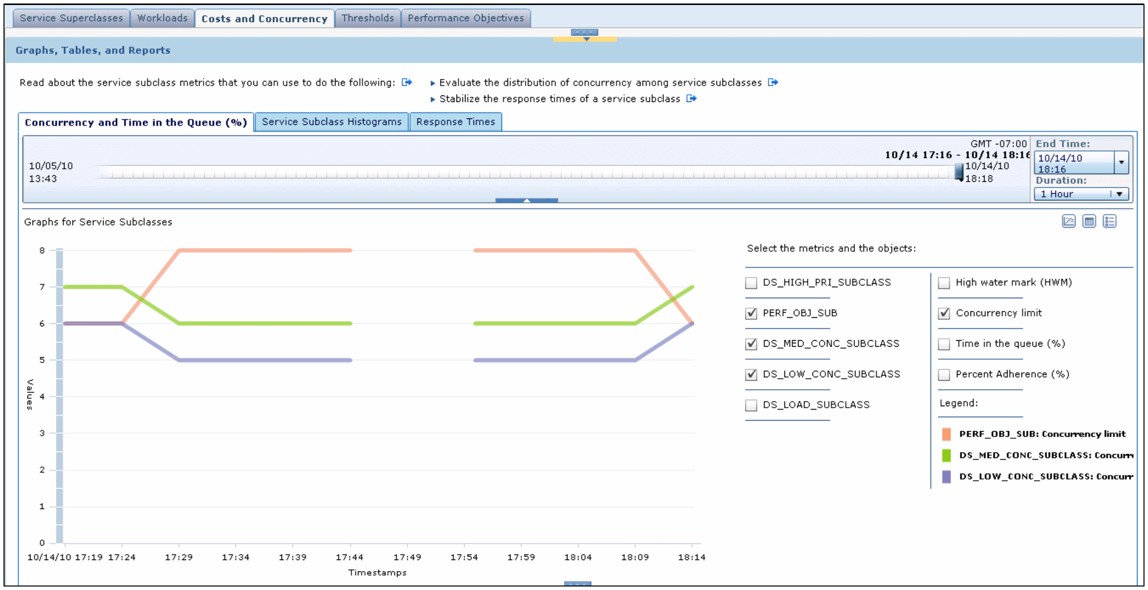


Figure 11-36 Concurrency limits changes for performance objective and discretionary service subclasses

The Response Times tab (Figure 11-37 on page 419) of the Costs and Concurrency page shows cumulative percentage number of service subclass activities that completed within the specified value for target time setting as well as cumulative percentage number of activities that completed outside target time setting.

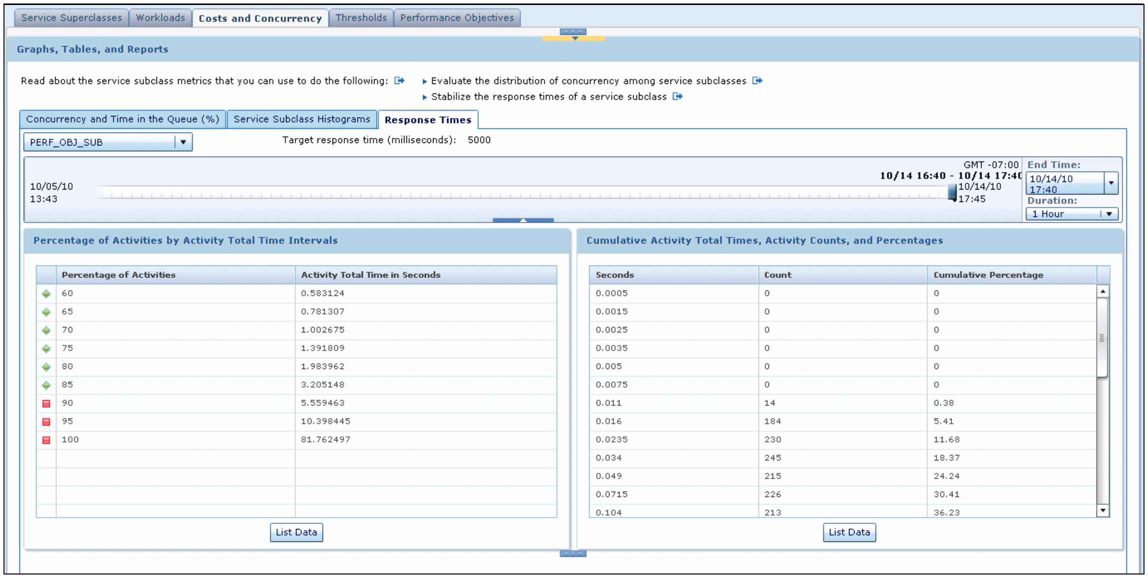


Figure 11-37 Response Times tab for performance objective service subclass

On the Performance Objectives page you can monitor the changes in the percent adherence value for the performance objective service subclass. This information closely corresponds to Concurrency and Time in queue graph from the Costs and Concurrency page.

Figure 11-38 on page 420 shows the Performance result tab of the Performance Objectives page.

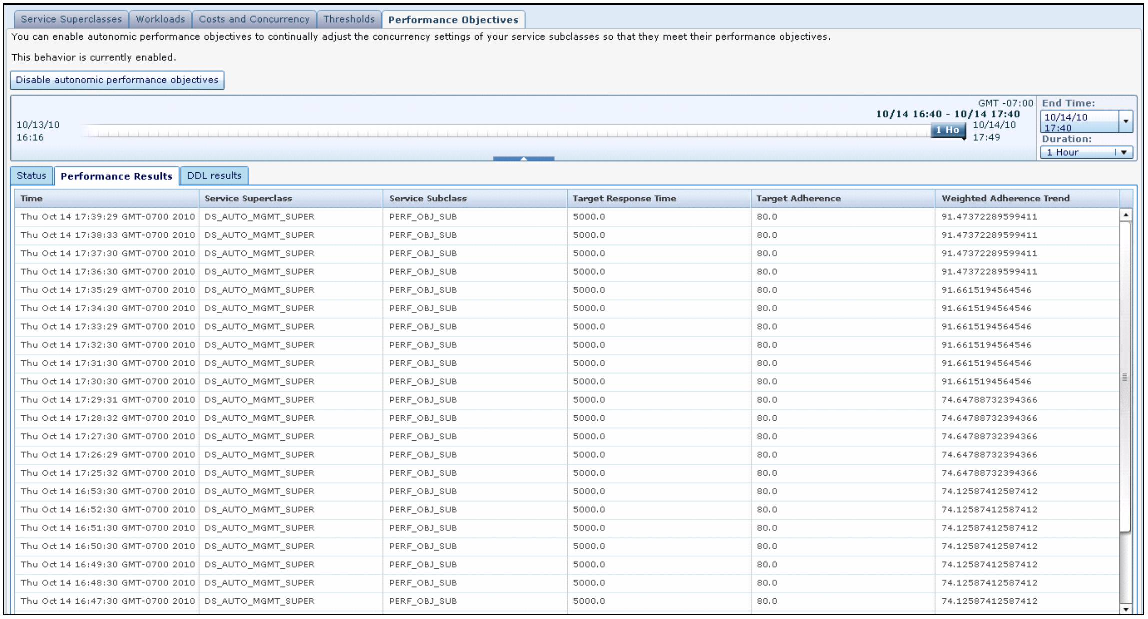


Figure 11-38 Performance results tab

The DDL tab (Figure 11-39 on page 420) of the Performance objective page displays the DDL statements that the autonomic performance objective service issues to enforce the performance objectives.

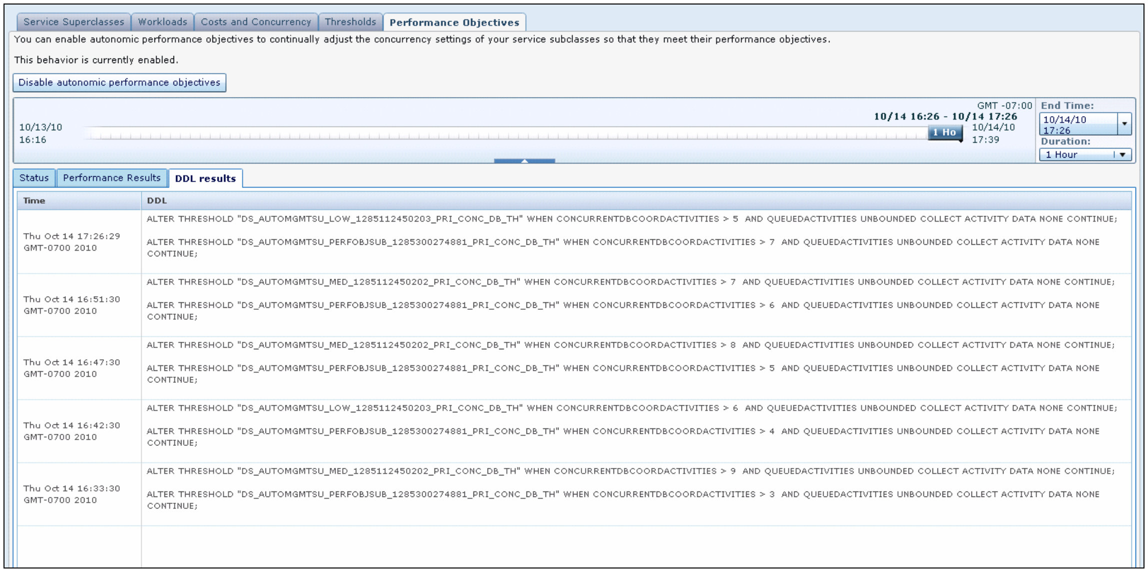


Figure 11-39 Performance objective DDL tab

Use the Status tab (Figure 11-40) of the Performance objective page to find out whether:

- autonomic performance objective service is running.
- Optim Performance Manager cannot adjust services subclasses to meet the performance objective of a workload

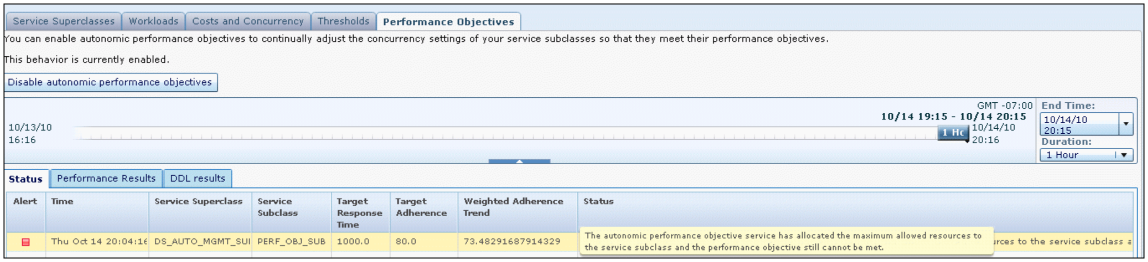


Figure 11-40 Performance objective status tab

12

Monitoring SAP environments

IBM works together with SAP closely to optimize Optim Performance Manager for monitoring an SAP environment. The optimizations that are implemented in Optim Performance Manager Extended Edition V4.1.0.1 include installation enhancements for Extended Insight client, lock monitor event detail levels, special DB2 monitor switch handling, watchdog for event monitors, and predefined system templates.

In this chapter we describe how to install and configure Optim Performance Manager Extended Edition to monitor an SAP environment. Within these steps we describe the implemented optimizations briefly including the benefit of them and give further configuration hints that result from the close work with SAP.

We also show you how to monitor response times and time spent details of transactions and SQL statements per SAP user, SAP source module or SAP transactions using Optim Performance Manager Extended Insight.

12.1 Installing Optim Performance Manager Extended Edition

Installing Optim Manager Extended Edition consists of the following:

- Installing and activating Optim Performance Manager
- Installing and configuring Optim Performance Manager Extended Insight client

To plan the installation of Optim Performance Manager, see Chapter 2, "Planning" on page 15. To install and activate Optim Performance Manager, use 3.2.1, "Installing Optim Performance Manager" on page 65 and 3.2.4, "Installing DB2 Performance Expert Client" on page 83.

SAP environments use CLI to access their DB2 databases. The path you need to choose to install and configure Optim Performance Manager Extended Insight depends on the setup of your SAP environment:

- In your SAP environment, if DB2 client packages are located on a central file share and copied to the SAP system during startup of the SAP application server, you can embed a configured Optim Performance Manager Extended Insight client into a DB2 client package. This way you ensure that Optim Performance Manager Extended Insight client is copied together with the DB2 client package to the SAP system. No additional configuration is required after the copy step. The embedded installation is described in 12.1.1, "Embedding Optim Performance Manager Extended Insight into an existing DB2 client package".
- In your SAP environment, if DB2 client packages are fix installed on the SAP system and not copied, then install Optim Performance Manager Extended Insight client and configure it for CLI as described in 3.4, "Installing and Configuring Extended Insight Client" on page 129.

12.1.1 Embedding Optim Performance Manager Extended Insight into an existing DB2 client package

The embedded installation of Optim Performance Manager Extended Insight uses a response file to embed and configure the product silently in a subdirectory under the DB2 Client Package installation path. When copying the DB2 Client Package to the SAP system during SAP application server startup, Extended Insight client is copied with it.

To embed Optim Performance Manager Extended Insight into an existing DB2 client package, perform these steps:

1. Update the properties shown in Example 12-1 in the `opmei_sample_response.rsp` file with the appropriate values of your environment, and save the file. The value for `DB2DSDRIVER_CONSROLLER_URL` is the same as the value for the `pdq.cmx.controllerURL` property in `pdq.properties` file of Optim Performance Manager which is set when activating Optim Performance Manager for Extended Insight.

   Example 12-1 Update properties

   ```
   #Has the license been accepted
   LICENSE_ACCEPTED=TRUE

   #CLI driver installation root directory
   CLI_DRIVER_ROOT=C:/temp/IBM/ibm_cli_driver/clidriver

   #flag for embedding Optim Performance Manager Extended Insight at the given
   CLI installation root directory
   EMBEDDED_INSTALL=TRUE

   #CMX controller url
   DB2DSDRIVER_CONTROLLER_URL=localhost:60000

   #location of the db2dsdriver.cfg file
   DB2_DSDRIVER_CFG=C:/temp/IBM/ibm_cli_driver/clidriver/cfg/db2dsdriver.cfg
   ```

   > **Note:** If you use a DB2 client package of DB2 V9.7 Fix Pack 3 or higher, specifying the `DB2_DSDRIVER_CFG` property is optional on Linux and UNIX. If the property value is blank, the installer uses the default location of the `db2dsdriver.cfg` file. For all other platforms and DB2 client package versions, specifying a value for `DB2_DSDRIVER_CFG` is mandatory.

2. From the directory of the Extended Insight installation image, run the following command with *<platform>* replaced with your platform, for example AIX:

   ```
   IBM_OPMEI_V4_1_0_1_<platform>.bin -i silent -f path_to_response_file
   ```

3. When the installation finished, verify that it was successful by reading the installation log files in the following directory:

   ```
   db2_client_package_installation_dir/opmei/logs
   ```

During the embedded installation, the value for the `connectionSupervisorLibrary` property in the `db2dsdriver.cfg` file is updated with the location of the `pqcmx` library. For DB2 Client Package V9.7.3 and later, the `connectionSupervisorLibrary` property is updated with a relative path to the `pqcmx` library so that the path remains valid even if the DB2 client package is copied to another location.

12.1.2 Validating Extended Insight Client configuration

If your DB2 client package is at DB2 9.7 Fix Pack 2 or higher, you can validate your Extended Insight client configuration for your SAP environment to ensure that Extended Insight data can be collected by Optim Performance Manager.

If you use the embedded install, then use this validation step for troubleshooting purposes if Optim Performance Manager does not collect Extended Insight data. Do the validation after the DB2 client packages were copied to the SAP system during startup of the SAP application server.

If you installed and configured Extended Insight client without the embedded install, we recommend to always perform the validation after you finished installation and configuration.

Call the validation routine using the following command:

`db2cli validate -database mydb:mydbserver:50000`

where *`mydb`*, *`myserver`* and *`50000`* are the database name, host name, and port number of your monitored database.

Run this command as the SAP user. The SAP user has the name as `<ssid>adm`, where `<ssid>` is the SAP system ID.

Before running the `db2cli` command on Linux or UNIX, you might have to adapt environment variables. Try the command first and if it fails, set the `DB2_CLI_DRIVER_INSTALL_PATH` variable to the path of the DB2 client package and include the lib directory of the DB2 client package in the `LIBPATH` variable. Example 12-2 shows an example of commands for an SAP system.

Example 12-2 Set environment variables

```
export DB2_CLI_DRIVER_INSTALL_PATH=/sapmnt/OLT/global/db6/AIX_64/db6_clidriver/

export
LIBPATH=/sapmnt/OLT/global/db6/AIX_64/db6_clidriver/lib:/usr/sap/OLT/SYS/exe/uc
/rs6000_64:/db2/db2olt/sqllib/lib64

cd /sapmnt/OLT/global/db6/AIX_64/db6_clidriver/bin

./db2cli validate -database OLT:db6lpar4:5912
```

Example 12-3 shows a sample output of the `db2cli` command. Look for messages prefixed with `PQCMX`. If you receive successfully connected messages, the Extended Insight client is correctly configured.

Example 12-3 Output of validation

```
IBM DATABASE 2 Interactive CLI Sample Program
(C) COPYRIGHT International Business Machines Corp. 1993,1996
All Rights Reserved
Licensed Materials - Property of IBM
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Header
:---------------------------------------------------------------------------
[ CLI Driver Version : 09.07.0000 ]
[ Informational Tokens : "DB2 v9.7.0.2", "s100514",IP23082","Fixpack 2" ]
[ CLI Driver Type : IBM DB2 Application Runtime Client
]--------------------------------------------------------------------------

Warning: The schema validation operation completed successfully.
The following data source name was not found in the db2cli.ini
file: "".

db2dsdriver.cfg Validation:
---------------------------------------------------------------------------
[ DB2DSDRIVER_CFG_PATH env var : unset ]
[ db2dsdriver.cfg Path : /db2/db2olt/sqllib/cfg/db2dsdriver.cfg ]
---------------------------------------------------------------------------
[ Keywords used by CLI for Database : OLT ]
Keyword Value
--------------------------------------------------
connectionSupervisorLibrary /opm_data/OPMEI07011/pureQuery/lib64/pqcmx
connectionSupervisorProperties
controllerURL=10.17.202.179:35055,dataSourceLookupInterval=1
---------------------------------------------------------------------------

CSC Information Section:
---------------------------------------------------------------------------
Monitored Database Name: OLT
Monitored Database Server: db6lpar4
Monitored Database Port: 5912
Platform Specific CSC Library Name:
/opm_data/OPMEI07011/pureQuery/lib64/libpqcmx.a
CSC library load: success
CSC initialization: success, 2.1
CSC Name: PQCMX
CSC Version: '9.7.0.2' 's100514' 'IP23082' '2'
PQCMX is attempting to connect to a controller server using the controllerURL
property fixed address: 10.17.202.179:35055.
PQCMX successfully connected to a controller server using the controllerURL
property fixed address: 10.17.202.179:35055 with a negotiated level: 2.
```

```
PQCMX datasource 1:db6lpar4:5912:OLT will use properties resolved after
connecting to the controller server. Resolved properties version: 1. Resolved
properties: monitorEnabled: 1, monitorServer: 10.17.202.179, monitorPort:
33366, monitorLevel: 1, monitorCollectionInterval: 60.
PQCMX monitoring for client datasource 1:db6lpar4:5912:OLT is enabled.
PQCMX datasource 1:db6lpar4:5912:OLT is attempting to connect to monitor server
10.17.202.179:33366.
PQCMX datasource 1:db6lpar4:5912:OLT is successfully connected to monitor
server 10.17.202.179:33366.
Monitoring status: on
End CSC Information Section
---------------------------------------------------------------------------

The validation completed.
```

12.2 Configuring Optim Performance Manager

Configuring Optim Performance Manager includes adding a monitored SAP database and specifying the monitoring configuration by either selecting a predefined system template or enabling monitoring profiles manually. As a result of the monitoring configuration, monitor switches and event monitors are turned on on the SAP database. Many of the SAP optimizations are implemented in the configuration part of Optim Performance Manager to ensure that

- The monitoring overhead on the SAP database does not impact the SAP system much.
- The monitoring settings that Optim Performance Manager changes on the SAP database do not change SAP required monitoring settings.
- The objects that Optim Performance Manager creates on the SAP database can easily be recognized.
- The event monitors are dropped even if Optim Performance Manger fails unexpectedly.

We describe the SAP specific configuration optimizations and hints in this section.

12.2.1 Predefined system templates for SAP

We recommend that you select one of the predefined SAP system templates when you configure the database for monitoring. The settings of the system templates result from the close work with SAP to optimize Optim Performance Manager for SAP environments. They ensure that monitoring overhead does not impact the SAP system much, but at the same time ensure that still valuable monitoring data is collected.

The predefined system templates are the following:

- SAP Business Information Warehouse production with low overhead
- SAP Business Information Warehouse production with all details
- SAP Enterprise Resource Planning production with low overhead
- SAP Enterprise Resource Planning production with all details

Those two *low overhead* templates set the same monitoring configurations, but differ in the predefined thresholds of the alerts. The same is true for two *with all details* templates.

All system templates include the following characteristics:

- The basic sampling interval is set to 5 minutes.
- The Basic, Locking, and I/O and Disk Space monitoring profiles are enabled.
  - Within the I/O and Disk Space monitoring profile, buffer pool, table spaces, and table data are collected, but no table space container data.
  - Within the Locking monitoring profile, deadlock events without history are enabled.
- The Active SQL and Connections, Workload Manager, CIM OS, and Performance Warehouse monitoring profiles are disabled
- The Extended Insight monitoring profile is enabled
  - Within the Extended Insight profile, only the collection on the client is enabled.

The *with all details* system templates have these additional characteristics:

- Within the Locking monitoring profile, lock wait information is collected with a sampling interval of 15 minutes.
- The Dynamic SQL profile is enabled with a sampling rate of 15 minutes.
- Within the Extended Insight monitoring profile, the collection of server metrics is enabled.

12.2.2 Event monitor settings in Locking monitoring profile

For a database running SAP workload, SAP recommends to set the detail level of lock event monitor data collected to `WITHOUT_HIST`. The database should not be configured to collect more details on lock events.

Therefore, during configuration, if Optim Performance Manager detects the database is a SAP database, it disallows the configuration of the lock event monitor details level in the Locking profile.

When you enable lock events in the Locking profile, Optim Performance Manager sets the corresponding database configuration parameter to the `WITHOUT_HIST` value. This is true for all lock events such as deadlock events, lock timeout events, or lock wait events. The `WITHOUT_HIST` setting minimizes the overhead that is created when the lock events are generated.

If you want to enable lock wait events, you can set a lock wait threshold in microseconds. In general, the suggested default value for a lock wait threshold is 5 000 000 microseconds, which is five seconds. If you want to set a value lower than the suggested default value, set the lock wait threshold to be 100,000 microseconds (100 milliseconds) or greater for an SAP environment. If you set a value that is lower than 100,000 microseconds, a very large number of lock wait alerts will be generated which can result in a certain overhead on the monitored database.

12.2.3 DB2 monitor switch settings

SAP requires that the snapshot switches be turned on for a DB2 database regardless of whether the database is monitored by Optim Performance Manager.

In general, Optim Performance Manager is designed to turn on DB2 monitor switches during configuration of a database based on the enabled monitoring profiles and to turn them off if you disable monitoring or unconfigure a database for monitoring. However, as a result of this requirement, none of the DB2 monitor switches are turned off by Optim Performance Manager during disabling or unconfiguring of an SAP database.

12.2.4 Controlling event monitors by using watchdog procedures

Depending on the enabled monitoring profiles, event monitors are created on the monitored SAP database. The watchdog procedures are used to drop event monitors if Optim Performance Manager is not reading and pruning the generated event monitor data on the SAP database due to a network disruption or similar system problem.

The watchdog procedures are stored procedures that Optim Performance Manager creates during configuration on an SAP database and configures to run automatically on an SAP database by registering them in the administrative task scheduler of DB2 that is available for DB2 V9.5 Fix Pack 2 or higher. If Optim

Performance Manager becomes unavailable, the watchdog procedures drop the event monitors and further data is not collected. When Optim Performance Manager becomes available again, the event monitors are re-created.

Activate the administrative task scheduler in DB2 for an SAP database before you configure it for monitoring using the following description from the DB2 Information Center:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.gui.doc/doc/c0054380.html

Table 12-1shows the monitoring profiles that create event monitors based on the settings in the profile, the corresponding event monitors, and watchdog stored procedures. *OPMID* identifies your Optim Performance Manager installation.

Table 12-1 Event monitors and corresponding watchdog procedures per monitoring profile

Monitoring profile	Event monitor	Watchdog procedure name
Extended Insight	Package Cache event monitor	`OPM_`*`OPMID`*`_PKGC_EVMON_WATCHDOG`
Extended Insight - Server metrics	UOW event monitor	`OPM_`*`OPMID`*`_UOW_EVMON_WATCHDOG`
Workload Manager	Statistic event monitor	`OPM_`*`OPMID`*`_WLMS_EVMON_WATCHDOG`
Locking	Deadlock event monitor	`OPM_`*`OPMID`*`_DLCK_EVMON_WATCHDOG`
Locking	Lock event monitor	`OPM_`*`OPMID`*`_NLCK_EVMON_WATCHDOG`

12.2.5 Identifying objects in SAP database

SAP demands that any objects created by third party products in a database that is running an SAP workload must be clearly identifiable. Therefore, Optim Performance Manager creates objects in its own schema and has defined naming conventions for the event monitors.

- Objects in the Optim Performance Manager schema:

  The objects are created in schema named *OPM*. You can easily identify the objects in your monitored database that are created by Optim Performance Manager, such as the table of event monitors.

- Naming conventions for event monitors:

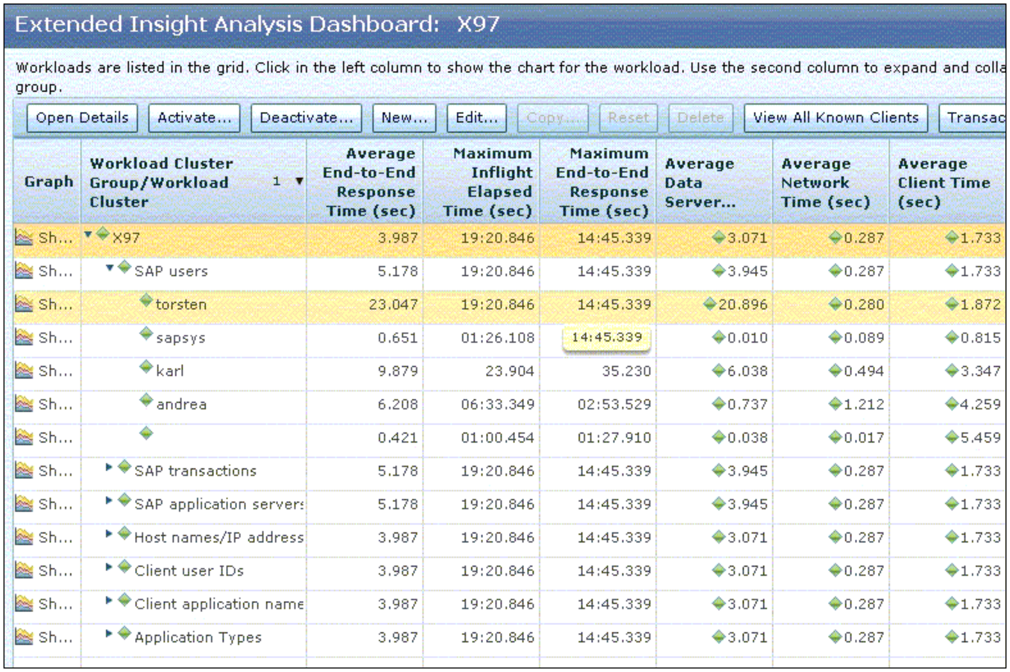
  Optim Performance Manager creates event monitors in a monitored database depending on the configuration of the monitoring profile. The event monitors are not associated with specific schemas. Therefore, Optim Performance

Manager uses specific names to identify the event monitors. The names of the event monitors that are created by Optim Performance Manager in the monitored databases all start with *OPM*.

12.3 Monitoring SAP workloads with Extended Insight

Extended Insight has predefined workload cluster groups to monitor SAP workloads, such as *SAP users* or *SAP transactions*. Using these groups helps you to pinpoint any problem and the source of the problem easily. For example, if a long running SAP Business Warehouse query slows down response time of other queries, you can find out which SAP user initiated the query, to which SAP transaction it belongs, or which SAP source modules it is using. In addition, you can find out the SQL text of this query and how this query is performing in the database.

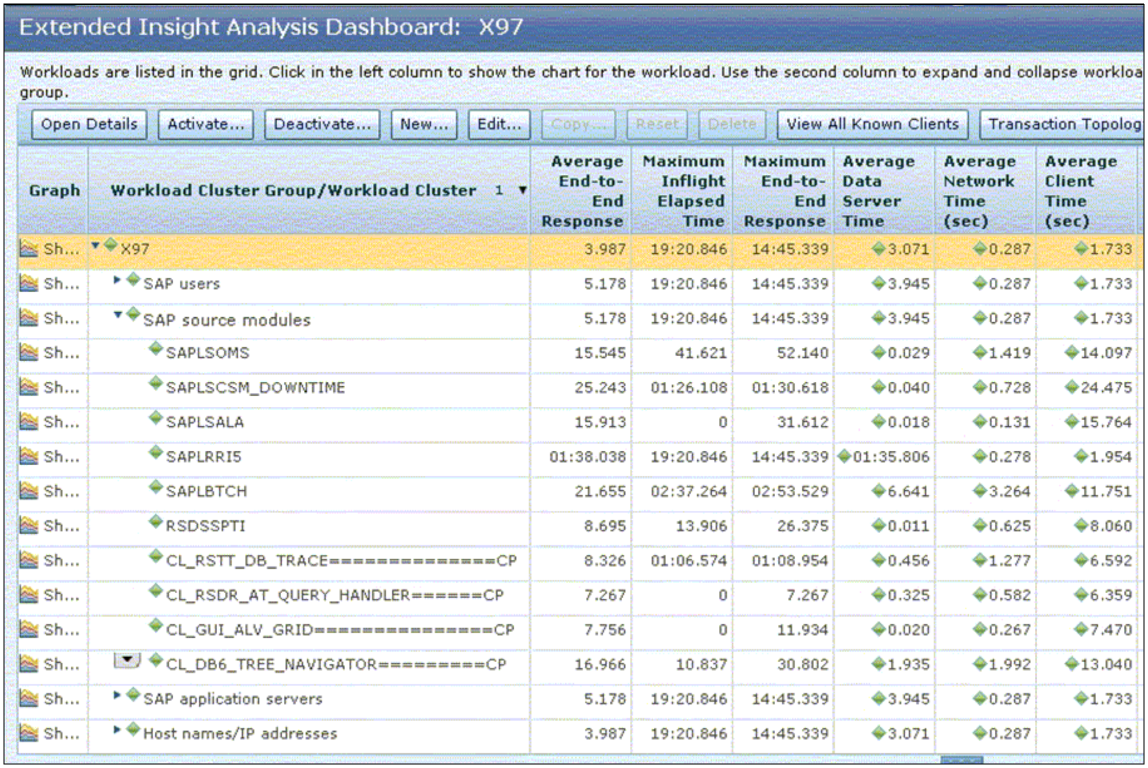
The Extended Insight dashboard in Figure 12-1 on page 432 shows a high average end-to-end response time for user `torsten`.



Graph	Workload Cluster Group/Workload Cluster	Average End-to-End Response Time (sec)	Maximum Inflight Elapsed Time (sec)	Maximum End-to-End Response Time (sec)	Average Data Server...	Average Network Time (sec)	Average Client Time (sec)
Sh...	X97	3.987	19:20.846	14:45.339	3.071	0.287	1.733
Sh...	SAP users	5.178	19:20.846	14:45.339	3.945	0.287	1.733
Sh...	torsten	23.047	19:20.846	14:45.339	20.896	0.280	1.872
Sh...	sapsys	0.651	01:26.108	14:45.339	0.010	0.089	0.815
Sh...	karl	9.879	23.904	35.230	6.038	0.494	3.347
Sh...	andrea	6.208	06:33.349	02:53.529	0.737	1.212	4.259
Sh...		0.421	01:00.454	01:27.910	0.038	0.017	5.459
Sh...	SAP transactions	5.178	19:20.846	14:45.339	3.945	0.287	1.733
Sh...	SAP application server:	5.178	19:20.846	14:45.339	3.945	0.287	1.733
Sh...	Host names/IP address	3.987	19:20.846	14:45.339	3.071	0.287	1.733
Sh...	Client user IDs	3.987	19:20.846	14:45.339	3.071	0.287	1.733
Sh...	Client application name	3.987	19:20.846	14:45.339	3.071	0.287	1.733
Sh...	Application Types	3.987	19:20.846	14:45.339	3.071	0.287	1.733

Figure 12-1 Extended Insight dashboard showing response times of SAP users

Let us see from Figure 12-2 on page 433 what SAP source modules are used for running the workload of these users. The workload spent much time in source module `SAPLRR15` because the average end-to-end response time for this source module is the highest.



Graph	Workload Cluster Group/Workload Cluster 1	Average End-to-End Response	Maximum Inflight Elapsed Time	Maximum End-to-End Response	Average Data Server Time	Average Network Time (sec)	Average Client Time (sec)
Sh...	X97	3.987	19:20.846	14:45.339	3.071	0.287	1.733
Sh...	SAP users	5.178	19:20.846	14:45.339	3.945	0.287	1.733
Sh...	SAP source modules	5.178	19:20.846	14:45.339	3.945	0.287	1.733
Sh...	SAPLSOMS	15.545	41.621	52.140	0.029	1.419	14.097
Sh...	SAPLSCSM_DOWNTIME	25.243	01:26.108	01:30.618	0.040	0.728	24.475
Sh...	SAPLSALA	15.913	0	31.612	0.018	0.131	15.764
Sh...	SAPLRRI5	01:38.038	19:20.846	14:45.339	01:35.806	0.278	1.954
Sh...	SAPLBTCH	21.655	02:37.264	02:53.529	6.641	3.264	11.751
Sh...	RSDSSPTI	8.695	13.906	26.375	0.011	0.625	8.060
Sh...	CL_RSTT_DB_TRACE==============CP	8.326	01:06.574	01:08.954	0.456	1.277	6.592
Sh...	CL_RSDR_AT_QUERY_HANDLER======CP	7.267	0	7.267	0.325	0.582	6.359
Sh...	CL_GUI_ALV_GRID===============CP	7.756	0	11.934	0.020	0.267	7.470
Sh...	CL_DB6_TREE_NAVIGATOR=========CP	16.966	10.837	30.802	1.935	1.992	13.040
Sh...	SAP application servers	5.178	19:20.846	14:45.339	3.945	0.287	1.733
Sh...	Host names/IP addresses	3.987	19:20.846	14:45.339	3.071	0.287	1.733

Figure 12-2 Extended Insight dashboard showing response times of SAP users

Let us drill down to user `torsten` by double clicking to see what SQL statements he is initiating in the SAP system. In Figure 12-3, we see that the top query the SAP user `torsten` is executing spends on average 11 minutes in the data server.

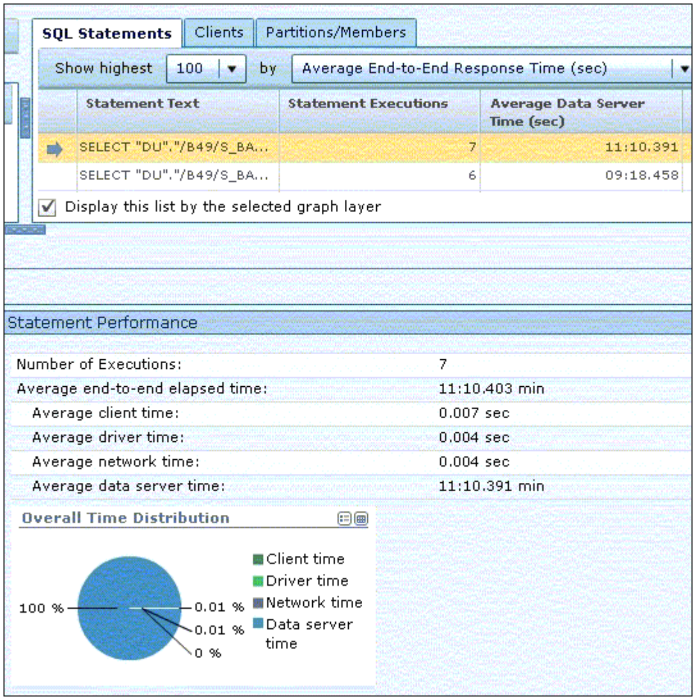


Figure 12-3 SQL statements of SAP user torsten

Figure 12-4 on page 435 shows the statement text.

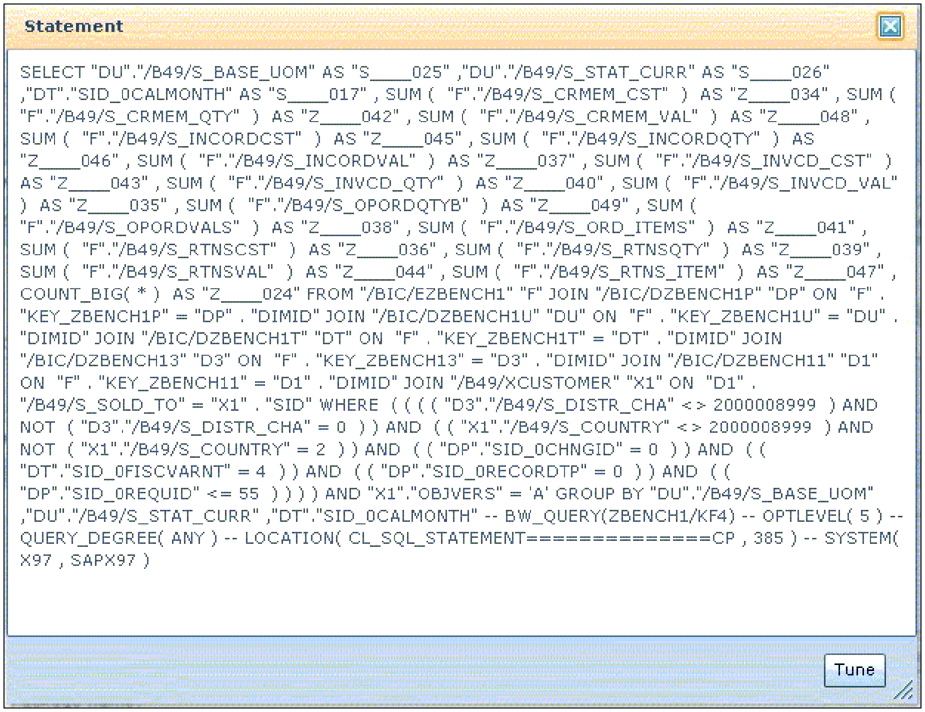


Figure 12-4 SQL statement text of long running SAP BW query

SAP applications use comments in the SQL statements that help for analysis and debugging purposes. You can identify the comments by looking for the double hyphen (--) at the end of the SQL statement. The SQL query in Figure 12-4 on page 435 includes several comments. Here is a general description of the comments that SAP applications use:

- OPTLEVEL: The optimization level which was set for DB2 Optimizer. The optimization level influences the SQL access plan.
- QUERY_DEGREE: This comment indicates the level of intra-partition parallelism which was used for execution of the query.
- LOCATION: This comment indicates the location in the SAP code, that is, method name or report name, and the code line which executed the given SQL query.
- SYSTEM: This comment indicates the SID of the SAP system where the SQL statement was executed.
- BW_QUERY: The comment is used by SAP Business Information Warehouse application after the code change implemented in the SAP Note 1379260. It has the following elements:

– BW_QUERY(*InfoProvider /QueryName*), where the *InfoProvider* element can be an InfoCube, a MultiProvider, or a DSO against which the query runs.

– *QueryName* is the name of the query.

To analyze this query further, using the Extended Insight dashboard you can check the execution details on the data server or launch Optim Query Tuner to obtain the access plan and tuning recommendations. Looking at the execution details on the data server shows that the query is reading a lot of rows, but does not return many. This is shown in Figure 12-5.

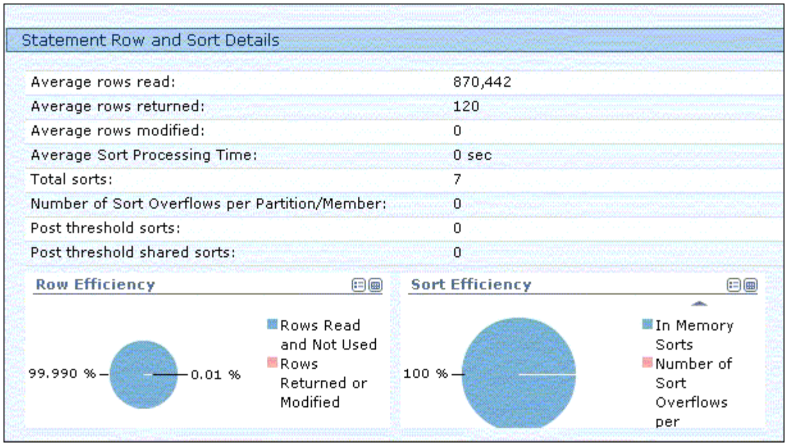


Figure 12-5 SQL exection metrics on data server

A

Managing the repository server and the repository database

This appendix contains useful information that helps you to manage Optim Performance Manager, especially, the repository server and the repository database, during run time.

The information we provide in this chapter consists of architectural concepts and troubleshooting tips and hints to maintain and administer the repository database.

A.1 Tables in the repository database

In this section, we introduce tables in the repository database. You learn

- which tables reside in which table space
- which tables store data for which monitoring profile
- which tables can get large
- which tables store collected monitoring data and which tables store metadata or configuration data

Note: This section does not explain the complete set of tables but focuses on important ones that contain collected monitoring data, meta data or configuration data.

A.1.1 Global metadata tables

The global metadata tables exist only once in the repository database. They reside in the `CONTROL` table space. Table A-1 list a few global metadata tables.

Table A-1 Global metadata tables

Table name	Schema	Description
CONNECTION_PROFILE	DB2PM	Defines the databases configured for monitoring.
DATABASES	DB2PM	Defines the databases to monitor. You can see the mapping between instance ID and monitored database by using the `D_I_INSTANCE_ID` and `D_DBNAME` columns.
INSTANCES	DB2PM	Defines the instances of the databases to monitor.
MT_COLUMN	DB2PM	Defines all columns of tables that are created if a new database is configured for monitoring. You can use this table to learn about the meaning of columns in tables. Important columns: ► `MC_COLUMN_NAME`: Column name ► `MC_TABLE_NAME`: Table the column belongs to ► `MC_DESCRIPTION`: Description for each column.

Table name	Schema	Description
MT_TABLE	DB2PM	Defines all tables that are created if a new database is configured for monitoring. You can use this table to learn about the purpose of a table Important columns: MT_TABLE_NAME: Table name MT_SCHEMA: Schema name MT_DESCRIPTION: Description for each table.
PE_SETUP	DB2PM	Defines the Optim Performance Manager level for each monitored database. A monitored database is identified using the instance ID (column ID). If you install a new version or fix pack of Optim Performance Manager, the database layout of the repository database might change, for example, new tables or new columns must be added. The migration of the database to the new layout is performed during installation. This table records the completed or uncompleted migration per instance ID by the Optim Performance Manager level per instance ID (column VERSION). At each startup of the repository server, it is checked whether the migration to the new layout is still outstanding for one or more instance IDs. If so, then the migration is performed during startup.
VERSION	DB2PM	Stores global system information such as the Optim Performance Manager version or the DB2 version.
MANAGED_DATABASE	IBMPDQ	Defines the databases that you add on the "Manage database connections" panel.
MANAGED_DATABASE_PROPS	IBMPDQ	Defines properties of the databases that you add on Manager database connections panel.

A.1.2 Metadata and protocol tables for the monitored databases

These metadata tables and protocol tables store database-specific information and there are a set of these tables each monitored database. The schema is DB2PM_<instance_id>. The metadata tables contain configuration and setup details and reside in the CONTROL table space. The protocol tables record the timestamps about the collected history data and the time frames of alerts. They reside in the SHORTTERM_<instance_id> table space.

Table A-2 Metadata and protocol tables

Table Name	Table space	Description
DB_PARTIONS	CONTROL	Contains partition information of the monitored database.
HISTORY_DATA	CONTROL	Contains the specification by which the data is collected and in which intervals. This table is changed if you change the collection intervals in the monitoring profiles using the “Configure Monitoring” wizard.
INSTANCE_INFO	CONTROL	Contains version information about the DB2 instance of the monitored database.
PARAMETER	CONTROL	Contains detailed information about configured monitoring profiles and other configurations. Any changes to the PARAMETER table are effective immediately using triggers. The PARAMETER table is changed if you change monitoring profiles using the “Configure Monitoring’” wizard. For troubleshooting purposes, you might be asked by support to change settings in the PARAMETER table.
PARTITION_ROLES	CONTROL	Contains the predefined and user-defined partition roles.
PARTITION_SETS	CONTROL	Contains partition set definitions.
PARTITION_TO_ROLES	CONTROL	Assigns roles to partitions.
PARTITION_TO_SETS	CONTROL	Assigns partitions to partition sets.
PE_THRESHOLDDEF	CONTROL	Contains the alert threshold definitions.
SMTPDESTINATION	CONTROL	Contains e-mail notification destinations.
SMTPNOTIFICATION	CONTROL	Contains e-mail notification settings
HISTORY_TOC	SHORTTERM_ <instance_id>	Contains timestamps of collected history data.
E2E_HISTORY_TOC	SHORTTERM_ <instance_id>	Contains timestamps of collected and aggregated Extended Insight data.
PE_EXCPLOG	SHORTTERM_ <instance_id>	Contains the occurred performance alerts.
PE_EXCPLOGDETAIL	SHORTTERM_ <instance_id>	Contains details about the occurred performance alerts.

A.1.3 Monitoring data tables for the monitored database

In this section, we describe the tables that store the collected data by each monitoring profile. A few tables are used by multiple monitoring profiles. For example, some tables of the Basic monitoring profile are also used for the I/O and Disk Space monitoring profile. For simplicity we list each table once only.

Basic monitoring profile

Table A-3 list the tables that belong to the Basic monitoring profile. These tables have the schema `DB2PM_<instance_id>` and reside in the `SHORTTERM_<instance_id>` table space.

Table A-3 Tables for Basic monitoring profile

Table name	Description
DB2	Contains instance information from the DBM snapshot.
DBASE	Contains database activity information from the Database snapshot.
DBCFG	Contains database configuration data.
DBMCFG	Contains database manager configuration data.
DB_STORAGE_GROUP	Contains storage group information from the Database snapshot.
DETAIL_LOG	Contains detail log information from the Database snapshot.
FCM	Contains FCM information from the DBM snapshot.
FCM_NODE	Contains FCM node information from the DBM snapshot.
HADR	Contains HADR information from the Database snapshot.
MEMSTATPOOL	Contains memory pool information from the Database and DBM snapshot.
OS_DATA_DB2	Contains memory and CPU utilization information about the monitored system retrieved by using the DB2 ENV_SYS_RESOURCE administrative procedure.
PROGRESS	Contains utility progress information from the DBM snapshot.
ROLLFORWARD	Contains rollforward information from the Database snapshot.
UTILITY_INFO	Contains utility information from the DBM snapshot.

Locking monitoring profile

Table A-4 on page 442 lists the tables that belong to the Locking monitoring profile. These tables have the schema `DB2PM_<instance_id>` and reside in the `SHORTTERM_<instance_id>` table space.

Table A-4 Tables for Locking monitoring profile

Table name	Description
APPLINLOCKCONF	Contains application locking conflict data from the Application and Lock snapshot. If you disable the collection of Lock wait information in the Locking profile then the Lock snapshot is not taken and this table is not filled.
LOCKEDRES	Contains resource locking conflict data from the Lock snapshot. If you disable the collection of Lock wait information in the Locking profile then the Lock snapshot is not taken and this table is not filled.
EVMON_*	All tables with prefix `EVMON_` contain data collected from the Lock event monitor of DB2 9.7 or higher. The tables are filled if you enabled the lock wait, lock timeout or deadlock alert in the Locking monitoring profile and if such alerts occurred.
EV_*	All tables with prefix `EV_` contain data collected from the Deadlock event monitor of DB2 9.1 or DB2 9.5. These tables are filled if you enabled deadlock alerts in the Locking monitoring profile and if such alerts occurred.

Active SQL and Connection monitoring profile

Table A-5 lists the tables that belong to the Active SQL and Connection monitoring profile. These tables have the schema `DB2PM_<instance_id>` and reside in the `SHORTTERM_<instance_id>` table space.

Table A-5 Tables for Active SQL and Connection monitoring profile

Table name	Description
APPL	Contains connection information from the Application snapshot. This table can get very large dependent on the number connections you have in the monitored database. If you run into disk space shortages then it might help to increase the collection interval of this monitoring profile.
STATEMENT	Contains statement information from the Application snapshot.
SUBSECTION	Contains statement subsection information from the Application snapshot.

I/O and Disk Space monitoring profile

Table A-6 lists the tables that belong to the I/O and Disk Space monitoring profile. These tables have the schema `DB2PM_<instance_id>` and reside in the `SHORTTERM_<instance_id>` table space.

Table A-6 Tables for I/O and Disk Space monitoring profile

Table name	Description
BUFFERPOOL	Contains buffer pool activity data from the Bufferpool snapshot.

Table name	Description
LCONTAINERS	Contains table space container information from the table space snapshot. This table can get very large dependent on the number of table space containers you have in the monitored database. If you run into disk space shortages then it might help to change the collection settings of this monitoring profile. You can for example disable the collection of container information, but continue to collect other table space information like table space activity data.
NODEIFBP	Contains buffer pool node information from the Bufferpool snapshot
NODEIFTBSP	Contains table space node information from the Tablespace snapshot. This table can get very large dependent on the number of table spaces you have in the monitored database. If you run into disk space shortages then it might help to change the collection settings of this monitoring profile. You can for example disable the collection of table space information.
TABLE	Contains table activity information from the Table snapshot. This table can get very large dependent on the number of tables you have in the monitored database. If you run into disk space shortages then it might help to change the collection settings of this monitoring profile. You can for example disable the collection of table information.
TABLEREORG	Contains table reorganization information from the Table snapshot.
TABLESPACE	Contains table space activity information from the Tablespace snapshot. This table can get very large dependent on the number of table spaces you have in the monitored database. If you run into disk space shortages then it might help to change the collection settings of this monitoring profile. You can for example disable the collection of table space information.
TBSPQUIESCER	Contains table space quiesce information from the Tablespace snapshot.
TBSPRANGES	Contains table space range information from the Tablespace snapshot

Workload Manager monitoring profile

Table A-7 lists the tables that belong to the Workload Manager monitoring profile. These table have the schema `DB2PM_<instance_id>` and reside in the `SHORTTERM_<instance_id>` table space.

Table A-7 Tables for Workload Manager monitoring profile

Table name	Description
WORK*	All tables with prefix WORK contain configuration information about defined Workload Manager objects such as WORKLOADS, WORKCLASSES, WORKACTIONS, and so on.
SERVICECLASSES	Contains configuration information about defined service classes.

Table name	Description
THRESHOLDS	Contains configuration information about defined thresholds.
HISTOGRAMBIN	Contains histogram information collected from the Statistic event monitor.
SCSTATS	Contains service class statistics collected from the Statistic event monitor.
WCSTATS	Contains work class statistics collected from the Statistic event monitor.
WLSTATS	Contains workload statistics collected from the Statistic event monitor.

Dynamic SQL monitoring profile

Table A-8 lists the tables that belong to the Dynamic SQL monitoring profile. These table have the schema DB2PM_<instance_id> and reside in the SHORTTERM_<instance_id> table space.

Table A-8 Tables for Dynamic SQL monitoring profile

Table name	Description
DYNSQL	Contains SQL data from the Dynamic SQL snapshot. This table can get very large dependent on the number of Dynamic SQL statements in the package cache. If you have run into disk space shortages then it might help to change the collection settings of the Dynamic SQL monitoring profile.

Extended Insight monitoring profile

Table A-9 lists the tables that belongs to the Dynamic SQL monitoring profile. These tables have the schema DB2PM_<instance_id> and reside in the SHORTTERM_<instance_id> table space.

Table A-9 Tables for Extended Insight monitoring profile

Table name	Description
E2E_STATEMENT_ EXECUTIONS_x	Partitioned fact table containing statement execution information from Extended Insight client. The x represents the aggregation level 1,2,3, or 4.
E2E_STATEMENT_SRV_ EXECUTIONS_x	Partitioned fact table containing statement execution information collected from the package event monitor on DB2 9.7 Fix Pack 1 or higher. The x represents the aggregation level 1,2,3, or 4.
E2E_TRANSACTION_ EXECUTIONS_x	Partitioned fact table containing transaction execution information collected from Extended Insight client and from the unit of work event monitor on DB2 9.7 Fix Pack 1 or higher. The x represents the aggregation level 1,2,3, or 4.

Table name	Description
E2E_TRANSACTION_ EXECUTIONS_M_x	Partitioned fact table containing transaction execution information on partitions or members collected from the unit of work event monitor on DB2 9.7 Fix Pack 1 or higher. The x represents the aggregation level 1,2,3, or 4.
E2E_CLIENT_ RUNTIMES_x	Partitioned fact table containing client runtime information from Extended Insight client. The x represents the aggregation level 1,2,3, or 4.
E2E_HISTOGRAMBIN_x	Partitioned fact table containing transaction execution histogram information. The x represents the aggregation level 1,2,3,or 4
E2E_*	All other tables starting with `E2E_` are dimension tables to the fact tables containing statement text, user ids, applications, and so on.

CIM OS data monitoring profile

Table A-10 lists the tables that belong to the CIM OS data monitoring profile. These tables have the schema `DB2PM_<instance_id>` and reside in the `SHORTTERM_<instance_id>` table space. Data collected by this monitoring profile is displayed on Performance Expert Client only. To collect this data you must setup a CIMON server on the monitored system that Optim Performance Manager accesses to get the data.

Table A-10 Tables for CIM OS data monitoring profile

Table name	Description
CPU	Contains CPU information.
CPUSTATISTICS	Contains CPU statistic information.
DISK	Contains information about disk drives.
DISKSTATISTICS	Contains statistical information about disk drives.
FILESYSTEM	Contains information about file systems.
OSCFG	Contains operating system information.
OSSTATISTICS	Contains operating system information.
PROCESSES	Contains processes information.

Performance Warehouse monitoring profile

Table A-11 on page 446 lists the tables that belong the Performance Warehouse monitoring profile. These table have the schema `PWH_<instance_id>` and reside in the `LONGTERM_<instance_id>` table space.

Table A-11 Tables for Performance Warehouse monitoring profile

Table name	Description
EVM_*	Contains statement or activity information collected from the statement or activity event monitor if you start SQL or Workload Manager activity traces from Performance Expert Client.
*	All tables that have the same table name in schema `PWH_<instance_id>` as a table in schema `DB2PM_<instance_id>` contain the same information, but the data is aggregated. See chapter A.3.2, "Aggregating inflight and workload manager data" on page 452 for information on data aggregation for long-term history.

A.2 How the repository server works

In Figure 1-1 on page 12 we introduced the architecture of Optim Performance Manager. Optim Performance Manager consists of a console server and a repository server component. The repository server collects monitoring data from the monitored database and stores it in the repository database. The repository server starts up if you call the **`pestart`** command or on Windows use the **`Start the Repository Server`** entry from the program start menu. The repository server logs the startup in the `db2pesrv.log` file and writes it to the console if you used the `pestart` command.

Let us use the log messages to explain how the repository server works. The repository server is a java process which starts for each monitored database a set of threads. The number of threads in this set depend on the enabled monitoring profiles for a monitored database. Each thread has it's own purpose. The set of threads that the repository server java process starts for one monitored database is called a monitoring instance. Therefore each monitored database has it's assigned instance ID which is used at several places like in the table schema name and in the table space names belonging to one monitored database, for example schema `DB2PM_<instance_id>` or table space `<SHORTTERM_<instance_id>`. The repository server writes global log messages and log messages for each monitored database to the log file. Example 12-4 shows one global log message and all log messages for one monitored database with instance ID 1 that has all monitoring profiles enabled except `CIM OS Data`.

Example 12-4 Startup log messages for one monitored database

```
[17:34:29.687][0]The Extended Insight controller server is started on port
7777.
[17:34:38.984]Starting the monitoring for [1] ...
[17:34:54.125]The monitoring of [1] is started. [Node: NODE7507, Host:
BL3AED4M, Port: 50000, OS: WINDOWS, DB2: DB2 v9.7.100.177]
```

```
[17:34:54.203][1]Initializing ...
[17:34:54.203][1]Databases [PEDEMO]
[17:34:54.203][1]Working directory = C:\Program
Files\IBM\OPM\RepositoryServer\instances\DB2\NODE7507
[17:34:54.203][1]Trace file       = C:\Program
Files\IBM\OPM\RepositoryServer\instances\DB2\NODE7507\db2pesrv.trc
[17:34:54.203][1]Stored procedures trace file = C:\Program
Files\IBM\OPM\RepositoryServer\instances\DB2\NODE7507\fpesnap.trc
[17:34:54.203][1]CIM Trace file  = C:\Program
Files\IBM\OPM\RepositoryServer\instances\DB2\NODE7507\fpecim.trc
[17:34:54.437]Status of [1]: initializing.
[17:35:01.359][1]Extended Insight data aggregator started.
[17:35:02.968][1]Extended Insight data loader started.
[17:35:09.75 ][1]Extended Insight data merger started.
[17:35:10.14 ][1]Extended Insight started.
[17:35:10.171][1]The Extended Insight monitor server is started on port 2232.
[17:35:11.89 ][1]Performance warehouse server started.
[17:35:12.625][1]Threshold alert processor started.
[17:35:12.765][1]Performance warehouse data loader started.
[17:35:13.078][1]Data cleaner started.
[17:35:13.718][1]Snapshot data collector started.
[17:35:32.64 ][1]Workload management statistic collector started.
[17:35:32.64 ][1]Workload management definition collector started.
[17:35:32.64 ][1]Operating system data collector started.
[17:35:33.39 ][1]Deadlock event alert processor started.
[17:35:37.703][1]Extended Insight statement text collector started.
[17:35:38.093][1]Extended Insight transaction metric collector started.
[17:35:38.546][1]Extended Insight statement metric collector started.
[17:35:39.343]Status of [1]: alive (2/4).
```

The number in brackets, in our example `[0]` or `[1]` tell you the instance ID. Instance ID `[0]` means that it is a global message and not belonging to a specific database.

The first messages you see for instance ID 1 gives you the info which database is associated with instance ID 1 and which trace files are used to write trace messages. The remaining messages tell you which threads are started to monitor the database. We describe the threads in the following.

A.2.1 Extended Insight threads

Before describing the Extended Insight threads, we give a few sentences about the architecture of Extended Insight within the repository server. Extended Insight consists of a global controller and a monitor server for each monitored database for that the Extended Insight monitoring profile is enabled. The controller listens on a port that you specify when you activate Extended Insight on Optim Performance Manager. The port is saved in the `pdq.properties` file. When you configure an Extended Insight client, you specify this port as well and it is saved in the `pdq.properties` file on the Extended Insight client machine. In this way, the communication between Extended Insight client and the repository server is established. When an application that you monitor with Extended Insight client starts and connects to the monitored database, the Extended Insight client accesses the controller and asks for the Extended Insight monitor server port number that is listening for the Extended Insight data for that monitored database. From that point on, the Extended Insight client sends the collected data to the monitor server for that database over the communicated monitor server port.

In our example, we see messages telling that the controller and the monitor server for database with instance ID 1 is started and which port numbers are used:

```
[0]The Extended Insight controller server is started on port 7777
[1]The Extended Insight monitor server is started on port 2232.
```

The port number of the monitor server is determined dynamically unless you set it explicitly in the Extended Insight monitoring profile.

The Extended Insight threads include:

- *Data aggregator*: This thread aggregates Extended Insight data into the next aggregation level. For details, see A.3.1, “Aggregating extended insight data” on page 450.
- *Data loader*: Stores collected Extended Insight data in the appropriate tables in the repository database.
- *Data merger*: Combines received data from Extended Insight client with collected unit of work event monitor data from the monitored database in order to insert the data from both sources into a single table.
- *Statement text collector*: Extended Insight client does not send the complete statement text of the SQL statements the applications execute to the monitor server. It sends a unique hash code. This thread retrieves SQL statements from the dynamic SQL statement cache of the monitored database either through the dynamic SQL snapshot or the package cache event monitor dependent on the DB2 version of the monitored database. It then calculates a hash code for them and stores them in the appropriate table. Through mapping the hash codes, the SQL statements that the applications execute are identified.
- *Transaction metric collector*: This thread collects unit or work event monitor data from the monitored database for DB2 9.7 Fix Pack 1 or higher.

- *Statement metric collector*: This thread collects data server statement execution details for static and dynamic statements using the package cache listing table function `MON_GET_PKG_CACHE_STMT` on the monitored database for DB2 9.7 Fix Pack 1 or higher.

A.2.2 Other threads

Other threads in the repository database include:

- *Performance Warehouse server*: This thread is only started if you enabled the Performance Warehouse monitoring profile. It allows scheduling Performance Warehouse processes and running SQL or Workload Manager activity traces from Performance Expert Client.
- *Performance Warehouse data loader*: This thread is only started if you enabled the Performance Warehouse monitoring profile. It aggregates data from the history tables into the long-term history tables. For more details see A.3.2, “Aggregating inflight and workload manager data” on page 452
- *Threshold alert processor*: This thread checks in fixed intervals whether threshold alerts occurred or ended
- *Data cleaner*: This thread deletes history data after the retention time is reached. For more details see A.4.1, “How the repository server deletes history data” on page 454
- *Snapshot™ data collector*: This thread is started whenever an inflight monitoring profile is enabled. It collects snapshot data from the monitored database in specified intervals. To collect the data, it executes a stored procedure. This stored procedure attaches to the DB2 instance of the monitored database, retrieves the snapshot using the DB2 snapshot API, detaches and stores the retrieved snapshot data in the appropriate tables. Stored procedures execute in the `db2fmp` processes started by DB2. Therefore, you see on the Optim Performance Manager machine multiple `db2fmp` processes running and consuming memory and CPU. The amount of memory and CPU they consume per collection depend on the amount of retrieved monitoring data.
- *Workload Management statistic collector*: This thread is started if the Workload Manager monitoring profile is enabled. It collects statistic event monitor data from the monitored database.
- *Workload Management definition collector*: This thread is started if the Workload Manager monitoring profile is enabled. It collects Workload Manager objects definition data from the catalog tables in the monitored database.
- *Operating System data collector*: This thread is started if the Basic monitoring profile is enabled. In the specified collection interval, it collects CPU and

memory consumption data from the monitored database using the DB2 `ENV_SYS_RESOURCES` administrative procedure.

- *Deadlock event alert processor*: This thread is started if the lock or deadlock event monitor is enabled in the Locking profile. It collects lock or deadlock event monitor data from the monitored database.

A.3 Data aggregation concepts

The repository server of Optim Performance Manager aggregates collected history data over time in order to provide a long-term history of data that can be used for trend and capacity purposes. This history data does not consume much space in the repository database. At time of writing this book, the inflight history data and extended insight history data have different aggregation concepts:

- Extended insight history data: Optim Performance Manager aggregates the data using four aggregation levels to achieve this goal: The older the data, the more it is aggregated.
- Inflight and workload manager history data: Optim Performance Manager uses the legacy Performance Warehouse functionality available from the legacy Performance Expert Client to aggregate data.

The report data in Optim Performance Manager is not aggregated at the time of writing this book. The reports that you can create from the Optim Performance Manager web console are based on the detailed history data that is deleted automatically after the retention period is reached. To achieve that you can create reports on data collected over a longer period of time than the retention period of the detailed history data you can configure a monitored database twice. One configuration collects performance data in short intervals with a retention time of a few days. The second configuration collects performance data in long intervals with a retention time of a few weeks or month. The second configuration is used for reports. The following technote describes how you can configure a monitored database in Optim Performance Manager twice to use the reports for trend and capacity purposes:

http://www-01.ibm.com/support/docview.wss?uid=swg21429964

A.3.1 Aggregating extended insight data

Optim Performance Manager uses four aggregation levels to aggregate and store extended insight data. Each aggregation level has a storage period. Optim Performance Manager deletes the data from an aggregation level automatically after the storage period is over. The storage period for aggregation level 1 is the shortest and for aggregation level 4 the longest.

Every minute Optim Performance Manager receives extended insight data from the Extended Insight clients and stores it in aggregation level 1. The extended insight data it receives contains transaction response time data that the Extended Insight client has aggregated already within one minute. Aggregation means that average and maximum response times, including the time breakdown and SQL statements for transactions with the same connection attributes, are calculated. After 15 minutes of collection, Optim Performance Manager reads the data from aggregation level 1 and aggregates it into aggregation level 2. After one hour of collection, Optim Performance Manager reads the data of aggregation level 2 and aggregates it into aggregation level 3. The aggregation to level 4 happens first after one day of collection. Figure A-1 on page 451 shows the aggregation concept.

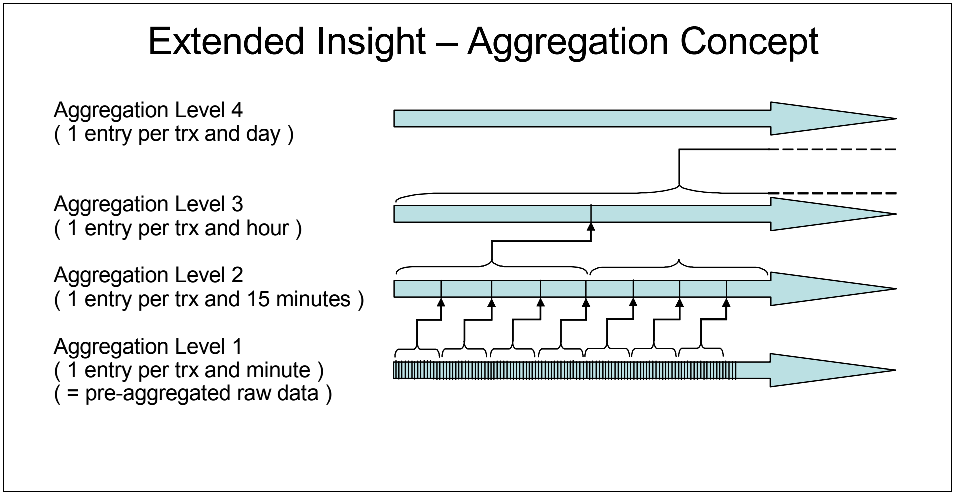


Figure A-1 Extended Insight data aggregation

Until the storage period for aggregation level 1 is reached, data for all four aggregation levels exist in parallel. If the data becomes older than the storage period of aggregation level 1, it only exists in aggregation level 2 to 4, if the data becomes older than the storage period of aggregation level 2, it only exists in aggregation levels 3 and 4; and so on.

Optim Performance Manager stores the data in the partitioned tables. One partitioned table exists for each aggregation level. The aggregation level is appended to the table name. For example the transaction execution information is stored in the following tables:

- `E2E_TRANSACTION_EXECUTIONS_1` - Contains transaction execution information for aggregation level 1.
- `E2E_TRANSACTION_EXECUTIONS_2` - Contains transaction execution information for aggregation level 2.

- `E2E_TRANSACTION_EXECUTIONS_3` - Contains transaction execution information for aggregation level 3.
- `E2E_TRANSACTION_EXECUTIONS_4` - Contains transaction execution information for aggregation level 4.

To display the extended insight data on Extended Insight Dashboard, Optim Performance Manager chooses the optimal aggregation level. For example, if you select a monitoring time frame of a few hours in history and the end of storage period of aggregation level 1 is within these hours, then the data from aggregation 2 is displayed for the whole monitoring timeframe. Another example is that if you select a long monitoring time frame and the data of aggregation level 2 is available for the whole monitoring time frame but would result in too many data points to be displayed on the Extended Insight dashboard, then Optim Performance Manager uses data from aggregation level 3 to display.

A.3.2 Aggregating inflight and workload manager data

Optim Performance Manager aggregates inflight history data only if you enable the Performance Warehouse monitoring profile during monitoring configuration of a database. A screenshot of the Performance Warehouse profile is shown in Figure A-2. The `Aggregation period in minutes` specifies how many data entries are aggregated. The default setting of 60 minutes means that all data that Optim Performance Manager collected within 60 minutes is aggregated to one data entry. The `Aggregation run interval in hours` specifies how often the aggregation step takes places. The default setting of two hours means that every two hours, the collected data of the last two hours is read and aggregated according to the aggregation period setting.

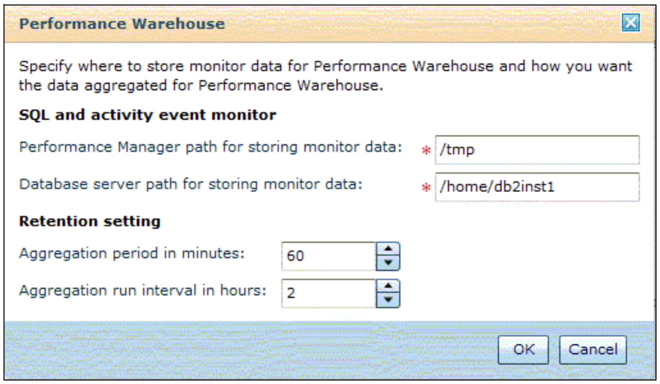


Figure A-2 Performance Warehouse monitoring profile

> **Note:** The event monitor paths that you specify in the Performance Warehouse profile have nothing to do with the aggregation. From Performance Expert Client, you can run SQL or activity traces based on event monitors. These trace functions belong to Performance Warehouse but the collected data is not involved in any aggregation.

Optim Performance Manager stores the aggregated data in tables in schema `PWH_<instance_id>`. The table name is the same as for the non-aggregated data, only the schema is different. For example, the inflight data about the database activity is saved in `DB2PM_<instance_id>.DBASE` and the aggregated data is saved in `PWH_<instance_id>.DBASE`.

Aggregation is only performed for the inflight data that is used in the Performance Warehouse analysis functions, such as trend analysis, reporting, queries, or rules-of-thumb. You can use Performance Warehouse analysis functions from Performance Expert Client only.

The following data is aggregated:

- Inflight data about database, buffer pool, table space, and table activity
- Database manager and database configuration data
- DB2 Workload Manager data
- Optional operating system data if CIM OS Data monitoring profile is enabled

The following data is not aggregated:

- Inflight data about connections, locks and dynamic SQL
- Inflight data about table space disk usage and containers

A.4 Deleting data from the repository database

Optim Performance Manager's repository server deletes the collected data that resides in the `SHORTTERM_<instance_id>` table space automatically once the specified retention period is reached. The `SHORTTERM_<instance_id>` table space stores all the data that you can monitor from the Optim Performance Manager web console. In this chapter, we refer to this data as *history data*. You can set the retention period using configuration wizard opened from the Manage Database Connection panel. In A.4.1, "How the repository server deletes history data", we provide the algorithm that the Optim Performance Manager repository server uses to delete the data.

If disk space for the history data is a concern, you can manage disk space consumption by adapting the monitoring profiles using the Configure Monitoring wizard. You can

- shorten the retention time.
- increase the collection intervals.
- disable monitoring profiles completely or partially.

2.4, “Capacity planning” on page 34 provides the formulas that you can use to calculate your disk space consumption in advance. A.1, “Tables in the repository database” on page 438 provides which tables are affected if you increase the collection intervals of monitoring profiles or disable monitoring profiles. For example, if you have already collected data for a while, then you know which tables are the biggest. Based on that, you could adapt the appropriate collection settings.

If you have to delete data from the history data immediately, for example, you receive the disk full error messages in the `db2pesrv.log` file, you can use the `delhistory` script that comes with Optim Performance Manager. A.4.2, “Deleting history data using the delhistory script” describes how to use this script. The `delhistory` script does not delete extended insight data. To delete extended insight data manually, we provide a few hints in A.4.3, “Deleting extended insight data manually” on page 458

If you collect long-term history data by having the Performance Warehouse monitoring profile enabled, you have to maintain the data manually. The long-term history data is saved in the `LONGTERM_<instance_id>` table space. The size of the `LONGTERM_<instance_id>` table space is usually much smaller than of the `SHORTTERM_<instance_id>` table space because Optim Performance Manager stores aggregated data in the long-term history. It is useful to keep this data for a long time (for example, months or even years) for trend analysis or capacity planning purposes. Refer to A.4.4, “Deleting aggregated long-term history data manually” on page 459 for maintaining the long-term history data manually.

A.4.1 How the repository server deletes history data

When configuring a database for monitoring, you can specify the retention time and storage period of the collected data as shown in Figure A-3 on page 455. The data retention time is used for history data that you can display on the Inflight Dashboards, the Workload Manager configuration, or in the Reports. The storage period is used for data that you can display on the Extended Insight Dashboard.

When the retention or storage period is reached, The repository server of Optim Performance Manager deletes the data automatically.

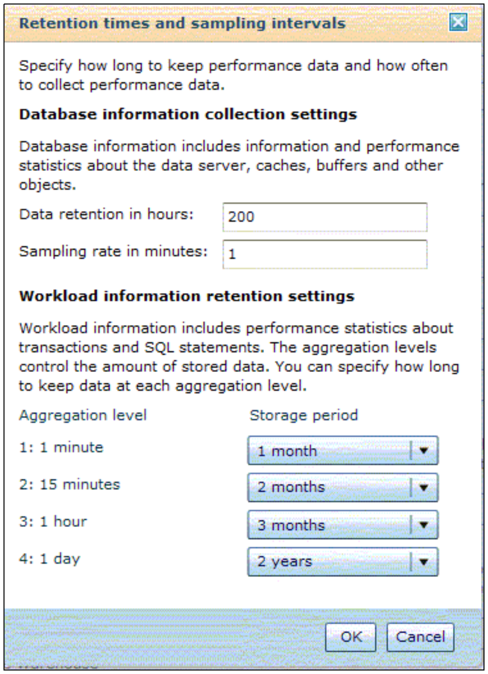


Figure A-3 Retention times and storage period settings

Deleting inflight, workload manager, and report history data

The inflight and report history data is stored in the tables with schema `DB2PM_<instance_id>` and reside in the table space `SHORTTERM_<instance id>`. All tables containing monitoring data in schema `DB2PM_<instance_id>` that do not begin with prefix `E2E` belong to this category. Optim Performance Manager uses the following approach to delete the history inflight, workload manager, and report data:

- Every half an hour the repository server checks for data that should be deleted. The first check takes place after the repository server run for half an hour.
- The retention time is calculated between the newest data and the oldest data in a table containing monitoring performance data for a monitored database. If a database is currently not monitored, no new data is coming in. In this case the history data is not deleted. This avoids having an empty history if a database is not monitored for a while.
- If the repository server runs once a while only, then the data volume that exceeds the retention period and should be deleted can be high. If the repository server would try to delete all the to-be-deleted data at once, it can lead to log full problems in the repository database. To avoid log full problems,

the repository server deletes, at one time, only the amount of data that was collected in a time frame which is twice of the cleanup interval. The cleanup interval is 30 minutes (see first bullet). This means that the repository server deletes every half hour data of a one-hour interval. It can take a while for the repository server to catch up with the data deletion if it does not run constantly.

Deleting Extended Insight history data

Extended Insight history data is stored in partitioned fact tables in schema DB2PM_<instance_id> and reside in table space SHORTTERM_<instance_id>. The table names begin with prefix `E2E`. Optim Performance Manager uses the following approach to delete the Extended Insight history data:

- Every half an hour Optim Performance Manager repository server checks for data that should be deleted. The first check takes place after the repository server run for half an hour.
- The storage period is calculated between the newest data and the data of the oldest partition in a partitioned table storing extended insight data for a monitored database. If a database is currently not monitored then no new data is coming in. In this case the history data is not deleted. This avoids having an empty history if a database is not monitored for a while. The time frame.
- Each partition contains data of a fixed range. Only if the complete range of a partition exceeds the storage period the partition is detached and dropped. The partition ranges for the four aggregation levels are the following:
  - Aggregation Level 1: 2 hours
  - Aggregation Level 2: 12 hours
  - Aggregation Level 3: 168 hours
  - Aggregation Level 4: 720 hours

A.4.2 Deleting history data using the `delhistory` script

You can use the `delhistory` script to delete history data that you can display on the Optim Performance Manager web console on the Inflight Dashboards, Workload Manager configuration or in the Reports.

> **Note:** The `delhistory` script does not delete history data that you can monitor from the Extended Insight dashboard.

The `delhistory` script is available in the `Repository/bin` directory of the Optim Performance Manager installation. On Windows, it is named `delhistory.bat` and on Linux and UNIX, it is named `delhistory.sh`. Run the `delhistory` script without any parameters to show the syntax. See Example A-1 for Windows.

Example A-1 delhistory syntax

```
C:\Program Files\IBM\OPM\RepositoryServer\bin>delhistory
Usage: delhistory nodename hours [dbname]
Description:
Delete all rows from history tables in the specified monitored instance, which
are older than the specified number of hours.
Parameter 'nodename' specifies the nodename of the remote monitored instance as
defined in OPM Server (displayed by 'peconfig').
Optional parameter: 'dbname' is the name of the OPM database, default is
'PERFDB
'Examples:
  delhistory NODE9194 24    : deletes all rows older than 1 day
  delhistory NODE9194 0     : deletes all rows
Attention: Run this script only when the server does not currently collect
history for this monitored instance.
```

The script exports the remaining data that should not be deleted into the temporary directory, for example `/tmp` on Linux and UNIX, and imports it back using the `REPLACE` option. Using the **`EXPORT`** and **`IMPORT`** commands, the data deletion is not logged and therefore avoids that you run into log full problems during data deletion.

> **Tip:** Before you run the `delhistory` script check the available free space in the temporary directory and increase it if necessary.

If your table spaces are SMS table spaces, the free disk space is given back immediately to the operating system after deleting the data. But if your table spaces are DMS or Automatic Storage table spaces, the disk space is released back to the operating system only if the table space high water mark is lowered as well.

Refer to DB2 Information Center for more information about the high water mark:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.dbobj.doc/doc/c0055399.html

Refer to DB2 Information Center for more information about how the storage is released back to the operating system:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.dbobj.doc/doc/c0055392.html

A.4.3 Deleting extended insight data manually

Extended Insight history data is stored in partitioned fact tables in schema DB2PM_<instance_id> and reside in table space SHORTTERM_<instance_id>. The table names begin with prefix E2E. If you must delete data immediately because of disk spaces shortages and you cannot wait until Optim Performance Manager deletes the data automatically, use the following approach:

- Use the **db2look** command to determine the DDL and the partition names of the following partitioned tables, where <x> refers to the aggregation level:
  - E2E_STATEMENT_EXECUTIONS_<x>
  - E2E_STATEMENT_SRV_EXECUTIONS_<x>
  - E2E_TRANSACTION_EXECUTIONS_<x>
  - E2E_TRANSACTION_EXECUTIONS_M_<x>

  Example A-2 is an example for a DDL returned by the **db2look** command. The columns names are replaced by dots since they are not important here.

  Example A-2 Extended Inisght history data table DDL

```
CREATE TABLE "DB2PM_7 "."E2E_STATEMENT_SRV_EXECUTIONS_3"  (....)
VALUE COMPRESSION
PARTITION BY RANGE("COLLECTION_TIMESTAMP")
(PART "PART0" STARTING(MINVALUE) ENDING('1970-01-01-00.00.00.000000')
EXCLUSIVE IN "SHORTTERM_7", PART "1282348800000"
STARTING('2010-08-21-00.00.00.000000') ENDING('2010-08-28-00.00.00.000000')
EXCLUSIVE IN "SHORTTERM_7", PART "1282953600000"
STARTING('2010-08-28-00.00.00.000000') ENDING('2010-09-04-00.00.00.000000')
EXCLUSIVE IN "SHORTTERM_7", PART "1283558400000"
STARTING('2010-09-04-00.00.00.000000') ENDING('2010-09-11-00.00.00.000000')
EXCLUSIVE IN "SHORTTERM_7", PART "1284163200000"
STARTING('2010-09-11-00.00.00.000000') ENDING('2010-09-18-00.00.00.000000')
EXCLUSIVE IN "SHORTTERM_7", PART "1284768000000"
STARTING('2010-09-18-00.00.00.000000') ENDING('2010-09-25-00.00.00.000000')
EXCLUSIVE IN "SHORTTERM_7", PART "1287187200000"
STARTING('2010-10-16-00.00.00.000000') ENDING('2010-10-23-00.00.00.000000')
EXCLUSIVE IN "SHORTTERM_7");
```

- Detach a partition by altering the tables and then drop the table created from the detach step. Start with the oldest partition in a table.

  This is an example for detaching and dropping a partition from the DDL listed in Example A-2:

```
ALTER TABLE DB2PM_7.E2E_STATEMENT_SRV_EXECUTIONS_3 DETACH PART
“1282348800000” INTO junk;
DROP TABLE junk;
```

A.4.4 Deleting aggregated long-term history data manually

Here we provide a script for you to delete the aggregated long-term history data. Long-term history data is stored if the Performance Warehouse monitoring profile is enabled. The data is stored in tables in schema `PWH_<instance_id>`. The tables reside in the `LONGTERM_<instance_id>` table space. The script shown in Example A-3 contains a set of `DELETE` statements that delete data older than a specified timestamp from the long-term history tables.

Example A-3 Long-term data deletion script

```
#! /usr/bin/ksh

###############################################################
#   call the script as DB2 instance owner
#
# Syntax:
# -d PE database name   (required)
# -i instance ID        (required)
# -t deletion timestamp in format yyyy-mm-dd (required)
###############################################################

errtrap()   {
         es=$?
         echo "ERROR line $1: Command exited with status $es."
         exit $es
         }
trap 'errtrap $LINENO' ERR
cleanup() {
         exit 256
         }
trap cleanup INT TERM
SYNTAX ()
{
cat << EOF
Syntax
-d PE database name      (required)
-i instance ID           (required)
-t deletion timestamp in format yyyy-mm-dd  (required)
EOF
}
#***** Main line code******#

while getopts ":od:i:t:" opt; do
        case $opt in
             d ) PEDBNAME=$OPTARG
             if [ -z "$PEDBNAME" ]; then
             SYNTAX
             exit -1
             fi;;
             i ) PEINSTID=$OPTARG
             if [ -z "$PEINSTID" ]; then
             SYNTAX
             exit -1
             fi;;
             t ) PEDATE=$OPTARG
             if [ -z "$PEDATE" ]; then
             SYNTAX
             exit -1
             fi;;
             o ) set -o xtrace;;
             /? ) SYNTAX
             exit -1
```

```
        esac
done

## connect the PE server database
trap - ERR
(db2 -o- CONNECT TO $PEDBNAME)
PECONNECTED=$?
trap 'errtrap $LINENO' ERR
if (( $PECONNECTED!=0 )); then
    print "Failed to connect to the database return code="$PECONNECTED
    exit $PECONNECTED
    fi
#################

## delete from the PWH tables

(db2 -t -n "DELETE from PWH_$PEINSTID.DBASE where INTERVAL_TO
             < '$PEDATE-00.00.00.000000';")
(db2 -t -n "DELETE from PWH_$PEINSTID.BUFFERPOOL where INTERVAL_TO
             < '$PEDATE-00.00.00.000000';")
(db2 -t -n "DELETE from PWH_$PEINSTID.TABLE where INTERVAL_TO
             < '$PEDATE-00.00.00.000000';")
(db2 -t -n "DELETE from PWH_$PEINSTID.TABLESPACE where INTERVAL_TO
             < '$PEDATE-00.00.00.000000';")
(db2 -t -n "DELETE from PWH_$PEINSTID.DBCFG where INTERVAL_TO
             < '$PEDATE-00.00.00.000000';")
(db2 -t -n "DELETE from PWH_$PEINSTID.DBMCFG where INTERVAL_TO
             < '$PEDATE-00.00.00.000000';")

## delete data from PWH OS tables. They only contain data if CIM is enabled

#(db2 -t -n "DELETE from PWH_$PEINSTID.OSCFG where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.CPU where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.DISK where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.OSSTATISTICS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.CPUSTATISTICS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.DISKSTATISTICS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.FILESYSTEM where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")

## delete data from PWH WLM tables. They only contain data if DB2 V9.5
## is monitored

#(db2 -t -n "DELETE from PWH_$PEINSTID.SCSTATS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.WCSTATS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.WLSTATS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.HISTOGRAMBIN where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.SERVICECLASSES where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.THRESHOLDS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.WORKACTIONS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.WORKACTIONSETS where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.WORKCLASSES where INTERVAL_TO
#             < '$PEDATE-00.00.00.000000';")
```

```
#(db2 -t -n "DELETE from PWH_$PEINSTID.WORKCLASSSETS where INTERVAL_TO
#            < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.WORKLOADCONNATTR where INTERVAL_TO
#            < '$PEDATE-00.00.00.000000';")
#(db2 -t -n "DELETE from PWH_$PEINSTID.WORKLOADS where INTERVAL_TO
#            < '$PEDATE-00.00.00.000000';")

## delete data from SQL activity traces

#(db2 -t -n "DELETE from DB2PM_$PEINSTID.LOADLOG where LL_STARTTS
#            < '$PEDATE-00.00.00.000000' and LL_LOADTYPE='STMT';")

## delete data from WLM activity traces

#(db2 -t -n "DELETE from DB2PM_$PEINSTID.LOADLOG where LL_STARTTS
#            < '$PEDATE-00.00.00.000000' and LL_LOADTYPE='ACTIVITY';")

(db2 -o- connect reset)
exit 0
```

A.5 Automatic runstats and reorganization in the repository database

The Optim Performance Manager installer creates the repository database and enables automatic table maintenance in the database configuration. For DB2 9.7, this action results to the configuration parameter settings for the repository database as shown in Figure A-4.

```
Automatic maintenance                          (AUTO_MAINT) = ON
  Automatic database backup                (AUTO_DB_BACKUP) = OFF
  Automatic table maintenance              (AUTO_TBL_MAINT) = ON
    Automatic runstats                      (AUTO_RUNSTATS) = ON
      Automatic statement statistics      (AUTO_STMT_STATS) = ON
    Automatic statistics profiling        (AUTO_STATS_PROF) = OFF
      Automatic profile updates             (AUTO_PROF_UPD) = OFF
    Automatic reorganization                   (AUTO_REORG) = ON
```

Figure A-4 Automatic maintenance settings

Automatic table maintenance includes automatic reorganization and automatic statistic collection. The automatic statistic collection is an online activity, but the automatic reorganization is an offline activity. This means that you must define an offline maintenance window to let the reorganization activity take place. You can specify an offline maintenance window using the Configure Automatic Maintenance wizard from DB2 Control Center.

For online activities, a default maintenance window is already set. You can change it using the Configure Automatic Maintenance wizard.

Note: Optim Performance Manager requires that the automatic maintenance is enabled for the repository database. If you disable the automatic maintenance manually in the database configuration, then Optim Performance Manager enables it again.

A.6 Backing up the repository database

Whether you implement a backup strategy for the repository database depends on the importance of the collected data. Here are a few considerations that may help to decide how often you back up the repository database:

- As a minimum backup, initially, back up the repository database after you configured your monitored databases. This action prevents you from loosing any configurations you have done. Additionally, back up the repository database whenever you have performed major changes to the configuration.
- If you collect performance data mainly for troubleshooting purposes, and have set retention times of a few days only, then the minimum backup concept might be suitable for you.
- If you collect long-term data or have set long retention times for trend and capacity planning purposes then backup that repository database regularly in order to not loose the long-term data.

A.7 Changing database configuration of the repository database

To benefit from the DB2 autonomic features, Optim Performance Manager creates the repository database with self tuning memory set to on. In addition, all database configuration parameters that are set to AUTOMATIC by default are not changed by Optim Performance Manager.

For some non-automatic configuration parameters, Optim Performance Manager requires initial settings and changes the configuration parameters after creating the repository database. Examples of changed parameters are:

- `AUTO_MAINT`
- `AUTO_RUNSTATS`
- `AUTO_REORG`
- `AUTO_TBL_MAINT`
- `CATALOGCACHE_SZ`
- `CHNGPGS_THRESH`

- LOGFILSIZ
- LOGBUFSZ
- LOGPRIMARY
- LOGSECOND
- SOFTMAX
- UTIL_HEAP_SZ
- LOCKTIMEOUT

In general, you can adapt database configuration parameters of the repository database if your environment requires that. Optim Performance Manager checks whether the parameters that require initial settings are changed. If numeric parameters are changed to lower values or string parameters are set to a different value, Optim Performance Manager changes them back to the setting it requires. If the numeric parameters are changed to higher values than required, Optim Performance Manager does not touch them. Optim Performance Manager does this check during the following operations:

- You start the **peconfig** configuration tool.
- You configure a database for monitoring in the web console or change an existing configuration.
- You enable or disable monitoring of a database in the web console.

If you have to change configuration parameters, it is most likely that you have to increase parameters that affect logging such has LOGFILSIZ, LOGPRIMARY, or LOGSECOND. The amount of log records depend on the amount of monitoring data that is inserted and deleted from the repository database for all monitored database. If the repository database runs out of log space then you receive the SQL0964N error message in the db2pesrv.log file. If you get this error repeatedly then increase the following configuration parameters:

- LOGFILSIZ
- LOGPRIMARY
- LOGSECOND

Make sure that you have enough space available in the path used for the log files. Check the Path to log files in the database configuration which path is used. We recommend that you separate log files into a separate path that does not include table spaces of the repository database in order to avoid I/O bottlenecks. You can change the log path by setting the NEWLOGPATH configuration parameter.

By default, the repository database is set up for circular logging. If you prefer another log method, you can change the database configuration appropriately.

A.8 Enabling row compression for the repository database

Currently, Optim Performance Manager stores the data in an uncompressed format. You can enable row compression for the repository database to save disk space. The following technote describes how to do that:

http://www-01.ibm.com/support/docview.wss?uid=swg21442681

> **Tip:** Take a look at this technote even if you don't want to enable row compression. This technote lists which tables of the repository database store data for which monitoring profiles.

A.9 Setting up HADR for the repository database

If you want to set up high availability disaster recovery (HADR) for the repository database then you get step-by-step guidance from the following technote:

http://www-01.ibm.com/support/docview.wss?uid=swg21442697

A.10 Using the configuration program peconfig

The **peconfig** program was part of DB2 Performance Expert and is still supported by Optim Performance Manager. Using `peconfig` you can perform the following configuration tasks in an interactive mode or in a silent mode:

- Add remote DB2 instances for monitoring.
- Remove local and remote DB2 instances from monitoring.
- Enable, disable, or change monitoring for DB2 instances.
- Add databases for monitoring.
- Remove databases from monitoring.
- Enable or disable deadlock event monitoring.
- List added DB2 instances and databases.
- Restart monitoring for single monitored DB2 instances.

To use `peconfig` in an interactive mode, you can run it as command line program or graphical user interface program. From the `RepositoryServer/bin` directory, call **peconfig** to start the graphical user interface or call **peconfig -console** to start it as command line program. If you have installed Optim Performance Manager on Windows, then the program start menu has the following entries to start `peconfig` in graphical or console mode:

- Connection configuration
- Connection configuration (Command line)

`peconfig` still follows the approach of DB2 Performance Expert architecture where you added DB2 instances for monitoring and not single databases as in Optim Performance Manger. The advantage of this configuration method is that after you register a DB2 instance for monitoring, you can add all its cataloged databases for monitoring in one step. However, all databases of a DB2 instance that you add by using the `peconfig` program have the same default monitoring values such as retention times and collection intervals. You can change the monitoring settings later using the Configure Monitoring wizard from the Optim Performance Manager web console. The changes for one database apply to all other monitored database of the DB2 instance that are added using `peconfig`. Depending on your monitoring requirements, this can be considered as a disadvantage.

To get a list of commands that you can use in the `peconfig` command line prgram, refer to the Information Center at

http://publib.boulder.ibm.com/infocenter/idm/docv3/topic/com.ibm.datatools.perfmgmt.installconfig.doc/reference_configuration.html

You can also perform the configuration using `peconfig` in silent mode. This is a way to configure multiple DB2 instances and databases for monitoring in unattended batch mode. To perform a silent configuration, follow these steps:

- Prepare a response file.

  A sample response file, `peresp.txt`, can be found in the `bin` subdirectory of the installation directory. This file contains detail information about performing a configuration in silent mode, including instructions, syntax, a template, and examples. For the purpose of this book, we have the Optim performance Manager installed on an AIX machine and the path to the response file is `/opt/IBM/OPM/RepositoryServer/bin`.

- Execute the silent configuration.

  Run the following command from the `bin` subdirectory of the installation directory:

  ```
  peconfig -silent /opt/IBM/OPM/RepositoryServer/bin/peresp.txt
  ```

  where `peresp.txt` represents the name of your response file. Performing silent configuration will create the log file `pesilent.trc` in `<working_dir>/<InstName>`, where *`InstName`* represents the name of the DB2 instance on which the Optim Performance Manager Server runs.

Additionally to the interactive and silent mode peconfig provides external commands that you can use in scripts or batch files to automate configuration changes. Using external commands you can perform the following tasks:

- Enable or disable monitoring for DB2 instances.
- Change passwords to be used to access monitored DB2 instances.
- Add databases for monitoring.
- Remove databases from monitoring.
- Enable or disable deadlock event monitoring.
- List added DB2 instances and databases.
- Crypt passwords in silent configuration files.
- Restart monitoring for single monitored DB2 instances.

The external commands are described in the DB2 Performance Expert Information Center using the following link:

http://publib.boulder.ibm.com/infocenter/idm/v2r1/index.jsp

A.11 Changing Java heap size parameters of the repository server

The Java heap size parameters specify the minimum amount of memory that the repository server allocates at start up and the maximum amount of memory that the repository server uses at run time.

By default, the minimum value for the JVM heap size parameter is 128 MB; the maximum value is 1024 MB on UNIX and Linux and 768 MB on Windows.

Increasing or decreasing the values depend on these conditions:

- You might have to increase the minimum value if many applications and processes run on the same system, on which repository server is installed. This ensures that the repository server process can allocate a specific amount of memory to run without out-of-memory errors.
- You might have to decrease the minimum value if the workload is small, for example, if you use Performance Expert only for a test system.
- You might have to increase the maximum value if you have many monitored DB2 instances with many partitions or database objects, such as `LCONTAINERS`, or if many monitoring functions are active. This ensures that the repository server can allocate enough memory to run without out-of-memory errors.
- You might have to decrease the maximum value if you experience that a lower value is sufficient.

The db2pesrv.log file indicates if you have to change the Java heap size parameters. A sample error message for this situation is the following:

```
There is insufficient memory to process all statement metrics data from
this database. Optim Performance Manager shuts down monitoring for this
database.
User response: Increase maximum Java heap size parameter and restart the
Optim Performance Manager Repository Server.
```

In our book test environment, we resolve this error message by increasing the minimum value of the JVM heap size parameter to ensure that the heap size required for the repository server is allocated right after startup.

If you must increase or decrease minimum or the maximum value of the JVM heap size parameter, complete the following steps. The steps are for a Linux or UNIX environment, and can be applied to Windows accordingly:

1. Stop repository server with the **pestop** command.
2. Log on as root.
3. Change to the directory <*install dir server*>/RepositoryServer/bin, where <install dir server> is the installation directory of Optim Performance Manager.
4. Open the script pestart and search for this line:

   ```
   -Xms128m -Xmx1024m
   ```

   This means that the Java parameter for minimum heap size is set to 128 megabytes, and that the Java parameter for maximum heap size is set to 1024 megabytes.
5. Edit the line by changing the value of the parameter to -Xms128m or -Xmx1024m accordingly.
6. Log on as the DB2 instance owner.
7. Start repository server again with **pestart** command.

A.12 Determining collection interval bottlenecks

Optim Performance Manager collects monitoring data in intervals. A collection step includes retrieving the monitoring data from the monitored database and inserting it into the repository database. Depending on the amount of monitoring data retrieved from the monitored database and the memory and CPU resources of the Optim Performance Manager machine the interval might be too short to complete the collection step. The collection steps are executed in sequence, one does not start before the previous completed. If the interval is too short to complete the collection step then collection steps are queued which results in a permanently busy Optim Performance Manager machine and in unsteady collected monitoring data.

The `db2pesrv.log` file indicates too short collection intervals by posting appropriated messages.

A.12.1 Determining inflight and report data collection bottlenecks

If the collection interval of inflight and report data is too short to complete the collection step, you find messages similar to the following in the `db2pesrv.log` file or on the `pestart` console:

```
[20:08:10.42 ][5]Snapshot time[s][137] exceeds iteration time. Tip: Increase
snapshot interval.
```

`[5]` indicates that this message belongs to monitored database with instance ID 5 and `[137]` indicates that the collection step took 137 seconds.

If you see such messages for the same monitored database often, increase collection intervals using the following approach:

1. Determine to which database this message belongs. Call **`peconfig -list`** from the `<install dir>/RepositoryServer/bin` directory and look up the monitored database having the instance ID as posted in the message, in our example instance ID 5.
2. In the Optim Performance Manager web console, go to the Manager Database Connection panel, open the Configure Monitoring wizard for this database and move to step 2 in this wizard.
3. Either increase the base collection interval or increase the collection interval of single monitoring profiles.
   - To increase the base collection interval, open the *Retention times and sampling intervals* dialog from step 2 of the Configure Monitoring wizard and increase the sampling rate in minutes. Because changing the base sampling rate overrides all custom sampling rates, we recommend to choose this method if you have not customized sampling rates of the single monitoring profiles before.
   - To increase sampling rates of single monitoring profiles, edit the following monitoring profiles and specify a higher sampling rate:
     - Locking - Snapshot data per single lock and per connection
     - Active SQL and Connection - Snapshot data per connection
     - Dynamic SQL - Snapshot data per statement in the package cache

     The sampling rate must be a multiplier of the base sampling rate. You do not have to increase the sampling rate for all of these monitoring profiles at once. For example, increase the sampling rate of one monitoring profile and after a period of a few intervals, check whether you still receive this message. If so, then increase another one. Unfortunately, the message does not tell you for which monitoring profile you receive this message, therefore, you have to try out to determine the one that you have to increase the value. All of them are candidates for a high amount of monitoring data. We recommend that you start with the one where you typically expect the highest amount of monitoring data.

4. If the monitored database is a partitioned database, reducing the number of monitored partitions also helps to avoid collection bottlenecks. You can reduce the number of monitored partitions by defining partition sets. A partition set contains the partitions for that you want to collect monitoring data. Partition sets can be defined in step 5 of the Configure Monitoring wizard.

A.12.2 Determining Extended Insight data collection bottlenecks

If you collect Extended Insight data for a monitored database of DB2 9.7 Fix Pack 1 or higher and within the Extended Insight monitoring profile, you have the collection of transaction metrics on data server enabled, the unit of work event monitor is created on the monitored database. Optim Performance Manager uses a fixed interval to read and flush data from the unformatted event monitor table. See Appendix B, "Optim Performance Manager footprint" on page 471 to learn more about how Optim Performance Manager uses this event monitor. If it takes too long to read and flush the data from the event monitor table and transfer it to the Optim Performance Manager machine, you might see an error message similar to the following in the `db2pesrv.log` file.

```
[10:28:26.617][7] Error: The monitored data could not be processed within
[180000] milliseconds by Monitor [opmuow1].

Explanation: Possible reason is that the workload is too high for the monitored
DB Server and/or network to deliver the monitoring data to the OPM Server in
sufficient time for it to be processed.

User response: If the problem persists, increase the network and/or DB Server
capacity.
```

`[7]` indicates that this message belongs to monitored database with instance ID 7. To check which monitored database is associated with the instance ID posted in this message, call **`peconfig -list`** from the `<install dir>/RepositoryServer/bin` directory and look up the monitored database having this instance ID.

If you receive this message repeatedly, we recommend to analyze further in order to improve the performance of reading and flushing data from the event monitor table. Some hints to improve performance are:

- If you monitor a partitioned database, then the amount of unit of work events that Optim Performance Manager has to read and flush correspond to the number of partitions that you monitor. You can reduce the number of monitored partitions by defining partition sets. A partition set contains the partitions for that you want to collect monitoring data. Partition sets can be defined in step 5 of the Configure Monitoring wizard.
- To avoid that the operations on the event monitor table impacts the workload on your monitored database as well as to avoid that the workload on your monitored database impacts reading data from the event monitor table, we recommend that you assign a dedicated table space and buffer pool for the event monitors that Optim Performance Manager creates. You can specify the table space to use in the Configure Monitoring wizard.
- We recommend that you use DB2 9.7 Fix Pack 2 or higher for the monitored database. Fix pack 2 includes a fix that reduces the CPU consumption of the unit of work event monitor.
- Monitor CPU and memory consumption over time on the Optim Performance Manager machine to check whether Optim Performance Manager has enough resources to retrieve and store monitoring data fast enough.

B

Optim Performance Manager footprint

To collect the performance information from the monitored DB2 databases, Optim Performance Manager must take snapshots and run event monitors. Depending on the monitoring profile, Optim Performance Manager may introduce overhead on the monitored database and monitored application. In this appendix, we describe what Optim Performance Manager does on your monitored database and the monitored application. We highlight the monitors that might have overhead for your attention.

B.1 Optim Performance Manager footprint on the monitored database

Optim Performance Manager retrieves information from monitored database by two means: calling snapshot and reading from the DB2 event monitors. When you configure the monitoring profile for a monitored database, the DB2 monitor switches and DB2 event monitors are turned on or turned off.

Figure B-1 on page 473 shows the step 2 of the configuring monitoring profile. When you enable a monitoring profile in this step, the corresponding DB2 monitor switches or DB2 event monitors are turned on on the monitored database to allow Optim Performance Manager to get snapshots or read event monitors to provide the monitoring information you require. When you disable a monitoring profile, the corresponding DB2 monitor switches or DB2 event monitors are turned off on the monitored database.

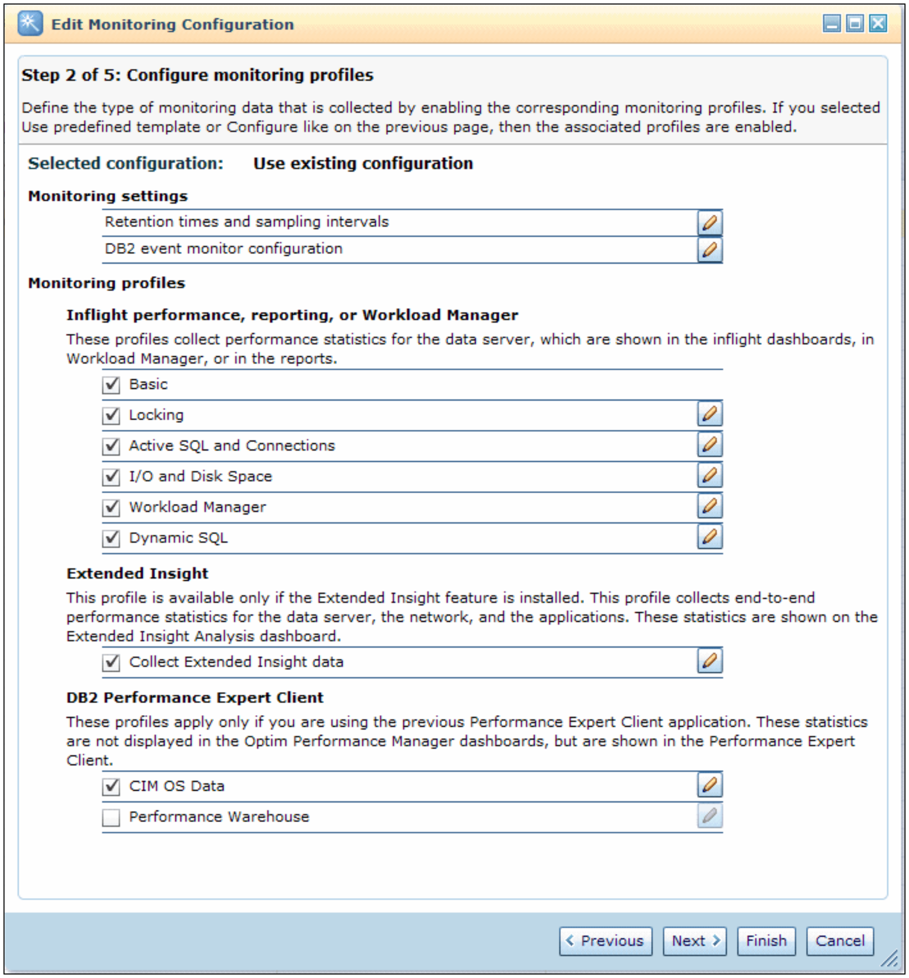


Figure B-1 Configuring monitoring profile

Turning on DB2 monitor switches or enabling DB2 event monitors on monitored database may cause some performance overhead. In step 3 of configuring monitoring profile (Figure B-2 on page 474), Optim Performance Manager uses a small yellow triangle to highlight the DB2 snapshots and event monitors that may cause unwanted performance overhead on the monitored database.

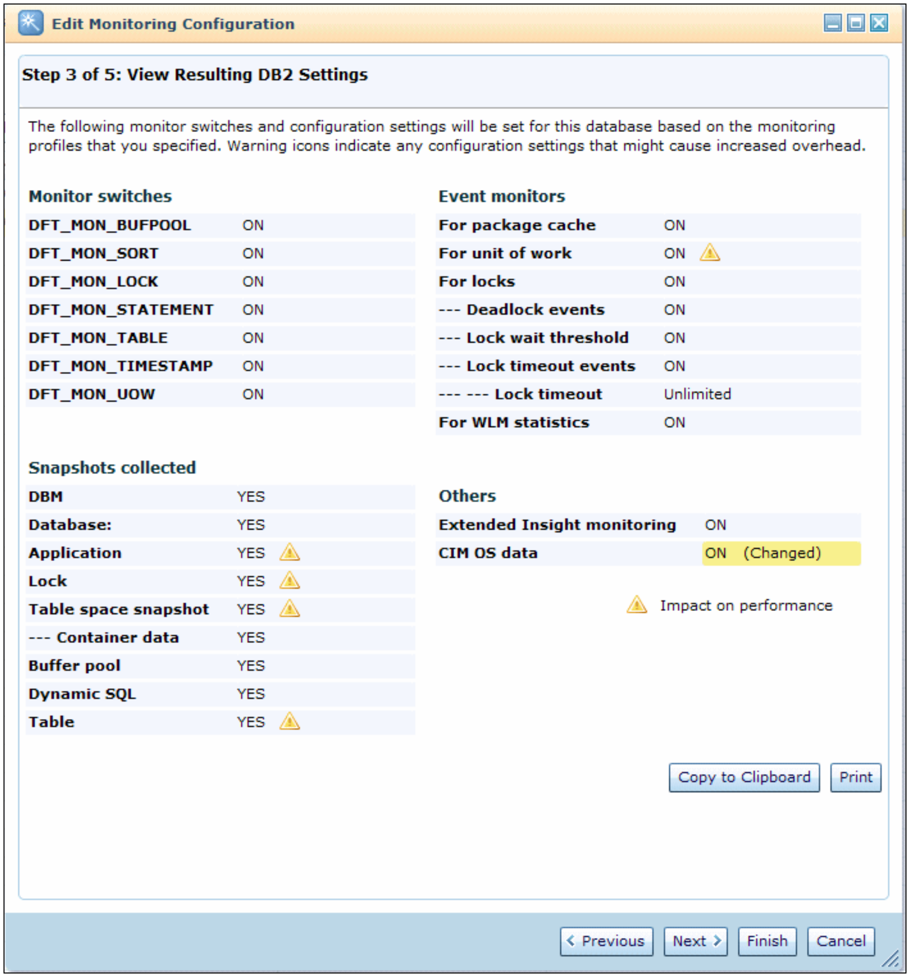


Figure B-2 DB2 settings

When you move the mouse over the yellow triangle, a message pops up to indicate the possible overhead. Figure B-3 on page 474 shows the pop-up message when we move the mouse over the yellow triangle of Table.

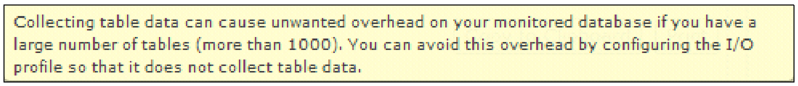


Figure B-3 Hint about monitoring overhead

Table B-1 on page 475 lists the monitoring profiles and their corresponding monitor switches, snapshots, and event monitors that will be turned on or created when a monitoring profile is enabled. When the monitoring profile is disabled, the corresponding DB2 event monitor will be dropped.

Table B-1 Monitoring profiles, monitoring switches, snapshots, and event monitors

Enabled monitoring profile	Enabled secondary monitoring profile	DB2 monitor switches to be turned on	DB2 snapshots collected	DB2 event monitor to be created
Basic	N/A	DFT_MON_SORT, DFT_MON_TIMESTAMP	DBM, Database	N/A
Lock	Lock waiting alert	N/A	N/A	Lock event monitor for lock wait
Lock	Lock timeout alert	N/A	N/A	Lock event monitor for lock timeout
Lock	Deadlock alert	N/A	N/A	Lock event monitor for deadlock, or deadlock event monitor for DB2 9.1 or DB2 9.5
Lock	Collect lock wait information	DFT_MON_LOCK, DFT_MON_STMT, DFT_MON_TIMESTAMP	Application, Lock	N/A
Active SQL and Connections	N/A	DFT_MON_SORT, DFT_MON_STMT, DFT_MON_TIMESTAMP, DFT_MON_UOW	Application	N/A
I/O and Disk Space	Collect table information	DFT_MON_BUFPOOL, DFT_MON_TABLE, DFT_MON_TIMESTAMP	DBM, Database, Buffer pool, Table	N/A
I/O and Disk Space	Collect table space information without container information	DFT_MON_BUFPOOL, DFT_MON_TIMESTAMP	DBM, Database, Table space snapshot without Container data, Buffer pool	N/A
I/O and Disk Space	Collect table space information with container information	DFT_MON_BUFPOOL, DFT_MON_TIMESTAMP	DBM, Database, Table space snapshot with Container data, Buffer pool	N/A
Workload Manager	N/A	N/A	N/A	Statistic event monitor
Dynamic SQL	N/A	DFT_MON_STMT, DFT_MON_TIMESTAMP	Dynamic SQL	N/A
Collect Extended Insight Data	Collect statement and transaction metrics on client	DFT_MON_STMT (DB2 9.1 or DB2 9.5)	N/A	Package cache event monitor (DB2 9.7 or higher)

Enabled monitoring profile	Enabled secondary monitoring profile	DB2 monitor switches to be turned on	DB2 snapshots collected	DB2 event monitor to be created
Collect Extended Insight Data	Collect statement and transaction metrics on client and Collect statement metrics on data server	N/A	N/A	Package cache event monitor
Collect Extended Insight Data	Collect transaction metrics on data server	N/A	N/A	Unit of work event monitor

Table B-2 shows the relationship between monitoring profiles and dashboards. When a monitoring profile is enabled, the related monitoring data will be presented on corresponding dashboards.

Table B-2 Monitoring profile verses dashboard

Dashboard	Monitoring profile						
	Basic	Locking	Active SQL and Connections	I/O and Disk Space	Workload Manager	Dynamic SQL	Extended Insight
Overview	Y	Y	Y				Y
Active SQL			Y				
BPIO	Y			Y			
Locking		Y					
Logging	Y						
Memory	Y		Y				
System	Y						
Utilities	Y						
Workload	Y						
Extended Insight							
Report			Y	Y	Y	Y	
Workload Manager configuration					Y		

B.1.1 DB2 monitor switches

Optim Performance Manager turns on or off the DB2 monitor switches by updating the database manager configuration parameters with the following commands:

```
db2 update dbm cfg using switch_name on
db2 update dbm cfg using switch_name off
```

However, if the monitored system is an SAP system, the monitor switches are not turned off when the monitoring profile on Optim Performance Manager is disabled. To learn more about monitoring SAP with Optim Performance Manager, see Chapter 12, “Monitoring SAP environments” on page 423.

Table B-3 shows the information retrieved by Optim Performance Manager (DBM) when each DB2 monitor switch is turned on respectively.

Table B-3 Information retrieved by Optim Performance Manager when DB2 Monitor Switch is turned on

Monitor switch	Database manager configuration parameter	Information Optim Performance Manager obtains
BUFFERPOOL	DFT_MON_BUFPOOL	Number of reads and writes, time taken
SORT	DFT_MON_SORT	Number of heaps used, sort performance
LOCK	DFT_MON_LOCK	Lock wait times, deadlocks
STATEMENT	DFT_MON_STMT	Start/stop time, statement identification
TABLE	DFT_MON_TABLE	Measure of activity (rows read/written)
TIMESTAMP	DFT_MON_TIMESTAMP	Timestamps
UOW	DFT_MON_UOW	Start/end times, completion status

For DB2 for Linux, UNIX, and Windows Version 9.1, 9.5, and 9.7, by default, all the switches listed in Table B-3 are turned off, except `DFT_MON_TIMESTAMP`, which is turned on.

To learn more about these switches, refer to the relative topics in DB2 Information Center:

- DB2 9.1:

  http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/c0005719.html

- DB2 9.5:

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.mon.doc/doc/c0005719.html

- DB2 9.7:

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.mon.doc/doc/c0005719.html

B.1.2 DB2 event monitors

Optim Performance Manager creates the event monitor tables under the schema `OPM` on each monitored database.

Package cache event monitor

The package cache event monitor is supported on DB2 9.7 Fix Pack 1 or above. Package cache event monitor captures data related to statement entries that have been flushed from the database package cache. It provides the history of the contents of the package cache that which can help with SQL query performance analysis and problem determination issues. To learn more about package cache event monitor, refer to DB2 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.mon.doc/doc/c0056443.html

If the monitoring profile results in package cache event monitor created, Optim Performance Manager creates one package cache event monitor on the monitored database. The package cache event monitor is named as `OPMP` followed by a hash code that uniquely identifies the Optim Performance Manager server which created the event monitor. For example, a package cache event monitor may be named as `OPMPAZ1AKZ`. When a record is flushed from the DB2 package cache, the package cache event monitor writes data into an unformatted event table which is created in schema `OPM`. The event table has the same name as the event monitor. Once every minute, the Optim Performance Manager Server reads the contents of the unformatted event table, processes the records, and then prunes the table by deleting the processed rows. This table is re-created only when the configuration profile is changed.

The package cache event monitor will be most active when the monitored database server has a low package cache hit ratio - meaning that SQL statements are regularly being flushed out of the package cache.

The DB2 package cache event monitor was designed to collect information about SQL statements with minimal impact to the database. However, there are guidelines you can follow to minimize this impact still further:

- Create a dedicated 32K table space (across all partitions) and a dedicated buffer pool, and specify that the package cache event table is to be created in the newly created table space when configuring EI monitoring profile, as shown in Figure B-4 on page 479. The table space should use 32K pages because the package cache event table contains a BLOB column and a 32K page table spaces will allow the BLOB data to be stored inline (in most cases). Inlined large object data allows the BLOB to flow through the buffer pool and thus improves efficiency. In addition, you can specify the table space as auto-resize or specify a maximum percent, which is 90% by default.

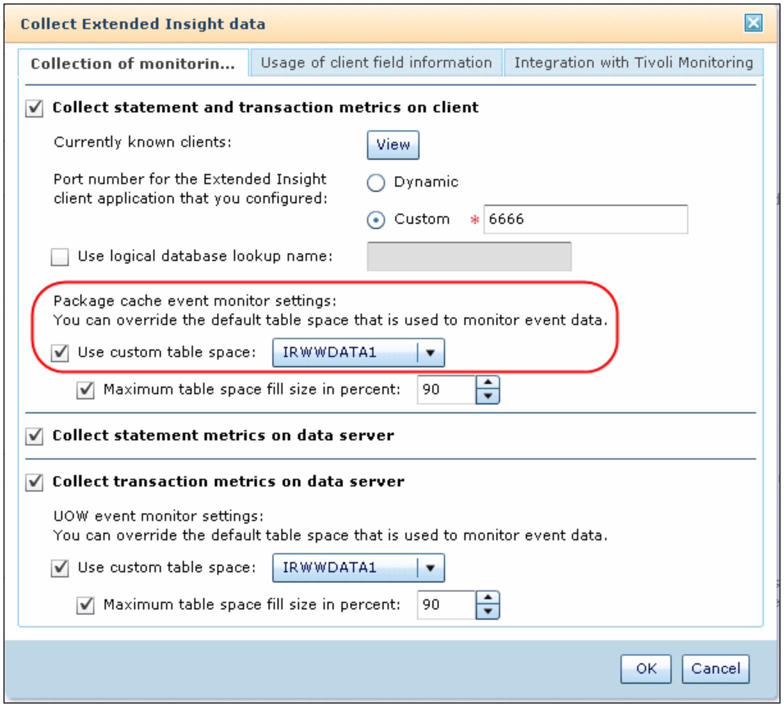


Figure B-4 Use custom table space for the package cache event table

- To account for peak periods, size the buffer pool to hold approximately two to three minutes worth of package cache event monitor data. The average size one-minute size of the package cache event monitor data can be calculated by estimating the transaction size in one minute, which is dependent on how much workload is running and monitored. This will allow Optim Performance Manager to read the monitor data from the buffer pool and thus reduce disk I/O.
- If at all possible, locate the package cache monitor table space on a set of dedicated disks.

Unit of work event monitor

The unit of work (UOW) event monitor is supported on DB2 9.7 Fix Pack 1 and above. However, to improve efficiency of this monitor, we strongly recommend installing DB2 9.7 Fix Pack 2. The unit of work event monitor records an event whenever a unit of work is completed, that is, whenever there is a commit or a rollback within the database. The recorded event is inserted, as a record, into the corresponding UOW unformatted event table. This historical information about individual units of work is useful for chargeback purposes (charging by CPU usage) and for monitoring compliance with response time service level objectives. To learn more about unit of work event monitor, refer to DB2 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.mon.doc/doc/c0055273.html

If the monitoring profile results in creating unit of work event monitor, Optim Performance Manager creates two unit of work event monitors on each monitored database. One unformatted event table is created for each unit of work event monitor to store information. Each unit of work event monitor is named with prefix followed by a hash code that uniquely identifies the Optim Performance Manager server that creates the event monitor. If the monitored database is a non-HADR database, the prefix is `OPMU`. If the monitored database is an HADR database, the prefix is `OPMO`. Since there are always two unit of work event monitors created, the event monitor name ends with “1” or “2” respectively. For example, on a non-HADR monitored database, the event monitors name could be `OPMU@AZ1AKZ1` and `OPMU@AZ1AKZ2`.

The two event monitors do not work concurrently, instead, one is always active, whilst the other is dormant. The active monitor is collecting information about units of work executed and inserting this information (one record per UOW) into the corresponding unformatted event table. After 30 seconds, the monitors switch and the dormant monitor becomes active, whilst the active monitor becomes dormant. At this point the unformatted event table of the dormant monitor is read by Optim Performance Monitor, the data processed, and the event monitor is then re-created. When 30 seconds has elapsed, the monitors switch once again and the process is repeated. The event monitor is dropped and then re-created because this is more efficient than using the SQL DELETE statements to prune the data - primarily because DELETEs will be logged by the database server.

On non-HADR database and HADR database, the unformatted event tables are pruned using different approaches:

- Non-HADR database: The event monitors and the event tables are re-created.
- HADR database: The event monitor is pruned by deleting records in the event tables but not being re-created. In the case of HADR with read-on-standby, all the methods for effective pruning (dropping table and let event monitor recreate it) will cause the active standby to be placed on the replay-only

window, which will terminate all application connections. Therefore, if Optim Performance Manager detects that the monitored database is set up for HADR, it prunes the unformatted event table by reading and deleting the data from the event tables instead of recreating them.

On a partitioned system, the amount of data retrieved from the UOW event monitor will depend upon on the number of partitions touched by each transaction. If a transaction touches many partitions, then information related to that transaction will be placed into the UOW event table on each partition. Consequently there could be significantly more data for Optim Performance Manager to process.

The DB2 unit of work event monitor was designed to collect transaction information with minimal impact to the database. However, there are guidelines you can follow to minimize this impact still further:

- Create a dedicated 32K table space (across all partitions) and a dedicated buffer pool, and specify that the UOW event tables are to be created in the newly created table space when configuring Extended Insight monitoring profile, as shown in Figure B-5 on page 482. The table space should use 32K pages because the UOW event table contains a BLOB column and a 32K page table spaces will allow the BLOB data to be stored inline (in most cases). Inlined large object data allows the BLOB to flow through the buffer pool and thus improves efficiency. In addition, you can specify the table space as auto-resize or specify a maximum percent, which is 90% by default.

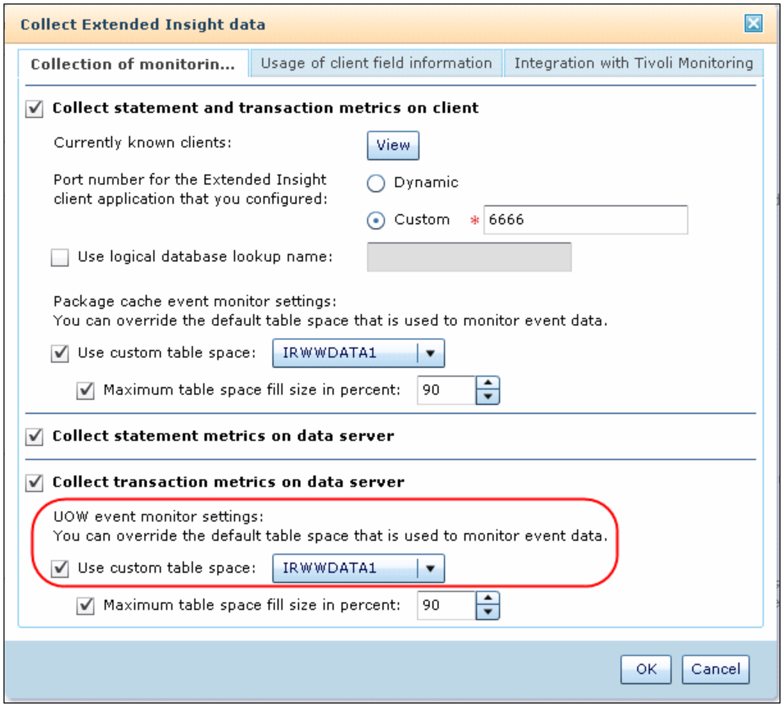


Figure B-5 Use custom table space for UOW event tables

- Size the buffer pool to hold approximately two to three minutes worth of UOW event monitor data - to account for peak periods. This will allow Optim Performance Manager to read the monitor data from the buffer pool and thus reduce disk I/O.
- If at all possible, locate the UOW table space on a set of dedicated disks.
- A record is inserted into the UOW event table for each completed UOW and each of these inserts are logged by the database manager. Therefore, enabling the UOW event monitor will cause additional logging on the monitored database server. The amount of additional logging incurred is dependant upon the mix of read and write transactions within the workload being monitored. We recommended that you monitor logging activity on the monitored database server to ensure the additional logging does not become a bottleneck, and if so, tune logging related parameters accordingly.

Deadlock event monitor and lock event monitor

The deadlock event monitor, also called legacy deadlock event monitor, is supported on DB2 9.1 and 9.5. Since DB2 9.7, the lock event monitor, also called new lock monitor, is supported. When you monitor DB2 9.7 or above on Optim

Performance Manager, you can choose to use legacy deadlock or to use new lock event monitor when configuring monitoring profile, as shown in Figure B-6 on page 483.

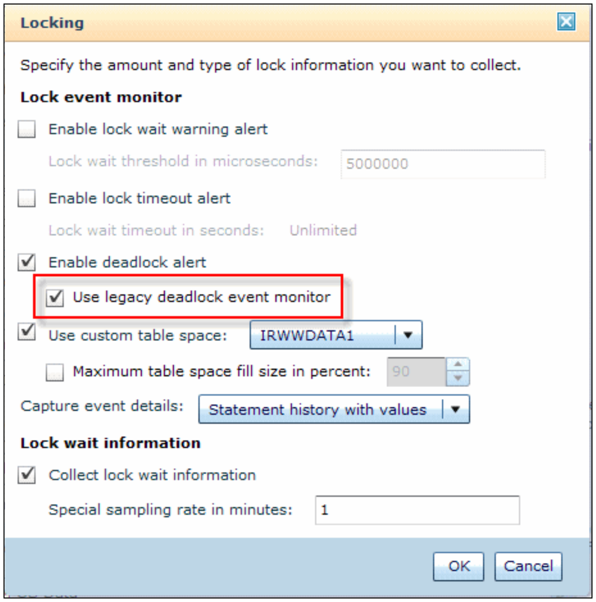


Figure B-6 Choose to use legacy deadlock event monitor

The deadlock event monitor collects information about the applications involved in the deadlock and the locks in contention and places the data in the deadlock event tables. When creating a deadlock event monitor, you can specify the following event monitor types:

- DEADLOCK WITH DETAILS event monitor:

  This event monitor collects the comprehensive information regarding the involved applications, including the identification of participating statements (and statement text) and a list of locks being held. Generally, the event monitor would cause some overhead on the database, dependent on how many statements are involved in deadlocks.

- DEADLOCKS WITH DETAILS HISTORY event monitor:

  This event monitor collects all information reported in a `DEADLOCKS WITH DETAILS` event monitor, along with the statement history for the current unit of work of each application owing a lock participating in a deadlock scenario for the database partition where that lock is held. Usually the event monitor would cause some minor overhead on the database.

- DEADLOCK WITH DETAILS HISTORY VALUES event monitor:

This event monitor collects all information reported in a deadlock with details and history, along with the values provided for any parameter markers at the time of execution of a statement. This event monitor collects the most deadlock information among the three deadlock even monitors. Therefore, in general, the DEADLOCK WITH DETAILS HISTORY VALUES event monitor causes higher overhead on the database than the other two types of event monitor.

To learn more about dead lock event monitor, refer to DB2 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.perf.doc/doc/t0055033.html

Lock event monitor collects lock timeout, lock wait, and deadlock information to help identify and resolve locking problems. To learn more about lock event monitor, refer to DB2 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.mon.doc/doc/t0055093.html

If the monitoring profile results in deadlock event monitor or lock event monitor created, Optim Performance Manager creates deadlock or lock event monitors on the monitored database.

- **Deadlock event monitor**

  If the monitored database is DB2 9.1 or 9.5, Optim Performance Manager creates one deadlock event monitor and some event tables in schema `OPM` per each monitored database. The number of event tables created depends on the detail level you configure.

  – For deadlock with details event monitoring, event tables are created for `DEADLOCK`, `DLCONN`, `DLLOCK`, and `CONNHEADER`,

  – For deadlocks with details history event monitoring, event tables are created for `STMTHIST`.

  – For deadlock with details history values event monitoring, event tables are created for `STMTVALS`.

  The deadlock event monitor is named with prefix `OPMD` followed by a hash code that uniquely identifies the Optim Performance Manager server that creates the event monitor. The table name consists of the data group name (`DEADLOCK`, `DLCONN`, `DLLOCK`, `CONNHEADER`, `MTMTHIST`, or `STMTVALS`) followed by the event monitor name. Here is an example of creating a deadlock event monitor when deadlock event monitor is enabled with details, history, and values:

```
CREATE EVENT MONITOR OPMDAZ1AKZ FOR DEADLOCKS WITH DETAILS HISTORY VALUES
WRITE TO TABLE
   DEADLOCK (TABLE OPM.DEADLOCK_OPMDAZ1AKZ),
```

```
DLCONN (TABLE OPM.DLCONN_OPMD7AZ1AKZ),
DLLOCK (TABLE OPM.DLLOCK_OPMD7AZ1AKZ),
CONNHEADER (TABLE OPM.CONNHEADER_OPMDAZ1AKZ),
STMTHIST (TABLE OPM.STMTHIST_OPMDAZ1AKZ),
STMTVALS( TABLE OPM.STMTVALS_OPMDAZ1AKZ) BUFFERSIZE 64
```

Optim Performance Manager reads and deletes data from these event tables in every 30 seconds.

- **Lock event monitor**

  If the monitored database is DB2 9.7 or above, Optim Performance Manager creates two lock event monitors per monitored database and one unformatted event table for each lock event monitor. Each lock event monitors is named with prefix followed by a hash code that uniquely identifies the Optim Performance Manager server that creates the event monitor. If the monitored database is a non-HADR database, the prefix is `OPMN`. If the monitored database is an HADR database, the prefix is `OPML`. Since there are always two lock event monitors created, the event monitor name ends with “1” or “2” respectively. For example, on a non-HADR monitored database, the event monitors name could be `OPMN@AZ1AKZ1` and `OPML@AZ1AKZ2`.

  Similar to the unit of work event monitors, the two lock event monitors also work switchingly and iteratively. Optim Performance Manager prunes the lock event tables iteratively in an interval of 30 seconds. Suppose that at the beginning of one iteration, the first event monitor (event monitor A) is active, collecting data, and the second event monitor (event monitor B) is waiting. Then event monitor A is deactivated and Optim Performance Manager reads data from its unformatted event table (`UET_A`); at the same time, event monitor B is activated to collect data. After Optim Performance Manager obtains the data from `UET_A`, `UET_A` is pruned. Then this iteration ends up. In the next iteration, event monitor A and event monitor B switches.

  On non-HADR database and HADR database, the unformatted event tables are pruned in different way: on a non-HADR database, the event monitors and the event tables are re-created, while for a HADR database, the event monitor is pruned by deleting records in the event tables but not being re-created.

  Lock event monitor can collect information of lock wait, lock timeout, and deadlock, depending on your monitoring profile and wether the following DB2 configuration parameters are enabled:

  – MON_LOCKWAIT: when this parameter is on, lock wait events at the database level is collected for the lock event monitor.

  – MON_LOCKTIMEOUT: when this parameter is on, lock timeout events at the database level is collected for the lock event monitor.

– MON_DEADLOCK: when this parameter is on, deadlock events at the database level is collected for the lock event monitor.

There are guidelines you can follow to minimize this impact further:

- Create a dedicated 32K table space (across all partitions) and a dedicated buffer pool, and specify that the legacy deadlock event table or lock event table is to be created in the newly created table space when configuring locking monitoring profile, as Figure B-7 on page 486 shows. The table space should use 32K pages because the event tables contain BLOB and CLOB columns and a 32K page table spaces will allow the BLOB and CLOB data to be stored inline (in most cases). Inlined large object data allows the BLOB and CLOB to flow through the buffer pool and thus improves efficiency. In addition, you can specify the table space as auto-resize or specify a maximum percent, which is 90% by default.

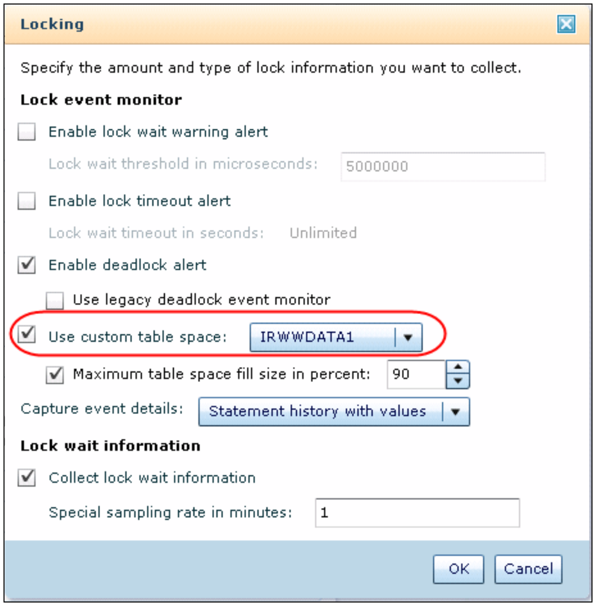


Figure B-7 Use custom table space for lock event monitor or legacy deadlock event monitor

- Size the buffer pool to hold approximately two to three minutes worth of lock event monitor data or legacy deadlock event monitor data - to account for peak periods. This will allow Optim Performance Manager to read the monitor data from the buffer pool and thus reduce disk I/O.
- If at all possible, locate the table space on a set of dedicated disks.

Statistic event monitor

The statistic event monitor is supported on DB2 95 and above. This monitor serves as a low-overhead alternative to capturing detailed activity information by collecting aggregate data (for example, the number of activities completed and average execution time). The aggregate data includes histograms for a number of activity measurements including lifetime, queue time, execution time, and estimated cost. You can use histograms to understand the distribution of values, identify outliers, and compute additional statistics such as averages and standard deviations. The histograms can help you understand the variation in lifetime that users experience. The average life time alone does not reflect what a user experiences if there is a high degree of variability. To learn more about WLM event monitor, refer to the DB2 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.wlm.doc/doc/c0052603.html

If the monitoring profile results in statistic event monitor created, Optim Performance Manager creates one statistic event monitor on each monitored database. The event monitor is named as `OPMW` followed by a hash code that uniquely identifies the Optim Performance Manager server that creates the event monitor. Six event tables are created on the monitored database. The name of the tables is started with `CONTROL_`, `HISTOGRAMBIN_`, `QSTATS_`, `SCSTATS_`, `WCSTATS_`, `WLSTATS_`, respectively, and is followed by the event monitor name. For example, a statistic event monitor could be named as“`OPMWAZ1AKZ`, and the tables could be `CONTROL_OPMWAZ1AKZ`, `HISTOGRAMBIN_OPMWAZ1AKZ`, `QSTATS_OPMWAZ1AKZ`, `SCSTATS_OPMWAZ1AKZ`, `WCSTATS_OPMWAZ1AKZ`, `WLSTATS_OPMWAZ1AKZ`. Optim Performance Manager reads and deletes data from the event tables iteratively and the interval is defined when configuring monitoring profile, as shown in Figure B-8.

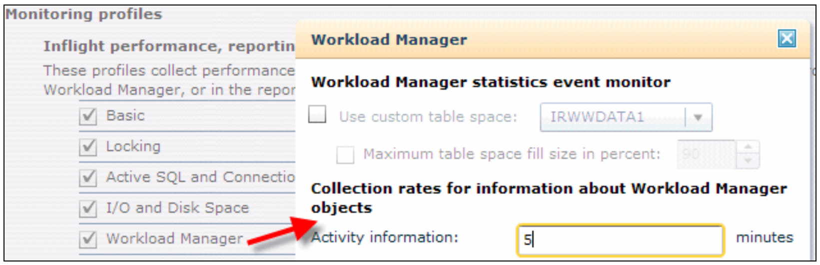


Figure B-8 Workload Manager data collection interval

Keep in mind that DB2 only supports one active statistic event monitor. This means that if you have already started a statistic event monitor on the monitored database, Optim Performance Manager cannot start another one to obtain other statistic information for you. Attempting to start more than one will result an error message in the `db2pesrv.log` file indicating that there is already one active statistic event monitor and Optim Performance Manager failed to start another one. Therefore, make sure that no statistic event monitor is running before you start Optim Performance Manager with the Workload Manager profile enabled.

B.1.3 Monitoring overhead considerations

When you configure monitoring profile, if you enable a monitoring profile, on the monitored database, the corresponding monitor switches will be turned on, the corresponding event monitors will be created, and the corresponding snapshots will be called. On the contrary, if you disable a monitoring profile, on the monitored database, the corresponding monitor switches will be turned off, the corresponding event monitors will be dropped, and the corresponding snapshots will not be called.

For enabled Inflight monitoring profiles, Optim Performance Manager takes snapshots or reads from DB2 event monitors periodically and you can specify the sampling rate. Generally, the bigger the sampling rate is, the less monitoring overhead is made to the monitored database.

For enabled Extended Insight monitoring profiles, Optim Performance Manager Extended Insight reads from DB2 event monitors in fixed frequency and you can not change it. The package cache event table is read by Optim Performance Manager once every minute. Each of the two UOW unformatted event tables is read by Optim Performance Manager once every minute. When the Optim Performance Manager repository server is stopped successfully by running `pestop`, the event monitors created by Optim Performance Manager would be dropped.

Enabling Dynamic SQL profile (or any other monitoring profile which takes Dynamic SQL snapshots) may cause unwanted overhead on the monitored database when the package cache is large - because each snapshot retrieves all the data from the package cache at the time of the snapshot. If you wish to collect Dynamic SQL data and the monitored database package cache is large, it will be necessary to increase the sampling interval for Dynamic SQL collection.

Consider the following potential monitoring overhead on the monitored database server:

- *Collecting application snapshots*: collecting SQL data can cause unwanted overhead on your monitored database if you have a large number of concurrent applications (more than 1000).
- *Collecting lock snapshots*: collecting lock snapshots can cause unwanted overhead on your database if you have a large number of locks on the database.

- *Collecting table space snapshots*: collecting table space data can cause unwanted overhead on your monitored database if you have a large number of table spaces or table space containers (more than 1000). You can avoid this overhead by configuring the I/O profile so that it does not collect table space data or does not collect table space container data.
- *Collecting table snapshots*: collecting table data can cause unwanted overhead on your monitored database if you have a large number of tables (more than 1000). You can avoid this overhead by configuring the I/O profile so that it doesn't collect data.
- *Enabling event monitors for unit of work*: collecting extended server insight data can cause unwanted overhead on your monitored database if you have a very large number of transactions running on your system (more than 20000 per minute).

B.2 Optim Performance Manager footprint on the monitored application

Optim Performance Manager collects client-side information regarding the statements and transactions executed by an application if the “Collect statement and transaction metrics on client” profile is enabled when configuring Extended Insight monitoring.

Optim Performance Manager has hooks into the JDBC drivers, as well as the DB2 CLI driver which allow it to collect this client side information. These hooks record information about the statements and transactions executing on that client into an in-memory table. Every second, the information within this table is hashed and aggregated in order to reduce its memory footprint. For example, every second a hash code is calculated based on the SQL statement text, and the hash code is then used to aggregate this new data with data already in the table. If the newly calculated hash code already exists in the table, the record is updated, if the hash code does not exist, a new record is inserted. Note that in order to reduce memory consumption, the SQL statement text is not kept, only the hash code. For transactions, a unique transaction ID is used for aggregation, rather than calculation of a hash code based on SQL statement.

Once every minute, this in-memory data is sent to the Optim Performance Manager Server where it is processed and inserted into the repository server database. The in-memory hash table is then cleared as soon as the data is sent to the Optim Performance Manager server (it does not wait for confirmation of receipt from the server) and processing starts again.

When the data arrives at the Optim Performance Manager server, Optim Performance Manager attempts to match the hash code for the SQL statements with a hash code the server computed from either a dynamic SQL snapshot, or a package cache event monitor. When a match is found, the record can be inserted into the Optim Performance Manager repository. This matching of client and server side information allows Optim Performance Manager to correlate client and server side metrics for individual statements – which in turn provides the full end-to-end view of execution metrics for each statement executed. A similar process occurs for transactions, except these are matched on transaction ID.

B.2.1 Footprint

Client side CPU and memory are consumed as Optim Performance Manager Extended Insight client component records information about the transactions and SQL statements executed by the application, performs the aggregation and sends this information to the Optim Performance Manager server. In most cases, the additional CPU and memory required to perform this processing will not be noticeable. However, the following points should be noted:

- Applications which issue a large number of unique SQL statements during a 1-minute interval are likely to generate more work for Optim Performance Manager Extended Insight client component. Since many of the statements are unique, it will not be possible to group the statements together in the in-memory table, and consequently the table will grow, as will the memory required to monitor the application. This may be the case if the application uses literals instead of parameter markers. Using literals will cause many of the SQL statements to be unique and best practice dictates that applications should use parameter markers wherever possible.
- The higher the rate at which transactions and SQL statements are executed, the more resources Optim Performance Manager Extended Insight client component will consume. Therefore, OLTP type applications are likely to create more work for Optim Performance Manager to deal with than traditional BI type workloads - simply because an OLTP application will execute far more statements and transaction per minute than a BI application.

Much of Optim Performance Manager Extended Insight client processing is done asynchronously. This greatly reduces the potential impact client side monitoring will have on elapsed time of the transactions and SQL statements - and therefore, reduce the impact client side monitoring has on application throughput.

Index

C

D

Z

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

Other publications

These publications are also relevant as further information sources:

- Database Monitoring Guide and Reference, *SC27-2458*
- Troubleshooting and Tuning Database Performance, *SC27-2461*

Online resources

These Web sites are also relevant as further information sources:

- Integrated Data Management Information Center

  `http://publib.boulder.ibm.com/infocenter/idm/v2r2/topic/com.ibm.dstudio.nav.doc/topics/welcome.html`
- DB2 Information Center

  `http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/`
- Database and Information Management home page

http://www.ibm.com/software/data/

- DB2 developerWorks

  http://www.ibm.com/developerworks/db2/

- IBM Tivoli Composite Application Manager for Transactions 7.2.0.2

  http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcamt.doc_7.2.0.2/ic-homepage.html

- IBM Tivoli Composite Application Manager for Application Diagnostics 7.1.0.2

  http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcamfad.doc_7101/ic-homepage.html

- IBM Tivoli Monitoring (latest ver)

  http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.2fp2/welcome.htm

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

IBM

Redbooks

IBM Optim Performance Manager for DB2 for Linux,

(1.0" spine)
0.875"<->1.498"
460 <-> 788 pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

Draft Document for Review January 13, 2011 8:13 pm

IBM Optim Performance Manager for DB2 for Linux, UNIX, and Windows

Installation, configuration, and integration with ITCAM and, Optim Query Tuner

Detailed functions and features usage guide

Performance tuning scenarios

Optim Performance Manager Extended Edition, a follow-on to DB2 Performance Expert, is one of the key product of the IBM Optim Solution. Optim Performance Manager Extended Edition provides a comprehensive, proactive performance management approach. It helps organizations resolve emergent database problems before they impact the business.

This book describes the architecture and the components of Optim Performance Manager Extended Edition. We provide information for planning the deployment of Optim Performance Manager and detail steps for successful installation, activation, and configuration of Optim Performance Manager and the Extended Insight client.

Optim Performance Manager delivers new paradigm in terms of how it is used to monitor and manage database and database application performance issues. We describe individual product dashboards and reports and discuss, with various scenarios, how they can be used to identify, diagnose, prevent, and solve database performance problems.

.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7925-00 ISBN