Monitoring DB2 Servers

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

DB2 Universal Database (UDB) Enterprise Server Edition (ESE) is a multiuser version of DB2 that allows you to create and manage single- partitioned or partitioned database environments. Partitioned database systems can manage high volumes of data and provide benefits such as increased performance and high availability.

Before attempting to understand how to monitor a DB2 UDB server, it is essential to get acquainted with its architecture.

On the client side, either local or remote applications, or both, are linked with the DB2 Universal Database™ client library. Local clients communicate using shared memory and semaphores; remote clients use a protocol such as Named Pipes (NPIPE), TCP/IP, NetBIOS, or SNA.

On the server side, activity is controlled by engine dispatchable units (EDUs). EDUs are implemented as threads in a single process on Windows-based platforms and as processes on UNIX. DB2 agents are the most common type of EDUs. These agents perform most of the SQL processing on behalf of applications. Prefetchers and page cleaners are other common EDUs.

- Prefetchers retrieve data from disk and move it into the buffer pool before applications need the data. For example, applications needing to scan through large volumes of data would have to wait for data to be moved from disk into the buffer pool if there were no data prefetchers. Agents of the application send asynchronous read- ahead requests to a common prefetch queue. As prefetchers become available, they implement those requests by using big-block or scatter-read input operations to bring the requested pages from disk to the buffer pool. If you have multiple disks for storage of the database data, the data can be striped across the disks. Striping data lets the prefetchers use multiple disks at the same time to retrieve data.
- Page cleaners move data from the buffer pool back out to disk. Page cleaners are background EDUs that are independent of the application agents. They look for pages from the buffer pool that are no longer needed and write the pages to disk. Page cleaners ensure that there is room in the buffer pool for the pages being retrieved by the prefetchers.
- A set of subagents might be assigned to process the client application requests. Multiple subagents can be assigned if the machine where the server resides has multiple processors or is part of a partitioned database. For example, in a symmetric multiprocessing (SMP) environment, multiple SMP subagents can exploit the many processors.

All agents and subagents are managed using a pooling algorithm that minimizes the creation and destruction of EDUs.

Buffer pools are areas of database server memory where database pages of user table data, index data, and catalog data are temporarily moved and can be modified. Buffer pools are a key determinant of database performance because data can be accessed much faster from memory than from disk. If more of the data needed by applications is present in a buffer pool, less time is required to access the data than to find it on disk.

The configuration of the buffer pools, as well as prefetcher and page cleaner EDUs, controls how quickly data can be accessed and how readily available it is to applications. Without the independent prefetchers and the page cleaner EDUs, the application agents would have to do all of the reading and writing of data between the buffer pool and disk storage.

All these integral components need to function like well-oiled machines in order to enable the DB2 server to quickly and efficiently process data requests. The slightest of problems in the configuration or operation of these components, if not swiftly resolved, can severely degrade database performance, and can even render the database server inaccessible to users. In order to prevent such adversities, it is imperative that the DB2 server is monitored round-the-clock, and problems brought to the attention of the administrators before irreparable damage is caused.

Moreover, a Database Partitioning Feature (DPF) is additionally available in the DB2 UDB Enterprise Server Edition (ESE). With DPF your database is scalable as you can add new machines and spread your database across them. This means more CPUs, more memory and more disks from each of the additional machines for your database.

If DB2 is installed in a DPF environment, the management challenges grow! Instead of monitoring a database on a single machine, each database now has to be monitored across all its partitons, which may be spread across multiple machines, to ascertain the health of the transactions to the database.

To address the unique monitoring requirements of the different DB2 installations, the eG Enterprise suite prescribes three exclusive monitoring models for the IBM DB2 server – one for version 8.0 (and above) of the DB2 server, one for versions 6.0 and 7.0, and one for the DB2 server that is installed in a DPF environment. These models thoroughly scrutinize every layer of a DB2 server for issues, and proactively alert administrators of probable problem conditions.

The sections to come discuss each of these models in great detail.

Chapter 2: How to Configure and Monitor DB2 UDB Server – Version 8.0 and above Using eG Enterprise?

eG Enterprise can monitor the DB2 UDB server in an agent-based or an agentless manner. In case of the agentless approach, the remote agent used to monitor the DB2 UDB server should be deployed on a remote Windows host in the environment.

The broad steps for monitoring DB2 UDB using eG Enterprise are as follows:

- Managing the DB2 UDB server (V8.0 and above)
- Configuring the tests

2.1 Managing the DB2 UDB server (V8.0 and above)

The DB2 UDB server (V8.0 and above) cannot be automatically discovered by eG Enterprise. This implies that you will have to manually add the server into the eG Enterprise system to manage it. Follow the steps below to achieve the same:

These steps have been discussed in this topic.

1. Log into the eG administrative interface.
2. eG Enterprise cannot automatically discover the DB2 UDB server. You need to manually add the eDirectory netware using the **COMPONENTS** page (see Figure 2.1) that appears when the Infrastructure -> Components -> Add/Modify menu sequence is followed. Remember that components manually added are managed automatically.

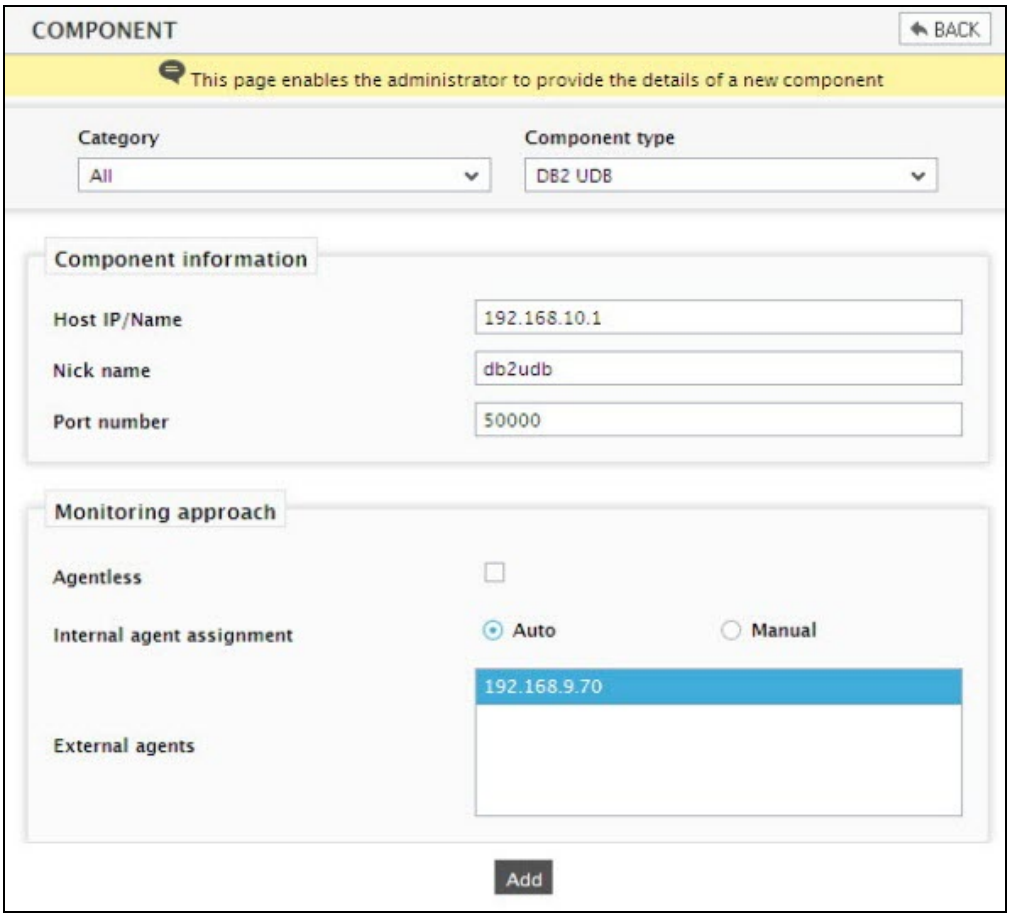

Figure 2.1: Adding the DB2UDB server

3. Specify the **Host IP/Name** and **Nick name** of the DB2 UDB server that you want to manage.
4. The **Port number** will be set as 50000 by default. If the DB2 UDB server is listening on a different port in your environment, then override this default setting.
5. Finally, click the **Add** button to add the DB2 UDB server for monitoring.

2.2 Configuring the tests

1. When you attempt to sign out of the eG administrative interface, a list of unconfigured tests appears (see Figure 2.2).

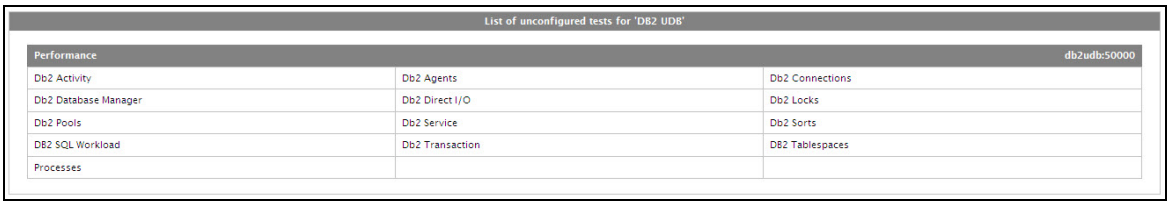

Figure 2.2: List of unconfigured tests for the DB2 UDB server

2. Click on the **Db2 Activity** test in Figure 2.2 to configure it. To know how to configure the test, refer to Section **3.3.1**.
3. Now try to signout, it will prompt to configure the **Db2 Service** test. To know details on configuring this test, refer to Section **3.3.3**.
4. Finally, click the **Update** button to register the changes, and then try signing out of the admin interface. This time it will prompt to configure the **Processes** test for the DB2 UDB server. Refer to *Monitoring Apache Web servers* for details on configuring the **Processes** test and specifying the **PROCESSPATTERN**.

Chapter 3: Monitoring IBM DB2 Server Version 8.0 (and above)

Figure 3.1 depicts the specialized *DB2 UDB* monitoring model that eG Enterprise offers for monitoring an IBM DB2 server ver. 8.0 (or above).

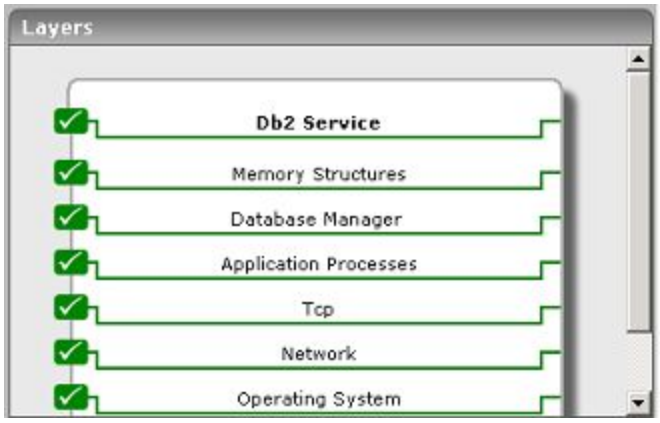

Figure 3.1: The layer model of an IBM DB2 server version 8.0 (or above)

Every layer of Figure 3.1 is mapped to a wide variety of tests that a single eG agent executes on the DB2 server; these tests extract a wealth of performance metrics from the DB2 server.

One of the key qualities of this eG agent is that it is capable of measuring the internal health of the DB2 server from anywhere in the environment! In other words, the eG agent need not be present on the DB2 server to monitor its internal operations. The eG Enterprise system supports “agentless monitoring” of a DB2 server, by means of which, the agent can be installed on any remote host in your environment, and can be easily configured to pull out statistics of interest from within the DB2 server.

The statistics so collected enable administrators to find quick and accurate answers to the following performance queries:

- Is the DB2 database server available? How quickly does it respond to user requests?
- What is the current connection load on the DB2 database manager? How many of these connections are local, and how many are remote connections?
- How quickly does the DB2 server process requests from client applications? Are there sufficient agents in the agent pool to service all the client requests?
- Are the agents in the pool utilized optimally, or are too many agents idle?

- Does the database server perform sorting efficiently? Has adequate sort heap space been allocated to the database manager to enable this?
- Are sort overflows kept at a minimum?
- Does sorting take too long?
- Are lock escalations occurring too frequently on the database?
- Are too many deadlocks been detected?
- Do applications obtain locks quickly, or do they have to wait too long for locks?
- Are the database buffer pools adequately sized?
- Are the page cleaners and prefetchers been utilized effectively?
- Are too many rollbacks happening on the database?
- Have too many SQL statements failed?

The sections to come elaborate on each layer of Figure 3.1, the tests associated with them, and the statistics they extract.

3.1 The Database Manager Layer

Using the tests associated with this layer, the following can be monitored:

- Critical activities performed by the database manager
- Client connections to the database manager
- Usage of the agent pools on the database manager

Figure 3.2: The tests associated with the Database Manager layer

3.1.1 Db2 Agents Test

An agent is a process or thread that carries out the requests made by a client application. Each connected application is served by exactly 1 coordinator agent and possibly, a set of subordinator agents or subagents. Subagents are used for parallel SQL processing in partitioned databases and on SMP machines.

For partitioned database environments and environments with intra-partition parallelism enabled, each partition (that is, each database server or node) has its own pool of agents from which subagents are drawn. Because of this pool, subagents do not have to be created and destroyed each time one is needed or has finished its work. The subagents can remain as associated agents in the pool and be used by the database manager for new requests from the application they are associated with.

The **Db2 Agents** test monitors how effectively the agent pool has been utilized.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total agents:	Indicates the total number of agents currently registered in the database manager instance that is being monitored (Sum of coordinator agents and subagents).	Number	This measure can be used to evaluate the setting for MAXAGENTS configuration parameter.
Idle agents :	Indicates the percentage of agents in the agent pool that is currently unassigned to an application and are, therefore, “idle”.	Number	Having idle agents available to service requests for agents can improve performance. So you can use this measure to help set the NUM_ POOLAGENTS configuration parameter.
Agents waiting on token:	Indicates the percentage of agents waiting for a token so they can execute a transaction in the database manager.	Percent	You can use this element to help evaluate your setting for the MAXCAGENTS configuration parameter. Each application has a dedicated coordinator agent to process database requests within the database manager. Each agent has to get a token before it can execute a transaction. The maximum number of agents that can execute database manager transactions is limited by the configuration parameter MAXCAGENTS.
Agents creation ratio:	Indicates the ratio of number of agents assigned directly from agent pool to the total number of agents used	Percent	A high percentage indicates the effectiveness of the agent pool. A consistent low value indicates that the number of agents in the agent pool are not adequate to service

Measurement	Description	Measurement Unit	Interpretation
	to service requests.		requests. You might want to consider increasing the NUM_ POOLAGENTS setting in this case.
Stolen Agents:	Indicates the number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.	Number	If this value is high, consider increasing the NUM_ POOLAGENTS configuration parameter.

3.1.1.1 Creating a Special User for Monitoring DB2

All the tests that the eG agent runs on a DB2 UDB server (version 8.0 and above) or on a DB2 DPF server, should be configured with the credentials of a user who has any of the following privileges to the monitored **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON.

It is recommended that you create a special user for this purpose. The steps for creating such a user are detailed below:

1. Create a user group - say, eg_mon_grp - on the operating system hosting the DB2 server.
2. Create a user - say, eg_user - at the OS-level, and add this new user to the group created in step 1.
3. Next, you need to grant the SYSADM or SYSCTRL or SYSMAINT or SYSMON privilege to the group created in step 1. For instance, to grant the SYSMON privilege to the eg_mon_grp in our example, do the following:
   - Go to the DB2 prompt.
   - Issue the following command at the prompt:

     UPDATE DBM CFG USING SYSMON_GROUP eg_mon_grp;
   - Finally, restart the DB2 instance by issuing the following commands, one after another:

db2stop force

db2start

3.1.2 Db2 Connections Test

The Db2 Connections test reports key statistics pertaining to the local and remote connections to the DB2 database manager.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1** topic.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total connections:	Indicates the total number of local and remote connections that are currently present in	Number	

Measurement	Description	Measurement Unit	Interpretation
	the database manager.		
Local connections:	Indicates the number of local applications that are currently connected to a database within the database manager instance being monitored.	Number	This number can help you determine the level of concurrent processing occurring in the database manager. This number only includes applications that were initiated from the same instance as the database manager. The applications are connected, but may or may not be executing a unit of work in the database. When used in conjunction with the Remote connections measurement, this measure can help you adjust the setting of the MAXAGENTS configuration parameter.
Remote connections:	Indicates the percentage of agents waiting for a token so they can execute a transaction in the database manager.	Percent	This number does not include applications that were initiated from the same instance as the database manager. When used in conjunction with the Local connections measure, this measure can help you adjust the setting of the MAX_ COORDAGENTS configuration parameter.
Local connections in exec:	Indicates the number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.	Number	This number can help you determine the level of concurrent processing occurring in the database manager. This number only includes applications that were initiated from the same instance as the database manager. When used in conjunction with the Remote connections in exec measure, this measure can help you adjust the

Measurement	Description	Measurement Unit	Interpretation
			setting of the MAXCAGENTS configuration parameter.
Remote connections in exec:	Indicates the number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.	Number	This number can help you determine the level of concurrent processing occurring on the database manager. This number does not include applications that were initiated from the same instance as the database manager. When used in conjunction with the Local connections in exec measure, this metric can help you adjust the setting of the MAXCAGENTS configuration parameter.

3.1.3 Db2 Database Manager Test

The database manager includes the database engine and the facilities to access data, such as the command line processor and the application interfaces. This test reports key statistics pertaining to the health of the DB2 database manager.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid

privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.

5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Sorts post threshold:	Indicates the number of sorts per second that have requested heaps after the sort heap threshold has been exceeded.	Sorts/Sec	Under normal conditions, the database manager will allocate sort heap using the value specified by the SORTHEAP configuration parameter. If the amount of memory allocated to sort heaps exceeds the sort heap threshold (SHEAPTHRES configuration parameter), the database manager will allocate sort heap using a value less than that specified by the SORTHEAP configuration parameter. Each active sort on the system allocates memory, which may result in sorting taking up too much of the system memory available. Sorts that start after the sort heap threshold has been reached may not receive an optimum amount of memory to execute, but, as a result, the entire system may benefit. By modifying the sort heap threshold and sort heap size configuration parameters, sort operation performance and overall system performance can be improved. If this element's value is

Measurement	Description	Measurement Unit	Interpretation
			high, you can: a. Increase the sort heap threshold (SHEAPTHRES), or, b. Adjust applications to use fewer or smaller sorts via SQL query changes.
Piped Sorts Requested:	A sort is classified as piped sort if the sorted information can return directly without requiring a temporary table to store a final, sorted list of data. This measure reports the number of piped sorts that have been requested per second..	Sorts/Sec	Piped sorts may reduce disk I/O. Allowing more piped sorts therefore, can improve the performance of sort operations and possibly the performance of the overall system.
Piped Sorts Rejected:	Indicates the percentage of piped sort requests that have been rejected.	Percent	When the number of rejected piped sorts are high, you can improve sort performance by adjusting one or both of the following configuration parameters: a. SORTHEAP b. SHEAPTHRES If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold, then there is the

Measurement	Description	Measurement Unit	Interpretation
			possibility that more memory will remain allocated for sorting. This could cause the paging of memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.
Hash Join Post Threshold:	Indicates the total number of times that a hash join heap request was limited due to concurrent use of shared or private sort heap space.	Hits/Sec	If this value is large, the sort heap threshold should be increased.

3.2 The Memory Structures Layer

The tests mapped to the **Memory Structures** layer (see Figure 3.3), report critical statistics that reveal:

- How efficiently the locking and sorting activities occur on the monitored DB2 database
- How well the buffer pools are managed
- The level of I/O activity on the DB2 database

Figure 3.3: The tests associated with the Memory Structures layer

3.2.1 Db2 Locks Test

Typically, locking activity is governed by the following factors:

- Concurrency and granularity
- Lock compatibility
- Lock conversion
- Lock escalation
- Lock waits and timeouts
- Deadlocks

In the event of an application slowdown, the measures reported by the **Db2 Locks** test enable administrators to accurately determine whether/not any of the above-mentioned factors have adversely impacted application performance, and if so, to what extent.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Locks Held:	Indicates the total number of locks currently held by all applications in the database.	Number	
Locks Escalated:	Indicates the number of times every second that locks have been escalated from several row locks to a table lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. The amount of lock list space available is determined by the maxlocks and locklist configuration parameters. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it will then use space in the lock list	Escalations/Sec	There are several possible causes for excessive lock escalations: a. The lock list size (locklist) may be too small for the number of concurrent applications b. The percent of the lock list usable by each application (maxlocks) may be too small c. One or more applications may be using an excessive number of locks. d. To resolve these problems, you may be able to: e. Increase the locklist configuration parameter value. f. Increase the maxlocks configuration parameter value. Identify the applications with large numbers of locks or those that are holding too much of the lock list. These applications can also cause lock escalations in other applications by using too large a portion of the lock list. These applications may need to resort to using table locks instead of row

Measurement	Description	Measurement Unit	Interpretation
	allocated for other applications. When the entire lock list is full, an error occurs. This data item includes a count of all lock escalations, including exclusive lock escalations.		locks, although table locks may cause an increase in lock_waits and lock_wait_time.
Exclusive Lock Escalations:	Indicates the number of times per second that locks have been escalated from several row locks to one exclusive table lock, or the number of times (per second) an exclusive lock on a row caused the table lock to become an exclusive lock.	Escalations/Sec	Other applications cannot access data held by an exclusive lock; therefore it is important to track exclusive locks since they can impact the concurrency of your data. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the locklist and maxlocks configuration parameters. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it will then use space in the lock list allocated for other applications. When the entire lock list is full, an error occurs. See Lock escalations for possible causes and resolutions to excessive exclusive lock escalations. An application may be using exclusive locks when share locks are sufficient. Although share

Measurement	Description	Measurement Unit	Interpretation
			locks may not reduce the total number of lock escalations share lock escalations may be preferable to exclusive lock escalations.
Locks Timedout:	Indicates the number of times that a request to lock an object timed-out instead of being granted.	Timeouts/Sec	This measurement can help you adjust the setting for the locktimeout database configuration parameter. If the number of lock time-outs becomes excessive when compared to normal operating levels, you may have an application that is holding locks for long durations. In this case, this element may indicate that you should analyze some of the other lock and deadlock monitor elements to determine if you have an application problem. You could also have too few lock time- outs if your locktimeout database configuration parameter is set too high. In this case, your applications may wait excessively to obtain a lock.
Lock Waits:	Indicates the total number of times per second that applications or connections waited for locks.	Waits/Sec	If the value is consistently high, find t the applications or connections causing lock waits and fine tune the appropriate SQL queries.
Average Lock Wait Time:	Indicates the average time that all the applications were waiting for a lock.	Secs	If the average lock wait time is high, you should look for applications that hold many locks, or have lock escalations, with a

Measurement	Description	Measurement Unit	Interpretation
			focus on tuning your applications to improve concurrency, if appropriate.
Percent of Application in Lock Wait:	Indicates the percentage of applications waiting for the release of lock.	Percent	If this value is high, the applications may have concurrency problems, and you should identify applications that are holding locks or exclusive locks for long periods of time.
Deadlocks:	Indicates the total number of deadlocks that have been detected per second.	Deadlocks/Sec	This element can indicate that applications are experiencing contention problems. These problems could be caused by the following situations: a. Lock escalations are occurring for the database b. An application may be locking tables explicitly when system-generated row locks may be sufficient. c. An application may be using an inappropriate isolation level when binding d. Catalog tables are locked for repeatable read e. Applications are getting the same locks in different orders, resulting in deadlock You may be able to resolve the problem by determining in which applications (or application processes) the deadlocks are

Measurement	Description	Measurement Unit	Interpretation
			occurring. You may then be able to modify the application to enable it to execute concurrently. Some applications, however, may not be capable of running concurrently.

3.2.2 Db2 Pools Test

A buffer pool is an area of memory into which database pages are read, modified, and held during processing.

Buffer pools improve database performance. If a needed page of data is already in the buffer pool, that page is accessed faster than if that page had to be read directly from disk. The database manager has agents whose tasks are to retrieve data pages from disk and place them in the buffer pool (prefetchers), and to write modified data pages from the buffer pool back to disk (page cleaners).

The reading and writing of data pages to and from disk is called disk input/output (I/O). Avoiding the wait associated with disk I/O is the primary way to improve the performance of the database. How you create the buffer pool, and configure the database manager and the agents associated with the buffer pool, controls the performance of the database. Through SQL and configuration parameters, you can control the size of the buffer pool, the number of prefetchers and page cleaners that move data pages into and out of the buffer pool, the size of the data pages, and the number of data pages that can be moved at one time.

The statistics reported by the Db2 Pools test help administrators analyze the usage of the buffer pools, and provides them with useful pointers to fine-tune the configuration of the buffer pools.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed

2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Buffer pool hit ratio:	Indicates the percentage of the requested pages that is readily available in the memory without doing disk I/O.	Percent	This measure is an overall indicator of how well the buffer pool is performing. Lower values indicate that more physical I/O is happening than logical. Since physical I/O costs more, maintaining higher buffer hit ratio is desired. Typically a hit ratio over 90% shows that the buffer pool is performing well. If this measure is consistently low, increase the size of the buffer pool by increasing BUFFPAGE configuration value.
Buffer Pool Hit Ration (Data):	Indicates the the percentage of the requested data pages that is readily available in the memory without doing disk I/O.	Percent	This measure is an indicator of how well the buffer pool is performing for the data page requests. Lower values indicate that more physical I/O is happening than logical. Since physical I/O costs more, maintaining higher buffer hit ratio is desired. Typically a hit ratio over 90% shows

Measurement	Description	Measurement Unit	Interpretation
			that the buffer pool is performing well. If this measure is consistently low, increase the size of the buffer pool by increasing BUFFPAGE configuration value.
Buffer Pool Hit Ratio (Index):	Indicates the percentage of the requested index pages that is readily available in the memory without doing disk I/O.	Percent	This measure is an overall indicator of how well the buffer pool is performing. Lower values indicate that more physical I/O is happening than logical. Since physical I/O costs more, maintaining higher buffer hit ratio is desired. Typically a hit ratio over 90% shows that the buffer pool is performing well. If this ratio is really low and the overall ratio is relatively high, then break the index (s) out into their own tablespace/buffer pool.
Pre Fetch Ratio:	The ratio of asynchronous reads to synchronous reads. The value indicates how effectively DB2 database manager is populating the buffer pools through the use of prefetchers.	Percent	High value indicates more asynchronous I/O is happening than synchronous. The value can be used to tune the num_ ioservers configuration parameter.
Percent Log Cleans:	Indicates the percentage of times a page cleaner was invoked because the logging space used had reached a predefined criterion for the database.	Percent	If this value is high (say > 40%), this could mean that page cleaners are constantly being utilized to clean the log and aren't available for other page cleaning activities, hampering performance. On the other hand, if the value is low, (say < 10%) then the page cleaners aren't being

Measurement	Description	Measurement Unit	Interpretation
			triggered as often for this activity. this means that they would be more available for the other types of page cleaning activities, which is great for buffer pool performance.
Percent Dirty Page Cleans:	Indicates the percentage of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.	Percent	The threshold is set by the chngpgs_ thresh configuration parameter. It is a percentage applied to the buffer pool size. When the number of dirty pages in the pool exceeds this value, the cleaners are triggered. If this value is set too low, pages might be written out too early, requiring them to be read back in. If set too high, then too many pages may accumulate, requiring users to write out pages synchronously.
Percent Victim Cleans:	Indicates the percentage of times the page cleaner (s) were triggered to oust a victim page from the buffer pool. A victim page is a clean or dirty page in the buffer pool that is removed simply because DB2 needs to make room for incoming pages. If a victim page is a dirty page then the information must be written out to disk. Any page that is removed will most likely cause	Percent	If the ratio is higher then the above two then that is typically a good indicator that the buffer pool needs to be larger since there never seems to be enough room for new pages to be brought in. This could also be a sign that dirty pages are staying in the buffer pool too long which could mean that the changed pages threshold (CHNGPGS_ THRESH) is set too high. Even the SOFTMAX parameter could be set too high and too much of the changed pages that are logged are not getting flushed out to make way for new pages. If this ratio is low, it may indicate that you have defined

Measurement	Description	Measurement Unit	Interpretation
	more physical I/O to occur in order to retrieve it again at later time when DB2 is ready to use it.		too many page cleaners. If your chngpgs_thresh is set too low, you may be writing out pages that you will dirty later. Aggressive cleaning defeats one purpose of the buffer pool, that is to defer writing to the last possible moment.
Catalog Cache Hit Ratio:	Indicates the percentage of time the requested information for table descriptor or authorization was readily availble in catalog cache without requiring to perform disk I/O.	Percent	The catalog cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. If the ratio is greater than 80%, then the catalog cache is performing well. A smaller value indicates that the catalog cache size should be increased by tuning the parameter CATALOGCACHE_ SZ in the database configuration. The value may be low immediately following the first connection to the database. The execution of Data Definition Language (DDL) SQL statements involving a table,view, or alias will evict the table descriptor information for that object from the catalog cache causing it to be re-inserted on the next reference. Therefore, the heavy use of DDLs may also increase the value of the measure.
Package Cache Hit Ratio:	The package and section information required for the execution of dynamic and static SQL statements are placed in	Percent	If the hit ratio is high (more than 80%), the cache is performing well. A smaller ratio may indicate that the package cache size (pckcachesz) should be increased.

Measurement	Description	Measurement Unit	Interpretation
	the package cache as required. This information is required whenever a dynamic or static statement is being executed. The ratio indicates the effectiveness of package cache hit ratio.		

3.2.3 Db2 Sorting Test

Sorting represents organizing the rows in a table into the order of one or more of its columns, optionally eliminating duplicate entries. Sorting is required when:

- No index exists to satisfy a requested ordering (for example a SELECT statement that uses the ORDER BY clause).
- An index exists but sorting would be more efficient than using the index
- An index is created.
- An index is dropped, which causes index page numbers to be sorted.

Because queries often require sorted or grouped results, sorting is often required, and the proper configuration of the sort heap areas is crucial to good query performance. Using the Db2Sort test, administrators can figure out whether/not the sort heap allocations are sufficient to facilitate efficient sorting.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server

3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test:

Measurement	Description	Measurement Unit	Interpretation
Sort Heap Allocated:	Indicates the sum of sort heap space allocated for all sorts in all active databases in the database manager.	Pages	Normal memory estimates do not include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required. You may use the information returned at the database manager level to help you tune the SHEAPTHRES configuration parameter. If the element value is greater than or equal to SHEAPTHRES, it means that the sorts are not getting the full sort heap as defined by the SORTHEAP parameter.
Average Sort	Indicates the average	Pages	If the SORTHEAP configuration

Measurement	Description	Measurement Unit	Interpretation
Heap Space Used:	sort heap space used by each sort.		parameter is substantially larger than the average sort heap used, you may be able to lower the value of this parameter.
Sort Rate:	Indicates the number of sort operations performed during the last measurement period.	Sorts / Sec	
Percent Sort Overflow:	Indicates the percentage of sorts that had to overflow to disk.	Percent	Sort overflows are sorts that ran out of sort heap and may have required disk space for temporary storage. When a sort overflows, additional overhead will be incurred because the sort will require a merge phase and can potentially require more I/O, if data needs to be written to disk. If this percentage is high, you may want to adjust the database configuration by increasing the value of sortheap.
Average Sort Time:	Indicates the average sort time for all sorts performed by all applications connected to a particular database.	Secs	A high value indicates the poor performance of sorting operations. Identify the statements that spend lot of time sorting. You may want to reduce the average sort time for these statements by increasing the sortheap parameter.

3.2.4 Db2 Direct I/O Test

This test monitors the I/O activity on the currently active DB2 database.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Direct Read Rate:	Indicates the total number of direct reads by the application per second. In other words, it is the number of read operations that do not use the buffer pool.	Reads/Sec	Direct reads are performed in units, the smallest being a 512- byte sector. They are used when: • Reading LONG VARCHAR columns • Reading LOB (large object) columns • Performing a backup A high value over a period of time may be indicative of a performance bottleneck.

Measurement	Description	Measurement Unit	Interpretation
Direct Write Rate:	Indicates the total number of direct writes by the application per second. In other words, it is the number of write operations that do not use the buffer pool.	Writes/Sec	Direct writes are performed in units, the smallest being a 512-byte sector. They are used when: • Writing LONG VARCHAR columns • Writing LOB (large object) columns • Performing a restore • Performing a load A high value over a period of time may be indicative of a performance bottleneck.
Buffer Pool IO Rate:	Indicates the rate at which the buffer pool I/O operations are being done in the database.	IOPS	In conjunction with the hit ratio statistics, and the characteristics of the applications executing, the I/O load may require adjustment of BUFFPAGE, or applications may require further tuning.

3.3 The Db2 Service Layer

Besides revealing the availability and responsiveness of the DB2 server, the tests associated with the **Db2 Service** layer indicate the level of SQL activity on the server, and the number and type of transactions that occur on the server. These measurements together serve as effective indicators of the processing ability of the DB2 server.

Figure 3.4: The tests associated with the Db2 Service layer

3.3.1 Db2 Activity Test

This test measures the level of SQL activity on the DB2 database server, and reveals how well the server processes SQL queries.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.

7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
Dynamic SQL Rate:	Indicates the rate of dynamic SQL statements that were attempted.	Stmts/Sec	This is an indication of throughput of the system during the monitoring period. A high value of dynamic SQLs and low value of failed SQLs indicate good throughput.
Select SQL Rate:	Indicates the rate at which SELECT SQL statements were executed during the last measurement period.	Stmts/Sec	This measure can be used to determine the level of database activity.
UID SQL Rate:	Indicates the rate at which update/delete/insert statements were issued during the last measurement period.	Stmts/Sec	This measure can be used to determine the level of database activity.
Failed SQL Rate:	Indicates the rate at which SQL statements failed.	Stmts/Sec	A relatively high value indicates a problem. Failed SQL statements waste system resources. Hence, the value of this measure should be very low.
Percent Failed SQL:	Indicates the percentage of SQL statements that failed during the interval. This value includes all SQL statements that received a negative SQLCODE	Percent	
Percent DDL SQL:	Denotes the percentage of SQL statements that were DDL(Data Definition Language) during the last measurement period.	Percent	This value should normally be low.

Measurement	Description	Measurement Unit	Interpretation
Percent UID SQL:	Indicates the percentage of update/insert/delete statements executed during the last measurement period.	Percent	This measure can be used to determine the level of database activity.

3.3.2 Db2 Transaction Test

This test tracks various statistics pertaining to the transactions executing on a DB2 database.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Commit Rate:	Indicates the transaction throughput. This measure is the sum of the committed statements attempted and internal commits (total number of commits initiated internally by the database manager) per second.	Commits/Sec	A decrease in this measure during the monitoring period may indicate that the applications are not doing frequent commits. This may lead to problems with logging and data concurrency. The cause has to be probed in the application.
Rollback Rate:	Indicates the rate of unit of work rollbacks.	Rollbacks/Sec	A high rollback rate is an indicator of bad performance, since work performed up to the rollback point is wasted. The cause of the rollbacks has to be probed in the application.
Transaction Rate:	Indicates the rate of commits and rollbacks for the application using the DB2 Connect gateway.	Trans/Sec	A high transaction rate with high rollback rate indicates bad performance.

3.3.3 Db2 Service Test

This test monitors the availability and response time of each database on the target DB2 UDB server.

Target of the test : A DB2 database server

Agent deploying the test : An external agent; if you are running this test using the external agent on the eG manager box, then make sure that this external agent is able to communicate with the port on which the target DB2 server is listening. Alternatively, you can deploy the external agent that will be running this test on a host that can access the port on which the target DB2 server is listening.

Outputs of the test : One set of results for each database instance on the target DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test
8. **QUERY** - The test emulates a user executing a query on the specified DATABASE, and thus determines the availability and responsiveness of the database server. In the **QUERY** text box, specify the select query to execute.
9. **ADDITIONAL DATABASES** - By default, this parameter is set to *none*. This implies that the test will report the availability and responsiveness of the database specified against the DATABASE parameter. If you wish to monitor the availability and responsiveness of other databases in the target DB2 server, then, you can specify those databases as a comma-separated list against this parameter. This will ensure that the availability and responsiveness of the specified databases will be reported by this test apart from the database specified against the DATABASE parameter. The databases specified will be the descriptors of this test in this case.

Measurements made by the test:

Measurement	Description	Measurement Unit	Interpretation
Availability	Indicates the availability of this database.	Percent	The availability is 100% when the database is responding to a request and 0% when it is not. Availability

Measurement	Description	Measurement Unit	Interpretation
			problems may be caused by a misconfiguration/malfunctioning of the database server, or if the database has not been started.
Response Time	Indicates the time taken by this database to respond to a user query.	Secs	A sudden increase in response time is indicative of a bottleneck at the database server.

3.3.4 DB2 Tablespaces Test

This test auto-discovers the tablespaces on an IBM DB2 server, and monitors the space usage of each tablespace.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each tablespace on the DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

8. **QUERY** - The test emulates a user executing a query on the specified DATABASE, and thus determines the availability and responsiveness of the database server. In the QUERY text box, specify the select query to execute.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Tablespace type:	Indicates the tablespace type.		If the target tablespace is a System-managed tablespace, then this measure will report the value System Managed Storage. On the other hand, if the tablespace is a database managed tablespace, then the value of this measure will be Database Managed Storage. The SMS (System Managed Space) tablespaces allow the operating system to allocate and manage the space where the table data resides. Once the initial create has been completed, you cannot add or delete containers to an SMS tablespace. The data in the table spaces is striped by extent across all the containers in the system. An extent is a group of consecutive pages defined to the database. The file extension denotes the type of the data stored in the file. To distribute the data evenly across all containers in the table space, the starting extents for tables are placed in round- robin fashion across all containers. Such distribution of extents is particularly important if the database contains many small tables.

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>In a DMS (Database Managed Space) table space, the database manager controls the storage space. The storage model consists of a limited number of devices or files whose space is managed by DB2 Database for Linux, UNIX, and Windows. The database administrator decides which devices and files to use, and DB2 manages the space on those devices and files. The table space is essentially an implementation of a special purpose file system designed to best meet the needs of the database manager.

DMS table spaces are different from SMS table spaces in that space for DMS table spaces is allocated when the table space is created. For SMS table spaces, space is allocated as needed - i.e., on demand.

The numeric values that correspond to the tablespace types reported by this measure are as follows:

<table><tr><th>Numeric Value</th><th>Tablespace Type</th></tr><tr><td>0</td><td>Database Managed Storage</td></tr><tr><td>1</td><td>System Managed Storage</td></tr></table>
Note:

By default, this measure reports the</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			Types displayed in the table above as its value. In the graph of the measure however, the types are represented using their corresponding numeric equivalents - i.e., 0 and 1.
Usable space:	Indicates the amount of space allocated to this tablespace.	MB	This measure is available only for DMS tablespaces.
Used space:	Indicates the space used in this tablespace.	MB	Ideally, the value of this measure should be low. A very high value or a consistent increase in this value could indicate a potential contention for space.
Free space:	Indicates the free space in this tablespace.	MB	This measure is available only for DMS tablespaces. This is because, for SMS tablespaces, space is allocated on demand, and deallocated when not required; this implies that SMS tablespaces will at no time have any unused space. For DMS tablespaces, a high value is desired for this measure. A very low value or a gradual decrease in this value could be a cause for concern, as it indicates a slow, but steady space erosion.
Availability:	Indicates the percentage of free space in this tablespace.	Percent	As free space value is applicable only for DMS tablespaces, this will be available only for DMS tablespaces. The test will not report this measure for SMS tablepsaces. For DMS tablespaces, a high value

Measurement	Description	Measurement Unit	Interpretation
			is desired for this measure. A very low value or a gradual decrease in this value could be a cause for concern, as it indicates a slow, but steady space erosion.

3.3.5 DB2 SQL Workload Test

Nothing can degrade the performance of the server like a resource-hungry or a long-running query! When such queries execute on the server, they either hog almost all the available CPU, memory, and disk resources or keep the resources locked for long time periods, thus leaving little to no resources for carrying out other critical database operations. This can significantly slowdown the database server and adversely impact user experience with the server. To ensure peak performance of the DB2 UDB server at all times, such queries should be rapidly identified and quickly optimized to minimize resource usage. This is where the **DB2 SQL Workload** test helps.

At configured intervals, this test compares the usage levels and execution times of all queries that started running on the server in the last measurement period and identifies a 'top query' in each of the following categories - CPU usage, memory usage, disk activity, and execution time. The test then reports the resource usage and execution time of the top queries and promptly alerts administrators if any query consumes more resources or takes more time to execute than it should. In such a scenario, administrators can use the detailed diagnosis of this test to view the inefficient queries and proceed to optimize them to enhance server performance.

Target of the test : A DB2 UDB server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the DB2 UDB server monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test.
8. **QUERY** - The test emulates a user executing a query on the specified **DATABASE**, and thus determines the availability and responsiveness of the database server. In the **QUERY**text box, specify the select query to execute.
9. To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

   The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

   - The eG manager license should allow the detailed diagnosis capability
   - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum physical read rate	Indicates the number of physical disk reads performed by the top query per execution.	Seconds/read	If the value of this measure is abnormally high, you can use the detailed diagnosis of this measure to view the top- 5 (by default) queries generating maximum physical disk activity. From this, you can identify the top query in terms of number of physical disk reads. You may then want to

Measurement	Description	Measurement Unit	Interpretation
			optimize the query to reduce the disk reads.
Maximum physical write rate	Indicates the number of memory buffers used by the top query per execution.	Seconds/write	If the value of this measure is abnormally high, you can use the detailed diagnosis of this measure to view the top-5 (by default) queries consuming memory excessively. From this, you can easily pick that query which is consuming the maximum memory. You may then want to optimize the query to minimize memory usage.
Maximum user CPU time	Indicates the CPU time used for user level processing upon execution of the top query.	Seconds	If the value of this measure is over 30 seconds, you can use the detailed diagnosis of this measure to the top-5 (by default) queries hogging the CPU resources. From this, you can easily pick that query which is consuming the maximum CPU. You may then want to optimize the query to minimize CPU usage.
Maximum elapsed time	Indicates the running time of each execution of the top query.	Seconds	If the value of this measure crosses 10 seconds, you can use the detailed diagnosis of this measure to view the top-5 (by default) queries that are taking too long to execute. . From this, you can easily pick that query with the maximum execution time. You may then want to optimize the query to minimize execution time.
Maximum system CPU time	Indicates the CPU time used for system level	Seconds	If the value of this measure is over 30 seconds, you can use the

Measurement	Description	Measurement Unit	Interpretation
	processing upon execution of the top query.		detailed diagnosis of this measure to the top- 5 (by default) users consuming the maximum system CPU time. From this, you can easily pick that query which is consuming the maximum CPU. You may then want to optimize the query to minimize CPU usage.

Chapter 4: How to Configure and Monitor DB2 UDB Server – Version 6/7.x Using eG Enterprise?

To assure backward compatibility with older, less-used DB2 versions, the eG retains the monitoring model for DB2 Server 6.0/7.X versions. To configure the DB2 UDB server 6/7.x versions to work with the eG Agent, a set of pre-requisites should be fulfilled. These requirements are explained in the following section.

4.1 Configuring a DB2 Server 6.0/7.X to work with the eG Agent

This section discusses about how to configure DB2 Server 6.0/7.X versions on Unix and Windows environments.

4.1.1 Configuring a DB2 Server on Unix Environments

1. To monitor a specific DB2 instance on Unix environments, login as the owner of the database instance and go to the **/opt/egurkha/bin** directory. From this directory, run the **start_egdb2mon** script. This script, checks the default monitors of the DB2 server that are required by eG agents for monitoring this database server.
2. If the DB2 default monitors are not started (by default, they will not be started), then the **start_egdb2mon** script generates the messages as shown in Figure 4.1

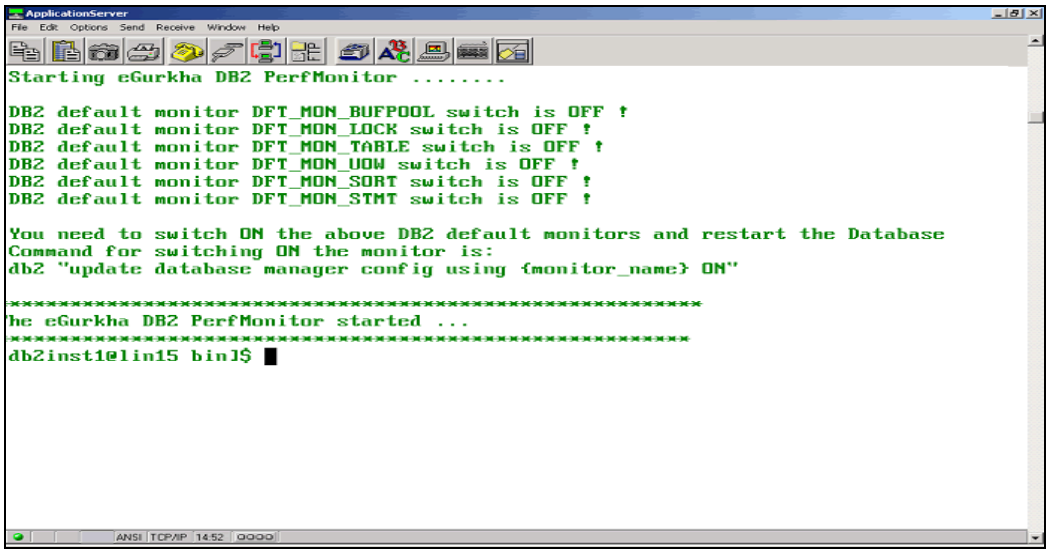

Figure 4.1: Messages indicating that the default monitors have not been turned on

3. Now, use the script **egdb2monswitch** from the directory **/opt/egurkha/bin**, to switch on all the necessary default monitors. This script can be run only by the DB2 instance owner.
4. Alternately, the same can be manually achieved by issuing the following commands at the command prompt:

```
[db2inst1@lin15 bin]$ db2 update database manager config using DFT_MON_BUFPOOL on
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
DB21025I  Client changes will not be effective until the next time the
application is started or the TERMINATE command has been issued.  Server
changes will not be effective until the next DB2START command.
[db2inst1@lin15 bin]$
```

5. Use the same method to start all the following default monitors.

   - DFT_MON_BUFPOOL
   - DFT_MON_LOCK
   - DFT_MON_TABLE
   - DFT_MON_UOW
   - DFT_MON_SORT
   - DFT_MON_STMT

   After all the default monitors have been started, remember to restart the database or else the changes will not take effect.

   To stop the DB2 instance, issue the command **db2stop**. To restart the instance, use the command **db2start** at the command prompt. These can normally be found under the **$INSTANCE/sqllib/adm** directory of the instance that is being monitored.

6. Sometimes, the DB2 database may not stop completely and instead, may send out a message stating that as some applications are already active, the DB2 instance could not be stopped. In that case, make sure that every required action is complete in that instance and then issue the following command to force the shutdown: **db2 force application all**
7. Now, try stopping the database and restart it. This will ensure that the DB2 Server has been completely configured to work with eG.

4.1.2 Configuring a DB2 Server on Windows 2000

Here again, create the eG DB2 PerfMonitor service

Once the service is created, follow the steps given below to start the service (Steps 3-5 are to be executed for the first time only):

1. Go to Control Panel > Administrative Tools -> Services
2. In the list of services, click on **eGDB2<port number>**. For example, if 50000 is the port number, then the name of the service will be **eGDB250000**.
3. Click the **Properties** button available in the Services panel to open the Service panel for the selected service.
4. Click on the **Log On** tab and choose the **This Account** selection.

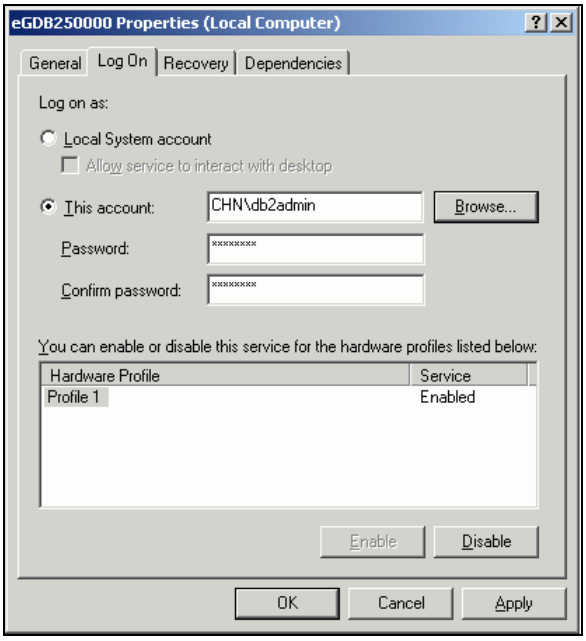

Figure 4.2: Selecting the This Account option from the Log On tab

5. Enter the username of the DB2 admin user in the **This Account** text box and the corresponding password against the **Password**text box.
6. Confirm the password by retyping it in the **Confirm password** text box. Then, click on the **OK** button at the bottom of the panel.
7. Start the selected service by clicking the **Start** button in the Services panel.

If the DB2 server is not configured with the performance monitors turned on, execute the command <EG_INSTALL_DIR>\lib\egdb2switch.bat.

Once the above-setup is done, manage the DB2 UDB - 6/7.x server that is auto discovered using the steps given in the Section **4.2**.

4.2 Managing the DB2 UDB – 6/7.x Server

eG Enterprise can automatically discover the DB2 UDB Server – Version 6/7.x in the environment. The discovered server can be managed using the following steps:

1. Log into the eG administrative interface.
2. If a DB2 UDB – 6/7.x server is already discovered, then directly proceed towards managing it using the **COMPONENTS – MANAGE / UNMANAGE** page (Infrastructure -> Components -> Manage/Unmanage).
3. However, if it is yet to be discovered, then run discovery (Infrastructure -> Components -> Discover) to get it discovered or add the component manually using the **COMPONENTS** page (Infrastructure -> Components -> Add/Modify). Remember that components manually added are managed automatically. Discovered components, however, are managed using the **COMPONENTS – MANAGE / UNMANAGE** page. Figure 4.3 and Figure 4.4 clearly illustrate the process of managing the *DB2 UDB – 6/7.x* server.

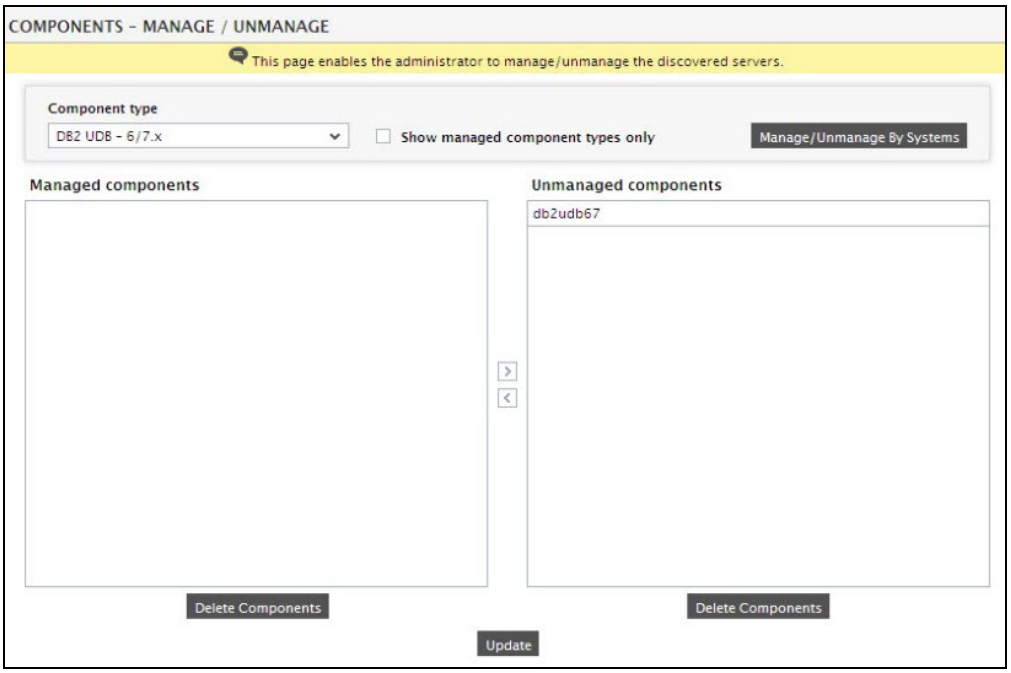

Figure 4.3: Viewing the list of unmanaged DB2 UDB - 6/7 . x servers in the COMPONENTS - MANAGE / UNMANAGE page

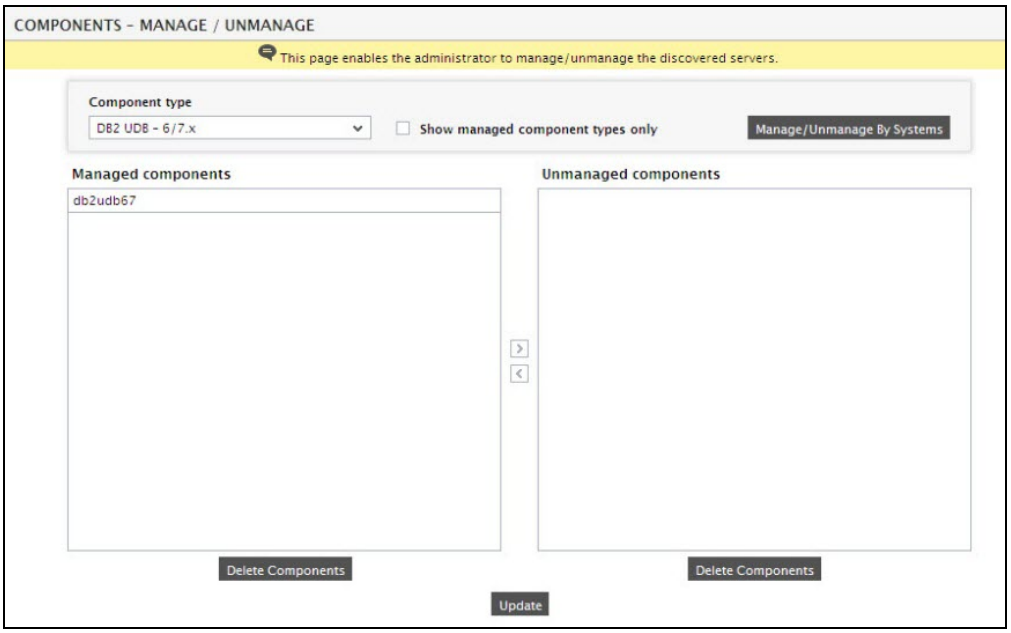

Figure 4.4: Managing a DB2 UDB - 6/7 . x server

4. Once you are done managing the DB2UDB-6/7.x server, sign out of the administrative interface.

Chapter 5: Monitoring DB2 Server Version 6.0/7.x

Figure 5.1 depicts the *DB2 UDB – 6/7.x* monitoring model that eG Enterprise prescribes for versions 6.0/7.x of the DB2 UDB server. This model has been deprecated, and is retained only to ensure backward compatability with older, less-used DB2 versions.

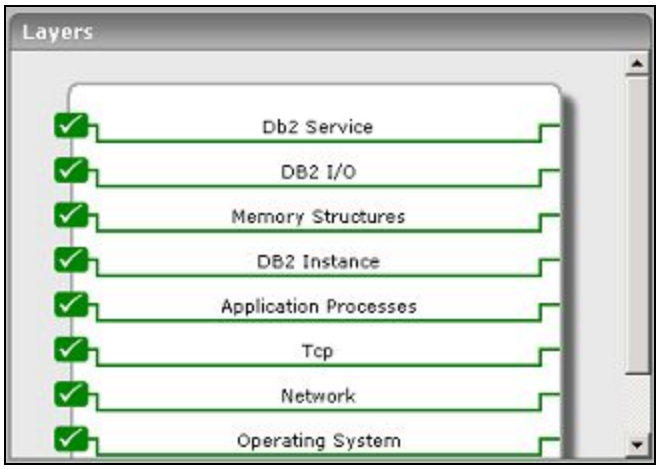

Figure 5.1: Layer model of the DB2 server version 6.0/7.x

The tests mapped to each of the layers of Figure 5.1 report critical statistics pertaining to the internal health of the DB2 server ver. 6.0/7.x. However, unlike the monitoring model discussed in **Monitoring IBM DB2 Server Version 8.0 (and above)**, this model requires that an eG agent be physically installed on the DB2 server for monitoring its performance. In other words, eG Enterprise cannot monitor a DB2 server ver. 6.0/7.x in an "agentless" manner.

The sections to come discuss each of the layers of Figure 5.1, elaborately.

5.1 The DB2 Instance Layer

Besides monitoring the availability of DB2 UDB databases, critical instance level measurements relating to the usage and responsiveness of each instance can be obtained on an on-going basis by an eG agent. This layer tracks the instance level measures of the DB2 UDB database server with the aid of the Db2Instance test (see Figure 5.2). In Figure 5.2, the Db2Instance test is used to monitor a specific instance called db2inst1 of the target DB2 database.

Figure 5.2: Tests mapping to the DB2 Instance layer

5.1.1 Db2 Instances Test

This test, executed by an internal agent, tracks various statistics at the instance level of a DB2 UDB database. The details of the test are provided below:

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Waiting agents:	Indicates the percentage of agents waiting to execute a unit of work.	Percent	It shows the percentage of agents that are waiting to execute (that is, that are asleep). Use this measure to adjust the MAXCAGENTS

Measurement	Description	Measurement Unit	Interpretation
			parameter setting for the database. If this value is high, you may consider increasing MAXCAGENTS.
Piped sorts accepted:	Indicates the percentage of piped sorts accepted during the interval.	Percent	Piped sorts reduce disk I/O and tend to be more efficient. A high percentage of piped sorts accepted indicate that efficient sorts are being performed. A low percentage indicates that sort heap and sort heap threshold may have to be adjusted. If this percentage is low, then increase SHEAPTHRES and possibly SORTHEAP.
Piped sort rejected:	Indicates the percentage of piped sort requests that were rejected.	Percent	Piped sorts reduce disk I/O and tend to be more efficient. A low percentage of piped sorts rejected indicate that efficient sorts are being performed. A high percentage of piped sorts rejected indicate that sort heap and sort heap threshold may have to be adjusted. If this percentage is high, then increase SHEAPTHRES and possibly SORTHEAP.
Agents registered:	Indicates the number of agents registered in the database manager instance that is being monitored (coordinator agents and subagents).	Number	This measure can be used to evaluate your setting for the MAXAGENTS configuration parameter.
Agents from empty pool:	Indicates how often an agent must be created because the pool is	Percent	A high value may indicate that the NUM_ POOLAGENTS configuration parameter should be

Measurement	Description	Measurement Unit	Interpretation
	empty.		increased. A low value suggests that NUM_POOLAGENTS is set too high, and that some of the agents in the pool are rarely used and are wasting system resources. A high percentage can also indicate that the overall workload for this node is too high. The workload can be adjusted by lowering the maximum number of coordinating agents specified by the MAXCAGENTS configuration parameter, or by redistributing data among the nodes.

5.2 The Memory Structures Layer

Next is the **Memory Structures** layer that tracks the health of the memory, lock, and buffer structures of the database server using the DB2LockAndDeadlock test and DB2BufferPool test (see Figure 5.3). Key performance metrics such as lock activity, buffer pool hit ratios, read and write rates to the database, average sorting time, rollback rates, etc. are being reported per database.

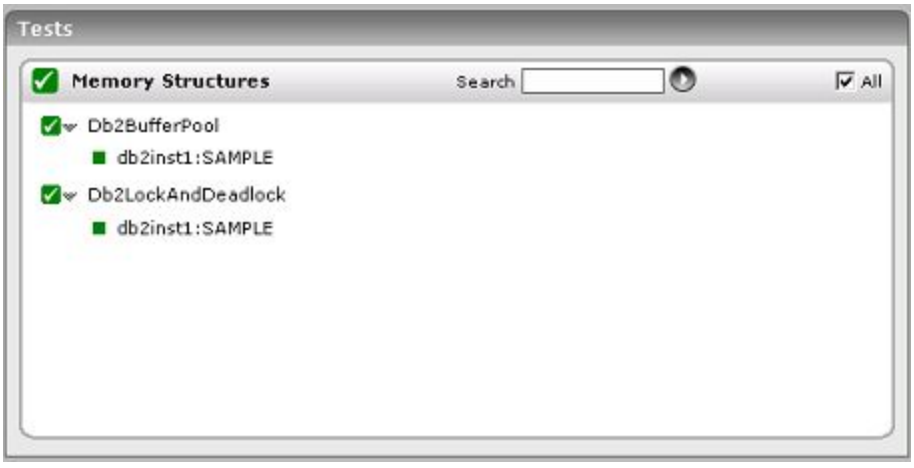

Figure 5.3: Tests mapping to the Memory Structures layer

5.2.1 Db2 Locks And Deadlocks Test

This test, executed by an internal agent, tracks various statistics pertaining to the locks and deadlocks in a DB2 database. The details of the test are provided below:

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Deadlocks:	Indicates the total number of deadlocks that have been detected.	Number	A high value is indicative of contention problems. These problems could be caused by the following reasons: • Lock escalations are occurring for the database • An application may be locking tables explicitly when system-generated row locks may be sufficient • An application may be using an inappropriate isolation level when binding • Catalog tables are locked for repeatable read

Measurement	Description	Measurement Unit	Interpretation
			• Applications are getting the same locks in different orders, resulting in deadlock. To resolve the problem, first determine the applications (or application processes) in which the deadlocks are occurring. Then, modify the application to enable it to execute concurrently. Some applications, however, may not be capable of running concurrently.
Exclusive lockescalations:	Indicates the number of times that locks have been escalated from several rowlocks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock. Other applications cannot access data held by an exclusive lock; therefore it is important to track exclusive locks since they can impact the concurrency of your data.	Number	A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the LOCKLIST and MAXLOCKS configuration parameters. A high value of this measure indicates that an application may be using exclusive locks when share locks are sufficient.
Locks held:	Indicates the total number of locks that have been currently held by all applications in/using the database.	Number	• This measure provides summary information about locking. A high value indicates that one or more of the applications needs to be tuned to improve performance. • You can also compare the value

Measurement	Description	Measurement Unit	Interpretation
			of this measure with the results of the following formula to determine the number of additional locks that may be requested. This comparison can help you determine if the configuration parameters need adjusting or your applications need tuning. (LOCKLIST * 4096 / 36) - locks held = # remaining where: LOCKLIST is the configuration parameter 4096 is the number of bytes in one 4K page 36 is the number of bytes required for each lock.
Lock timeouts:	When a unit of work exceeds the maximum allowable amount of time, a lock timeout occurs and the unit of work isn't granted the lock it has been waiting for. This measure indicates the total number of lock timeouts during a specific interval.	Number	If the number of lock timeouts becomes excessive when compared to normal operating levels, an application may be holding locks for long durations. This requires an adjustment in the LOCKTIMEOUT configuration parameter. Committing can also free locks. If the LOCKTIMEOUT database configuration parameter is set too high, it may result in too few lock timeouts. In such a case, your applications may wait excessively to obtain a lock.
Lock escalations:	Denotes the total number of times that	Number	A high value signifies a problem.

Measurement	Description	Measurement Unit	Interpretation
	locks have been escalated from several row locks to a table lock.		There are several possible causes for excessive lock escalations: • The lock list size (LOCKLIST) may be too small for the number of concurrent applications • The percent of the lock list usable by each application (MAXLOCKS) may be too small • One or more applications may be using an excessive number of locks. To resolve these problems, • Increase the LOCKLIST configuration parameter value. • Increase the MAXLOCKS configuration parameter value.
Percent of applications in lock wait:	Indicates the percentage of applications waiting for the release of lock.	Percent	A high value indicates that the applications are experiencing concurrency problems. Hence, the applications that are holding locks or exclusive locks for long periods of time have to be identified.

5.2.2 Db2 Buffer Pools Test

This test, executed by an internal agent, tracks various statistics pertaining to the buffer pool in a DB2 UDB database. The details of the test are provided below:

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Buffer pool hit ratio:	Indicates the effectiveness of the buffer pool. This indicates the percentage of the requested data blocks that is readily available in the memory without doing disk I/O.	Percent	The more a data block requested is found in the buffer pool, the better it is for performance since it avoids unnecessary disk input and output. The configuration of the buffer pool is the single most important tuning area, since most data manipulation for connected applications, excluding large objects and long field data, takes place here. If this measure is low (less than 90% for decision support, less than 50% for query- only and online transaction procession), increase the size of the buffer pool by increasing BUFFPAGE As a rule, the buffer pool should be as large as possible. Increase BUFFPAGE until you see an increase in swapping (indicated by an operating system monitor).
Buffer pool hit ratio:	Indicates the effectiveness of the buffer pool. This indicates the	Percent	The more data found in the buffer pool, the better it is for performance since it avoids unnecessary input and output.

Measurement	Description	Measurement Unit	Interpretation
	percentage of the requested index blocks that is readily available in the memory without doing disk I/O.		The configuration of the buffer pool is the single most important tuning area, since most data manipulation for connected applications, excluding large objects and long field data, takes place here. If this measure is low (less than 90% for decision support, less than 50% for query- only and online transaction procession), increase the size of the buffer pool by increasing BUFFPAGE. As a rule, the buffer pool should be as large as possible. Increase BUFFPAGE until you see an increase in swapping (indicated by an operating system monitor.
Catalog cache hit ratio:	Indicates the percentage of catalog cache hit ratio. This indicates the percentage of the requested catalog blocks that is readily available in the memory without doing disk I/O.	Percent	This measure includes both successful and unsuccessful accesses to the catalog cache. The catalog cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. If the ratio is greater than 80%, then the catalog cache is performing well. A smaller value indicates that the catalog cache size should be increased by tuning the parameter CATALOGCACHE_ SZ in the database configuration. The value may be low immediately following the first connection to the database.

Measurement	Description	Measurement Unit	Interpretation
			The execution of Data Definition Language (DDL) SQL statements involving a table, view, or alias will evict the table descriptor information for that object from the catalog cache causing it to be re-inserted on the next reference. Therefore, the heavy use of DDLs may also increase the value of the measure.

5.3 The DB2 I/O Layer

The DB2 I/O layer monitors the input/output activity happening on the DB2 UDB server with the Db2 I/O test shown in Figure 5.4.

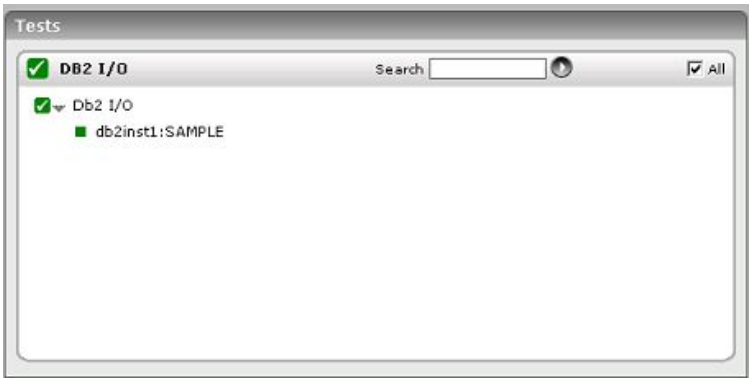

Figure 5.4: Tests mapping to the DB2 IO layer.

5.3.1 Db2 I/O Test

This test, executed by an internal agent, tracks various statistics pertaining to the inputs and outputs of a DB2 UDB database. The details of the test are provided below:

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

Measurements made by the test:

Measurement	Description	Measurement Unit	Interpretation
Direct read rate:	Indicates the total number of direct reads by the application per sec. In other words, it the number of read operations that do not use the buffer pool.	Reads/Sec	Direct reads are performed in units, the smallest being a 512- byte sector. They are used when: • Reading LONG VARCHAR columns • Reading LOB (large object) columns • Performing a backup A high value over a period of time may be indicative of a performance bottleneck.
Direct write rate:	Indicates the total number of direct writes by the application per sec. In other words, it is the number of write operations that do not use the buffer pool.	Writes/Sec	Direct writes are performed in units, the smallest being a 512-byte sector. They are used when: • Writing LONG VARCHAR columns • Writing LOB (large object) columns • Performing a restore • Performing a load. A high value over a period of time may be indicative of a performance

Measurement	Description	Measurement Unit	Interpretation
			bottleneck.
Buffer pool I/O rate:	Indicates the rate at which the buffer pool I/O operations are being done in the database.	IOs/Sec	In conjunction with the hit ratio statistics, and the characteristics of the applications executing, the I/O load may require adjustment of BUFFPAGE, or applications may require further tuning.

5.4 The DB2 Service Layer

This layer tracks the overall health of the service offered by the database server to clients with the help of DB2Transaction test, DB2Sort test, and DB2SQLActivity test (see Figure 5.5). The details of all the tests mentioned above are available in the following sections.

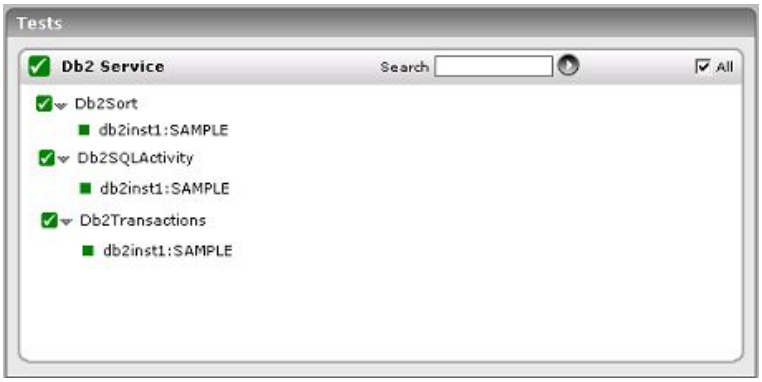

Figure 5.5: Tests mapping to the Db2 Service layer

5.4.1 Db2 Transactions Test

This test, executed by an internal agent, tracks various statistics pertaining to transactions in DB2 UDB database. The details of the test are provided below:

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

Measurements made by the test:

Measurement	Description	Measurement Unit	Interpretation
Commit rate:	Indicates the transaction throughput. This measure is the sum of the committed statements attempted and internal commits (total number of commits initiated internally by the database manager.) per sec.	Commits/Sec	A decrease in this measure during the monitoring period may indicate that the applications are not doing frequent commits. This may lead to problems with logging and data concurrency. The cause has to be probed in the application.
Rollback rate:	Indicates the rate of unit of work rollbacks.	Rollbacks /Sec	A high rollback rate is an indicator of bad performance, since work performed up to the rollback point is wasted. The cause of the rollbacks has to be probed in the application.
Transaction rate:	Indicates the rate of commits and rollbacks for the application using the DB2 Connect gateway.	Trans/Sec	A high transaction rate with high rollback rate indicates bad performance.

Note:

Since these measures pertain to the factors that are application dependent, no specific boundaries have been indicated for these values.

5.4.2 Db2 Sorts Test

This test, executed by an internal agent, tracks various statistics pertaining to the sorts in a DB2 UDB database. The details of the test are provided below:

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

Measurements made by the test:

Measurement	Description	Measurement Unit	Interpretation
Avg sort time:	Indicates the average sort time for all sorts performed by all applications connected to a particular database.	Ms/Sort	A high value can point to a database with possible sorting problem (for example, applications are spending too much time on sorts). See the SORT screen for this database. The sort information for this database manager should also be looked at instance level. It may be necessary to increase the SORTHEAP configuration parameter.
Percent Sort Overflow:	Denotes the percentage of sorts that overflowed.	Percent	Sort overflows are sorts that ran out of sort heap and required disk space for temporary storage. These sorts are not efficient, and when the value of this measure is

Measurement	Description	Measurement Unit	Interpretation
			consistently high for a number of intervals, then, it may be necessary to increase the SORTHEAP configuration parameter.

5.4.3 Db2 SQL Activity Test

This test, executed by an internal agent, tracks various statistics pertaining to the SQL activities happening in a DB2 UDB database. The details of the test are provided below:

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every database being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.

Measurements made by the test:

Measurement	Description	Measurement Unit	Interpretation
	Indicates the number of dynamic SQL statements that were attempted.	Stmts/Sec	This is an indication of throughput of the system during the monitoring period. A high value of dynamic SQLs and low value of failed SQLs indicate good throughput.
Select rate:	Indicates the number of SELECT SQL statements	Stmts/Sec	This measure can be used to determine the level of database

Measurement	Description	Measurement Unit	Interpretation
	that were executed.		activity at the application or database level. This measure is useful for analyzing the application activity and throughput.
Failed SQL rate:	Indicates the number of SQL statements that were attempted, but failed.	Stmts/Sec	This measure helps in determining reasons for poor performance, since, failed statements means resource wastage and lower throughput for the database.
UID SQL rate:	Indicates the number of SQL UPDATE, INSERT, and DELETE statements that were executed	Stmts/Sec	This information can be useful for analyzing application activity and throughput.
SQL failures:	Indicates the percentage of SQL statements that failed during the interval.	Percent	A relatively high value indicates a problem. The percentage of SQL statements that received a negative SQL code indicates a possible cause of poor performance. Failed SQL statements waste system resources. Hence, this value of this measure should be very low.
Percent of DDL SQLs:	Denotes the percentage of SQL statements that are DDL during a specific interval.	Percent	This value should normally be low.
Percent of UID SQLs:	Percentage of SQL statements that are update/insert/delete during the interval.	Percent	This measure can be used to determine the level of database activity at the application or database level.

Measurement	Description	Measurement Unit	Interpretation

Note:

The values for these measures are dependent on the type of application that the database is supporting. For example, in an On-Line Transaction Processing System (OLTP), we can expect a relatively high number of Updates, Inserts, and Deletes compared to a Decision Support System (DSS).

5.4.4 DB2 Backup Status Test

This test reports the statistics pertaining to the backups performed on the target DB2 database server. Using this test, administrators can figure out the count of backups that failed and the count of backup jobs that were completed successfully. In addition, this test reports the maximum time taken to complete the backup jobs. By closely monitoring the measures reported by this test, administrators can pin point the jobs that failed and the further analyze on why the backup jobs failed and rectify the failure at the earliest.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.

7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test
8. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:The eG manager license should allow the detailed diagnosis capabilityBoth the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total backup	Indicates the total number of backups performed during the last measurement period.	Number	
Completed backup	Indicates the number of backups completed successfully during the last measurement period.	Number	The detailed diagnosis of this measure lists the DB partition number EID, Start time, End Time, Duration (Minutes), Backup ID, Comment text, Number of tablespaces, Tablespace Names, Operation type, Object type, Location, Device type, SQL code, SQL warn and SQL state.
Failed backup	Indicates the number of backups failed during the last measurement period.	Number	Ideally, the value of this measure should be zero. Any value greater than zero, is a cause of concern, as it indicates the existence of a failed job. To know which job(s) has failed, use the detailed diagnosis capability of this measure. The detailed diagnosis of this measure lists the DB partition number EID, Start time, End Time, Duration

Measurement	Description	Measurement Unit	Interpretation
			(Minutes), Backup ID, Comment text, Number of tablespaces, Tablespace Names, Operation type, Object type, Location, Device type, SQL code, SQL warn and SQL state.
Maximum duration of completed backup	Indicates the maximum time taken to complete the backup jobs during the last measurement period.	Minutes	

5.5 Troubleshooting

If a DB2 server is in an unknown state, then do the following:

1. Execute the eG internal agent on the DB2 server's host.
2. In addition, a separate DB2 performance monitor script has to be started. One instance of this script has to be executed for each DB2 instance to be monitored. This script uses the DB2 snapshot-monitoring interface to extract and report performance data to the agent. This script should be executed from a user account that belongs to the DB2 administrator group.

Note:

Please refer to Section **4.1** above for more details on the DB2 performance monitor script

Chapter 6: How to Configure and Monitor DB2 DPF Server Using eG Enterprise?

eG Enterprise can monitor DB2 DPF server in an agent-based or an agentless manner. In case of the agentless approach, the remote agent used to monitor the DB2 DPF server should be deployed on a remote Windows host in the environment.

The broad steps for monitoring DB2 DPF server using eG Enterprise are as follows:

- Managing the DB2 DPF server
- Configuring the tests

6.1 Managing the DB2 DPF server

The DB2 DPF server cannot be automatically discovered by eG Enterprise. This implies that you will have to manually add the server into the eG Enterprise system to manage it. Follow the steps below to achieve the same:

1. Login into the eG administrative interface.
2. eG Enterprise cannot automatically discover the DB2 DPF server. You need to manually add the eDirectory netware using the **COMPONENTS** page (see Figure 6.1) that appears when the Infrastructure -> Components -> Add/Modify menu sequence is followed. Remember that components manually added are managed automatically.

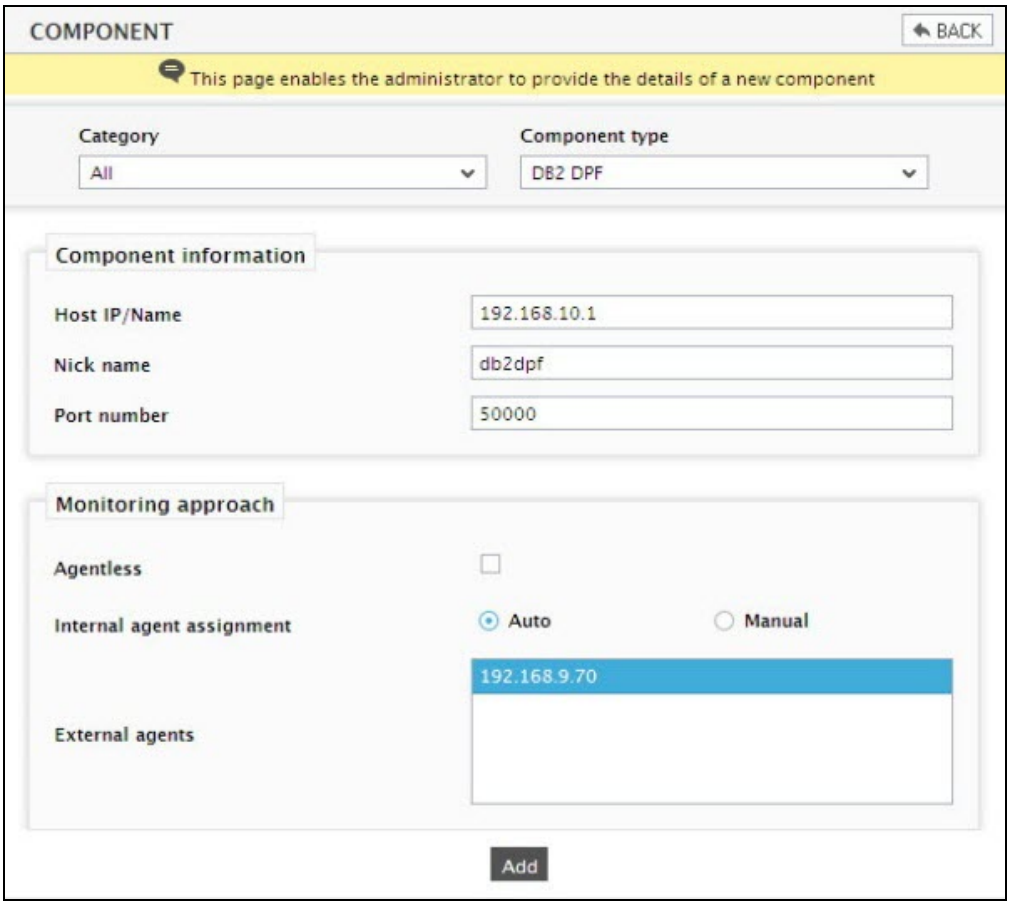

Figure 6.1: Adding the DB2 DPF server

3. Specify the **Host IP/Name** and **Nick name** of the DB2 DPF server that you want to manage.
4. The **Port number** will be set as 50000 by default. If the DB2 DPF server is listening on a different port in your environment, then override this default setting.
5. Finally, click the **Add** button to add the DB2 DPF server for monitoring.

6.2 Configuring the tests

1. When you attempt to sign out, a list of unconfigured tests appears (see Figure 6.2).

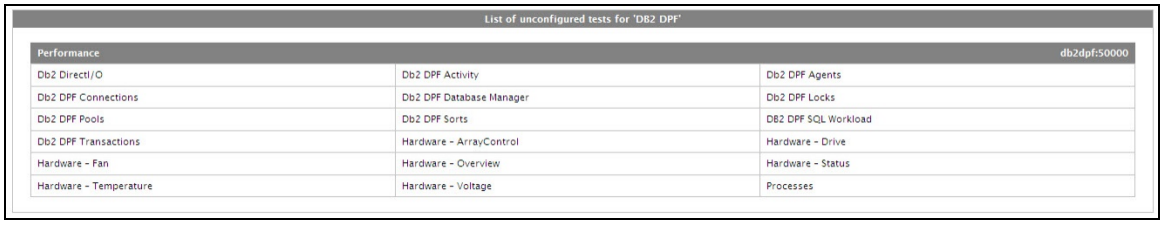

Figure 6.2: List of unconfigured tests for the DB2 DPF server

2. Click on the tests in Figure 6.2 to configure. To know how to configure the tests, refer to the **Monitoring the IBM DB2 Server in a DPF Environment** chapter.
3. Now, try signing out of the eG administrative interface. This time it will prompt to configure the **Processes** test. To know how to configure this test, refer to the *Monitoring Unix and Windows Servers* document.

Chapter 7: Monitoring the IBM DB2 Server in a DPF Environment

The Database Partitioning Feature (DPF) is available on DB2 UDB Enterprise Server Edition (ESE). With DPF your database is scalable as you can add new machines and spread your database across them. This means more CPUs, more memory and more disks from each of the additional machines for your database! DB2 UDB ESE with DPF is ideal to manage data warehousing, data mining and online analytical processing (OLAP) workloads. It can also work well with online transaction processing (OLTP) workloads.

When a database is partitioned, you split your database into different independent parts, each consisting of its own data, configuration files, indexes and transaction logs. Each of these parts is a database partition. You can assign multiple partitions to a single physical machine. These are called 'logical partitions' and they share the resources of the machine.

A single-partition database is a database with only one partition.

A multi-partition database (also referred to as a partitioned database), is a database with two or more partitions. Depending on your hardware environment, there can be several configurations in which you can partition your database. Figure 7.1 shows the configuration of one logical partition in a single SMP machine.

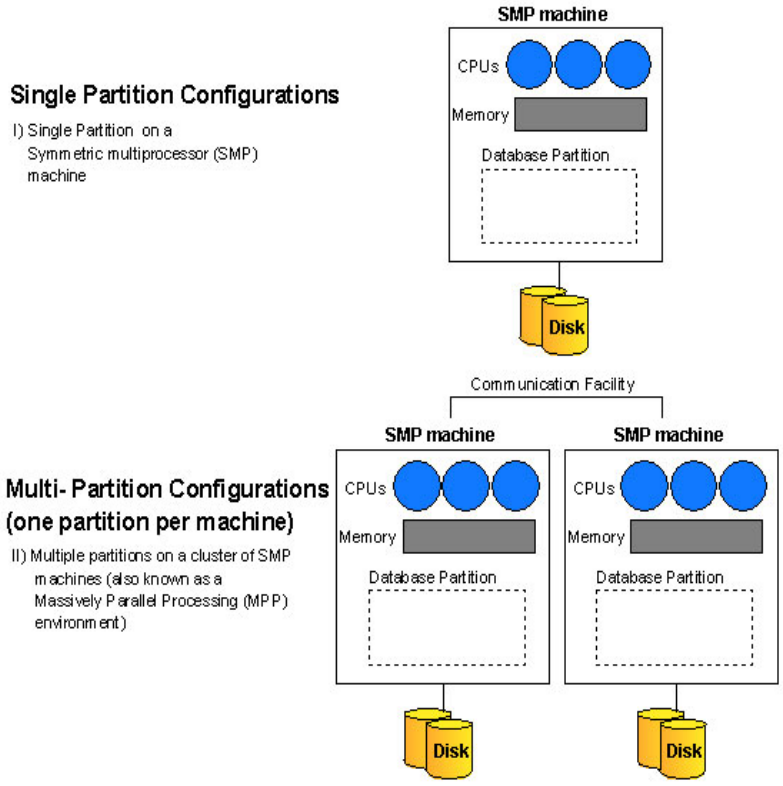

Figure 7.1: A Single-partition Configuration

Figure 7.2 shows more multi-partition configurations with several logical partitions in a machine.

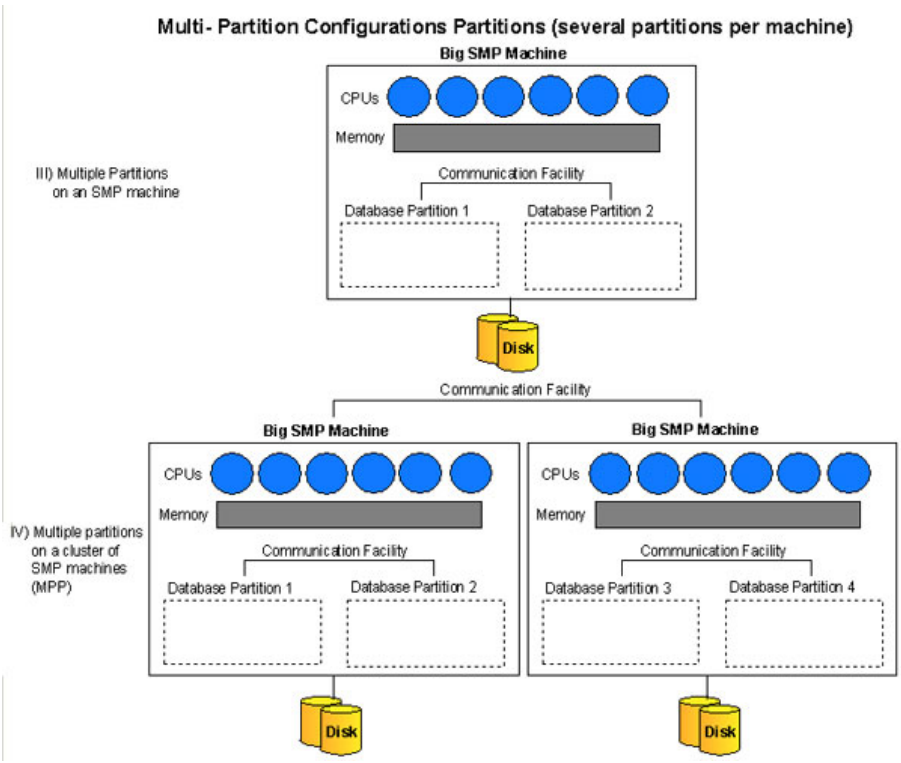

Figure 7.2: A multi-partition config

A user will connect to the database and issue queries as usual without a need to know that the database has been partitioned.

Figure 7.3 visualizes how a DB2 environment is split in a DPF system.

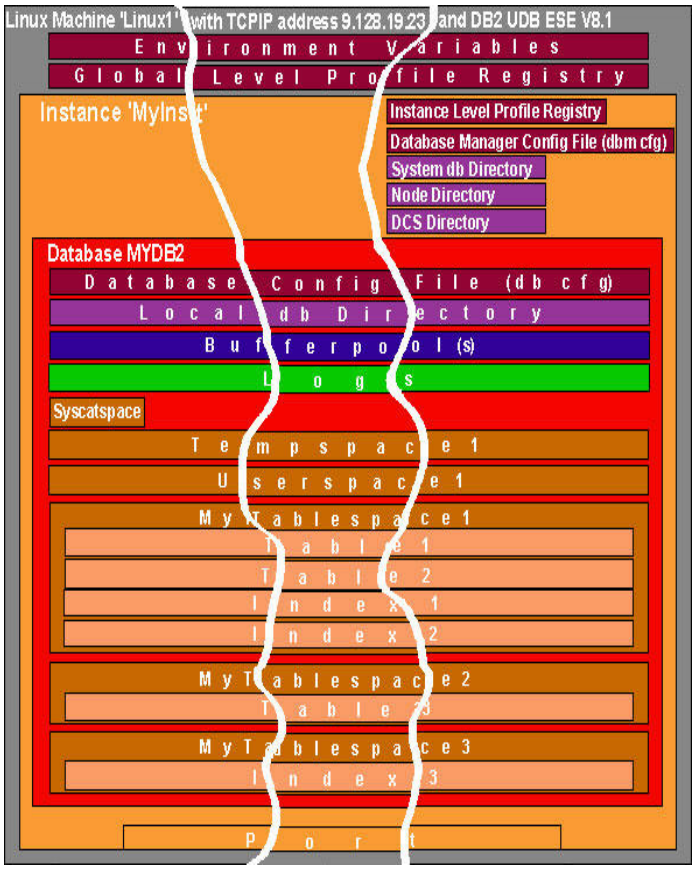

Figure 7.3: A visualization of a DPF system

eG Enterprise provides an exclusive *DB2 DPF* monitoring model that monitors the load on the DB2 server, and reveals whether the load is uniformly distributed across all the logical partitions.

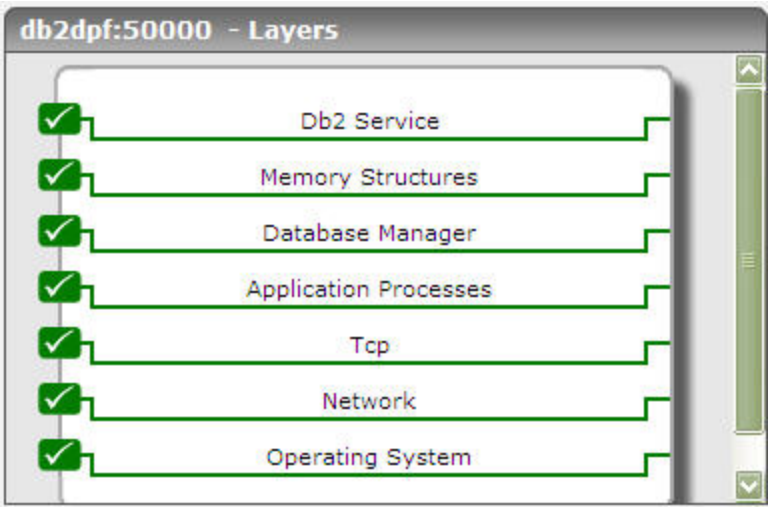

Figure 7.4: The DB2 DPF Monitoring Model

Each layer of this model is mapped to a variety of tests that report useful statistics that provide accurate answers to persistent performance queries:

- What is the current connection load on the DB2 database manager? How many of these connections are local, and how many are remote connections?
- How quickly does the DB2 server process requests from client applications? Are there sufficient agents in the agent pool to service all the client requests?
- Are the agents in the pool utilized optimally, or are too many agents idle?
- Does the database server perform sorting efficiently? Has adequate sort heap space been allocated to the database manager to enable this?
- Are sort overflows kept at a minimum?
- Does sorting take too long?
- Are lock escalations occurring too frequently on the logical partitions?
- Do applications obtain locks quickly, or do they have to wait too long for locks?
- Are the database buffer pools adequately sized for each logical partition?
- Are the page cleaners and prefetchers been utilized effectively by each logical partition?
- Are too many rollbacks happening on the logical partitions?
- Have too many SQL statements failed on any logical partition? If so, which one is it?

The sections to come elaborate on each layer of Figure 3.1, the tests associated with them, and the statistics they extract.

7.1 The Database Manager Layer

Using the tests associated with the **Database Manager** layer, the following can be monitored:

- Critical activities performed by the database manager
- Client connections to the database manager
- Usage of the agent pools on the database manager

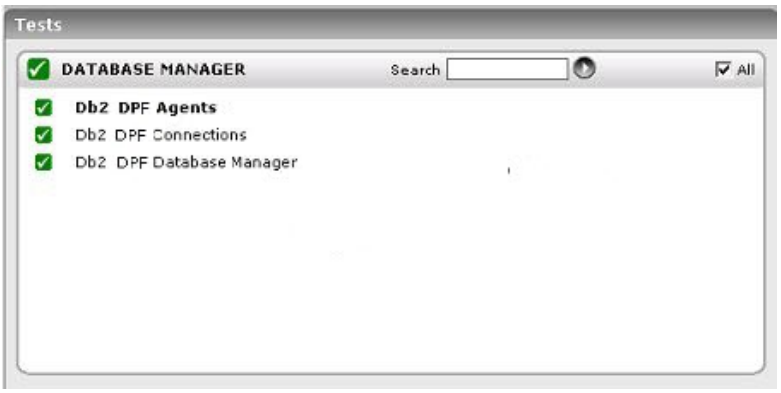

Figure 7.5: The tests associated with the Database Manager layer

7.1.1 Db2 DPF Agents Test

An agent is a process or thread that carries out the requests made by a client application. Each connected application is served by exactly 1 coordinator agent and possibly, a set of subordinator agents or subagents. Subagents are used for parallel SQL processing in partitioned databases and on SMP machines.

For partitioned database environments, each partition (that is, each database server or node) has its own pool of agents from which subagents are drawn. Because of this pool, subagents do not have to be created and destroyed each time one is needed or has finished its work. The subagents can remain as associated agents in the pool and be used by the database manager for new requests from the application they are associated with.

The Db2 DPF Agents test monitors how effectively the agent pool has been utilized.

Target of the test : A DB2 database server with DPF enabled

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The IP address of the DB2 server
3. **PORT** - The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified

DATABASE: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.

5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total Agents:	Indicates the total number of agents currently registered in the database manager instance that is being monitored (Sum of coordinator agents and subagents).	Number	This measure can be used to evaluate the setting for MAXAGENTS configuration parameter.
Idle Agents:	Indicates the percentage of agents in the agent pool that is currently unassigned to an application and are, therefore, “idle”.	Number	Having idle agents available to service requests for agents can improve performance. So you can use this measure to help set the NUM_ POOLAGENTS configuration parameter.
Agents waiting on token:	Indicates the percentage of agents waiting for a token so they can execute a transaction in the database manager.	Percent	You can use this element to help evaluate your setting for the MAXCAGENTS configuration parameter. Each application has a dedicated coordinator agent to process database requests within the database manager. Each agent has to get a token before it can execute a transaction. The maximum number of agents that

Measurement	Description	Measurement Unit	Interpretation
			can execute database manager transactions is limited by the configuration parameter MAXCAGENTS.
Agents creation ratio:	Indicates the ratio of number of agents assigned directly from agent pool to the total number of agents used to service requests.	Percent	A high percentage indicates the effectiveness of the agent pool. A consistent low value indicates that the number of agents in the agent pool are not adequate to service requests. You might want to consider increasing the NUM_ POOLAGENTS setting in this case.
Stolen Agents:	Indicates the number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.	Number	If this value is high, consider increasing the NUM_ POOLAGENTS configuration parameter.

7.1.2 Db2 DPF Connections Test

The Db2 Connections test reports key statistics pertaining to the local and remote connections to the DB2 database manager.

Target of the test : A DB2 database server with DPF enabled

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed

2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total Connections:	Indicates the totalnumber of local and remote connections that are currently present in the database manager.	Number	
Local Connections:	Indicates the number of local applications that are currently connected to a database within the database manager instance being monitored.	Number	This number can help you determine the level of concurrent processing occurring in the database manager. This number only includes applications that were initiated from the same instance as the database manager. The applications are connected, but may or may not be executing a unit of work in the database. When used in conjunction with the Remote connections measurement, this measure can help you adjust the setting of the MAXAGENTS configuration parameter.

Measurement	Description	Measurement Unit	Interpretation
Remote Connections:	Indicates the percentage of agents waiting for a token so they can execute a transaction in the database manager.	Number	This number does not include applications that were initiated from the same instance as the database manager. When used in conjunction with the Local connections measure, this measure can help you adjust the setting of the MAX_ COORDAGENTS configuration parameter.
Local connections in exec:	Indicates the number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.	Number	This number can help you determine the level of concurrent processing occurring in the database manager. This number only includes applications that were initiated from the same instance as the database manager. When used in conjunction with the Remote connections in exec measure, this measure can help you adjust the setting of the MAXCAGENTS configuration parameter.
Remote connections in exec:	Indicates the number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.	Number	This number can help you determine the level of concurrent processing occurring on the database manager. This number does not include applications that were initiated from the same instance as the database manager. When used in conjunction with the Local connections in exec measure, this metric can help you adjust the setting of the MAXCAGENTS configuration parameter.

7.1.3 Db2 DPF Database Manager Test

The database manager includes the database engine and the facilities to access data, such as the command line processor and the application interfaces. This test reports key statistics pertaining to the health of the DB2 database manager.

Target of the test : A DB2 database server with DPF enabled

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every DB2 database server being monitored

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Sorts post threshold:	Indicates the number of sorts per second that have requested heaps after the sort heap threshold has been exceeded.	Sorts/Sec	Under normal conditions, the database manager will allocate sort heap using the value specified by the SORTHEAP configuration parameter. If the amount of memory

Measurement	Description	Measurement Unit	Interpretation
			allocated to sort heaps exceeds the sort heap threshold (SHEAPTHRES configuration parameter), the database manager will allocate sort heap using a value less than that specified by the SORTHEAP configuration parameter. Each active sort on the system allocates memory, which may result in sorting taking up too much of the system memory available. Sorts that start after the sort heap threshold has been reached may not receive an optimum amount of memory to execute, but, as a result, the entire system may benefit. By modifying the sort heap threshold and sort heap size configuration parameters, sort operation performance and overall system performance can be improved. If this element's value is high, you can: • Increase the sort heap threshold (SHEAPTHRES), or, • Adjust applications to use fewer or smaller sorts via SQL query changes.
Piped Sorts Requested:	A sort is classified as piped sort if the sorted information can return directly without requiring a temporary	Sorts/Sec	Piped sorts may reduce disk I/O. Allowing more piped sorts therefore, can improve the performance of sort operations and possibly the performance of the overall system.

Measurement	Description	Measurement Unit	Interpretation
	table to store a final, sorted list of data. This measure reports the number of piped sorts that have been requested per second.		
Piped Sorts Rejected:	Indicates the percentage of piped sort requests that have been rejected.	Percent	When the number of rejected piped sorts are high, you can improve sort performance by adjusting one or both of the following configuration parameters: • SORTHEAP • SHEAPTHRES If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold, then there is the possibility that more memory will remain allocated for sorting. This could cause the paging of memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.
Hash Join Post Threshold:	Indicates the total number of times that a hash join heap request was limited due to concurrent use of shared or private sort heap space.	Hits/Sec	If this value is large, the sort heap threshold should be increased.

7.2 The Memory Structures Layer

The tests mapped to the Memory Structures layer (see Figure 7.6), report critical statistics that reveal:

- How efficiently the locking and sorting activities occur on the monitored DB2 database
- How well the buffer pools are managed
- The level of I/O activity on the DB2 database

Figure 7.6: The tests associated with the Memory Structures layer

7.2.1 Db2 DPF Locks Test

Typically, locking activity is governed by the following factors:

- Concurrency and granularity
- Lock compatibility
- Lock conversion
- Lock escalation
- Lock waits and timeouts
- Deadlocks

In the event of an application slowdown, the measures reported by the Db2 DPF Locks test enable administrators to accurately determine whether/not any of the above-mentioned factors have adversely impacted application performance, and if so, to what extent.

Target of the test : A DB2 database server with DPF enabled

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each logical partition of every database on the DB2 database server that is currently active.

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Locks Held:	Indicates the total number of locks currently held by all applications in this partition of this database.	Number	
Locks Escalated:	Indicates the number of times every second that locks have been escalated from several row locks to a table lock.	Number	There are several possible causes for excessive lock escalations: • The lock list size (locklist) may be too small for the number of concurrent applications

Measurement	Description	Measurement Unit	Interpretation
	A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. The amount of lock list space available is determined by the maxlocks and locklist configuration parameters. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it will then use space in the lock list allocated for other applications. When the entire lock list is full, an error occurs. This data item includes a count of all lock escalations, including exclusive lock escalations.		• The percent of the lock list usable by each application (maxlocks) may be too small • One or more applications may be using an excessive number of locks. • To resolve these problems, you may be able to: • Increase the locklist configuration parameter value. • Increase the maxlocks configuration parameter value. Identify the applications with large numbers of locks or those that are holding too much of the lock list. These applications can also cause lock escalations in other applications by using too large a portion of the lock list. These applications may need to resort to using table locks instead of row locks, although table locks may cause an increase in lock_waits and lock_wait_time.
Exclusive Lock Escalations:	Indicates the number of times per second that locks have been	Number	Other applications cannot access data held by an exclusive lock;

Measurement	Description	Measurement Unit	Interpretation
	escalated from several row locks to one exclusive table lock, or the number of times (per second) an exclusive lock on a row caused the table lock to become an exclusive lock.		therefore it is important to track exclusive locks since they can impact the concurrency of your data. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the locklist and maxlocks configuration parameters. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it will then use space in the lock list allocated for other applications. When the entire lock list is full, an error occurs. See Lock escalations for possible causes and resolutions to excessive exclusive lock escalations. An application may be using exclusive locks when share locks are sufficient. Although share locks may not reduce the total number of lock escalations share lock escalations may be preferable to exclusive lock escalations.
Locks Timedout:	Indicates the number of times that a request to lock an object timed-out instead of being granted.	Number	This measurement can help you adjust the setting for the locktimeout database configuration parameter. If the number of lock time-outs becomes excessive when compared to

Measurement	Description	Measurement Unit	Interpretation
			normal operating levels, you may have an application that is holding locks for long durations. In this case, this element may indicate that you should analyze some of the other lock and deadlock monitor elements to determine if you have an application problem. You could also have too few lock time- outs if your locktimeout database configuration parameter is set too high. In this case, your applications may wait excessively to obtain a lock.
Lock Waits:	Indicates the total number of times per second that applications or connections in this partition waited for locks.	Waits/Sec	If the value is consistently high, find the applications or connections causing lock waits and fine tune the appropriate SQL queries.
Average Lock Wait Time:	Indicates the average time that all the applications were waiting for a lock.	Seconds	If the average lock wait time is high, you should look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate.
Percent of Application in Lock Wait:	Indicates the percentage of applications in this partition waiting for the release of a lock.	Percent	If this value is high, the applications may have concurrency problems, and you should identify applications that are holding locks or exclusive locks for long periods of time.
Deadlocks:	Indicates the total	Deadlocks/Sec	This element can indicate that

Measurement	Description	Measurement Unit	Interpretation
	number of deadlocks that have been detected per second in this partition.		applications are experiencing contention problems. These problems could be caused by the following situations: • Lock escalations are occurring for the database • An application may be locking tables explicitly when system-generated row locks may be sufficient. • An application may be using an inappropriate isolation level when binding • Catalog tables are locked for repeatable read • Applications are getting the same locks in different orders, resulting in deadlock You may be able to resolve the problem by determining in which applications (or application processes) the deadlocks are occurring. You may then be able to modify the application to enable it to execute concurrently. Some applications, however, may not be capable of running concurrently.

7.2.2 Db2 DPF Pools Test

A buffer pool is an area of memory into which database pages are read, modified, and held during processing.

Buffer pools improve database performance. If a needed page of data is already in the buffer pool, that page is accessed faster than if that page had to be read directly from disk. The database manager has agents whose tasks are to retrieve data pages from disk and place them in the buffer pool (prefetchers), and to write modified data pages from the buffer pool back to disk (page cleaners).

The reading and writing of data pages to and from disk is called disk input/output (I/O). Avoiding the wait associated with disk I/O is the primary way to improve the performance of the database. How you create the buffer pool, and configure the database manager and the agents associated with the buffer pool, controls the performance of the database. Through SQL and configuration parameters, you can control the size of the buffer pool, the number of prefetchers and page cleaners that move data pages into and out of the buffer pool, the size of the data pages, and the number of data pages that can be moved at one time.

In Figure 3, we showed bufferpools split across the different partitions. Interpreting this figure for buffer pools, is different than for the other objects, because the data cached in the bufferpools is not partitioned as the figure may imply. What is actually happening is that buffer pools in a DPF environment can be tailored to the different partitions. Using the CREATE BUFFERPOOL statement with the DATABASE PARTITION GROUP clause, you can associate a bufferpool to a given partition group. What this means is that you have the flexibility to define the buffer pool to the specific partitions defined in the partition group. In addition, the size of the buffer pool on each partition in the partition group can be different if desired.

The statistics reported by the Db2 Pools test help administrators analyze the usage of the buffer pools, and provides them with useful pointers to fine-tune the configuration of the buffer pools.

Target of the test : A DB2 database server with DPF enabled

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every logical partition of each database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified

DATABASE: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.

5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Buffer Pool Hit Ratio:	Indicates the percentage of the pages requested that is readily available in the memory of this logical partition without doing disk I/O.	Percent	This measure is an overall indicator of how well the buffer pool is performing. Lower values indicate that more physical I/O is happening than logical. Since physical I/O costs more, maintaing higher buffer hit ratio is desired. Typically a hit ratio over 90% shows that the buffer pool is performing well. If this measure is consistently low, increase the size of the buffer pool by increasing BUFFPAGE configuration value.
Buffer Pool Hit Ration (Data):	Indicates the percentage of the data pages requested that is readily available in the memory of this logical partition without doing disk I/O.	Percent	This measure is an indicator of how well the buffer pool is performing for the data page requests. Lower values indicate that more physical I/O is happening than logical. Since physical I/O costs more, maintaing higher buffer hit ratio is desired. Typically a hit ratio over 90% shows that the buffer pool is performing well. If this measure is consistently low, increase the size of the buffer pool by increasing BUFFPAGE configuration value.

Measurement	Description	Measurement Unit	Interpretation
Buffer Pool Hit Ratio (Index):	Indicates the percentage of the index pages requested that is readily available in the memory of this logical partition without doing disk I/O.	Percent	This measure is an overall indicator of how well the buffer pool is performing. Lower values indicate that more physical I/O is happening than logical. Since physical I/O costs more, maintaing higher buffer hit ratio is desired. Typically a hit ratio over 90% shows that the buffer pool is performing well. If this ratio is really low and the overall ratio is relatively high, then break the index (s) out into their own tablespace/buffer pool.
Pre Fetch Ratio:	The ratio of asynchronous reads to synchronous reads. The value indicates how effectively the DB2 database manager is populating the buffer pools through the use of prefetchers.	Percent	High value indicates more asynchronous I/O is happening than synchronous. The value can be used to tune the num_ ioservers configuration parameter.
Percent Log Cleans:	Indicates the percentage of times a page cleaner was invoked because the logging space used had reached a predefined criterion for the database logical partition.	Percent	If this value is high (say > 40%), this could mean that page cleaners are constantly being utilized to clean the log and aren't available for other page cleaning activities, hampering performance. On the other hand, if the value is low, (say < 10%) then the page cleaners aren't being triggered as often for this activity. this means that they would be more available for the other types of page cleaning activities, which is great for buffer pool performance.

Measurement	Description	Measurement Unit	Interpretation
Percent Dirty Page Cleans:	Indicates the percentage of times a page cleaner was invoked because the buffer pool defined for this logical partition had reached the dirty page threshold criterion for the database.	Percent	The threshold is set by the chngpgs_ thresh configuration parameter. It is a percentage applied to the buffer pool size. When the number of dirty pages in the pool exceeds this value, the cleaners are triggered. If this value is set too low, pages might be written out too early, requiring them to be read back in. If set too high, then too many pages may accumulate, requiring users to write out pages synchronously.
Percent Victim Cleans:	Indicates the percentage of times the page cleaner (s) were triggered to oust a victim page from the buffer pool. A victim page is a clean or dirty page in the buffer pool that is removed simply because DB2 needs to make room for incoming pages. If a victim page is a dirty page then the information must be written out to disk. Any page that is removed will most likely cause more physical I/O to occur in order to retrieve it again at later time when DB2 is ready to use it.	Percent	If the ratio is higher then the above two then that is typically a good indicator that the buffer pool needs to be larger since there never seems to be enough room for new pages to be brought in. This could also be a sign that dirty pages are staying in the buffer pool too long which could mean that the changed pages threshold (CHNGPGS_ THRESH) is set too high. Even the SOFTMAX parameter could be set too high and too much of the changed pages that are logged are not getting flushed out to make way for new pages. If this ratio is low, it may indicate that you have defined too many page cleaners. If your chngpgs_thresh is set too low, you may be writing out pages that you will dirty later. Aggressive cleaning defeats one purpose of the buffer

Measurement	Description	Measurement Unit	Interpretation
			pool, that is to defer writing to the last possible moment.
Catalog Cache Hit Ratio:	Indicates the percentage of time the requested information for table descriptor or authorization was readily available in catalog cache without requiring to perform disk I/O.	Percent	The catalog cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. If the ratio is greater than 80%, then the catalog cache is performing well. A smaller value indicates that the catalog cache size should be increased by tuning the parameter CATALOGCACHE_ SZ in the database configuration. The value may be low immediately following the first connection to the database. The execution of Data Definition Language (DDL) SQL statements involving a table,view, or alias will evict the table descriptor information for that object from the catalog cache causing it to be re-inserted on the next reference. Therefore, the heavy use of DDLs may also increase the value of the measure.
Package cache hit ratio:	The package and section information required for the execution of dynamic and static SQL statements are placed in the package cache as required. This information is required whenever a dynamic or static statement is being	Percent	If the hit ratio is high (more than 80%), the cache is performing well. A smaller ratio may indicate that the package cache size (pckcachesz) should be increased.

Measurement	Description	Measurement Unit	Interpretation
	executed. The ratio indicates the effectiveness of package cache hit ratio.		

7.2.3 Db2 DPF Sorts Test

Sorting represents organizing the rows in a table into the order of one or more of its columns, optionally eliminating duplicate entries. Sorting is required when:

- No index exists to satisfy a requested ordering (for example a SELECT statement that uses the ORDER BY clause).
- An index exists but sorting would be more efficient than using the index
- An index is created.
- An index is dropped, which causes index page numbers to be sorted.

Because queries often require sorted or grouped results, sorting is often required, and the proper configuration of the sort heap areas is crucial to good query performance. Using the Db2Sort test, administrators can figure out whether/not the sort heap allocations are sufficient to facilitate efficient sorting.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every logical partition of each database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate

user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.

5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Sort Heap Allocated:	Indicates the sum of sort heap space allocated for all sorts in this database partition.	Pages	Normal memory estimates do not include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required. You may use the information returned at the database manager level to help you tune the SHEAPTHRES configuration parameter. If the element value is greater than or equal to SHEAPTHRES, it means that the sorts are not getting the full sort heap as defined by the SORTHEAP parameter.
Average Sort Heap Space Used:	Indicates the average sort heap space used by each sort.	Pages	If the SORTHEAP configuration parameter is substantially larger than the average sort heap used, you may be able to lower the value of this parameter.

Measurement	Description	Measurement Unit	Interpretation
Sort Rate:	Indicates the number of sort operations performed on this logical partition during the last measurement period.	Sorts / Sec	
Percent Sort Overflow:	Indicates the percentage of sorts in this logical partition that had to overflow to disk.	Percent	Sort overflows are sorts that ran out of sort heap and may have required disk space for temporary storage. When a sort overflows, additional overhead will be incurred because the sort will require a merge phase and can potentially require more I/O, if data needs to be written to disk. If this percentage is high, you may want to adjust the database configuration by increasing the value of sortheap.
Average Sort Time:	Indicates the average sort time for all sorts performed by all applications connected to this logical partition.	Seconds	A high value indicates the poor performance of sorting operations. Identify the statements that spend lot of time sorting. You may want to reduce the average sort time for these statements by increasing the sortheap parameter.

7.2.4 Db2 Direct I/O Test

This test monitors the I/O activity on the currently active logical partitions.

Target of the test : A DB2 database server with DPF enabled

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every logical partition of each database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Direct Read Rate:	Indicates the total number of direct reads by the application per second. In other words, it is the number of read operations that do not use the buffer pool.	Reads/Sec	Direct reads are performed in units, the smallest being a 512- byte sector. They are used when: • Reading LONG VARCHAR columns • Reading LOB (large object) columns • Performing a backup A high value over a period of time may be indicative of a performance bottleneck.
Direct Write Rate:	Indicates the total number of direct writes	Writes/Sec	Direct writes are performed in units, the smallest being a 512-byte

Measurement	Description	Measurement Unit	Interpretation
	by the application per second. In other words, it is the number of write operations that do not use the buffer pool.		sector. They are used when: • Writing LONG VARCHAR columns • Writing LOB (large object) columns • Performing a restore • Performing a load A high value over a period of time may be indicative of a performance bottleneck.
Buffer Pool IO Rate:	Indicates the rate at which the buffer pool I/O operations are being done in this logical partition.	IOPS	In conjunction with the hit ratio statistics, and the characteristics of the applications executing, the I/O load may require adjustment of BUFFPAGE, or applications may require further tuning.

7.3 The Db2 Service Layer

The tests associated with the **Db2 Service** layer indicate the level of SQL activity on the server, and the number and type of transactions that occur on the server. These measurements together serve as effective indicators of the processing ability of the DB2 server.

Figure 7.7: The tests associated with the Db2 Service layer

7.3.1 Db2 DPF Activity Test

This test measures the level of SQL activity on the logical partitions, and reveals how well the partitions process SQL queries.

Target of the test : A DB2 database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every logical partition of each database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.

7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Dynamic SQL Rate:	Indicates the rate of dynamic SQL statements that were attempted on this logical partition.	Statements/sec	This is an indication of throughput of the system during the monitoring period. A high value of dynamic SQLs and low value of failed SQLs indicate good throughput.
Select SQL rate:	Indicates the rate at which SELECT SQL statements were executed on this logical partition during the last measurement period.	Statements/sec	This measure can be used to determine the level of database activity.
UID SQL rate:	Indicates the rate at which update/delete/insert statements were issued on this logical partition during the last measurement period.	Statements/sec	This measure can be used to determine the level of database activity.
Failed SQL Rate:	Indicates the rate at which SQL statements to this logical partition failed.	Statements/sec	A relatively high value indicates a problem. Failed SQL statements waste system resources. Hence, the value of this measure should be very low.
Percent Failed SQL:	Indicates the percentage of SQL statements attempted on this partion that failed during the interval. This value includes all SQL statements that received a negative SQLCODE	Percent	
Percent DDL SQL:	Denotes the percentage	Percent	This value should normally be

Measurement	Description	Measurement Unit	Interpretation
	of SQL statements attempted on this logical partition that were DDL (Data Definition Language) during the last measurement period.		low.
Percent UID SQL:	Indicates the percentage of update/insert/delete statements executed on this logical partition during the last measurement period.	Percent	This measure can be used to determine the level of database activity.

7.3.2 Db2 DPF Transactions Test

This test tracks various statistics pertaining to the transactions executing on each logical partition of a DB2 database.

Target of the test : A DB2 DPF server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every logical partition of each database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.

5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Commit Rate:	Indicates the transaction throughput. This measure is the sum of the committed statements attempted on this logical partition and internal commits (total number of commits initiated internally by the database manager) per second.	Commits/Sec	A decrease in this measure during the monitoring period may indicate that the applications are not doing frequent commits. This may lead to problems with logging and data concurrency. The cause has to be probed in the application.
Rollback Rate:	Indicates the rate of unit of work rollbacks.	Rollbacks/Sec	A high rollback rate is an indicator of bad performance, since work performed up to the rollback point is wasted. The cause of the rollbacks has to be probed in the application.
Transaction Rate:	Indicates the rate of commits and rollbacks for the application using the DB2 Connect gateway.	Transactions/Sec	A high transaction rate with high rollback rate indicates bad performance.

7.3.3 DB2 DPF SQL Workload Test

Nothing can degrade the performance of DB2 partition like a resource-hungry or a long-running query! When such queries execute on a logical partition, they either hog almost all the available

CPU, memory, and disk resources or keep the resources locked for long time periods, thus leaving little to no resources for carrying out other critical database operations. This can significantly slowdown the partition and adversely impact user experience with the partition. To ensure peak performance of the logical partitons at all times, such queries should be rapidly identified and quickly optimized to minimize resource usage. This is where the **DB2 DPF SQL Workload** test helps. At configured intervals, this test compares the usage levels and execution times of all queries that started running on the logical partitions in the last measurement period and identifies a 'top query' in each of the following categories - CPU usage, memory usage, disk activity, and execution time. The test then reports the resource usage and execution time of the top queries and promptly alerts administrators if any query consumes more resources or takes more time to execute than it should. In such a scenario, administrators can use the detailed diagnosis of this test to view the inefficient queries and proceed to optimize them to enhance server performance.

Target of the test : A DB2 DPF server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every logical partition of each database on the DB2 database server that is currently active

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test.
8. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the

detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

- The eG manager license should allow the detailed diagnosis capability
- Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum physical read rate	Indicates the number of physical disk reads performed by the top query per execution.	Seconds/read	If the value of this measure is abnormally high, you can use the detailed diagnosis of this measure to view the top- 5 (by default) queries generating maximum physical disk activity. From this, you can identify the top query in terms of number of physical disk reads. You may then want to optimize the query to reduce the disk reads.
Maximum physical write rate	Indicates the number of memory buffers used by the top query per execution.	Seconds/write	If the value of this measure is abnormally high, you can use the detailed diagnosis of this measure to view the top- 5 (by default) queries consuming memory excessively. From this, you can easily pick that query which is consuming the maximum memory. You may then want to optimize the query to minimize memory usage.
Maximum user	Indicates the CPU time used for user level processing upon	Seconds	If the value of this measure is over 30 seconds, you can use the detailed diagnosis of this measure

Measurement	Description	Measurement Unit	Interpretation
CPU time	execution of the top query.		to the top-5 (by default) queries hogging the CPU resources. From this, you can easily pick that query which is consuming the maximum CPU. You may then want to optimize the query to minimize CPU usage.
Maximum elapsed time	Indicates the running time of each execution of the top query.	Seconds	If the value of this measure crosses 10 seconds, you can use the detailed diagnosis of this measure to view the top-5 (by default) queries that are taking too long to execute. . From this, you can easily pick that query with the maximum execution time. You may then want to optimize the query to minimize execution time.
Maximum system CPU time	Indicates the CPU time used for system level processing upon execution of the top query.	Seconds	If the value of this measure is over 30 seconds, you can use the detailed diagnosis of this measure to the top-5 (by default) users consuming the maximum system CPU time. From this, you can easily pick that query which is consuming the maximum CPU. You may then want to optimize the query to minimize CPU usage.

7.3.4 DB2 DPF Backup Status Test

This test auto-discovers the logical partitions on the target DB2 database serverand for each logical partition, this test reports the statistics pertaining to the backups performed. Using this test, administrators can figure out the count of backups that failed and the count of backup jobs that were completed successfully. In addition, this test reports the maximum time taken to complete the backup jobs on each logical partition. By closely monitoring the measures reported by this test,

administrators can pin point the logical partition on which maximum number of backup jobs failed and further analyze the reason behind such failures.

Target of the test : A DB2 database server with DPF enabled

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every logical partition of each database on the DB2 database server

Configurable parameters for the test

1. **TEST PERIOD** – How often should the test be executed
2. **HOST** – The IP address of the DB2 server
3. **PORT** – The port number through which the DB2 server communicates. The default port is 50000.
4. **USER** - Specify the name of the user who has any of the following privileges to the specified **DATABASE**: SYSADM or SYSCTRL or SYSMAINT or SYSMON. You can create a separate user on the OS hosting the DB2 server for this purpose, and assign any of the aforesaid privileges to that user. The steps for the same are detailed in the Section **3.1.1.1**.
5. **PASSWORD** - Enter the password of the specified **USER** in the **PASSWORD** text box.
6. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it here.
7. **DATABASE** - Specify the name of the database on the monitored DB2 server to be used by this test
8. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:The eG manager license should allow the detailed diagnosis capabilityBoth the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total backup	Indicates the total number of backups performed for this logical partition during the last measurement period.	Number	This measure is a good indicator of backup load on the logical partition.
Completed backup	Indicates the number of backups completed successfully for this logical partition during the last measurement period.	Number	The detailed diagnosis of this measure lists the DB partition number EID, Start time, End Time, Duration (Minutes), Backup ID, Comment text, Number of tablespaces, Tablespace Names, Operation type, Object type, Location, Device type, SQL code, SQL warn and SQL state.
Failed backup	Indicates the number of backups failed for this logical partition during the last measurement period.	Number	Ideally, the value of this measure should be zero. Any value greater than zero, is a cause of concern, as it indicates the existence of a failed job. To know which job(s) has failed, use the detailed diagnosis capability of this measure. The detailed diagnosis of this measure lists the DB partition number EID, Start time, End Time, Duration (Minutes), Backup ID, Comment text, Number of tablespaces, Tablespace Names, Operation type, Object type, Location, Device type, SQL code, SQL warn and SQL state. Compare the value of this measure across the logical partitions to figure out the logical partition on which the maximum number of backup jobs failed.
Maximum duration of	Indicates the maximum	Minutes	Compare the value of this measure

Measurement	Description	Measurement Unit	Interpretation
completed backup	time taken to complete the backup jobs on this logical partition during the last measurement period.		across logical partitions to figure out the logical partition on which the backup took too long to be completed.

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.