



IBM® Optim™ Performance Manager Extended Edition V4.1.0.1

Best Practices

*Deploying Optim Performance Manager in
large scale environments*

Ute Baumbach (bmb@de.ibm.com)
*Optim Performance Manager Development
IBM Research and Development Germany*

Simon Harris (siharris@au1.ibm.com)
*Data Studio Quality Assurance
IBM Australia*

Last updated: April 2011

| | | |
|------------|---|-----------|
| 1 | <u>EXECUTIVE SUMMARY</u> | 3 |
| 2 | <u>PLANNING THE OPTIM PERFORMANCE MANAGER DEPLOYMENT</u> | 5 |
| 2.1 | SAMPLE 1: MONITORING MULTIPLE NON-PARTITIONED DATABASES | 6 |
| 2.1.1 | SAMPLE 1A) MONITORING 15 DATABASES | 6 |
| 2.1.2 | SAMPLE 1B) MONITORING 30 DATABASES | 6 |
| 2.2 | SAMPLE 2: MONITORING A PARTITIONED DATABASE | 6 |
| 2.2.1 | SAMPLE 2A) MONITORING A PARTITIONED DATABASE WITH 24 PARTITIONS | 6 |
| 2.2.2 | SAMPLE 2B) MONITORING A PARTITIONED DATABASE WITH 48 PARTITIONS | 6 |
| 2.3 | SAMPLE 3: MONITORING DATABASES WITH HIGH TRANSACTION RATE | 7 |
| 2.3.1 | SAMPLE 3A) MONITORING 3 DATABASES | 7 |
| 2.3.2 | SAMPLE 3B) MONITORING A PARTITIONED DATABASE WITH 48 PARTITIONS WITH EXTENDED INSIGHT ENABLED | 7 |
| 3 | <u>BEST PRACTICES FOR CONFIGURING OPTIM PERFORMANCE MANAGER TO MONITOR LARGE & ACTIVE SYSTEMS</u> | 8 |
| 3.1 | BEST PRACTICES FOR CONFIGURING MONITORING PROFILES – CONTROLLING THE TYPE OF INFORMATION COLLECTED | 9 |
| 3.1.1 | BASIC PROFILE | 10 |
| 3.1.2 | LOCKING PROFILE | 10 |
| 3.1.3 | ACTIVE SQL AND CONNECTIONS PROFILE | 12 |
| 3.1.4 | I/O AND DISK SPACE PROFILE | 12 |
| 3.1.5 | WORKLOAD MANAGER PROFILE | 13 |
| 3.1.6 | DYNAMIC SQL PROFILE | 14 |
| 3.2 | BEST PRACTICES FOR CONFIGURING MONITORING GRANULARITY – CONTROLLING THE LEVEL OF DETAIL COLLECTED | 15 |
| 3.3 | BEST PRACTICES FOR CONFIGURING MONITORING SAMPLING RATE – CONTROLLING THE FREQUENCY OF DATA COLLECTION | 16 |
| 3.4 | BEST PRACTICES FOR CONFIGURING EXTENDED INSIGHT MONITORING | 17 |
| 3.4.1 | COLLECT STATEMENT AND TRANSACTION METRICS ON CLIENT | 18 |
| 3.4.2 | COLLECT STATEMENT METRICS ON DATA SERVER | 21 |
| 3.4.3 | COLLECT TRANSACTION METRICS ON DATA SERVER | 21 |
| 3.5 | CHARACTERISTICS OF THE DATABASE BEING MONITORED | 23 |
| 3.6 | CHARACTERISTICS OF THE APPLICATION OR WORKLOAD BEING MONITORED (EXTENDED INSIGHT CLIENT) | 23 |
| 3.7 | RESOURCE CONSUMPTION ON THE SYSTEMS BEING MONITORED | 23 |
| 4 | <u>BEST PRACTICES FOR CONFIGURING OPTIM PERFORMANCE MANAGER TO MONITOR A LARGE NUMBER OF DATABASES</u> | 24 |
| 4.1.1 | RETENTION TIMES AND SAMPLING INTERVALS | 25 |
| 5 | <u>BEST PRACTICES FOR MONITORING PARTITIONED DATABASE ENVIRONMENTS</u> | 25 |

| | | |
|------------|--|-----------|
| 6 | <u>BEST PRACTICES FOR MONITORING DB2 PURESCALE™ SYSTEMS</u> | 29 |
| 7 | <u>RUNNING OPTIM PERFORMANCE MANAGER</u> | 30 |
| 7.1 | VERIFYING OPTIM PERFORMANCE MANAGER SET UP | 30 |
| 8 | <u>SUMMARY: MONITORING IS A BALANCING ACT!</u> | 31 |
| 9 | <u>NOTICES</u> | 32 |
| 9.1 | TRADEMARKS | 33 |

1 Executive Summary

This document describes best practices for deploying Optim™ Performance Manager Extended Edition V4.1.0.1 (OPM) in large scale environments.

For the purposes of this paper, the following environments are considered as large scale:

- Optim Performance Manager monitoring a large number of databases (greater than 30)
- Optim Performance Manager monitoring one or more partitioned databases with more than 24 partitions
- Optim Performance Manager with Extended Insight monitoring databases with a transaction rate greater than 10,000 transactions per minute
- Any combination of any of the above.

Each monitored database, or database partition in the case of partitioned database environments, requires operating system resources (memory, CPU, disk space and network) on the Optim Performance Manager server to collect, process and store the monitoring data. It is important to ensure that your Optim Performance Manager server has sufficient capacity to monitor your intended environment.

Before installing and configuring Optim Performance Manager, gather some basic information about the databases and applications that you want to monitor by considering questions such as:

- How many databases and database partitions do I want to monitor?
- How often do I want to collect the data? Which data?
- How long do I want to keep the collected data?

Based on the answers to the above questions, what type of computer do I need for the Optim Performance Manager server?

After setting up Optim Performance Manager, you might need to consider questions such as:

- How can I change collection settings to maintain good Optim Performance Manager performance, for example, if more databases are added or if the amount of monitoring data increases?
- What else can I do to maintain efficient Optim Performance Manager performance?

This document provides guidelines to help answer these questions.

In addition to reading this document, refer to the Optim Performance Manager Redbook for valuable information about architecture, planning, installation, configuration, and maintenance. This document complements the Redbook by providing information that is useful when deploying Optim Performance Manager in a large scale environment.

Download the Optim Performance Manager Redbook here:

<http://www.redbooks.ibm.com/redpieces/abstracts/sg247925.html?Open>

2 Planning the Optim Performance Manager deployment

Before you start planning, read the Redbook *Chapter 1: Optim Performance Manager overview* for an overview of product features, components, packaging, and architecture.

The most important topic when planning the deployment of Optim Performance Manager in a large scale environment is to understand the system resources that Optim Performance Manager requires. The remainder of this chapter shows sample estimations for the system resources that Optim Performance Manager needs for monitoring different large scale environments. We are refining the calculations for estimating disk space, memory, and CPU requirements continuously. Take the examples below as initial guidelines for planning purposes, but if you need sizing estimations for your specific environment then contact IBM. We will use a questionnaire to help provide a sizing estimation. For additional details on the calculations that are used to estimate the system resources, refer to the Redbook *Chapter 2.4: Capacity Planning*.

In addition to system resource estimations, you need to understand how to conform to Optim Performance Manager system prerequisites, comply with security requirements, and choose the best storage option for collected data. These topics are all handled in the Redbook *Chapter 2: Planning*.

The system resources that Optim Performance Manager needs depend mostly on the following parameters:

- The type of monitoring information that is being collected (for example what kind of snapshot, event monitor data, or Extended Insight data)
- The granularity of the monitoring information that is collected
- The interval at which the monitoring information is being collected
- The retention time that the collected monitoring information is stored
- The number of users that use the Optim Performance Manager web console concurrently
- The characteristics of the database that is being monitored (for example, the number of database objects)
- The characteristics of the application or workload that is being monitored (for example, the rate of executing transactions and statements)

For the following capacity estimation examples, we use the following common parameters. :

- Concurrent users to Optim Performance Manager web console: **10**
- Collect dynamic SQL snapshot information: **Yes**
- Collect snapshot information about connections: **Yes**
 - Concurrent connections: **200**
- Collect snapshot information about database objects: **Yes**
 - Number of table spaces: **100**
 - Number of buffer pools: **10**
 - Number of table space containers: **500**

- Number of tables : 1000

2.1 Sample 1: Monitoring multiple non-partitioned databases

For simplicity, we assume that the parameters for all databases are as listed above. Extended Insight is not used.

Additionally we assume the following:

- Sampling interval: **2 min**
- Retention time: **100 hours**

2.1.1 Sample 1a) Monitoring 15 databases

Calculated resource estimations:

- Disk space: **120 GB**
- Memory: **8 GB**
- CPU: **4 CPUs PV_5_2 64-bit 2*1.9GHz (or equivalent)**

2.1.2 Sample 1b) Monitoring 30 databases

Calculated resource estimations:

- Disk space: **240 GB**
- Memory: **10 GB**
- CPU: **6 CPUs PV_5_2 64-bit 2*1.9GHz (or equivalent)**

2.2 Sample 2: Monitoring a partitioned database

For simplicity we assume that all partitions have the same number of monitored objects as listed above. Extended Insight is not used.

Additionally we assume the following:

- Sampling interval: **30 min**
- Retention time: **200 hours**

2.2.1 Sample 2a) Monitoring a partitioned database with 24 partitions

Calculated resource estimations:

- Disk space: **25 GB**
- Memory: **7 GB**
- CPU: **3-4 CPUs PV_5_2 64-bit 2*1.9GHz (or equivalent)**

2.2.2 Sample 2b) Monitoring a partitioned database with 48 partitions

Calculated resource estimations:

- Disk space: **30 GB**
- Memory: **9 GB**
- CPU: **6-7 CPUs PV_5_2 64-bit 2*1.9GHz (or equivalent)**

2.3 Sample 3: Monitoring databases with high transaction rate

We assume the same parameters as for Sample 1.

Additionally Extended Insight is fully enabled for each monitored database. The monitored databases are at level DB2® V9.7 Fix Pack 1, otherwise the server part of Extended Insight that collects transaction and statement execution details on the data server cannot be enabled. We assume the same Extended Insight parameters for each database, as follows:

- Extended Insight client used: **Yes**
- Extended Insight server enabled: **Yes**
- Transaction rate per minute: **40000**
- Unique SQL statement rate per minute: **1600**
- Extended Insight retention times:
 - Level 1: **1 day**
 - Level 2: **1 month**
 - Level 3: **3 months**
 - Level 4: **2 years**

2.3.1 Sample 3a) Monitoring 3 databases

Calculated resource estimations:

- Disk space: **140 GB**
- Memory: **9 GB**
- CPU: **7 CPUs, PV_5_2 64-bit 2*1.9GHz (or equivalent)**

2.3.2 Sample 3b) Monitoring a partitioned database with 48 partitions with Extended Insight enabled

Because this is a partitioned instance we assume a different number of transactions and statements:

- Transaction rate per minute: **1000**
- Unique SQL statement rate per minute: **4000**

Calculated resource estimations:

- Disk space: **36 GB**
- Memory: **11 GB**
- CPU: **8 CPUs PV_5_2 64-bit 2*1.9GHz (or equivalent)**

However, Optim Performance Manager best practices recommend the use of partition sets to monitor large partitioned instances. If a partition set was defined to monitor 12 of the 48 partitions, then the Optim Performance Manager sizing would be reduced to:

- Disk space: **25 GB**
- Memory: **7 GB**
- CPU: **5 CPUs PV_5_2 64-bit 2*1.9GHz (or equivalent)**

3 Best Practices for configuring Optim Performance Manager to monitor large & active systems

After you have installed and activated Optim Performance Manager (and optionally, Extended Insight) you configure Optim Performance Manager. The configuration of Optim Performance Manager consists of specifying the database that you want to monitor and defining the data collection settings (for example, what type of data to collect, and how often to collect it).

Optim Performance Manager collects monitoring information from the database server and application (Extended Insight only) using a variety of techniques. For the database server, Optim Performance Manager generally employs the use of the DB2 snapshot facility along with various event monitors including deadlock, unit of work, and package cache. For application monitoring with Extended Insight, Optim Performance Manager makes use of hooks in the DB2 client (both Java and CLI) to record information about the SQL statements and transactions that are being executed.

Any form of monitoring will consume additional resources on the systems that are being monitored. These additional resources are required as the monitoring application gathers and processes the monitoring information. Optim Performance Manager has been carefully designed to minimize the impact this collection and processing has on both the application and database tiers. The amount of additional resources (and the impact this might have on application or database performance) varies from one environment to another. This section explains the main factors that influence the consumption of these resources and outlines best practices for configuring Optim Performance Manager for monitoring.

The additional resources required by Optim Performance Manager when monitoring application and database activity generally depend upon:

1. The Optim Performance Manager configuration, including:
 - i. The type of in-flight monitoring information that is being collected
 - ii. The granularity of the monitoring information that is being collected
 - iii. The interval at which the monitoring information is being collected
 - iv. The type of Extended Insight monitoring information that is being collected (if any).
2. Characteristics of the database that is being monitored
3. Characteristics of the application or workload that is being monitored
4. Resource consumption of the systems that are being monitored.

You define data collection settings by enabling *monitoring profiles*. There are multiple monitoring profiles available that collect different types of monitoring information, for example, locking, connection, and I/O information, or Extended Insight data.

The general rule of thumb is: the more monitoring profiles that are enabled, or the greater the level of detail collected, or the more frequently the data is collected, the greater the resources that are required (on the database server, client, and Optim Performance Manager server) to collect and process the monitoring information.

The remainder of this section will cover each of the four items listed above in more detail.

3.1 Best practices for configuring monitoring profiles – controlling the type of information collected

The types of monitoring information that is being collected directly influences the resources that are required to gather the information on the application (Extended Insight only) and database tiers. Additionally the types of collected monitoring information influences the resources required on the Optim Performance Manager server to process that data. Optim Performance Manager uses monitoring profiles to control the type of monitoring information that is collected. There are seven profiles, which are illustrated in Figure 1 below. (Note: there are additional monitoring profiles for the Performance Expert Client that are not considered in this document).

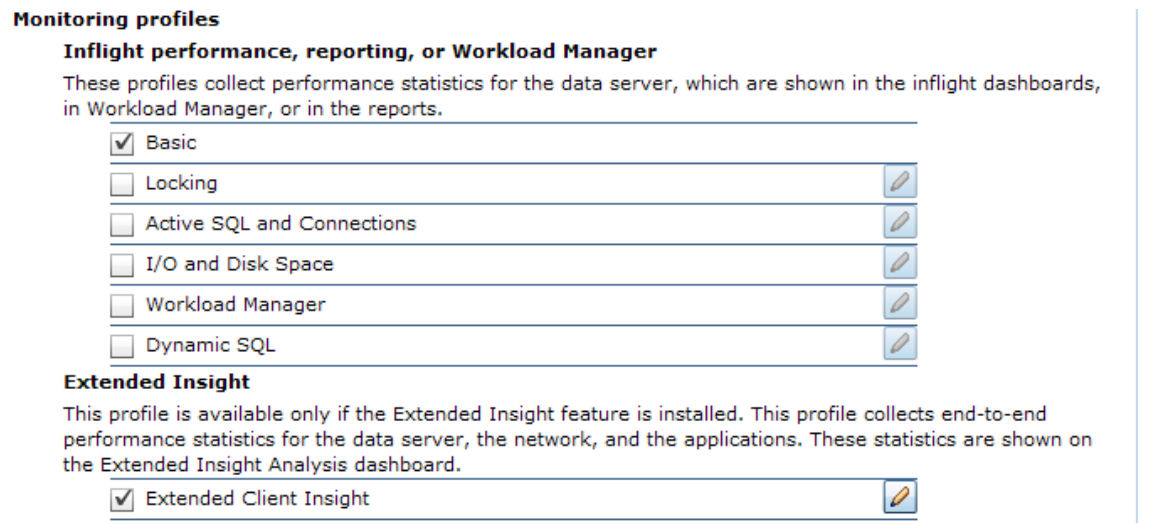


Figure 1 – Monitoring profiles

Not surprisingly, the more profiles that are enabled, the greater the resources required to collect that information on the monitored systems and the greater the resources required to process and store that information on the Optim Performance Manager server.

Optim Performance Manager best practices recommend that you should start by collecting the appropriate level of monitoring information that meets the immediate needs of the business, and no

more. After the monitoring configuration has been validated in your environment, additional monitoring information can be collected later if necessary.

The next sections describe each of the monitoring profiles in more detail. For additional information about each of the monitoring profiles, please refer to the Optim Performance Manager documentation or the Redbook *Chapter 3.3: Configuring Optim Performance Manager*.

3.1.1 Basic profile

The Basic profile uses the DB2 database manager, database snapshots, and DB2 APIs to collect database and database manager configuration information and basic operating system load. It is a very low overhead monitor and should be enabled for all databases, regardless of activity or size.

3.1.2 Locking profile

The Locking profile can be used to provide information about locking issues within your databases. Enable this profile if you suspect, or know that your monitored database has locking problems such as deadlocks, lock waits or lock timeouts. Figure 2 below shows the options that are available under the Locking profile if your monitored database is at DB2 V9.7 level or above:

The screenshot shows a dialog box titled "Locking" with a close button in the top right corner. The main instruction reads: "Specify the amount and type of lock information you want to collect." Below this, there are two sections: "Lock event monitor" and "Lock wait information".

Lock event monitor

- Enable lock wait warning alert
Lock wait threshold in microseconds: * 5000000
- Enable lock timeout alert
Lock wait timeout in seconds: Unlimited
- Enable deadlock alert
 - Use legacy deadlock event monitor
- Use custom table space: SYSTOOLSPACE | v
- Maximum table space fill size in percent: 90 | v
- Capture event details: Without statement history | v

Lock wait information

- Collect lock wait information
Special sampling rate in minutes: 1

At the bottom right, there are "OK" and "Cancel" buttons.

Figure 2 – Locking monitoring profile

Basic locking information can be collected by the lock event monitor using “Enable lock wait warning alert” and selecting the “Without statement history” option for the “Capture event details” field. This setting collects information for lock events requiring minimum additional resources on the monitored database and minimum resources on the Optim Performance Manager server to collect and store the data.

If you need to analyze multiple locking problems, use “Enable lock wait warning alert” and select the “Statement history” or “Statement history with values” option for the “Capture event details” field. These settings collect low level locking information and allow fine grained analysis of locking issues. However, they will increase the overhead on the monitored database because the DB2 database manager will keep all statements in memory in case a lock wait, deadlock, or time-out occurs. At the same time, Optim Performance Manager has more data to collect and process when an event occurs.

If you need to analyze deadlocks and timeouts, select “Enable lock timeout alert” or “Enable deadlock alert” respectively. These are additional options of the lock event monitor and do not increase the overhead of the event monitor on the monitored database much compared to enabling the lock event monitor by selecting “Enable lock wait warning alert”,

The lock event monitor is not available with DB2 V9.5 or lower. You can only enable the collection of deadlock alerts. In that case, Optim Performance Manager turns on the deadlock event monitor.

Selecting “Collect lock wait information” (which is based on lock and application snapshots) is useful if your locking problems consist mainly of lock waits and lock timeouts. With the information that is collected, you can analyze the lock tree along with all involved connections and statements. However, on a system with a lot of locking activity, the amount of monitoring data collected can be very large. It is recommended to only enable this option to solve specific locking issues, and after they are resolved, “Collect lock wait information” should be disabled.

Tip: If you select “Collect lock wait information”, ensure that the sampling interval is smaller than the lock timeout value of the monitored database. If the sampling interval is higher, Optim Performance Manager might not catch all lock wait situations.

If any of the event monitors are enabled (lock wait warning alert, lock timeout alert, or deadlock alert), follow these best practices:

- Create a dedicated 32K table space (across all partitions) and a dedicated buffer pool, and specify that the event tables are to be created in the newly created table space when you configure the locking profile.

The table space should use 32K pages because the event table contains a BLOB column and a 32K page table space allows 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.

- If your monitored database is a non-HADR (High Availability Disaster Recovery) database, create the table space with the option **DROPPED TABLE RECOVERY OFF**.

- To account for peak periods, size the buffer pool to hold approximately two to three minutes worth of event monitor data. This allows Optim Performance Manager to read the monitor data from the buffer pool and thus reduces disk I/O.
- If at all possible, locate the event monitor table space on a set of dedicated disks.

On active systems, during normal operation, the Locking profile should be disabled, or only “Enable lock wait warning alert” with “Without statement history” option information should be collected. Only enable collection of more detailed locking data if you suspect or know that your monitored database has locking problems. After the problem has been solved, collection of detailed data should once again be disabled.

3.1.3 Active SQL and Connections profile

The Active SQL and Connections profile uses DB2 application snapshots to collect monitoring data from the database server. Collecting this data can be useful to analyze the performance of long running SQL statements.

Optim Performance Manager will only catch SQL statements that are currently executing when the snapshot is taken. Consequently, Optim Performance Manager might not catch SQL statements if their execution time is shorter than the sampling interval.

Tip: In OLTP environments with short running statements, many of the statements might not be caught. Therefore, consider disabling this profile in these environments and use Extended Insight, which will provide the necessary information.

On very active systems that have a large number of database connections, the amount of monitoring data that is collected for this profile can be considerable. In these environments, consider specifying a higher sampling rate than the global rate if collection of this data is deemed appropriate, as illustrated in figure 3.

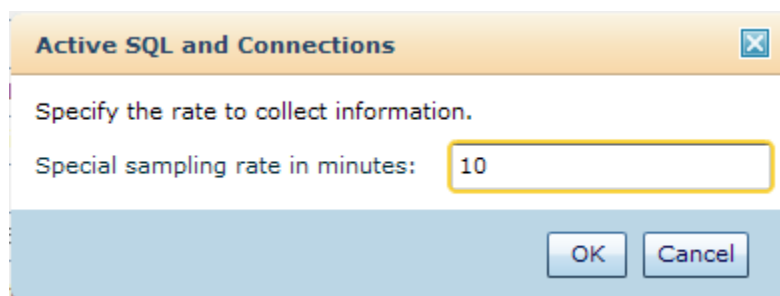


Figure 3 – Active SQL and Connections monitoring profile

3.1.4 I/O and Disk Space profile

Enabling the I/O and Disk Space profile collects information on buffer pool activity within your database and it is likely to be one of the key metrics for determining the health and performance of the monitored database. Optim Performance Manager uses a combination of the database manager, database, and buffer

pool snapshots to collect this information. Because buffer pool activity is a key health metric, it is recommended that this profile be enabled for all of your monitored databases.

You can also collect more detailed information for tables, table spaces and table space containers (which use the underlying DB2 table and table space snapshots) using this profile.

However, on large systems – that is, systems with a large number of tables, table spaces or table space containers – it is recommended to disable the collection of table, table space and container data during normal operation, and only enable it for detailed debugging of suspected issues involving these objects. This recommendation is due to the large quantities of monitoring data that these options can produce.

As an example, consider a partitioned database environment that has 32 data partitions, each partition contains 80 table spaces (the same 80 on each partition), and each table space has eight containers. On this system, Optim Performance Manager would collect information for $32 \times 80 = 2560$ table spaces and $32 \times 80 \times 8 = 20480$ containers during each sampling interval.

Similarly, collection of data for tables on systems which contain many tables (for example SAP, which regularly contains 50,000+ tables) should not be enabled by default because this would require Optim Performance Manager to collect, process, and store information for all accessed objects during the sampling interval.

It is often more useful to collect long term trend data for tables, table spaces and table space containers so that very frequent collection of this data is not necessary. As a result, IBM has enhanced the I/O and Disk Space profile to enable customization of the collection interval for tables, table spaces and table space containers. This feature can be enabled by applying APAR PK99928.

3.1.5 Workload Manager profile

Collect this data only if you want to use the Workload Manager Configuration Tool within Optim Performance Manager. If you enable this profile, a low overhead statistics event monitor is started on the monitored database. If you enable it for a high number of monitored databases or for a partitioned database, it is a best practice to increase the sampling rate for statistics – for example 10 minutes for a large number of monitored databases and 30 minutes for a partitioned database, as illustrated in figure 4. This is the interval in which Optim Performance Manager flushes the statistics event monitor data and inserts the collected data into the repository database. Setting the sampling rate to a higher value saves disk space on the Optim Performance Manager machine without the loss of much information.

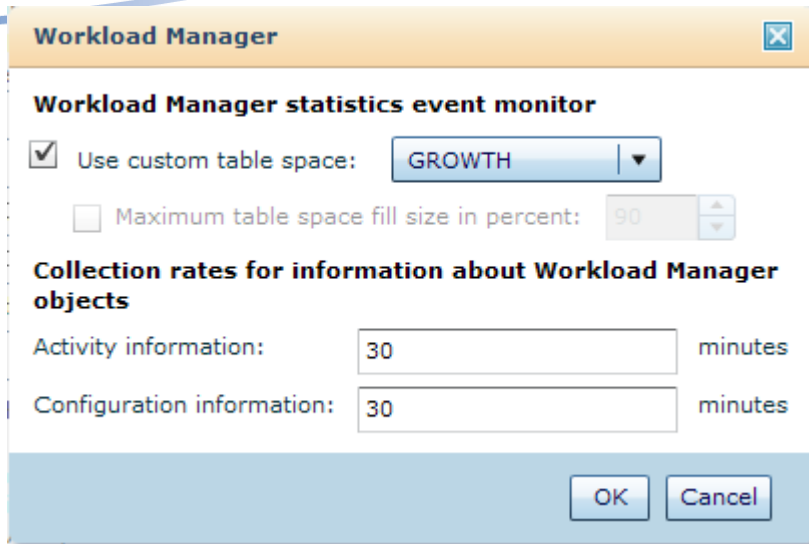


Figure 4 – Workload Manager monitoring profile

Tip: Keep in mind that DB2 software only supports one active statistic event monitor per monitored database. Therefore, if you have already started a statistic event monitor on the monitored database, Optim Performance Manager cannot start another one. Attempting to start more than one statistic event monitor will result in an error message in the db2pesrv.log file indicating that there is already an active statistic event monitor. Ensure that no statistic event monitor is running before you start Optim Performance Manager with the Workload Manager profile enabled.

As for any event monitor that Optim Performance Manager creates on the monitored database, IBM recommends that you use a dedicated table space (across all partitions) on the monitored database.

3.1.6 Dynamic SQL profile

The “Dynamic SQL” profile uses the Dynamic SQL snapshot to collect execution statistics for SQL statements that are currently stored in the package cache. This data is not used directly by any in-flight dashboards and is only used by the “Dynamic SQL Statement” report. The amount of data that is collected depends upon the number of SQL statements in the package cache -- and the number of SQL statements in the package cache is dependant upon the size of the cache and the length of the SQL statements within it.

During each sampling interval, the entire contents of the package cache is retrieved and stored by Optim Performance Manager – even if there has been no activity on the database. Therefore, this profile has the potential to generate very large volumes of monitoring data. Because the Dynamic SQL profile data is only used in the “Dynamic SQL Statement” report, this option should be disabled unless you are specifically collecting the information to view in the report.

If you do want to collect this data and you have a large package cache containing a large number of statements, then you should increase the sampling interval for this profile. This scenario could occur on a very active system that is executing many SQL statements per second, or on a partitioned database

environment that has a large package cache. On these types of systems, IBM recommends that you specify a sampling interval for this profile of 10 minutes or greater.

3.2 Best practices for configuring monitoring granularity – controlling the level of detail collected

Optim Performance Manager provides capabilities to control the granularity of the information that is collected within the individual monitoring profiles – some of this has already been discussed. Collection of detailed data will allow low level analysis of database and application activity – but is likely to incur additional overhead and will require more resources on the Optim Performance Manager Server to process and store the information.

As an example, let’s look at the “I/O and Disk Space” profile. Figure 5 below shows the options that are available to collect finer detailed data for “I/O and Disk Space”:

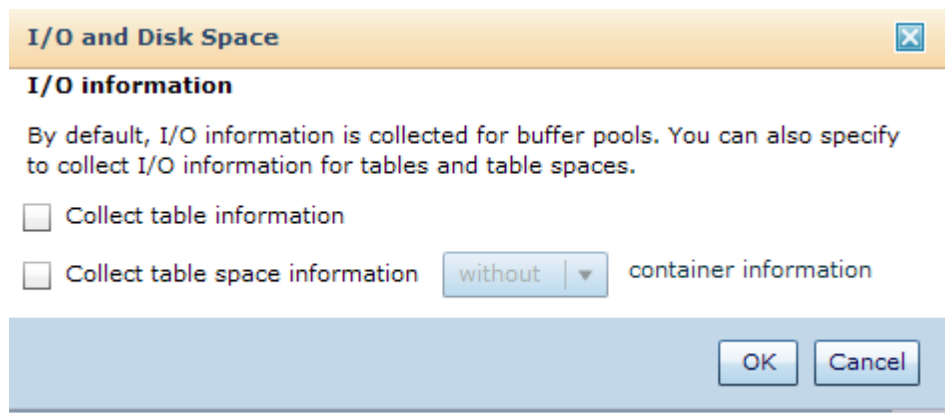


Figure 5 – I/O and Disk Space monitoring profile

By default, I/O information is collected for the buffer pools and this is likely to satisfy most requirements. However, it is possible to collect table, table space and container information during each sampling interval. Although enabling the collection of this information on most systems will not be an issue, it might require significant additional resources if there are very large numbers of these objects in your database. For example, SAP databases regularly contain more than 50,000 tables. If “Collect table information” is enabled, Optim Performance Manager will gather information on all of the tables that are accessed (using snapshots) during each sampling interval. Therefore, Optim Performance Manager will potentially have to gather, process, and store the information for a large number of tables during each monitoring interval.

As a consequence, it is important to understand the implications of collecting such data within very large or very active systems. If you do choose to collect fine grained detail on these systems, then you should balance this with an increase in the sampling rate of the monitored data to allow Optim Performance Manager sufficient time to process and store the data before the next iteration.

3.3 Best practices for configuring monitoring sampling rate – controlling the frequency of data collection

The global sampling rate controls how frequently in-flight monitoring information is requested from the database server. The default, and minimum, is 1 minute. The sampling rate for the Extended Insight Client and Server profiles is not configurable and is set to 1 minute.

The sampling rate can be set at the global level, and also fine tuned under many of the individual monitoring profiles, as illustrated in figures 6 and 7:

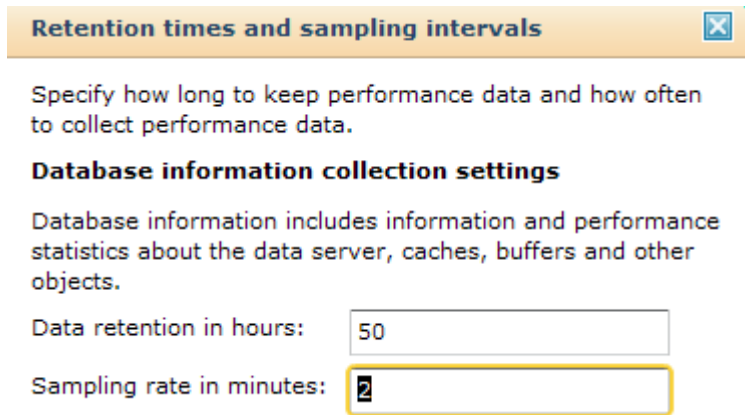


Figure 6 – Retention times and sampling rate setting dialog

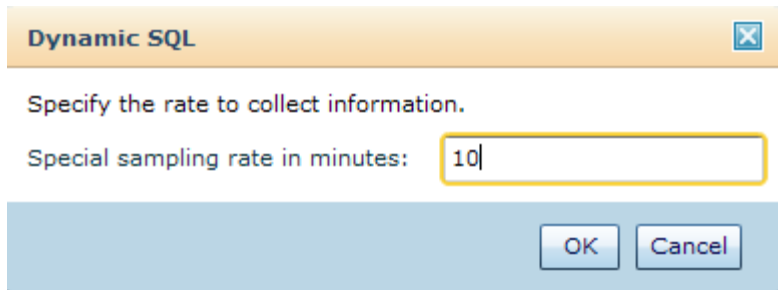


Figure 7 – Dynamic SQL monitoring profile

Figures 6 and 7 above show a global sampling rate of 2 minutes, but a customized sampling rate for Dynamic SQL data collection of 10 minutes.

Tip: Note that any customized sampling rate must be an integer multiple of the global sampling rate. If you change the global sampling rate, then any customized sampling rate will be overwritten and set to the global sampling rate. Therefore, you will need to re-set the customized sampling rates if the global sampling rate is changed.

Because in-flight data is not aggregated, the sampling rate has a direct impact on the amount of disk space that is required on the Optim Performance Manager server to hold the monitoring data -- that is, halving the sampling rate will double the amount of disk space that is required to hold the data.

There is also a linear relationship between the retention time and disk space that is required by Optim Performance Manager to store the in-flight monitoring data. So if you double the retention time, the amount of disk space that is required to hold the in-flight data doubles.

The shorter the sampling rate, the more frequently the data will be collected from the database server. Consequently, Optim Performance Manager will consume more resources on the monitored systems, and more resources will be required on the Optim Performance Manager server to process the data in a timely manner.

During the sampling interval, it must be possible for Optim Performance Manager to gather the data from the monitored data server, process the data, and store it. If Optim Performance Manager finds that this is not possible, a message is printed in the log and data collection is interrupted to prevent an ever increasing backlog of monitoring data that needs processing. To resolve the issue, either increase the sampling rate or reduce the amount of monitoring data collected.

The optimal sampling rate is a fine balance between the need to perform fine grained analysis of the monitoring data, the impact on the monitored system, and the resources available on the Optim Performance Manager server to process the data in a timely manner. Best practices recommend to start with a relatively high sampling rate which will meet the majority of your requirement and fine tune the rate on individual profiles to meet specific monitoring requirements.

3.4 Best practices for configuring Extended Insight monitoring

Extended Insight monitoring is a combination of three different sub-profiles of monitoring - "Collect statement and transaction metrics on client", "Collect statement metrics on data server" and "Collect transaction metrics on data server", as illustrated in figure 8:

Figure 8 – Extended Insight monitoring profile

It is not possible to enable “Collect statement metrics on data server” without first enabling “Collect statement and transaction metrics on client”.

3.4.1 Collect statement and transaction metrics on client

The “Collect statement and transaction metrics on client” sub-profile (often called Extended Client Insight) provides DBAs with valuable information by monitoring SQL activities that originate on the application or client tier. The profile does this by recording information about SQL statements and transactions at the client-side and periodically sending this information to the Optim Performance Manager Server. Information regarding execution of the same SQL statements is gathered on the database server by using the dynamic SQL snapshot feature in DB2 V9.1 and V9.5, and the package cache event monitor in DB2 V9.7 or higher. Both client and server side details for the SQL statements are then matched at the Optim Performance Manager server, which is then able to provide the full end-to-end view of each individual SQL statement.

Optim Performance Manager has hooks into the DB2 JDBC and CLI drivers which allow it to collect this client side information. These hooks record information about the statements and transactions that are 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 is kept. For transactions, a unique transaction ID is used for aggregation rather than calculation of a hash code. 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 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-side 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.

Applications that issue a large number of unique SQL statements during a 1-minute interval are likely to generate more work for the Optim Performance Manager Extended Insight client component. Because many of the statements are unique, it is not possible to group the statements together in the in-memory table, and consequently the table and the memory required to monitor the application will grow. This situation might occur if the application uses literals instead of parameter markers. Using literals will cause many of the SQL statements to be unique and a best practice recommends that applications should use parameter markers wherever possible.

For example, consider a workload that issues 1,000 SQL statement per minute, with 500 unique statements within the workload (that is, 50% are unique). The aggregated hash table for monitoring this application will have 500 entries during any 1-minute monitoring interval. Now, consider an almost identical workload with 1,000 statements per minute, but only 10 unique statements. The aggregated hash table for this workload will contain only 10 entries and will occupy approximately 50 times less memory on the client. It will also require less network bandwidth to send this data to the Optim Performance Manager server and fewer resources on the Optim Performance Manager server to process this data.

In addition, the higher the rate at which transactions and SQL statements are executed, the more resources the 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 BI type workloads, simply because an OLTP application will execute far more statements and transaction per minute than a BI application.

Much of the 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, reduces the impact client-side monitoring has on application throughput. In most cases, any overhead introduced on the application by Extended Insight (EI) Client monitoring will be negligible. Only if you are monitoring a very high volume OLTP workload

with very short transaction elapsed times (measured in milliseconds) are you likely to notice a small increase in transaction time.

On the database server side, we have already discussed best practices for collecting the information on DB2 V9.1 and DB2 V9.5 under the Dynamic SQL profile section of this document. For DB2 V9.7 and above, Optim Performance Manager collects the server-side information using the package cache event monitor. This event monitor captures information when entries are flushed from 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 that you can follow to minimize this impact even 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 the figure 9 below.
- 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.
- If your monitored database is a non-HADR database, create the table space with the option **DROPPED TABLE RECOVERY OFF**

The screenshot shows a dialog box titled "Collect Extended Insight data" with three tabs: "Collection of monitorin...", "Usage of client field information", and "Integration with Tivoli Monitoring". The "Collection of monitorin..." tab is active. It contains several sections with checkboxes and input fields:

- Collect statement and transaction metrics on client**
 - Currently known clients: [View]
 - Port number for the Extended Insight client application that you configured: Dynamic, Custom * 6666
 - Use logical database lookup name: []
 - Package cache event monitor settings:** You can override the default table space that is used to monitor event data.
 - Use custom table space: IRWWDATA1
 - Maximum table space fill size in percent: 90
- Collect statement metrics on data server**
- Collect transaction metrics on data server**
 - UOW event monitor settings: You can override the default table space that is used to monitor event data.
 - Use custom table space: IRWWDATA1
 - Maximum table space fill size in percent: 90

At the bottom right, there are "OK" and "Cancel" buttons.

Figure 9 – Extended Insight monitoring profile, package cache event monitor settings

- To account for peak periods, size the buffer pool to hold approximately two to three minutes worth of package cache event monitor data. This will allow Optim Performance Manager to read the monitored data from the buffer pool and thus reduce disk I/O.
- If possible, locate the package cache monitor table space on a set of dedicated disks.

A record is inserted into the package cache event table for each statement that is expelled from the cache, and each of these inserts is logged by the database manager. Therefore, enabling this profile will cause additional logging on the monitored database server. The amount of additional logging incurred is dependent upon the rate at which SQL statements are expelled from the cache – the higher the expulsion rate, the greater the additional logging. Only in very extreme cases where there is a very low package cache hit ratio with a high volume transaction workload will the additional logging incurred be noticeable. In these cases, it is important to monitor logging activity on the database server to ensure that the additional logging does not become a bottleneck, and if it does, tune logging related parameters accordingly.

3.4.2 Collect statement metrics on data server

It is only possible to enable this option when the “Collect statement and transaction metrics on client” is also enabled. The “Collect statement metrics on data server” also uses the DB2 package cache event monitor to gather monitoring information from the database server. Specifically, when this option is enabled, Optim Performance Manager gathers information from the event monitor about the statement *metrics*. As described in the previous section, the “Collect statement and transaction metrics on client” option also uses the DB2 package cache event monitor, but in this case, it is used to collect information about the statement *text*, and not the statement metrics.

Consequently, all of the best practices detailed previously for using the DB2 package cache event monitor to gather data for Optim Performance Manager also apply when enabling this profile.

3.4.3 Collect transaction metrics on data server

The “Collect transaction metrics on data server” sub-profile is used to gather information about the transactions that are executed on the database server. When this sub-profile is enabled, Optim Performance Manager uses the DB2 unit of work event monitor to gather information. 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 unformatted event table.

The unit of work event monitor is supported on DB2 V9.7 Fix Pack 1 and above. However, to improve efficiency of this monitor, it is highly desirable to install DB2 V9.7 Fix Pack 2 on the monitored database server.

On a partitioned system, the amount of data retrieved from the unit of work event monitor depends upon the number of partitions touched by each transaction. If a transaction touches many partitions, information related to that transaction will be placed into the event table on each partition. Consequently, there might 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 best practices you can follow to minimize this impact even further:

- Create a dedicated 32K table space (across all partitions) and a dedicated buffer pool, and specify that the event tables are to be created in the newly created table space when configuring this monitoring profile, as shown in the figure 10 below. The table space should use 32K pages because the 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.
- If your monitored database is a non-HADR database, create the table space with the option **DROPPED TABLE RECOVERY OFF**

The screenshot shows a dialog box titled "Collect Extended Insight data" with three tabs: "Collection of monitorin...", "Usage of client field information", and "Integration with Tivoli Monitoring". The "Collection of monitorin..." tab is active. It contains several sections with checkboxes and input fields:

- Collect statement and transaction metrics on client** (checked): Includes a "View" button, radio buttons for "Dynamic" and "Custom" (selected), and a text input field containing "6666".
- Use logical database lookup name:** (unchecked) with an empty text input field.
- Package cache event monitor settings:** Includes a note: "You can override the default table space that is used to monitor event data."
 - Use custom table space:** (checked) with a dropdown menu showing "IRWWDATA1".
 - Maximum table space fill size in percent:** (checked) with a spinner set to "90".
- Collect statement metrics on data server** (checked)
- Collect transaction metrics on data server** (checked): This section is circled in red.
 - UOW event monitor settings:** Includes a note: "You can override the default table space that is used to monitor event data."
 - Use custom table space:** (checked) with a dropdown menu showing "IRWWDATA1".
 - Maximum table space fill size in percent:** (checked) with a spinner set to "90".

At the bottom right, there are "OK" and "Cancel" buttons.

Figure 10 – Extended Insight monitoring profile, UOW event monitor settings

- Size the buffer pool to hold approximately two to three minutes worth of unit or work event monitor data - to account for peak periods. This will allow Optim Performance Manager to read the monitor data directly from the buffer pool and thus reduce disk I/O.
- If possible, locate the table space on a set of dedicated disks.

A record is inserted into the event table for each completed unit of work and each of these inserts is logged by the database manager. Therefore, enabling this profile will cause additional logging on the monitored database server.

The amount of additional logging incurred is dependent upon the transaction rate – the higher the transaction rate, the greater the additional logging. However, the proportional increase you will experience in logging depends upon the mix of read and write transactions within your workload. If your workload is heavily read biased, then enabling this option may result in a proportionally significant increase in logging activity. This is because the read transactions will be recorded in the event table and thus incur logging – whereas, before monitoring, these same read transactions would not have incurred any logging.

As a consequence, it is important to monitor logging activity on the database server to ensure the additional logging does not become a bottleneck, and if it does, tune logging related parameters accordingly.

3.5 Characteristics of the database being monitored

The main factor influencing Optim Performance Manager’s consumption of resources on the database server is the number of objects being monitored -- and quite obviously, the greater the number of objects being monitored, the greater the resources required to gather that data. In addition, more monitored objects will require greater resources on the Optim Performance Manager server to process the data,

For example, enabling the Extended Insight Server monitoring profile is likely to require more resources when monitoring an OLTP system executing 1,000 dynamic SQL statements per second than it will when monitoring a BI workload executing just a handful of SQL statements per second. Because in this case monitoring of the OLTP system will produce more data to process, the Optim Performance Manager server will also require more resources.

Interestingly, the volume of data in the monitored database does *not* have a direct impact on resources that are required to monitor that database.

3.6 Characteristics of the application or workload being monitored (Extended Insight Client)

The key characteristics of the application or workload which influence monitoring have been covered in the “Best practices for configuring Extended Insight monitoring” section of this document.

3.7 Resource consumption on the systems being monitored

Any form of monitoring will consume machine resources on the systems being monitored. The main resources consumed by Optim Performance Manager on the application tier and database server are CPU, memory, disk and network.

In some cases, resource utilization on the application server or database server can influence monitoring overhead. This is only likely to be the case when the enablement of Optim Performance Manager monitoring on one of the servers requires resources which are already constrained on that machine. In some extreme cases, when the resource required is severely constrained, the overhead of enabling

monitoring can be significant. In all cases, it is imperative to ensure there are sufficient resources on the monitored servers to allow Optim Performance Manager to collect the required data. Reducing the amount of monitoring data collected is also an option, but freeing up system resources is likely to be more effective.

One common case where resource constraints might occur is during batch processing. Batch jobs are usually designed to run the machine to near full capacity. Because batch jobs rarely have the same monitoring requirements as on-line transactions, the same requirements on the business to monitor performance of the batch jobs does not usually exist. Therefore, during these periods the amount of Optim Performance Manager monitoring can be greatly reduced or even disabled altogether.

4 Best practices for configuring Optim Performance Manager to monitor a large number of databases

The number of databases Optim Performance Manager can monitor depends upon the resources of the Optim Performance Manager server and the level of monitoring data being collected. However, the more databases that are being monitored, the more work Optim Performance Manager has to do in parallel to process and store the data, which could lead to contention for one or more Optim Performance Manager server resources.

When monitoring a large number of systems, or few very large and active systems, particular attention has to be paid to the bandwidth of the network adapter card on the Optim Performance Manager server. If the Optim Performance Manager server's network card is running near full capacity, then Optim Performance Manager will be unable to receive the data from the monitored data servers in a timely manner and consequently the data will not be processed or stored. In these cases, it may be necessary to reduce the amount of monitoring data that is being collected by adjusting monitoring profiles and sampling intervals, or to increase the capacity of the network adapter card, or to split the monitoring of your database environment across multiple Optim Performance Manager servers.

In addition, the actual size of the Optim Performance Manager repository database needs to be considered. Generally, the repository database will scale well up to approximately 250GB in size. However, this is not a hard limit because Optim Performance Manager repository databases that monitor a large number of databases will generally scale better than Optim Performance Manager repository databases of a similar size that monitor only a few databases.

Best practices recommend that you start with conservative data collection settings if you want to monitor a large number of databases. Optim Performance Manager offers predefined system templates that already have monitoring profile settings predefined associated with the specific type of workload that you run on your monitored database. Start by choosing one of the 'low overhead' system templates for each of the databases depending upon the kind of workload and refine the settings later if necessary. The current 'low overhead' monitoring profiles are:

- OLTP production with Low Overhead
- BI production system with Low Overhead
- SAP Business Information Warehouse production with Low Overhead
- SAP Enterprise Resource Planning production with Low Overhead

'Low overhead' means that the collection settings defined in these profiles minimize the overhead on the monitored database itself, which also minimizes the work Optim Performance Manager has to do to collect and store the data.

Best practices recommend that in the case of monitoring partitioned databases with many partitions or monitoring databases with a high transaction rate using Extended Insight, the number of monitored databases per Optim Performance Manager server should not exceed a single digit number.

4.1.1 Retention times and sampling intervals

The greater the number of databases being monitored by a single Optim Performance Manager instance, the higher you should set the global sampling rate for each of the databases. The sampling rate should be set such that Optim Performance Manager is able to process and store all monitoring data for each of the databases within the sampling period. If the sampling rate is too small, or there are insufficient resources on the Optim Performance Manager server, then Optim Performance Manager will not be able to process the data within the sampling period. In this case a message will be written to the Optim Performance Manager log file db2pesrv.log indicating that processing could not complete in the time allowed. When these messages are issued, there are three options to resolve the issue, which can be implemented individually or together:

1. Increase the sampling rate to allow more time for the data to be processed
2. Decrease the amount of monitoring data being collected by disabling one or more of the monitoring profiles
3. Increase the resources available to the Optim Performance Manager server.

Best practices recommend that you should initially set the global sampling rate to at least two minutes and specify lengthier sampling rates for heavy monitoring profiles in terms of amount of collected data, for example, Active SQL and Connections or Dynamic SQL. If you find that data could not be collected in time although you have set bigger intervals for heavy monitoring profiles then increase the global sampling rate.

Tip: Increasing the global sampling interval overwrites each customized sampling rate for all other monitoring profiles. Therefore, you might need to go back into the profiles and reset the customized sampling rate.

5 Best practices for monitoring partitioned database environments

In the previous chapter, we described general considerations for Optim Performance Manager monitoring profiles. These same considerations also apply when you are monitoring partitioned database environments. The remainder of this section discusses considerations specific to monitoring partitioned database environments.

Partitioned database environments are not commonly used to process high volumes of transactions. However, the very nature of partitioned database environments means that they can potentially contain very large numbers of objects – particularly tables, table spaces and table space containers. For example, a single logical table in a partitioned database environment will result in Optim Performance Manager collecting and processing information for that table across each partition – so instead of collecting information for one object, Optim Performance Manager is actually collecting information for ‘n’ objects (where ‘n’ is the number of partitions the table is spread across). Therefore, on partitioned database environments particular attention needs to be paid to the number of objects being monitored, in particular, when the I/O and Disk Space profile and the Dynamic SQL profiles are enabled. If the monitored database has a large number of partitions and contains a large number of these objects, configure a higher sampling interval for these profiles and reduce the data collected in the I/O and Disk Space profile.

It is also considered a best practice for a database with a large number of partitions to increase the global sampling interval when configuring monitoring, for example, start with 30 minutes. If you enable the Workload Manager profile, then set the sampling rate for statistics to 30 minutes as well. The sampling rates can be fine tuned at a later date after you are sure that there are sufficient resources on the database and Optim Performance Manager server to gather and process the monitoring data.

After you have configured a partitioned database for monitoring, open the ‘Configure Monitoring wizard’ again and go to Step 5 of the wizard. This step was not displayed when you first configured the database because at that point Optim Performance Manager had not detected that the database was partitioned.

Step 5 allows you to annotate roles to partitions, create partition sets, and select the active partition set. A partition set contains a subset of all available partitions. The active partition set represents the set of partitions for which Optim Performance Manager is currently collecting monitoring data. By defining and using partition sets in this manner, you reduce the amount of monitoring data that is collected and therefore you optimize for performance and reduce disk space consumption. Essentially, you define and use partition sets to collect a representative sample of monitoring data from a subset of the partitions.

Annotating roles to partitions is useful if you have dedicated coordinator, data, ETL or catalog partitions. After you annotate roles to partitions, a dedicated partition set for each role is predefined by Optim Performance Manager. You can assign different alert thresholds to different roles and on some dashboards you can select a role partition set to display data just for that role.

The following screenshots illustrate how to annotate partitions and create partition sets.

First, select a role as show in figure 11:

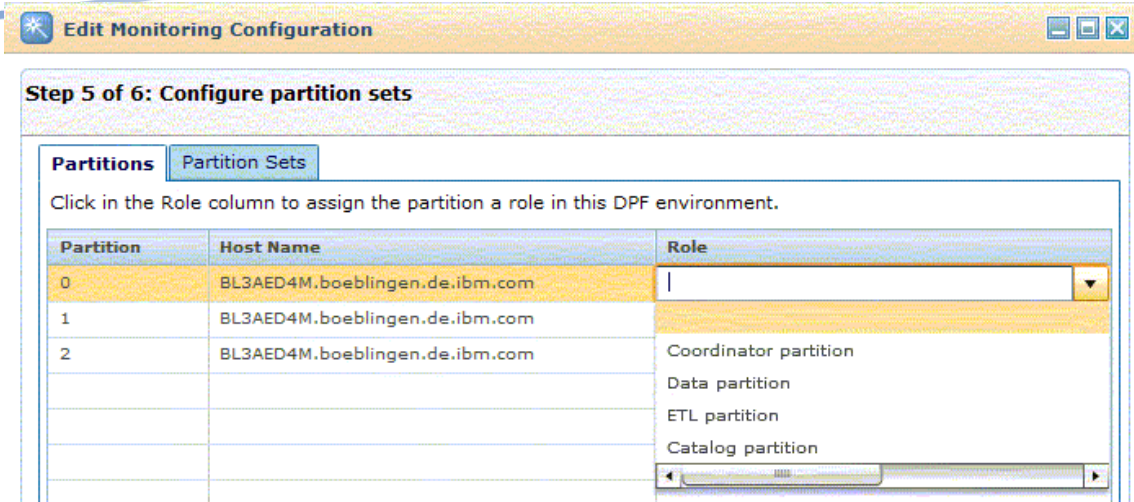


Figure 11 – Role annotation to partitions

When role selection for each partition is complete, each role is displayed, as shown in figure 12:

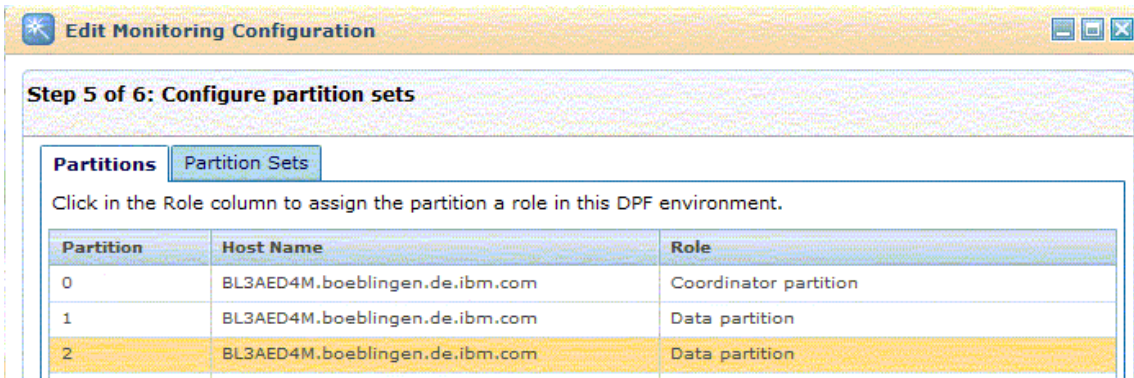


Figure 12 – Complete role annotation to partitions

Then select the Partition Sets tab to define partition sets, as shown in figure 13:

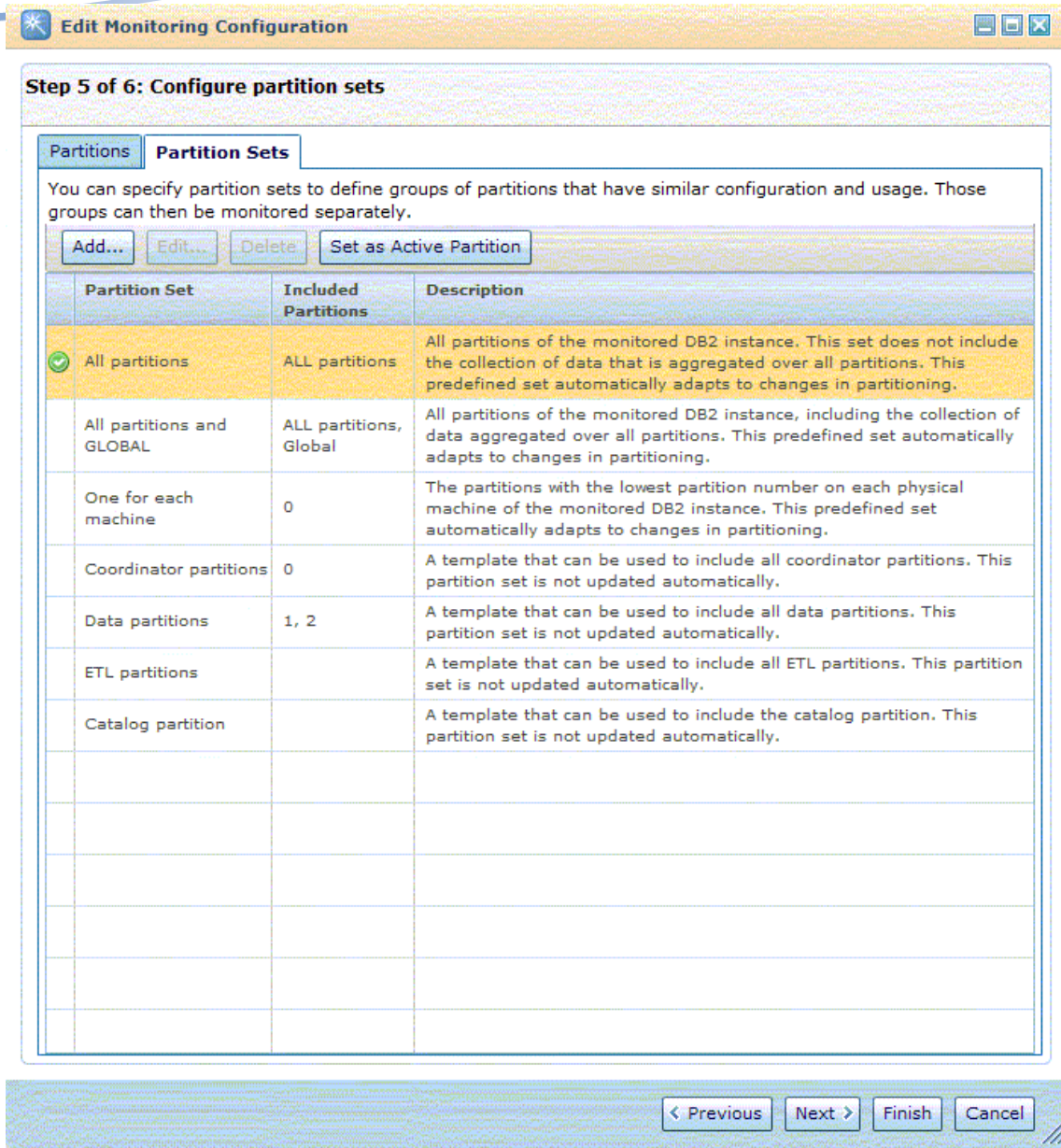


Figure 13 – Partition set selection and specification

By default Optim Performance Manager has three predefined partition sets (All partitions, All partitions and GLOBAL and One for each machine). 'All partitions' is the active partition set. This means that Optim Performance Manager collects data for all available partitions. The role partition sets (Coordinator partitions, Data partitions, and so on) are templates and only become valid partition sets to use if you annotate roles to partitions.

The GLOBAL snapshot is a high overhead and time consuming snapshot on the monitored database because the monitoring data is aggregated over all partitions before it is returned to Optim Performance Manager. In addition, the Optim Performance Manager web console dashboards do not display data that is collected by the GLOBAL snapshot because Optim Performance Manager aggregates the data itself

over the collected partitions before it displays the data. The only exception to this is the Dynamic SQL Report, which uses the GLOBAL snapshot to show aggregated statement data. Additionally, Performance Expert (PE) Client displays GLOBAL snapshot data.

Unless the globally aggregated data in the Dynamic SQL Report is required, or there is a requirement to use the PE Client then best practices recommend that you do not collect data from the GLOBAL snapshot, and therefore do not set the partition set 'All partitions and GLOBAL' as the active partition set.

It is possible to disable collection of the GLOBAL snapshot even if the "All partitions and global" partition set is accidentally set as active.

To disable the collection of global snapshots, perform the following steps:

1. Connect to the repository database.

2. Update the PARAMETER table by executing this SQL statement:

```
update db2pm_<instance ID>.parameter set PA_FLAGVALUE='Y' where  
PA_KEY='DISABLE_GLOBAL_SNAPSHOT'
```

where <instance ID> is a unique number that identifies the monitored DB2 database for which you want to disable global snapshots.

To view the instance IDs of all registered DB2 instances, use the '**peconfig -list**' command from the `RepositoryServer/bin` directory of your Optim Performance Manager installation.

For all but the smallest of partitioned database environments, Optim Performance Manager best practices recommend the use of partition sets to provide a representative sampling of the monitored data across the database partitions and thus reduce the amount of data collected and processed by Optim Performance Manager. On medium to large partitioned database environments, the use of partition sets should be considered mandatory.

If your partitioned database environment consists of multiple logical partitions on multiple physical hosts, for example, four logical partitions on each of eight physical hosts resulting in 32 partitions, then best practices recommend the use of the partition set 'One for each machine'. Assuming that the data is well partitioned without data skews across the logical partitions of one physical host, using this partition set will provide a representative monitoring sampling.

6 Best practices for monitoring DB2 pureScale™ systems

Currently Optim Performance Manager provides DB2 pureScale monitoring for either the whole data sharing group or for specific members. Dedicated monitoring of the cluster caching facility is not yet available. Therefore DB2 pureScale monitoring can be considered similar to monitoring a partitioned database environment in terms of the amount of data collected and the work Optim Performance Manager has to do to process and store the data. However, there are a few key differences.

Typically DB2 pureScale systems run high-volume OLTP workloads. Best practices recommend starting with the 'OLTP production with Low overhead' system template and increasing the global sampling rate. Additionally, edit the Locking profile and enable only the 'Lock Wait Warning Alert'. When enabling collection of Extended Insight data, particular attention needs to be paid to the volume of transactions

and SQL statements that are being processed by the DB2 pureScale system. If the volume is very large, Optim Performance Manager might not be able to keep pace with the volume of monitoring data generated and will shut down monitoring of these components to prevent the monitoring data from overwhelming the system. In these cases, the same best practices as documented in the “Best practices for configuring Extended Insight monitoring” section apply.

As for partitioned database environments, particular attention needs to be paid to the number of objects being monitored. In particular, this applies when the I/O and Disk Space profile and the Dynamic SQL profiles are enabled. If the monitored database has a large number of members and contains a large number of these objects, configure a higher sampling interval for these profiles and reduce the data that is collected in the I/O and Disk Space profile.

The following limitations currently apply when monitoring DB2 pureScale systems:

- Partition sets for DB2 pureScale systems do not apply - the data for all members is collected.
- The DB2 PureScale Feature does not currently support the DB2 Workload Manager. Therefore, do not use ‘Workload Manager Configuration’ within Optim Performance Manager and always ensure that the Workload Manager monitoring profile is disabled.

7 Running Optim Performance Manager

In this chapter are some hints and tips for managing Optim Performance Manager and ensuring that it runs with optimal performance.

Refer to Redbook *Appendix A. Managing the repository server and the repository database*. Important sections in this context are:

- A.2 How the repository server works
- A.4 Deleting data from the repository database
- A.5 Automatic runstats and reorganization in the repository database
- A.7 Changing database configuration of the repository database
- A.8 Enabling row compression for the repository database

7.1 Verifying Optim Performance Manager set up

After you have done the initial configuration and let Optim Performance Manager monitor the databases for a while we recommend checking the db2pesrv.log to see whether Optim Performance Manager runs with optimal performance.

Besides listing which monitoring threads are started for each monitored database, the db2pesrv.log file indicates error conditions such as out of memory errors or indicates if there are collection interval bottlenecks where the collection of data took longer than expected.

To learn more about collection interval bottlenecks and how to handle them, refer to the Redbook *Chapter A.12 Determining collection interval bottlenecks*.

To learn more about memory bottlenecks refer to the Redbook *Chapter A.11 Changing Java heap size parameters of the repository server*.

Memory bottlenecks can also occur if the system memory or the ulimit parameter of the DB2 fenced user or instance owner of the DB2 instance Optim Performance Manager is running on is too small. If this happens, you will see messages like this in the db2pesrv.log:

- 15:38:34.284][1] the snapshot category [locking] is disabled due to memory problems. Optim™ Performance Manager will try to enable the category again in a few minutes.
[15:38:34.284][1]If the problem persists, increase the physical memory for Optim Performance Manager or disable the snapshot category [locking] permanently.
- [13:04:51.204][1]Terminating: Error while taking history snapshot. PMGETLIST failed, rc=-101: Allocation of 134104960 bytes in file: tablemem.c at line: 286 failed.

To resolve the problem, do these tasks:

- Increase the physical memory for Optim Performance Manager.
- On Linux and UNIX, do the following:
 1. Set the ulimit parameter for the fenced user of the DB2 instance on which Optim Performance Manager is running to a higher value or to unlimited.
If a fenced user is not specified, check the ulimit parameter of the DB2 instance on which Optim Performance Manager is running and increase the value of the ulimit parameter or set it to unlimited.
 2. Restart the DB2 instance on which Optim Performance Manager is running for the changes to become effective.
- If these steps fail, and if the problem is caused by the snapshot category locking, permanently disable the collection of lock wait information in the Locking monitoring profile.

8 Summary: Monitoring is a balancing act!

Monitoring is a balancing act. You need to balance the need for monitoring information with the impact the collection and processing of this information has on the systems being monitored. At the end of the day, monitoring is there to fulfill a specific purpose within your organization, so be sure to fully understand the organization's requirements for monitoring and plan the Optim Performance Manager implementation accordingly.

Monitoring best practices recommend that you should start by collecting the level of information that will meet the immediate monitoring needs of the business. If additional business requirements surface which require additional information, then collection of the required information can be enabled at that time.

Important: Do not start by collecting all monitoring information unless this has been deemed necessary by the business needs. Collecting all monitoring information will require more resources to gather and process the data, and it is highly likely that your business only requires a subset of that data. It is also

very likely that you will not re-visit the configuration and disable collection of those metrics which are not useful in your environment.

Every case is unique because the applications, database servers, and environments deployed by businesses are unique. Even the same application deployed by two businesses can have widely differing uses and therefore widely differing monitoring needs.

Optim Performance Manager supports these best practices by providing fine grained control over what monitoring information is collected and when it is collected – thus allowing each business to collect the appropriate level of monitoring information.

9 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 grant 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.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment do so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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: © Copyright IBM Corporation 2011. All Rights Reserved.

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.

9.1 Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.