OMi Management Pack for Microsoft SQL Server

Software Version: 1.01
For the Operations Manager i for Linux and Windows® operating systems

Reference Guide

Document Release Date: April 2017

Software Release Date: December 2015

Hewlett Packard Enterprise

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notice

Trademark Notices

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com/.

This site requires that you register for an HPE Passport and to sign in. To register for an HPE Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HPE Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support site at: https://softwaresupport.hpe.com/.

Most of the support areas require that you register as an HPE Passport user and to sign in. Many also require a support contract. To register for an HPE Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HPE Passport login page.

To find more information about access levels, go to: https://softwaresupport.hpe.com/web/softwaresupport/access-levels.

HPE Software Solutions Now accesses the Solution and Integration Portal website. This site enables you to explore HPE product solutions to meet your business needs, includes a full list of integrations between HPE products, as well as a listing of ITIL processes. The URL for this website is https://softwaresupport.hpe.com/km/KM01702731.

Contents

Chapter 1: OMi Management Pack for Microsoft SQL Server

The OMi Management Pack for Microsoft SQL Server (OMi MP for Microsoft SQL Server) works with Operations Manager i (OMi) that enables you to monitor Microsoft SQL Server database in your environments and its underlying infrastructure. It includes Indicators - Health Indicators (HIs), Event Type Indicators (ETIs) and Correlation Rules that analyze the events that occur in the Microsoft SQL Server databases and report the health status. It provides out-of-the-box Management Templates for monitoring different types of Microsoft SQL Server environments (standalone and cluster) and also includes capabilities to monitor the health and performance of the systems. These Management Templates consist of a wide range of Aspects which enable the monitoring of Microsoft SQL Server components and the system components.

These Management Templates can be deployed by administrators for monitoring Microsoft SQL Server databases in an environment. Subject Matter Experts (SMEs) and developers can customize the Microsoft SQL Server Management Templates.

The OMi MP for Microsoft SQL Server works with OMi and provides the following additional functionality to support a unified monitoring solution:

- Microsoft SQL Server instance-based deployment and configuration.
- Supports agent and agentless monitoring of Microsoft SQL Server instances.

OMi MP for Microsoft SQL Server Metrics

This chapter provides information about the Microsoft SQL Server collections, metrics, and data store tables which can be used to configure the data-collection procedure.

CheckpointpagesPersec

Description: This metric monitors the number of pages flushed by checkpoint or other operations that require all dirty pages to be flushed.

Collection interval: VERYHIGH

Policy: MSSQLServer_3415

Aspect: Microsoft SQL Server Buffer Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DeadlocksRate

Description: This metric monitors the Deadlock rate for each object type.

Collection interval: NORUN

Policy: MSSQLServer_3271

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: MAJOR / 3

Message Text: Deadlocks rate for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of lock requests per second has resulted in a deadlock for each object type: Extent, Key, Page, Table, RID, Database.

The probable causes could be:

- Two or more processes access data in different methods
- Size of a transaction is too large.

In any multi-use environment, occasional lock collisions are normal. Excessive lock collisions affect performance.

Performance may be affected since one of the deadlocked processes will be terminated by the server.

Suggested Action(s) : Deadlocks affect performance for two reasons. First, the deadlocked process needs to be rolled back. Second, it probably has to be done again.

Action depends on situation. You may need to restructure indexes or reschedule load processes when readers are not running, or make transactions shorter or smaller. Optimize the queries. This is often a process design problem. The automatic action report for this metric shows which users are connected to the SQL Server.

LongTransaction

Description: This metric monitors the Long running transaction.

Collection interval: LOW

Policy: MSSQLServer_3033

Aspect: Microsoft SQL Server Backup

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MINOR / 0.5

Message Text: <OPTION(count)> databases missing since last backup for more than or equal to <OPTION(hours_passed)> hours. [Policy: <NAME>]

Instruction Text:

This metric counts the number of databases which are not backed up since last backup for defined hours. If the defined hours indicates 876000, the number of databases have not been backed up for a long time.

Probable Cause(s) : A database backup has not been taken and it has exceeded the threshold time of backup.

Suggested Action(s) : Do a complete database backup.

DBSpaceUsedPct

Description: This metric monitors the percentage of database space used.

Collection interval: MEDIUM

Policy: MSSQLServer_3218

Aspect: Microsoft SQL Server Space

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: CRITICAL / 95

Message Text: % database space used (<VALUE>%) in database <OPTION(database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Percentage of space used in a database compared to the total size of the database.
- Database container is filled out.

Suggested Action(s) :

- Use ALTER DATABASE to increase the size of the data segment.
- Drop objects from the database.
- Delete rows from tables in the database.

- Add space to the database by executing ALTER DATABASE command. If there is no free space available on the existing database devices, create a new device using the DISK INIT command, or exit the device by running the DISK RESIZE command.

The automatic action report for this metric shows other database statistics through the sp_helpdb.

LockTimeoutRate

Description: This metric monitors the Lock timeout rate for each object type.

Collection interval: NORUN

Policy: MSSQLServer_3270

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 10

Message Text: Lock timeout rate for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of lock requests per second has timed out, including internal requests for NOWAIT locks for each object type: Extent, Key, Page, Table, RID, Database.

Locks are being held for a long duration indicates a locking contention problem. The processes do not complete properly and will abort.

Suggested Action(s) : Analyze the SQL code and look for unnecessary exclusive locks, holdlocks or overly long transactions. This is often a process design problem.

The automatic action report for this metric shows which users are connected to SQL Server.

UserConnectPct

Description: This metric monitors the percentage of users who are connected.

Collection interval: VERYHIGH

Policy: MSSQLServer_3011

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 98, MINOR / 90

Message Text: Rule1: % of current users connected (<VALUE>%) too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: % of current users connected (<VALUE>%) too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of current user connections is too high compared with the total number of user connections.

Suggested Action(s) : As the DBA, you can set a 'max' threshold for this metric based on a number of user connections that is normally not exceeded. So any 'connection leaks' in applications can be found before they begin to affect the performance of the server.

This metric determines open connections and not the connections that are working (processing queries, DML, and so on). See metric 3026 for percentage of connections that are active.

The automatic action report for this metric shows the maximum number of user connections allowed and which users are connected to the SQL Server.

A set of graphs (default) are launched from this event. See the 'Users' graph to understand about performance over a period of time.

ReadsOutstdRate

Description: This metric monitors the number of read requests issued to the operating system that are not complete.

Collection interval: VERYHIGH

Policy: MSSQLServer_3007

Aspect: Microsoft SQL Server Input and Output Utilization

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 2

Message Text: Reads outstanding rate (<VALUE>/min) too high (>=<THRESHOLD>/min) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of SQL Server 'read' requests that are too high compared with the Windows I/O subsystem requests that are not complete.

Suggested Action(s) : This helps you to identify I/O as a bottleneck. As I/O increases, disk speed will have a limit on the performance.

If this is a problem, it may be necessary to replace disks that are slower with faster ones. You can consider adding RAM (to increase the cache hit ratio) that would be more efficient than upgrading to a faster disk I/O subsystem.

This can be true, but also depends on the types of queries being issued and on how much RAM is already present in the system. If the system handles DSS-type of queries, which need to scan large amounts of data that are typically not held in cache, then the disk subsystem's throughput speed becomes a major factor in the overall performance of the server. In such a situation, increasing SQL Server's cache size typically does not benefit performance.

Beyond a certain total amount of RAM in the system, adding more RAM does not provide benefit because the time required to search that much cache RAM takes time required to read the data from disk. The break-even point, of course, depends on the relative speed of the CPU(s) in the system versus disk I/O time, but can occur at 1GB or less of total cache memory, and thus is something to be considered.

To put it in simple terms, upgrading a disk from 512MB RAM to 1GB can have positive effect on performance than upgrading the disk from 2GB to 2.5GB. If your server already has 2GB of RAM, you may need to look at other options.

Another possible solution is that the SQL 's background maintenance operations can cause extra I/O to occur. These include shrinking of databases and device files, and automatic execution of UPDATE STATISTICS on tables. You can disable these if you find it is causing performance hits on a production system.

The automatic action report for this metric shows the 'Input/Output' statistics and the users who are connected to the SQL Server.

You can launch a set of graphs (default) from this event. See the IO Utilization graph to know about the performance over a period of time.

ServiceMon

Description: This metric monitors the SQL Server service.

Collection interval: VERYHIGH

Policy: MSSQLServer_3058

Aspect: Microsoft SQL Server Availability

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Fault

Severity / Threshold: Common=>WARNING / 4.500000

Message Text: Rule1: <MSG_OBJECT> (SQL agent service) is having Status=<OPTION(Status)>. [Policy: <NAME>]

Message Text: Rule2: <MSG_OBJECT> (SQL agent service) is having Status=<OPTION(Status)>. [Policy: <NAME>]

Message Text: Rule3: <MSG_OBJECT> (SQL agent service) is having Status=<OPTION(Status)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

The SQL Server Agent service is not running. It can be in any of the following states:

SERVICE_START_PENDING, SERVICE_CONTINUE_PENDING,SERVICE_PAUSE_PENDING, SERVICE_STOP_PENDING,SERVICE_PAUSED or SERVICE_STOPPED,ERROR

Potential Impact: The SQL Server actions (mostly scheduled), which depend on the SQL agent, will fail.

Suggested Action(s) :

1. Verify that the SQL Agent service is running.
2. From the **Administrative Tools -> Services**, verify that the status of the particular SQL Agent service is Started.
3. If the SQL Agent service is not started, right-click the service and then click **Start** or **Resume**, as appropriate.
4. Please check the SQL Server Agent service status for the instance.

VirtDevSpUsdPct

Description: This metric monitors the percentage of space used on a specific virtual device.

Collection interval: MEDIUM

Policy: MSSQLServer_3215

Aspect: Microsoft SQL Server Space

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: CRITICAL / 99, MAJOR / 95, MINOR / 90

Message Text: Rule1: % of space used (<VALUE>%) on virtual device <OPTION(virtual_device)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: % of space used (<VALUE>%) on virtual device <OPTION(virtual_device)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule3: % of space used (<VALUE>%) on virtual device <OPTION(virtual_device)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Allocated storage has reached the total storage with the potential impact of the inability to extend databases.

Suggested Action(s) : If the device size limit has been reached, re-evaluate the initial space estimation. Allocate more disk and analyze why the disk filled up unexpectedly. Autogrow is a useful feature that prevents databases from being filled when the device is completely used. However, most administrators prefer a certain level of control in being able to plan for the database's next level of growth rather than letting SQL Server taking the control.

In other words, as a DBA, you must know when a database allocate disk space on its own, an alarm is generated. The alarm also alerts you when the autogrow settings are inadequate (causing the database to autogrow more often), database usage is expected to change or the remaining disk space changes. For instance, if the database is about to autogrow and completely fill up a disk drive, it might be more efficient to plan to create new file devices to store the database. You should always plan database systems and their device usages based on the amount of data being stored and the amount of load expected for the database, to plan its size and expected growth.

In other words, growth should be a planned event. Since there is always the unexpected to be dealt with, autogrow becomes a great assist in preventing a potential problem (the database stopping). However, it is possible that a database can grow so fast that the log will autogrow before anyone intervenes. This could be the result of an unexpected increase in usage of the database; or, it could be someone running an incorrect query. In the worst case, this could cause autogrow to execute not once but repeatedly until the disk fills up.

Another potential problem is that if the device files increase and decrease several times, disk fragmentation can result and this could affect the performance. It is recommended to pre-allocate databases to the specified size, so that autogrow is triggered rarely. So, to use autogrow on a database, this metric is not necessary to trigger a major alarm, but would be helpful to monitor it.

The automatic action report for this metric shows all virtual devices assigned to each database and the percentage of space used.

AvailableReplicaMsgByteSentPerSec

Description: This metric monitors the total number of database message bytes enqueued to be sent over the network to the replica.

Collection interval: HIGH

Policy: MSSQLServer_3432

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

ReadWriteErrCnt

Description: This metric monitors the number of SQL Server read/write errors since the last refresh.

Collection interval: MEDIUM

Policy: MSSQLServer_3023

Aspect: Microsoft SQL Server Error

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Fault

Severity / Threshold: WARNING / 0.5

Message Text: # of SQL Server read/write errors since the last probing (<VALUE>) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- The number of disk read or write errors encountered by SQL Server since the last probing is too high.
- Disk I/O failure

Suggested Action(s) : If the problem persists, the disk hardware has to be examined.

The automatic action report for this metric shows SQL Server information using the `sp_monitor` command.

By default, a set of graphs are launched from this event. See the Errors graph to understand about performance over a period of time.

LockMemoryPct

Description: This metric monitors the percentage of lock memory in use.

Collection interval: MEDIUM

Policy: MSSQLServer_3075

Aspect: Microsoft SQL Server Locks

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 37

Message Text: % of lock memory in use (<VALUE>%) too high (>=<THRESHOLD>%) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The SQL Server lock memory has reached its reconfiguration threshold.

Suggested Action(s) : Modify the number of locks available to a specified value using the sp_ configure command.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Locks and its Memory Utilization graph to understand about performance over a period of time.

PacketErrorCnt

Description: This metric monitors the number of packet errors while reading or writing packets.

Collection interval: MEDIUM

Policy: MSSQLServer_3024

Aspect: Microsoft SQL Server Error

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Fault

Severity / Threshold: WARNING / 0.5

Message Text: (<VALUE>) packet errors while reading or writing packets for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

There are network errors since the last reading.

This metric allows a DBA to monitor current user activity for SELECT queries.

Suggested Action(s) : Not Available

Use the 'sp_monitor' command to see the automatic action report for this metric that shows the SQL Server information.

By default, a set of graphs are launched from this event. See the Errors graph to understand about performance over a period of time.

CPUUsedPct

Description: This metric monitors the percentage of CPU time used by the SQL Server.

Collection interval: VERYHIGH

Policy: MSSQLServer_3025

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 95

Message Text: % CPU time used by SQL Server (<VALUE>% of <OPTION(NUM_CPUS)> CPU(s)) too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percentage reported is the number of seconds of CPU time used by SQL Server in ratio to the total amount of elapsed time since the last probing.

Cause: SQL Server CPU load running at 100% indicates a problem. Either SQL Server has excessive load or a thread is in the endless CPU loop.

This percentage is aggregated for all CPUs in the system.

Suggested Action(s) :

- Add CPUs to the server.
- For Runaway process, locate the thread that is causing a CPU loop and use the KILL command to kill it. If it does not help, restart the SQL Server.

Use the sp_monitor command to see which users are connected to the SQL Server information.

By default, a set of graphs are launched from this event. See the Server Status graph to understand about performance over a period of time.

DBLogShrinksCnt

Description: This metric monitors the number of transaction log shrinks per database.

Collection interval: NORUN

Policy: MSSQLServer_3267

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 10.0

Message Text: # of transaction log shrinks for <OPTION(database_name)> (<VALUE>) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of times the transaction log for each database has decreased are too high.

Suggested Action(s) : The administrator will have to disable the automated process and expand the log back to its original size.

The automatic action report shows a list of users who are connected to the SQL Server.

FileGrpSpaceFree

Description: This metric monitors the free space (MB) in each filegroup for each database.

Collection interval: NORUN

Policy: MSSQLServer_3279

Aspect: NA

CIT: NA

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_REPORT / DBSPI_MSS_REPORT

Message Category: MSSQLServer_Admin

Severity / Threshold: CRITICAL / 50.0, MAJOR / 100.0, MINOR / 150.0, CRITICAL / 50.0, MAJOR / 100.0, MINOR / 150.0

Message Text: Rule1: filegroup space MB free (<VALUE> MB) for transaction log filegroup in database <OPTION(database_name)> too low (<=<THRESHOLD> MB) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: filegroup space MB free (<VALUE> MB) for transaction log filegroup in database <OPTION(database_name)> too low (<=<THRESHOLD> MB) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule3: filegroup space MB free (<VALUE> MB) for transaction log filegroup in database <OPTION(database_name)> too low (<=<THRESHOLD> MB) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule4: filegroup space MB free (<VALUE> MB) for filegroup <OPTION(filegroup_name)> in database <OPTION(database_name)> too low (<=<THRESHOLD> MB) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule5: filegroup space MB free (<VALUE> MB) for filegroup <OPTION(filegroup_name)> in database <OPTION(database_name)> too low (<=<THRESHOLD> MB) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule6: filegroup space MB free (<VALUE> MB) for filegroup <OPTION(filegroup_name)> in database <OPTION(database_name)> too low (<=<THRESHOLD> MB) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Database filegroup is getting filled up.

Suggested Action(s) :

- Use ALTER DATABASE to increase the size of the filegroup or add a new filegroup.
- Delete objects from the database.
- Delete rows from tables in the database.
- Add space to the database by executing the ALTER DATABASE command.
- Create a new device using the DISK INIT command or increase the size of the database using the DISK RESIZE command if there is no free space available on the existing database devices.

The automatic action report for this metric will show statistics for each filegroup in the database.

LogshippingBackupJobs

Description: This metric monitors the backup job in primary instance of logshipping configuration.

Collection interval: HIGH

Policy: MSSQLServer_3291

Aspect: Microsoft SQL Server Logshipping

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 2.5, MAJOR / 1.5, CRITICAL / 0.5

Message Text: Rule1: Backup job is not enabled in the primary instance. Primary ID = <OPTION (primary_id)>, Primary DB name = <OPTION(primary_db)>, Enabled= <OPTION(enabled) > [Policy: <NAME>]

Message Text: Rule2: Last execution of a backup job is success, but backup job is not running as per the scheduled time span. Primary ID = <OPTION(primary_id)>, Primary DB name = <OPTION (primary_db)>, Elapsed time = <OPTION(elapsed_time) seconds > [Policy: <NAME>]

Message Text: Rule3: Last execution of backup Job is failed in primary instance. Primary ID = <OPTION(primary_id)>, Primary DB name = <OPTION(primary_db)>, Detailed Error= <OPTION (detailed_error) > [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Log shipping is not configured properly.
- Log backup folder might not be accessible.
- Permission denied or not found.
- No space left on the device.
- Server may not be the primary. Log shipping job may not be enabled and scheduled properly.
- The date or time (or both) on the primary server is modified such that the date or time on the primary server is significantly ahead between consecutive transaction log backups

Suggested Action(s) :

1. Check agent log and logshipping monitor information.
2. Check the job history. Restart the SQL Server Agent, if required.
3. Check the configuration of the backup job in primary instance.
4. Check the properties of the shared backup folder in the primary and secondary servers.

MemoryMenagerStolenServerMemory

Description: This metric monitors the amount of memory the server is currently using for purposes other than monitoring the database pages.

Collection interval: VERYHIGH

Policy: MSSQLServer_3423

Aspect: Microsoft SQL Server Memory and Memory Manager (Add-on)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

> **Note:** This metric is supported on SQL Server version 2012 onwards.

PrincipalSysadminCnt

Description: This metric counts the number of principals who are members of the sysadmin fixed server role (M3090_PrincipalSysadminCnt).

Collection interval: LOW

Policy: MSSQLServer_3090

Aspect: Microsoft SQL Server Audit

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 10

Message Text: Number of principals having system admin role (<VALUE>) goes beyond threshold value (<THRESHOLD>) for <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

This metric counts the number of principals who are members of the sysadmin fixed server role. SQL Server depends on role-based security to manage permissions. If multiple IT system administrators have permissions to set up new SQL Server logins, they might be inclined to do so as part of the sysadmin role. Adding a normal user to the sysadmin role could pose a security risk and is not recommended unless the principal is highly trusted. If this count is higher than expected, contact system administrators. The number of principals can be retrieved from the server_role_members and server_principls tables.

Probable Cause(s) :

- There could be a security risk if the sysadmin role is assigned to a normal user.
- If multiple IT system administrators have permissions to set up new SQL Server logins, they might be inclined to do so as part of the sysadmin role.

Potential Impact: Unauthorized access of server will affect the server security.

Unauthorized access of server could lead to failure of system.

Suggested Action(s) :

1. Manage permissions on SQL server.
2. Add additional checks for server security.
3. Validate the role of the users having 'sysadmin' role.

Metric Desc:

1. This metric counts the number of principals who are members of the sysadmin fixed server role.

2. Adding a normal user to the sysadmin role could pose a security risk and is not recommended unless the principal is highly trusted.

MemoryManagerTargetServerMemory

Description: This metric monitors the ideal amount of memory the server is willing to consume.

Collection interval: HIGH

Policy: MSSQLServer_3425

Aspect: Microsoft SQL Server Memory and Memory Manager (Add-on)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DatabaseReplicaLogSndQueue

Description: This metric monitors the amount of logs (in KB) that is waiting to be sent to the database replica.

Collection interval: HIGH

Policy: MSSQLServer_3441

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

SuspectDBCnt

Description: This metric monitors the number of databases marked as suspect.

Collection interval: MEDIUM

Policy: MSSQLServer_3028

Aspect: Microsoft SQL Server Availability

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Fault

Severity / Threshold: CRITICAL / 0.5

Message Text: <VALUE> databases marked as suspect for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : If the SQL Server cannot access a database device or recover a database at startup, it marks this database as 'suspect'. The 'suspect' status prevents users from accessing the database.

One or more databases on SQL Server have been marked as 'suspect'.

Suggested Action(s) :

Use the sp_resetstatus stored procedure to turn off the suspect flag on a database leaving all other database options intact.

Caution: Use sp_resetstatus only when directed by your primary support provider or this manual. Otherwise, you might damage your database. If the suspect database is damaged and can not be recovered, remove the database using DBCC DBREPAIR: dbcc dbrepair(database_name,dropdb).

The automatic action report for this metric shows other database statistics through the sp_helpdb.

By default, a set of graphs are launched from this event. See the Errors graph to understand about performance over a period of time.

ServerConnect

Description: This metric monitors the ability to connect to a server (M097_ServerConnect).

Collection interval: VERYHIGH

Policy: MSSQLServer_3097

Aspect: Microsoft SQL Server Availability

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

DatabaseReplicaRermainingRedonBytePerSec

Description: This metric monitors the amount of log bytes remaining to be redone to complete the reverting phase received by the availability.

Collection interval: HIGH

Policy: MSSQLServer_3436

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

AvailableReplicaMsgEnqSentPerSec

Description: This metric monitors the total number of messages enqueued to be sent over the network to the replica.

Collection interval: HIGH

Policy: MSSQLServer_3430

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

BlokingSesions

Description: This metric monitors the existence of the processes that block a resource because of multiple sessions trying to access the same resource with lock-based concurrency (M3095_ BlokingSesions).

Collection interval: VERYHIGH

Policy: MSSQLServer_3095

Aspect: Microsoft SQL Server Processes and Statistics (Add-on)

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: CRITICAL / 0.5

Message Text: <VALUE> sessions blocked for longer time (<OPTION(wait_time)> seconds) which is more than threshold (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Blocking Sessions are processes that block a resource because of multiple sessions trying to access the same resource in relational database management system (RDBMS) with lock-based concurrency. Blocking occurs when one session holds a lock on a specific resource and a second session attempts to acquire a conflicting lock type on the same resource. Blocking is an unavoidable characteristic of any relational database management system (RDBMS) with lock-based concurrency. However, too much blocking can cause performance issues (M095_BlokingSesions).

Probable Cause(s) :

When locking and blocking increase to the point where there is a considerable impact on the system performance, it is usually due to one of the following reasons:

- A session holds locks on a set of resources and never releases them. This is usually caused by canceled queries that are not rolled back and orphaned transactions
- A session (identified by a session_id or “SPID”) holds locks on a set of resources for an extended period of time before releasing them. This is usually caused by long-running transactions, lack of appropriate indexes, inappropriate use of locking hints, and other issues related to poor application design.

Potential Impact: Performance issues caused because of longer lock periods.

Suggested Action(s) :

- Kill the session at the start of the blocking chain.
- Shorten transaction times.
- Create proper indexes.
- Use locking hints. See SQL Server Books Online.
- Use row versioning-based Isolation levels.
- Configure SQL Server settings (memory settings, lock timeouts and so on).
- Change the thresholds on the monitor for this specific database or all databases
- Disable the monitor for this specific database or all databases if blocked sessions are not a concern for the database.

DBLogGrowthsCnt

Description: This metric monitors the number of transaction log expansions for each database.

Collection interval: NORUN

Policy: MSSQLServer_3266

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 10.0

Message Text: # of transaction log expansions for <OPTION(database_name)> (<VALUE>) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Number of times the transaction log for each database has been increased is too high.
- SQL Server has run out of space for the transaction log.
- A system administrator has increased the log to provide more space.

Suggested Action(s) : This depends on whether expansion is required. If the expansion is because of increased activity on the database, database will continue to expand.

The changes may be retained or the frequency of backup transaction commands may need to be increased. If the changes are due to a single unique event, an administrator may consider truncating the log and shrinking the database back to its original size.

The automatic action report for this metric shows which users are connected to the SQL Server.

ServiceMon

Description: This metric monitors the SQL Agent Service.

Collection interval: VERYHIGH

Policy: MSSQLServer_3057

Aspect: Microsoft SQL Server Availability

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Fault

Severity / Threshold: Common=>WARNING / 4.500000

Message Text: Rule1: <MSG_OBJECT> (SQL service) is having Status=<OPTION(Status)>. [Policy: <NAME>]

Message Text: Rule2: <MSG_OBJECT> (SQL service) is having Status=<OPTION(Status)>. [Policy: <NAME>]

Message Text: Rule3: <MSG_OBJECT> (SQL service) is having Status=<OPTION(Status)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : SQL Server service is not running. It can be in any of the following states: `SERVICE_START_PENDING, SERVICE_CONTINUE_PENDING,SERVICE_PAUSE_PENDING, SERVICE_STOP_PENDING, SERVICE_PAUSED or SERVICE_STOPPED, ERROR`.

Potential Impact: Applications, which are trying to connect to the SQL Server, will fail.

Suggested Action(s) : Verify that the SQL Server service is running. From **Administrative Tools -> Services**, verify that the status of the particular SQL Server service is started. If it is not, right-click the service and then click **Start** or **Resume**.

Please check the SQL Server service status for the instance.

BatchReqstsRate

Description: This metric monitors the batch requests rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3074

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 500

Message Text: Batch requests rate (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Number of Transact-SQL command batches received per second are very high
- There is a increase in the number of requests on the server.

Suggested Action(s) : Check if the batch requests rate is high.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Server Status graph to understand about performance over a period of time.

SqlAdtSchMod

Description: This metric monitors the modification of database schema (M3093_SchemaMod).

Collection interval: HIGH

Policy: MSSQLServer_3093

Aspect: Microsoft SQL Server Audit

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> databases have modified schema since yesterday for <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

This metric is not important and viewing the data points on the Analysis page will not provide useful information. Use this metric to create an alert that will be raised when data is added to the schema or the existing schema is modified.

Spikes indicate that something has been added to the database schema or the existing schema has been modified. A spike will not occur if a schema object is deleted.

When you retrieve the modified date from the sys.objects table, the schema is modified.

Probable Cause(s) : The database schema is modified either by the user or application.

Potential Impact: It impacts system performance and Data integrity.

Suggested Action(s) : Validate the change in database schema.

Note: Add db_datareader role to agent user for all databases using “exec sp_addrolemember ‘db_datareader’, '<agent user>’”.

SecondaryReplicaFailoverReady

Description: This monitor checks whether the availability group has at least one secondary replica which is failover ready. (M3514_FailoverReady)

Collection interval: HIGH

Policy: MSSQLServer_3514

Aspect: Microsoft SQL Server High Availability (DR)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> secondary replica(s) are not set as automatic failover whereas primary replica is set as automatic failover for <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

Metric returns 'failover_mode' of the secondary replica which is failover ready (M3514_FailoverReady).

Probable Cause(s) : The availability group is not ready for automatic failover. The primary replica is configured for automatic failover; however, the secondary replica is not ready for automatic failover. The secondary replica that is configured for automatic failover might be unavailable or its data synchronization state is currently not `SYNCHRONIZED`.

Potential Impact: The primary replica is not failover ready.

Suggested Action(s) :

- Check if at least one secondary replica is configured as automatic failover. Otherwise, update the configuration of a secondary replica to be automatic failover target with synchronous commit.
- Check if the automatic failover target replica’s data synchronization state is `SYNCHRONIZED` through its monitor and resolve the issue at the availability replica.

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

DatabaseReplicaBlockedRedoPerSec

Description: This metric monitors the number of times redo is blocked.

Collection interval: HIGH

Policy: MSSQLServer_3439

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

IndxSearchsRate

Description: This metric monitors the Index searches rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3052

Aspect: Microsoft SQL Server Data Access Methods

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 1000

Message Text: Index searches rate (<VALUE>/sec) too high (>=<THRESHOLD>) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of index searches per second.

These are used to start range scans and single index record fetches and to reposition an index. In general, the number of Indexes are recommended to be high.

This means that more searches are being performed using indexes rather than full scans which is preferable.

However, on large databases where the data is constantly changing, this value may start to decrease and the full scan value may start to increase.

Suggested Action(s) : Update the statistics for the affected tables.

Use Enterprise Manager to reschedule when statistics samples are updated. They may not be occurring frequently or they may not be scheduled at all. In this case set 'auto update statistics' database option to ON on all your databases.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Data Access graph to understand about performance over a period of time.

DBLogShrinksCnt

Description: This metric monitors the number of transaction log shrinks for the server.

Collection interval: MEDIUM

Policy: MSSQLServer_3067

Aspect: Microsoft SQL Server Transactions

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 4.0

Message Text: # of transaction log shrinks for the entire server (<VALUE>) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of times the transaction log for has been decreased for the server instance. SQL Server has automated processes for many administration tasks. Among them are tasks to shrink a database in a specified percent of the allocated space is unused.

Suggested Action(s) : This depends on whether expansion is important. If the shrinkage was not intended, the administrator will have to disable the automated process and expand the log back to its original size.

The automatic action report for this metric shows which users are connected to SQL Server.

FullTextSearchInsufficientDiskSpace

Description: This metric monitors the count of databases with insufficient disk space for the Full Text Search Operation.

Collection interval: HIGH

Policy: MSSQLServer_3516

Aspect: Microsoft SQL Server Full-Text Search

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 0.5

Message Text: <VALUE> databases (>=1) have insufficient disk space (<OPTION(disk_space)>% used) for full text search on <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

This metric returns count of the databases whose used space is above the threshold.

Running out of disk space in can cause significant disruptions in the SQL Server production environment and can

prohibit applications that are running from completing operations.

Probable Cause(s) :

The size of a full-text catalog is usually less than 50 percent of the size of the raw data, that is, of all full-text indexable data in the catalogs. However, the master-merge process requires additional space.

Potential Impact: Significant disruptions in the SQL Server production environment.

Suggested Action(s) :

- Determine the total data size of the full-text indexed columns in all indexes that belong to the catalog.
- Allocate space for 100 percent of the raw data size for the full-text catalog, to further accommodate the master-merge process.
- Use the operating system cleanup utilities to reclaim disk space.

Note: Add db_datareader role to agent user for all databases using “exec sp_addrolemember ‘db_datareader’, '<agent user>’”.

RepnLatency

Description: This metric monitors the Replication Latency.

Collection interval: HIGH

Policy: MSSQLServer_3082

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 60.0

Message Text: Replication Latency <VALUE> for the Log Reader Agent with agent_id <OPTION (agent_id)> too high (>=<THRESHOLD>) for the publisher <OPTION(publisher)> [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Replication latency for one or more publishers is high.
- Delay in distribution of commands which are marked for replication. There may be some issues in the log reader or distribution agent.

Suggested Action(s) : Check for the distribution of commands which are marked for replication.

RepnAgentsStatus

Description: This metric monitors the status of the Replication Agent.

Collection interval: VERYHIGH

Policy: MSSQLServer_3081

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 0.5

Message Text: One or more replication agents failed. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- One or more Replication agents have failed.
- Server is not available due to deadlock, connection failure or time-out failure.

Suggested Action(s) :

- Check both the agent history and job history to see if issues have occurred around the same time.
- Verify the basic connectivity is working between the computers accessed by the agent.

The automatic action report for this metric will show the list of agents which are failed.

DistDeliveredTransPersec

Description: This metric monitors the number of transactions delivered to the Subscriber per second.

Collection interval: HIGH

Policy: MSSQLServer_3402

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DatabaseReplicaSyncState

Description: This monitor checks the data synchronization state of database replica (M3511_SyncStateOfDB).

Collection interval: HIGH

Policy: MSSQLServer_3511

Aspect: Microsoft SQL Server High Availability (DR)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> database replica(s) are not synchronized for <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

This metric shows data synchronization health state of the replica database. A value of `NOT SYNCHRONIZE` indicates the health state is not healthy and a value of `SYNCHRONIZED` indicates the health state is healthy.

This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

Probable Cause(s) : The data synchronization state of this availability database is unhealthy. On an asynchronous-commit availability replica, every availability database should be in the `SYNCHRONIZING` state. On a synchronous-commit replica, every availability database should be in the `SYNCHRONIZED` state.

- Availability replica might be disconnected
- The data movement might be suspended.
- The database may not be accessible.

- It might be a temporary delay issue due to network latency or the load on primary or secondary replica.

Potential Impact: Replica when needed may not be available. Unsynchronized database replica will impact the availability of accurate data.

Suggested Action(s) :

- Resolve any connection or data movement suspend issue.
- Check events for the issue using SSMS, look for database errors, and follow the troubleshooting for the specific error to resolve it.

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

AutoCloseSetting

Description: This metric monitors the Auto Close setting for the databases. If the settings do not meet the specified standard, an alert is generated to avoid performance issues caused by recompilation of all subsequent execution plans from plan cache which are cleared.

Collection interval: HIGH

Policy: MSSQLServer_3501

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> databases with 'is_auto_close_on' setting enabled for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Monitors the Auto Close setting for this database. If settings do not meet the specified standard, an alert is generated. Expected setting for this monitor is `AUTO_CLOSE = OFF`. The `AUTO_CLOSE` option is useful for desktop databases because it allows for database files to be managed as regular files. They can be moved, copied to make backups or even sent to other users using email. However, when the database is set to `AUTOCLOSE = ON`, an operation that initiates an automatic database shutdown clears the plan cache for the instance of SQL Server. Clearing the plan cache causes a recompilation of all subsequent execution plans and can cause a sudden, temporary decrease in query performance. Database mirroring requires `AUTO_CLOSE OFF`.

Probable Cause(s) : A warning alert is sent if the **Auto Close** option does not match the required setting. The monitor is configured to trigger an alert when this setting is set to `ON`.

Potential Impact: Performance issues caused by recompilation of all subsequent execution plans from plan cache which are cleared.

Suggested Action(s) : This issue may be resolved by:

- Changing the configuration setting for this database to match the expected value.
- Overriding the expected value for this unit monitor for this specific database or all databases.

Alternatively, if this monitor is not of concern for this database:

- Disabling the monitor using overrides for this specific database or all databases.
- Disabling the top-level aggregate configuration monitor using overrides for this specific database or all databases.

PhysIOByUsrPct

Description: This metric monitors the percentage of physical I/O used by a process ID.

Collection interval: VERYHIGH

Policy: MSSQLServer_3227

Aspect: Microsoft SQL Server Input and Output Utilization

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 95

Message Text: % of physical I/O used by process ID <OPTION(user_name)> (<VALUE>%) too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : This metric excludes SQLAgent processes.

The cumulative physical reads and writes (as a percentage) by a process is too high compared with all physical reads and writes by all the SQL Server users.

If a process performs heavy I/O activities then this may cause performance problems for other users trying to run their queries.

Suggested Action(s) : If this situation causes a significant performance degradation then that process should be closed.

The automatic action report for this metric shows which users are connected to SQL Server.

FileGrpUsedSpacePct

Description: This metric monitors the percentage of space used per filegroup and database.

Collection interval: HIGH

Policy: MSSQLServer_3278

Aspect: Microsoft SQL Server Space

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: CRITICAL / 99.0, MAJOR / 95.0, MINOR / 90.0, CRITICAL / 99.0, MAJOR / 95.0, MINOR / 90.0

Message Text: Rule1: % filegroup space used (<VALUE>%) for transaction log filegroup in database <OPTION(database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: % filegroup space used (<VALUE>%) for transaction log filegroup in database <OPTION(database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)> [Policy: <NAME>]

Message Text: Rule3: % filegroup space used (<VALUE>%) for transaction log filegroup in database <OPTION(database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule4: % filegroup space used (<VALUE>%) for filegroup <OPTION(filegroup_name)> in database <OPTION(database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule5: % filegroup space used (<VALUE>%) for filegroup <OPTION(filegroup_name)> in database <OPTION(database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule6: % filegroup space used (<VALUE>%) for filegroup <OPTION(filegroup_name)> in database <OPTION(database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Percentage of space available in each filegroup for each database.

Suggested Action(s) :

- Use ALTER DATABASE to increase the size of the filegroup, or add a new filegroup.
- Drop objects from the database.
- Delete rows from tables in the database.
- Add space to the database by executing the ALTER DATABASE command. Create a new database, using the DISK INIT command or increase space on an existing database by running the DISK RESIZE command.

The automatic action report for this metric will show statistics for each filegroup in the database.

AvailabilityReplicaConnState

Description: This monitor checks the connection state between availability replicas (M3515_ReplicaConnState).

Collection interval: HIGH

Policy: MSSQLServer_3515

Aspect: Microsoft SQL Server High Availability (DR)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> secondary replica(s) are not connected to the primary for <OPTION (dbserv)> [Policy: <NAME>]

Instruction Text:

Metric returns Connection state between availability replica. It is unhealthy when connection state is disconnected else it is healthy. (M3515_ReplcRlSt)

Probable Cause(s) :

- This secondary replica is not connected to the primary replica database. The connected state is DISCONNECTED.
- Connection port might be conflict with other application.
- There is a mismatch in the encryption type or algorithm.
- Connection endpoint is deleted or not started.
- Transport is disconnected.

Potential Impact: Availability replicas are disconnected.

Suggested Action(s) :

- Check if the database mirroring endpoint configuration is of primary replica and secondary replica instances. Update the configuration mismatch.
- Check if the port is in conflict with another application and change the port number.

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

DatabaseReplicaJoinState

Description: This monitor checks the join state of database replica (M3512_JoinStateofDB).

Collection interval: VERYHIGH

Policy: MSSQLServer_3512

Aspect: Microsoft SQL Server High Availability (DR)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> database replica(s) are not joined for <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

This metric shows the `JOIN` state of database replica. When the Replica database is not in the JOIN state, it is considered as Unhealthy. When the Replica database is in the JOIN state, it is considered as Healthy (M3512_JoinStateofDB).

Probable Cause(s) : The secondary database is not joined to the availability group. The configuration of the secondary database may be incomplete.

Potential Impact: Secondary database may be unavailable.

Suggested Action(s) : Join the database replica using T-SQL, SQLPS cmdlet or SSMS UI.

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

UploadedChangesPersec

Description: This metric monitors the number of rows per second replicated from the Subscriber to the Publisher.

Collection interval: HIGH

Policy: MSSQLServer_3406

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

CurAvgLatchWait

Description: This metric monitors the current average latch wait time.

Collection interval: MEDIUM

Policy: MSSQLServer_3076

Aspect: Microsoft SQL Server Latches

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 500

Message Text: Current average latch wait time (<VALUE> milliseconds) too high (>=<THRESHOLD> milliseconds) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

The average latch wait time (in milliseconds) for latch requests that had to wait during the current collection interval is too high. By default, it is 1 hour. See M069 for the average latch wait time since the server was started.

Many users try to access the same row at the same time. This is a performance bottleneck.

Suggested Action(s) : Analyze the database design, process design, and coding.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Latches graph to understand about performance over a period of time.

SQLServerCompilation

Description: This metric tracks the number of compilations per second. Indicates the number of times the compile code path is entered. Includes compiles caused by statement-level recompilations in SQL Server.

Collection interval:

Policy: MSSQLServer_3427

Aspect: Microsoft SQL Server Processes and Statistics (Add-on)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

ActiveConntnPct

Description: This metric monitors the percentage of total connections that are active compared to inactive.

Collection interval: VERYHIGH

Policy: MSSQLServer_3026

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 20

Message Text: % of total connections that are active vs sleeping (<VALUE>%) too low (<=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of current active connections that are too low as a percent of total connections that are active versus inactive.

If this metric's value is consistently low, SQL Server resources may be tied to a large number of idle, 'sleeping' connections.

Suggested Action(s) : If this metric is consistently low, you may want to adjust your 'front end' applications so that they do not keep idle connections for too long. Example: By default, MSAccess keeps idle connections for 10 minutes. This value can be adjusted to 3 or 5 minutes.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Users graph to understand about performance over a period of time.

ReportsFailed

Description: This metric monitors the number of reports (Reporting Services) that have failed.

Collection interval: MEDIUM

Policy: MSSQLServer_3080

Aspect: Microsoft SQL Server Reports

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MINOR / 1

Message Text: Number of reports failed <VALUE> too high (>=<THRESHOLD>) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : There may be a number of possible causes:

- DataSource is not configured properly.
- Report generation takes a long time.
- Schedule used to trigger the report has expired.
- The report is undeliverable (it is too big).
- The delivery extension specified in the subscription has been uninstalled or disabled.
- The credential settings changed from integrated to stored or prompted values.
- The parameter name or datatype is changed in the report definition and the report was republished. If the subscription includes a parameter that is no longer valid, the subscription becomes invalid.

Suggested Action(s) : You can do the following:

- Check the Application Log for errors related to Reporting Service.
- Ensure that the subscription is active.
- The credentials used to run the report are valid.
- Datasource used to connect to the DB is working.

SQLServerReCompilation

Description: For the SQL Recompilation, create a new query execution plan that will be optimal for the new database state and ensure better procedure performance.

Collection interval: VERYHIGH

Policy: MSSQLServer_3426

Aspect: Microsoft SQL Server Processes and Statistics (Add-on)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 10

Message Text: <VALUE> SQL Recompilations which is more than specified threshold (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

When a batch or remote procedure call (RPC) is submitted to SQL Server, before it begins executing the server checks for the validity and correctness of the query plan. If one of these checks fail, the batch may have to be compiled again to produce a different query plan. Such compilations are known as recompilations. These recompilations are necessary to ensure correctness and are often performed when the server determines that there could be a more optimal query plan due to changes in underlying data. Compilations by nature are CPU intensive and hence excessive recompilations could result in a CPU-bound performance problem on the system.

Probable Cause(s) : High number of recompilations as compared to compilations. Recompilation can happen due to various reasons, such as:

- Schema is changed
- Statistics are changed
- Deferred compile
- SET option is changed
- Temporary table is changed
- Stored procedure is created with the RECOMPILE query hint or which uses OPTION (RECOMPILE).

Potential Impact: SQL query/SP performance issues

Suggested Action(s) :

- You can use System Monitor (PerfMon) or SQL Trace (SQL Server Profiler) to detect excessive compilations and recompilations.
- Search for the following key data counters:
  - SQL Server: SQL Statistics: Batch Requests/sec
  - SQL Server: SQL Statistics: SQL Compilations/sec
  - SQL Server: SQL Statistics: SQL Recompilations/sec
- Start SQL Profiler and check SQL:StmtRecompile trace to identify root cause of issue.

LocksInUsePct

Description: This metric monitors the percentage of locks in use.

Collection interval: VERYHIGH

Policy: MSSQLServer_3013

Aspect: Microsoft SQL Server Locks

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Conf

Severity / Threshold: MINOR / 80

Message Text: % locks in use (<VALUE>%) too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percentage of total locks currently in use are too high compared with the total number of locks configured for SQL Server.

Suggested Action(s) :

The number of locks configured is fixed at server start-up.

When the number of locks are met, other processes requesting locks abort, and new users cannot connect. Evaluate why this is occurs, then increase number of available locks (sp_configure 'locks'), and reboot server for change to take effect.

Note: This uses memory.

The automatic action report for this metric will show all outstanding locks and which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Locks and its Memory Utilization graph to understand about performance over a period of time.

DatabaseReplicaLogBytRec

Description: This metric monitors the amount of logs received by the availability replica for the database.

Collection interval:

Policy: MSSQLServer_3442

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

DatabaseReplicaRecQueue

Description: This metric monitors the number of hardened log in kilobytes that is waiting to be redone on the secondary.

Collection interval: HIGH

Policy: MSSQLServer_3440

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

TransLogUsedPct

Description: This metric monitors the percentage of transaction log space used for each database.

Collection interval: HIGH

Policy: MSSQLServer_3216

Aspect: Microsoft SQL Server Space

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: CRITICAL / 99, MAJOR / 90, MINOR / 80

Message Text: Rule1: % of transaction log space used (<VALUE>%) in database <OPTION (database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: % of transaction log space used (<VALUE>%) in database <OPTION (database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule3: % of transaction log space used (<VALUE>%) in database <OPTION (database_name)> too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Percentage of space used in database transaction log to the total log size. When the transaction processing reaches 100%, the transaction processing will abort or suspend based on the database settings.

Suggested Action(s) : You can do one of the following:

- Periodically dump (truncate) the transaction log. If you do not do it periodically, the log will grow unchecked until it fills up. You can either turn on the database option (using sp_dboption) 'trunc. log on chkpt.', which is NOT recommended on a production machine, or you can dump the transaction log, which cleans the completed transactions.
- Automate cleaning the transaction log on a periodic basis before the threshold is reached.

  If you are cleaning the transaction log and it does not reduce the size, you have a long-running transaction which is not complete, or you may have an aborted transaction in the database which has not been marked as complete in the log.

- Shut down and restart the server. This will mark the incomplete transaction as complete (and rolled back).

Autogrow is a useful feature that prevents databases from being filled when the device is completely used. However, most administrators prefer a certain level of control in being able to plan for the database's next level of growth rather than letting SQL Server taking the control.

In other words, as a DBA, you must know when a database allocate disk space on its own, an alarm is generated. The alarm also alerts you when the autogrow settings are inadequate (causing the database to autogrow more often), database usage is expected to change or the remaining disk space changes. For instance, if the database is about to autogrow and completely fill up a disk drive, it might be more efficient to plan to create new file devices to store the database. You should always plan database systems and their device usages based on the amount of data being stored and the amount of load expected for the database, to plan its size and expected growth.

In other words, growth should be a planned event. Since there is always the unexpected to be dealt with, autogrow becomes a great assist in preventing a potential problem (the database stopping). However, it is possible that a database can grow so fast that the log will autogrow before anyone intervenes. This could be the result of an unexpected increase in usage of the database; or, it could be someone running an incorrect query. In the worst case, this could cause autogrow to execute not once but repeatedly until the disk fills up.

Another potential problem is that if the device files increase and decrease several times, disk fragmentation can result and this could affect the performance. It is recommended to pre-allocate databases to the specified size, so that autogrow is triggered rarely. So, to use autogrow on a database, this metric is not necessary to trigger a major alarm, but would be helpful to monitor it.

The automatic action report for this metric shows log space (in MB) and percentage used per database and other database statistics using the sp_helpdb command.

TransactionRate

Description: This metric monitors the server transaction rate.

Collection interval: VERYHIGH

Policy: MSSQLServer_3009

Aspect: Microsoft SQL Server Transactions

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 100

Message Text: Server transactions rate (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Rate of transactions for the entire server is too high.

This number increases with server modification statements.

Suggested Action(s) : As this increases, you need to determine whether you are CPU or IO bound. If you are CPU bound, add more processors. If you are IO bound, it's time to revisit the I/O subsystem or add RAM. If this corresponds with a new release, consider optimizing queries. You can consider replication (allocate data between two or more servers). Also check to see if the procedure cache is stressed.

The automatic action report for this metric shows I/O statistics and which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Transactions graph to understand about performance over a period of time.

DBActivTransCnt

Description: This metric monitors the number of active transactions for the each database.

Collection interval: NORUN

Policy: MSSQLServer_3264

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 5

Message Text: # of active transactions for the database <OPTION(database_name)> (<VALUE>) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of active transactions for each database in the server has increased.

Suggested Action(s) :

- Optimize queries
- Upgrade server hardware or migrate part of the data to a separate server.

The automatic action report for this metric shows active transactions by database and which users are connected to SQL Server.

SnapshotDeliveredCmdsPersec

Description: This metric monitors the number of Snapshot commands per second delivered to the Distributor.

Collection interval: HIGH

Policy: MSSQLServer_3407

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DBPageVerifySetting

Description: This metric monitors the number of databases with setting as 'page_verify_option' is not equal to CHECKSUM.

Collection interval: HIGH

Policy: MSSQLServer_3507

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

TransactionRate

Description: This metric monitors the database transaction rate.

Collection interval: NORUN

Policy: MSSQLServer_3209

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 100

Message Text: Database transactions rate for <OPTION(database_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Rate of transactions for each database is too high. This number increases with server modification statements.

Suggested Action(s) :

Determine whether you are designing a CPU or IO bound application. If you are designing a CPU bound application, add more processors. If you are designing a IO bound application, revisit the I/O subsystem or add RAM. If this corresponds with a new release, consider optimizing queries. You can consider replication (allocate data between two or more servers).

Also check to see if the stored procedure cache is stressed using the SQLQueryStress tool.

The automatic action report for this metric shows I/O statistics and which users are connected to SQL Server.

UserConnect

Description: This metric monitors the user connections count (M031_NumUsersCnt)

Collection interval: VERYHIGH

Policy: MSSQLServer_3031

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

MemoryAvailableMBytes

Description: This metric monitors the amount of physical memory(in MB) currently available for allocation to a process or for system use.

Collection interval: MEDIUM

Policy: MSSQLServer_3420

Aspect: Microsoft SQL Server Memory and Memory Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

LogReaderDelivLatency

Description: This metric monitors the time (in milliseconds) elapsed from when transactions are applied at the Publisher till the time they are delivered to the Distributor.

Collection interval: HIGH

Policy: MSSQLServer_3411

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 30000, WARNING / 10000

Message Text: Rule1: LogReader Delivery Latency for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: LogReader Delivery Latency for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Slow network could be an issue. Network latency has a substantial negative performance impact.
- Delivery latency can be high if publisher servers are overloaded.

Potential Impact: Performance issues at the publisher

Suggested Action(s) :

- Monitor Log Reader delivery rate which can be found from the mslogreader_history table in the distribution database. Poor log reader performance is caused by network issues.
- Check if transaction log is not growing or shrinking at the same time.
- Check if database backups are not running during your long latency period.

MemoryManagerTotalServerMemory

Description: This metric monitors the number of database pages (buffer manager) which are currently being occupied in the data cache.

Collection interval: HIGH

Policy: MSSQLServer_3421

Aspect: Microsoft SQL Server Memory and Memory Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

TransLogBackup

Description: This metric monitors the number of hours since last database transaction log backup.

Collection interval: LOW

Policy: MSSQLServer_3234

Aspect: Microsoft SQL Server Backup

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 876000.0, MINOR / 48, WARNING / 24

Message Text: Rule1: The transaction log for database <OPTION(database_name)> has never been backed up for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: The transaction log for database <OPTION(database_name)> has not been backed up for <VALUE> hours (>=<THRESHOLD> hours) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule3: The transaction log for database <OPTION(database_name)> has not been backed up for <VALUE> hours (>=<THRESHOLD> hours) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Database Transaction Log backup has exceeded the threshold.

The Transaction log for the indicated database does not have the **Truncate on checkpoint** option selected and has exceeded the backup threshold.

Suggested Action(s) : Take a backup of the Transaction Log for the indicated database.

FullScansRate

Description: This metric monitors the Full scans rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3051

Aspect: Microsoft SQL Server Data Access Methods

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 20

Message Text: Full scan rate (<VALUE>/sec) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of unrestricted full scans per second is too high. These can be either base-table or full-index scans

Full Scans occur when there are no restrictions placed on queries, or when the optimizer determines that a table scan is the most efficient way to this can be a heavy performance drain. However there may be also be many small tables that don't have indexes at all.

Note: A high scan rate does not necessarily indicate a major performance problem, since small table scans may account for most of the scans being performed. However, this metric can still be useful as an overall indicator of either an increased level of activity in the database, or a change in the types of users or the way they are using the database. So it can indicate that things are out of the ordinary with respect to the type or amount of activity going on. Also, this metric coupled with other such as #3007 (Reads Outstanding) may indicate that too many table scans on large tables are being performed, which warrants further investigation.

Suggested Action(s) : Do one of the following:

- Find the queries or procedures that cause the unrestricted scans.
- Build new indexes and update statistics.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Data Access graph to understand about performance over a period of time.

DatabaseSize

Description: This metric monitors the MSSQL Server Database Size (in MB) that is allocated and free.

Collection interval: MEDIUM

Policy: MSSQLServer_3240

Aspect: Microsoft SQL Server Space

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBSPI_MSS_REPORT / DBSPI_MSS_REPORT

DistDeliveredCmdsPersec

Description: This metric monitors the number of commands delivered to the Subscriber per second.

Collection interval: HIGH

Policy: MSSQLServer_3401

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

LockTimeoutRate

Description: This metric monitors the lock timeout rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3070

Aspect: Microsoft SQL Server Locks

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 10

Message Text: Lock timeout rate (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : There are Internal requests for NOWAIT locks for all combined objects.

Locks are applied for a long duration, which usually indicates a locking contention problem. The performance impact is that processes will not complete properly and will abort.

Suggested Action(s) :

- Analyze the SQL code.
- Look for unnecessary exclusive locks, holdlocks or overly long transactions. This is often a process design problem.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Lock Requests graph to understand about performance over a period of time.

DBTrustworthySetting

Description: This metric monitors the number of databases with 'is_trustworthy_on' setting is enabled.

Collection interval: HIGH

Policy: MSSQLServer_3509

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

DBLogGrowthsCnt

Description: This metric monitors the number of transaction log expansions for the server.

Collection interval: MEDIUM

Policy: MSSQLServer_3066

Aspect: Microsoft SQL Server Transactions

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 4.0

Message Text: # of transaction log expansions for the entire server (<VALUE>) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The transaction log has been expanded for the entire server for a number of times and these are too high.

SQL Server has no space for the transaction log and expanded it, or a system administrator has expanded the log to provide more space.

Suggested Action(s) : This depends on whether expansion is required. If the expansion is because of increased activity on the database, database will continue to expand.

The changes may be retained or the frequency of backup transaction commands may need to be increased. If the changes are due to a single unique event, an administrator may consider truncating the log and shrinking the database back to its original size.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Transactions graph to understand about performance over a period of time.

BufferManagerDBPages

Description: This metric monitors the total memory the server has committed to the SQL objects.

Collection interval: HIGH

Policy: MSSQLServer_3422

Aspect: Microsoft SQL Server Buffer Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DatabaseReplicaTxTermPerSec

Description: This metric monitors transaction termination (in milliseconds) waited for acknowledgment per second.

Collection interval: HIGH

Policy: MSSQLServer_3438

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

AvailabilityReplicaRole

Description: This monitor checks the role of availability replica (M3510_ReplicaRole).

Collection interval: HIGH

Policy: MSSQLServer_3510

Aspect: Microsoft SQL Server High Availability (DR)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> availability replica(s) are having invalid('Resolving') role for <OPTION (dbserv)> [Policy: <NAME>]

Instruction Text:

Metric returns role of availability replica (M3510_ReplicaRole).

Probable Cause(s) : The role of this availability replica is unhealthy. The replica does not have either the primary or secondary role.

Potential Impact: NA

Suggested Action(s) : This unit monitor does not contain any resolutions.

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

DBMirroring_LogGenRate

Description: This metric monitors the Log generation rate on the principal.

Collection interval: HIGH

Policy: MSSQLServer_3087

Aspect: Microsoft SQL Server Database Mirroring

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 10.000000

Message Text: Log generation rate on the principal database <OPTION(database_name)> is <VALUE> too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Rate at which incoming transactions are being entered into the principal's log (in KB) is too high per second.

The amount part of the log that is generated by the application is the amount of information that is sent across the wire through the network to the mirror database, and therefore has the most significant impact on performance.

Suggested Action(s) : As a rough guideline, the network bandwidth should be three times the maximum log generation rate.

LockAvgWaitTime

Description: This metric monitors the Average lock wait time.

Collection interval: MEDIUM

Policy: MSSQLServer_3073

Aspect: Microsoft SQL Server Locks

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 500

Message Text: Average lock wait time (<VALUE> milliseconds) too high (>=<THRESHOLD> milliseconds) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

Average amount of wait time (in milliseconds) for each lock request that resulted in a wait for all objects combined.

Many users try to access the same row at the same time. This is a performance bottleneck in preventing latches from being released.

Suggested Action(s) : Analyze the database design and coding for locking issues. Quick fixes include getting faster CPUs.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Lock Requests graph to understand about performance over a period of time.

TableSize

Description: This metric monitors the MSSQL Server Table Size (in MB) which is allocated and which is free.

Collection interval: MEDIUM

Policy: MSSQLServer_3241

Aspect: Microsoft SQL Server Space

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBSPI_MSS_REPORT / DBSPI_MSS_REPORT

DBMirroring_CurrSndRate

Description: This metric monitors the current send rate on the principal.

Collection interval: HIGH

Policy: MSSQLServer_3088

Aspect: Microsoft SQL Server Database Mirroring

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 10.000000

Message Text: Current send rate on the principal database <OPTION(database_name)> is <VALUE> too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Rate at which transactions are being sent to the mirror server instance in kilobytes (KB) per second.

It depends on the hardware, disk environment and network bandwidth.

Suggested Action(s) : Check hardware setup and network efficiency.

CmdQueueLenPct

Description: This metric monitors the percentage of command queue length used.

Collection interval: MEDIUM

Policy: MSSQLServer_3017

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 10

Message Text: % of command queue length used (<VALUE>%) too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : When SQL Server does not have a worker thread immediately available to execute a command, it places the command into Command Queue.

Value of this metric running above 0 indicates that the number of user connections exceed the maximum number of worker threads so that the SQL Server activated its thread pooling mechanism.

Too much of thread swapping may negatively affect SQL Server performance.

Suggested Action(s) : Consider increasing the number of available worker threads using the `max worker threads` configuration parameter and increase memory allocated to SQL Server.

You can restrict user connections from using the 'user connections' configuration parameter to decrease the workload on the SQL Server. User connections and worker threads are counted as overhead against the SQL Server memory allocation. So, plan accordingly when adjusting these values.

The automatic action report for this metric shows network statistics and the maximum configured worker threads.

By default, a set of graphs are launched from this event. See the Server Status graph to understand about performance over a period of time.

LocksWaitRate

Description: This metric monitors the wait rate of the Lock requests.

Collection interval: MEDIUM

Policy: MSSQLServer_3072

Aspect: Microsoft SQL Server Locks

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 10

Message Text: Locks wait rate (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of lock requests per second has resulted in a deadlock for each object type: Extent, Key, Page, Table, RID, Database.

Many users try to access the same row at the same time or there is a performance bottleneck due to increased activity.

Suggested Action(s) : This is normal unless you are experiencing excessive lock timeouts (as measured by user complaints). Then this becomes a very useful metric for understanding volumes. This may require analysis of the database design and coding.

The automatic action report for this metric shows a lock count for each object and which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Lock Requests graph to understand about performance over a period of time.

DBMirroring_UnrestoredLog

Description: This metric monitors the Unrestored log on the mirror.

Collection interval: HIGH

Policy: MSSQLServer_3086

Aspect: Microsoft SQL Server Database Mirroring

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 95.000000, WARNING / 90.000000

Message Text: Rule1: Size of unrestored log in the redo queue on the mirror database <OPTION (database_name)> is <VALUE> too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: Size of unrestored log in the redo queue on the mirror database <OPTION (database_name)> is <VALUE> too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The amount of log waiting in the redo queue (in kilobytes) is too high.

In any operating mode, the mirror server can develop a backlog of unrestored log records that have been written to the log file but still need to be restored on the mirror database. Normally it happens during failover time.

Suggested Action(s) : Check if the failover was successful.

MemoryManagerGrantsPending

Description: This metric monitors the number of processes waiting for a workspace memory grant.

Collection interval: VERYHIGH

Policy: MSSQLServer_3417

Aspect: Microsoft SQL Server Buffer Manager (Add-on)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

LogshippingRestoreJobs

Description: This metric monitors the restore job in secondary instance of logshipping configuration.

Collection interval: HIGH

Policy: MSSQLServer_3293

Aspect: Microsoft SQL Server Logshipping

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 2.5, MAJOR / 1.5, CRITICAL / 0.5

Message Text: Rule1: Restore job is not enabled in the secondary instance. Secondary ID = <OPTION(secondary_id)>, Secondary DB name = <OPTION(secondary_db)>, Enabled= <OPTION (enabled) ;> [Policy: <NAME>]

Message Text: Rule2: Last execution of a restore job is success, but restore job is not running as per the scheduled time span. Secondary ID = <OPTION(secondary_id)>, Secondary DB name = <OPTION(secondary_db)>, Elapsed time = <OPTION(elapsed_time) seconds ;> [Policy: <NAME>]

Message Text: Rule3: Last execution of restore Job is failed in secondary instance. Secondary ID = <OPTION(secondary_id)>, Secondary DB name = <OPTION(secondary_db)>, Detailed Error= <OPTION(detailed_error) > [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Primary and secondary configuration may not be in a proper state.
- Restore job mighty not be enabled and scheduled properly. This job might not run properly.
- Transaction log is full.
- If the secondary database instance is not configured as standby (read-only), then restore might fail.
- The primary and secondary configuration could be out of sync.
- Authentication mismatch between the primary and secondary instances.
- The shared folder does not exist.
- The primary and secondary configuration are not in sync.

Suggested Action(s) :

- Check agent log and logshipping monitor information.
- Take a full manual backup of the database and transaction log from the primary and restore it in the secondary. Then configure the logshipping. This might resolve out of sync error.
- Make sure the login credentials are same in both the primary and secondary instances and DB name also should be same.
- Check the job history.
- Check the configuration of the copy job.
- Check the properties of the shared backup folder in the primary and secondary.

ExtntsAllocRate

Description: This metric monitors the allocated rate for the Extents.

Collection interval: HIGH

Policy: MSSQLServer_3054

Aspect: Microsoft SQL Server Data Access Methods

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 300

Message Text: Extents allocated rate (<VALUE>/sec) too high (>=<THRESHOLD>) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of extents allocated (per second) are too high compared with database objects used for storing index or data records

This should be an indication of table creation such as the creation of temporary tables. If many new rows are added to indexed database tables, the index may have more than a usual rate of Extents allocated. The relationship between this metric and #3053 (Pages allocated rate) can be informative. If existing tables are growing, then 8 pages tend to be allocated for each extent. On the other hand, if smaller tables are created, only one or two pages may be allocated for each extent allocated. So an increase in one or both of these metrics can indicate a change in the type of activity being performed on a database.

Suggested Action(s) : If this is an indexing problem and a regular one, the addition of new index pages may be reduced by implementing a fill factor of the indexes. If the problem is frequent and related to the creation of tables, an alternative to using temporary tables may need to be found in SQL. - Query optimization and using fill factor is one suggestion. However, query optimization might not help, unless the problem is that `temp` tables are unnecessarily created. Applying a fill factor to tables might help in the short term, if pages are being allocated due to page splits. Unfortunately, there is no known method of accurately tracking page splitting, short of analyzing the entire transaction log.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Data Access graph to understand about performance over a period of time.

DelLatency

Description: This metric monitors the Delivery Latency.

Collection interval: HIGH

Policy: MSSQLServer_3083

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 60.0

Message Text: Delivery Latency <VALUE> for the Distribution Agent with agent_id <OPTION(agent_id)> too high (>=<THRESHOLD>) for the publisher <OPTION(publisher)> [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

Delivery latency for one or more publishers is high.

Delay in distribution of commands which are marked for replication. There can be issues related to the distribution agent.

Suggested Action(s) : Check for the distribution of commands which are waiting in the distribution database.

LazywritesPersec

Description: This metric monitors the number of buffers written by buffer manager's lazy writer.

Collection interval: VERYHIGH

Policy: MSSQLServer_3416

Aspect: Microsoft SQL Server Buffer Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

WritsOutstdRate

Description: This metric monitors the number of write requests issued to the OS not completed.

Collection interval: VERYHIGH

Policy: MSSQLServer_3008

Aspect: Microsoft SQL Server Input and Output Utilization

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 1

Message Text: Writes outstanding rate (<VALUE>/min) too high (>=<THRESHOLD>/min) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of SQL Server `write` requests that are too high compared with the Windows OS I/O subsystem that are not complete.

Suggested Action(s) : You can do the followings:

- Add more RAM.
- Install a faster IO system.
- Review the application's transaction management.

The automatic action report for this metric will show 'Input/Output' statistics and which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the IO Utilization graph to understand about performance over a period of time.

Pagelifeexpectancy

Description: This metric monitors the number of seconds a page will stay in the buffer pool without references.

Collection interval: VERYHIGH

Policy: MSSQLServer_3412

Aspect: Microsoft SQL Server Input and Output Utilization

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

SQLServerBatchRequest

Description: SQL Server's Batch Requests represents the number of SQL Statements that are being executed per second.

Collection interval:

Policy: MSSQLServer_3428

Aspect: Microsoft SQL Server Processes and Statistics (Add-on)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DBPageRecoveryMOdelSetting

Description: This metric monitors the number of databases that has the `recovery_model` setting is disabled.

Collection interval: HIGH

Policy: MSSQLServer_3508

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

DatabaseReplicaFileStreamByteRcvPerSec

Description: This metric monitors the amount of filestream data received by the availability replica for the database.

Collection interval: HIGH

Policy: MSSQLServer_3435

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

PagereadsPersec

Description: This metric monitors the number of physical database page reads issued.

Collection interval: VERYHIGH

Policy: MSSQLServer_3413

Aspect: Microsoft SQL Server Buffer Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DBMirroring_UnsentLog

Description: This metric monitors the unsent log on the principal.

Collection interval: HIGH

Policy: MSSQLServer_3085

Aspect: Microsoft SQL Server Database Mirroring

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 95.000000, WARNING / 90.000000

Message Text: Rule1: Size of unsent log in the send queue on the principal database <OPTION (database_name)> is <VALUE> too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: Size of unsent log in the send queue on the principal database <OPTION (database_name)> is <VALUE> too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The size of the unsent log (in KB) is too high.

During high-performance mode, a principal server can develop a backlog of unsent log records that still need to be sent from the principal server to the mirror server. However, this is also relevant for high-safety mode when mirroring is paused or suspended because the partners become disconnected.

Suggested Action(s) : Check if mirroring is paused, suspended or partner servers are disconnected.

DistDeliveryLatency

Description: This metric monitors the time (in milliseconds) elapsed from when transactions are delivered to the Distributor till the time they are applied at the Subscriber.

Collection interval: HIGH

Policy: MSSQLServer_3403

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 30000, WARNING / 10000

Message Text: Rule1: Distribution Delivery Latency for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: Distribution Delivery Latency for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Slow network could be an issue. Network latency has a substantial negative performance impact.
- Delivery latency can be high if subscribing servers are overloaded.

Potential Impact: Performance issues at the Subscriber.

Suggested Action(s) :

1. Check the size of Distribution database. Due to huge data load on distribution database, sometimes distribution agent faces issues while reading data which in turn affect the delivery rate metric.
2. Re-configure Agent values for CommitBatchSize and CommitBatchThreshold to increase the throughput.
3. Avoid horizontal filtering.

FullTextSearchPopulationBatches

Description: This metric monitors the count of outstanding population batches for the Full Text Search Service.

Collection interval: MEDIUM

Policy: MSSQLServer_3518

Aspect: Microsoft SQL Server Full-Text Search

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> currently outstanding population batches (>=1) for full text search on <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

The outstanding population batches are monitored and alert is generated for the count of outstanding population batches.

Probable Cause(s) : Low system resources are available for Full Text Search.

Potential Impact: Slow down in performance of Full Text Search.

Suggested Action(s) : Attempt to restart the SQL Server Full Text Search service after rebuilding the full text catalog.

DeadlocksRate

Description: This metric monitors the Deadlocks rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3071

Aspect: Microsoft SQL Server Locks

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MAJOR / 3

Message Text: Deadlocks rate (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of lock requests per second has resulted in a deadlock for each object type: Extent, Key, Page, Table, RID, Database.

The probable causes could be:

- Two or more processes access data in different methods
- Size of a transaction is too large.

In any multi-use environment, occasional lock collisions are normal. Excessive lock collisions affect performance.

Performance may be affected since one of the deadlocked processes will be terminated by the server.

Suggested Action(s) : Deadlocks affect performance for two reasons. First, the deadlocked process needs to be rolled back. Second, it probably has to be done again.

Action depends on situation. You may need to restructure indexes or reschedule load processes when readers are not running, or make transactions shorter or smaller. Optimize the queries. This is often a process design problem. The automatic action report for this metric shows which users are connected to the SQL Server.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Lock Requests graph to understand about performance over a period of time.

SnapshotDeliveredTransPersec

Description: This metric monitors the number of Snapshot transactions delivered to the Distributor per second.

Collection interval: HIGH

Policy: MSSQLServer_3408

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

AvailableReplicaByteRcvPerSec

Description: This metric monitors the total number of bytes received from the replica over the network.

Collection interval: HIGH

Policy: MSSQLServer_3434

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

CacheHitPct

Description: This metric monitors the percentage of times a data page was found in the cache.

Collection interval: VERYHIGH

Policy: MSSQLServer_3001

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 80

Message Text: Cache hit percentage (<VALUE>%) too low (<=<THRESHOLD>%) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Percentage of read requests that read information from memory rather than disk. A low value for the Cache Hit Ratio is an indication of high physical reads.

Suggested Action(s) :

- Measure I/O when the server is about to start to load the cache. The cache may not have enough space to store frequently used data pages.

  By default, the server data cache size is automatically set by the SQL Server. In this case, adding RAM could be the only remedy. However, this may not be the case when SQL Server is sharing its hardware with other memory-intensive OS processes, such as other server processes or if the amount of memory configured for SQL Server to use (maximum server memory) has been set (as per the recommendations for Microsoft Full-Text Search, for example).

- Check the SQL statements and correct the statements if they are wrong. If inaccurate index statistics, and so on cause excess table scans, correcting the SQL statements improve the cache hit ratio as well as overall server performance.

The automatic action report for this metric will show 'Least Recently Used' statistics and which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Least Recently Used graph to understand about performance over a period of time.

AvgLatchWaitTim

Description: This metric monitors the average latch wait time.

Collection interval: MEDIUM

Policy: MSSQLServer_3069

Aspect: Microsoft SQL Server Latches

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 500

Message Text: Average latch wait time (<VALUE> milliseconds) too high (>=<THRESHOLD> milliseconds) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Average latch wait time (in milliseconds) for latch requests that had to wait since the server started. See M076 for current average latch wait time.

Many users try to access the same row at the same time. This is a performance bottleneck.

Suggested Action(s) : Analyze the database design, process design, and coding.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Latches graph to understand about performance over a period of time.

LongRunningJobs

Description: This metric monitors the number of long running jobs.

Collection interval: VERYHIGH

Policy: MSSQLServer_3089

Aspect: Microsoft SQL Server Jobs (Add-on)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> jobs running for long duration (<OPTION(duration)> minutes) which is more than threshold (<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

This monitor returns the long running job count and checks for jobs which are running more than the defined threshold. A warning message is sent if the long running job runs for longer than the configured threshold. The specified query helps to list the count of long running jobs (M3089_LongRunningJobs).

Probable Cause(s) : Resource exhaustion and improperly designed jobs could lead to long running jobs, which is an unhealthy state that is caused by a SQL Server Agent job that has run longer than the defined threshold.

Potential Impact: Performance issues for upcoming scheduled jobs, which in turn affects overall SQL Server performance.

Suggested Action(s) : To identify what jobs are running for a long time, check SQL Server Management Studio.

Additionally if it is expected for some agent jobs to run for a longer period of time:

- Override the monitor to change the thresholds for this specific SQL instance.
- Disable the monitor for this specific SQL instance.

DBMirroring_State

Description: This metric monitors the mirroring state of the server instance.

Collection interval: VERYHIGH

Policy: MSSQLServer_3084

Aspect: Microsoft SQL Server Database Mirroring

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 0.5

Message Text: Mirroring is suspended for server instance <OPTION(dbserv)> for one or more database(s) with its mirroring partner instance <OPTION(mirror_instance)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Mirroring is suspended.

The principal database is available but does not send any logs to the mirror server. If the session is paused or there are redo errors on the mirror, the principal enters the SUSPENDED state.

Suggested Action(s) : Check for redo errors on the mirror or whether the mirroring session is paused.

ReportsUptime

Description: MSSQL Server Availability: Reports uptime information.

Collection interval: VERYHIGH

Policy: MSSQLServer_3243

Aspect: Microsoft SQL Server Availability

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBSPI_MSS_REPORT / DBSPI_MSS_REPORT

DBChainingSetting

Description: This metric monitors the number of databases with 'is_db_chaining_on' setting is disabled.

Collection interval: HIGH

Policy: MSSQLServer_3506

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

ConflictsPersec

Description: This metric monitors the number of conflicts per second during Publisher or Subscriber upload and download.

Collection interval: HIGH

Policy: MSSQLServer_3404

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 5, WARNING / 1

Message Text: Rule1: Number of conflicts per second for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: Number of conflicts per second for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Merge replication is a two-way replication in which data changes occur. The publisher and subscriber are merged at the time of synchronization. This value should always be zero. A nonzero value may require notification and appropriate action should be taken care to override the conflict. Change of data on both publisher and subscriber at the time of synchronization results in conflict.

Suggested Action(s) : Based on business requirement, you can specify merge replication to recognize conflicts at row-level or at column-level.In order to resolve conflict, SQL Server has a few options:

1. Default Resolver:

Default resolver resolves the conflict based on priority. At the subscriber end, you can assign the priority values which will determine the winner in case of conflict.

2. Custom Resolver:

If your business requirements do not meet the default resolver, the second option you have is custom resolver. Customer resolver is specific to tables. If you want to modify any table, modify the resolver.

3. Stored Procedure Conflict Resolver:

Is a type of customer resolver which uses stored procedure in T-SQL to implement business logic at the time of conflict. These stored procedures are applicable only for publisher and will be used to resolve update conflicts only.

CompletedJobs

Description: This metric generates a report on all failed and canceled jobs.

Collection interval: VERYHIGH

Policy: MSSQLServer_3277

Aspect: Microsoft SQL Server Jobs

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 3, MAJOR / 2

Message Text: Rule1: Job cancelled. Job name = <OPTION(job_name)>, Job id = <OPTION(job_id)>, run date = <OPTION(run_date)>, runtime = <OPTION(run_time) for <OPTION(dbserv)>>. [Policy: <NAME>]

Message Text: Rule2: Job failed. Job name = <OPTION(job_name)>, Job id = <OPTION(job_id)>, run date = <OPTION(run_date)>, runtime = <OPTION(run_time) for <OPTION(dbserv)>>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : A job has failed or has been canceled.

Suggested Action(s) : Check the annotations for the steps completed. If successful, all steps are completed. If failed or canceled, then it will show the steps completed before the failure or cancellation.

The automatic action for the metric generates a report on the completed steps for this job.

PgesAlloctdRate

Description: This metric monitors the pages allocated rate.

Collection interval: VERYHIGH

Policy: MSSQLServer_3053

Aspect: Microsoft SQL Server Data Access Methods

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 1000

Message Text: Pages allocated rate (<VALUE>/sec) too high (>=<THRESHOLD>) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of pages allocated (per second) are too high compared with the database objects used for storing index or data records.

A large amount of data is added to the database or data is moved in an unfortunate manner that causes splitting of pages. The relationship between this metric and #3054 (Extents allocated rate) can be informative. If existing tables are growing, then eight pages are allocated for each extent. On the other hand, if you create smaller tables, only one or two pages are allocated for each extent. So an increase in one or both of these metrics can indicate a change in the type of activity being performed on a database.

Suggested Action(s) : This may be normal, but watch the trends here, as it will help you determine how busy your I/O subsystem is. Query optimization will not help unless the `temp` tables are unnecessarily created. Applying a fill factor to tables might help in the short term, if pages are being allocated due to page splits. Unfortunately, there is no known method of accurately tracking page splitting, short of analyzing the entire transaction log.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Data Access graph to understand about performance over a period of time.

LockAvgWaitTime

Description: This metric monitors the average lock wait time per object type.

Collection interval: NORUN

Policy: MSSQLServer_3273

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 500

Message Text: Average lock wait time for object <OPTION(instance_name)> (<VALUE> milliseconds) too high (>=<THRESHOLD> milliseconds) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of lock requests per second has resulted in a deadlock for each object type: Extent, Key, Page, Table, RID, Database

Many users are trying to access the same row at the same time or there is a performance bottleneck preventing latches from being released.

Suggested Action(s) : This may require analysis of the database design and coding for locking issues. Quick fixes include getting faster CPUs.

The automatic action report for this metric shows which users are connected to SQL Server.

DBConnect

Description: This metric monitors the ability to connect to a database.

Collection interval: NORUN

Policy: MSSQLServer_3230

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Fault

Severity / Threshold: CRITICAL / 0.5

Message Text: Cannot connect to database <OPTION(database_name)> for <OPTION(dbserv)>; Failed with the error <OPTION(error_code)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Unable to connect to the database

The database may be corrupt or the 'single user' database option may be turned on.

Suggested Action(s) : Check database configuration options using sp_helpdb, and reconfigure if that's the problem. If the database is corrupt, reload the database from a backup.

The automatic action report for this metric shows other database statistics using the sp_helpdb command.

MemoryWrittenPagePerSec

Description: This metric monitors the number of pages read from or written to disk is collected.

Collection interval: VERYHIGH

Policy: MSSQLServer_3424

Aspect: Microsoft SQL Server Memory and Memory Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

AvailableReplicaByteSentPerSec

Description: This metric monitors the total number of bytes sent over the network to the Replica database.

Collection interval: HIGH

Policy: MSSQLServer_3433

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

AutoShrinkSetting

Description: Monitors the Auto Shrink setting for the databases. This monitor is a part of overall standards requirement hence if settings do not meet the specified standard, an alert needs to be generated to avoid performance issues in database.

Collection interval: HIGH

Policy: MSSQLServer_3503

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> databases with 'is_auto_shrink_on' setting enabled for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Auto Shrink specify whether the database files are available for periodic shrinking. Possible values are `True` and `False`. Monitor the Auto Shrink setting for the databases. If settings do not meet the specified standard, an alert needs to be generated to avoid performance issues in database. Expected setting for this monitor is IS_AUTO_SHRINK_ON = OFF.

Probable Cause(s) : A warning alert will be raised if the option does not match the required setting. Out of the box, the monitor is configured to alert when this setting is set to ON.

Potential Impact: Performance issues in database.

Suggested Action(s) : This issue may be resolved by:

- Changing the configuration setting for this database to match the expected value.
- Overriding the expected value for this unit monitor for this specific database or all databases.

Alternatively, if this monitor is not important for this database:

- Disable the monitor using overrides for this specific database or all databases.
- Disable the top-level aggregate configuration monitor using overrides for this specific database or all databases.

SqlAdtLstBckUp

Description: This metric monitors the size of last full database backup of each database (M3092_ LastBkSize).

Collection interval: MEDIUM

Policy: MSSQLServer_3092

Aspect: Microsoft SQL Server Audit

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 500

Message Text: Size of last full database backup (<VALUE> mb) is higher than threshold (<THRESHOLD> mb) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

The size of the last full database backup for each database (in MB). When reviewing the analysis graph, look for large, steep increase or decrease that indicate rapid changes in size. For example, you might expect backup size to track data size on a slow upward trend over a long time with seasonal drops relating to when data is archived. Monitoring system databases means that you can check to see if anyone accidentally creates a table in Master and fills it with data because they didn’t change their default database setting.

Retrieve top records whose `back_size` is changed to maximum from table `msdb.dbo.backupfile`. If no backup exists for the database, size returned is 0.

Probable Cause(s) :

- Large amount of data inserted or deleted from the database.
- Backup process did not complete correctly.

Potential Impact: Impacts server performance and efficiency.

Suggested Action(s) :

1. Identify unused large size data objects in database.

2. Check if the database backup is not corrupt and take manual backup in this case.

LogReaderTransPersec

Description: This metric monitors the number of LogReader transactions delivered to the Distributor per second.

Collection interval: HIGH

Policy: MSSQLServer_3410

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

BlckdProcessCnt

Description: This metric monitors the number of blocked processes.

Collection interval: VERYHIGH

Policy: MSSQLServer_3014

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 10

Message Text: # of blocked processes (<VALUE>) too high (>=<THRESHOLD> for 2 collection intervals) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of blocked processes in the server is greater than the threshold for more than one collection interval (usually 5 minutes). Potential impact is performance since blocked processes will wait until block is cleared.

Suggested Action(s) : Blocked processes are an indication of contention, which occurs frequently in OLTP and mixed use systems. You may need to restructure indexes, reschedule load processes when readers are not running or change page lock promotion thresholds. It is also an indication of a poorly designed application process.

The automatic action report for this metric will show processes that are the source of interlocking and which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Server Status for processes and transactions graph to understand about performance over a period of time.

PhysicalReadsWrites

Description: This metric monitors the MSSQL Server Workload: Number of physical reads and writes to the disk since the last collection for each tablespace.

Collection interval: VERYHIGH

Policy: MSSQLServer_3244

Aspect: Microsoft SQL Server Input and Output Utilization

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBSPI_MSS_REPORT / DBSPI_MSS_REPORT

LogshippingCopyJobs

Description: This metric monitors the copy backup job in secondary instance of logshipping configuration.

Collection interval: HIGH

Policy: MSSQLServer_3292

Aspect: Microsoft SQL Server Logshipping

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: WARNING / 2.5, MAJOR / 1.5, CRITICAL / 0.5

Message Text: Rule1: Copy Backup job is not enabled in the secondary instance. Secondary ID = <OPTION(secondary_id)>, Secondary DB name = <OPTION(secondary_db)>, Enabled= <OPTION (enabled) > [Policy: <NAME>]

Message Text: Rule2: Last execution of a copy backup job is success, but copy backup job is not running as per the scheduled time span. Secondary ID = <OPTION(secondary_id)>, Secondary DB name = <OPTION(secondary_db)>, Elapsed time = <OPTION(elapsed_time) seconds > [Policy: <NAME>]

Message Text: Rule3: Last execution of copy backup Job is failed in secondary instance. Secondary ID = <OPTION(secondary_id)>, Secondary DB name = <OPTION(secondary_db)>, Detailed Error= <OPTION(detailed_error) > [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

- Primary and secondary configuration may not be in a proper state.
- Log copy back up job mighty not be enabled and scheduled properly. This job might not run properly.
- Copy job might fail, if the secondary could not connect to the primary (network issue) there will not be any space on the device.
- The shared folder is not accessible.

Suggested Action(s) :

1. Check agent log and logshipping monitor information.
2. Check the job history.
3. Check the configuration of the restore job.
4. Check the properties of the shared backup folder in the primary and secondary.

MemryCnsmptn

Description: This monitor checks for the amount of memory used by the resource pool.

Collection interval: HIGH

Policy: MSSQLServer_3094

Aspect: Microsoft SQL Server Memory and Memory Manager (Add-on)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: CRITICAL / 90

Message Text: <VALUE> pages read from or written to disk is more than (<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

This metric returns the amount of memory used by resource pool. This monitor tracks the memory consumption. A warning or error alert is generated if the amount of memory is greater than configured threshold (M3094_MemryCnsmptn).

Probable Cause(s) : SQL Server In-Memory OLTP uses more memory and in different ways than SQL Server does.

It is possible that the amount of memory you installed and allocated for In-Memory OLTP becomes inadequate for your growing needs.

If so, you could run out of memory.

Potential Impact: Memory allocation performance issues.

Suggested Action(s) : To resolve your Low Memory or Out Of Memory condition you need to either free up existing memory by reducing usage, or make more memory available to your in-memory tables. Possible corrective actions may include:

- Check if there are long running transactions that is preventing garbage collection of memory. Consider ending the transaction and/or check the design of the application to see if you can reduce the duration of the transactions.

- Free up existing memory.
- Delete non-essential memory optimized table rows and wait for garbage collection.
- Move one or more rows to a disk-based table.
- Increase value of MAX_MEMORY_PERCENT on the resource pool.
- Increase memory available to SQL Server Instance by configuring the Max Server Memory to a higher value.
- Install additional memory.

FThreadCnt

Description: This metric monitors the number of free threads for the DB Engine process.

Collection interval: VERYHIGH

Policy: MSSQLServer_3096

Aspect: Microsoft SQL Server Processes and Statistics (Add-on)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: CRITICAL / 10

Message Text: <VALUE> free thread count for DB Engine process is less than specified threshold (<=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

SQL Server opens system thread for each query request. If the amount of threads exceed the specified max worker threads value, SQL Server pools the worker threads. When all worker threads are active with long running queries, SQL Server may not appear respond until a worker thread completes and becomes available. Though this is not a defect, this can sometimes be undesirable. The monitor analyzes amount of free threads and notifies if the amount is low.

Probable Cause(s) : Increased amount of work causing increase in utilized threads, this could indicate that SQL Server is working under significant load or an excessive number of queries running in parallel.

Potential Impact: Affects the response time and overall system performance.

Suggested Action(s) : Adjusting max worker threads is an advanced option. Thread pooling helps optimize performance when large number of clients are connected to the server. Usually, a separate operating system thread is created for each query request. However, with hundreds of connections to the server, using one thread for each query request can consume large amounts of system resources. The **max worker threads** option enables SQL Server to create a pool of worker threads to service a larger number of query requests, which improves performance.

VirtualDeviceSize

Description: This metric monitors the MSSQL Server Virtual Device size (in MB) allocated for the virtual device.

Collection interval: MEDIUM

Policy: MSSQLServer_3242

Aspect: Microsoft SQL Server Space

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: DBSPI_MSS_REPORT / DBSPI_MSS_REPORT

LocksWaitRate

Description: This metric monitors the locks wait rate per object type.

Collection interval: NORUN

Policy: MSSQLServer_3272

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 10

Message Text: Locks wait rate for object <OPTION(instance_name)> (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of lock requests per second that could not be satisfied immediately and required the caller to wait for each object type:

Extent, Key, Page, Table, RID, Database.

Many users are trying to access the same row at the same time or there is a performance bottleneck due to increased activity.

Suggested Action(s) : This is normal unless you are experiencing excessive lock timeouts (as measured by user complaints). Then this becomes a very useful metric for understanding volumes. This may require analysis of the database design and coding.

The automatic action report for this metric shows a lock count for each object and which users are connected to SQL Server.

RunableContnPct

Description: This metric monitors the percentage of total connections that are runnable.

Collection interval: MEDIUM

Policy: MSSQLServer_3032

Aspect: Microsoft SQL Server Processes and Statistics

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 50

Message Text: % of total connections that are runnable (<VALUE>%) too high (>=<THRESHOLD>%) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percentage of the total runnable connections is high.

If the percentage of runnable connections is greater, then it means that there is too much contention for MSSQL Server resources and the overall performance of the system is negatively affected.

Suggested Action(s) : Analyze the applications that are running by looking at the automatic action report and determine if the application needs correction.

The automatic action report for this metric shows which users are connected to SQL Server.

DownloadedChangesPersec

Description: This metric monitors the number of rows per second replicated from the Publisher to the Subscriber.

Collection interval: HIGH

Policy: MSSQLServer_3405

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

LatchWaitsRate

Description: This metric monitors the Latch waits rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3068

Aspect: Microsoft SQL Server Latches

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 10

Message Text: Latch waits rate (<VALUE>/sec) too high (>=<THRESHOLD>/sec) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of latch requests that could not be fulfilled immediately and had to wait before being fulfilled.

Many users try to access the same row at the same time. This is a performance bottleneck.

Suggested Action(s) : Analyze the database design, process design, and coding.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Latches graph to understand about performance over a period of time.

PagewritesPersec

Description: This metric monitors the number of physical database page writes issued.

Collection interval: VERYHIGH

Policy: MSSQLServer_3414

Aspect: Microsoft SQL Server Buffer Manager

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

DatabaseBackup

Description: This metric monitors the number of hours since last database backup.

Collection interval: NORUN

Policy: MSSQLServer_3233

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 876000.0, MINOR / 168, WARNING / 72

Message Text: Rule1: Database <OPTION(database_name)> has never been backed up for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule2: Database <OPTION(database_name)> has not been backed up for <VALUE> hours (>=<THRESHOLD> hours) for <OPTION(dbserv)>. [Policy: <NAME>]

Message Text: Rule3: Database <OPTION(database_name)> has not been backed up for <VALUE> hours (>=<THRESHOLD> hours) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Database backup has not been done and it has exceeded the threshold time of backup.

Suggested Action(s) : Take a complete database backup for the indicated database.

DBBackupCnt

Description: This metric monitors the number of databases that have not been backed up since the last backup.

Collection interval: VERYHIGH

Policy: MSSQLServer_3035

Aspect: Microsoft SQL Server Transactions

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: MINOR / 60

Message Text: Transaction <OPTION(trans)> in program <OPTION(prog)> (spid=<OPTION(spid)>) for user <OPTION(user)> too long (>=<THRESHOLD> seconds) for database <OPTION(db)> in <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : This metric will generate an alarm if the longest running transaction has been running longer than the defined threshold (in seconds). The message text identifies the longest running transaction.

There can be a variety of reasons for transactions taking a long time to execute, one of the most common reasons is blocking. Blocking occurs when one session (identified by SPID) holds a lock on a specific resource and a second session attempts to acquire a conflicting lock type on the same resource.

Suggested Action(s) : Using the SPID in the message, follow the steps to gather information about the process or transaction:

1. Identify the SPID at the head of the blocking chain.
2. Use the SQL Enterprise Manager as follows:
   a. Expand the server group and then expand the server.
   b. Expand Management and then expand Current Activity.
   c. Expand Locks/Process ID. In the details pane, the SPIDs, along with their blocking information are shown. The SPIDs that are blocking others will appear as `(Blocking)`.

   Note: It is sometimes necessary to use direct queries instead of Enterprise Manager, because some types of `tempdb` blocking problems may prevent you from running queries through Enterprise Manager, which uses temporary table operations. Using direct queries gives you the control necessary to avoid this problem.
3. Find the query that the blocking SPID is running.
4. Use the following command to determine the command issued by a particular SPID:

```
DBCC INPUTBUFFER(<spid>)
```

5. Alternately, you can use SQL Enterprise Manager as follows:

   a. Expand the server group and then expand the server.

   b. Expand 'Management' and then expand 'Current Activity'.

   c. Click 'Process Info.' The SPIDs are shown in the details pane.

   d. Double-click the SPID to see the last Transact-SQL command the SPID executed.
6. Find the type of locks the SPID is holding.

   You can determine this information by executing the sp_lock system stored procedure. Alternatively you can use Enterprise Manager as follows:

   a. Expand the server group; then expand the server.

   b. Expand 'Management'; then expand 'Current Activity'.

   c. Expand 'Locks/Process ID.' In the details pane, the SPIDs, along with the information on the locks they are holding, are shown.
7. Find the transaction nesting level and process status of the blocking SPID. The transaction nesting level of a SPID is available in the @@TRANCOUNT global variable. However, it can be determined from outside the SPID by querying the `sysprocesses` table as follows:

```
select open_tran from sysprocesses where spid=<blocking spid number>
```

The value returned is the @@TRANCOUNT value for the SPID. This value shows the transaction nesting level for the blocking SPID, which in turn can explain why it is holding locks. For example, if the value is greater than zero, the SPID is in the midst of a transaction (in which case it is expected to retain certain locks it has acquired, depending on the transaction isolation level).

AutoUpdateStatisticsAsyncSetting

Description: Monitors the Auto Update Statistics Asynchronously setting for the database. If settings do not meet the specified standard, an alert needs to be generated to avoid query performance issues.

Collection interval: HIGH

Policy: MSSQLServer_3504

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> databases with 'is_auto_update_stats_async_on' setting disabled for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Monitor the Auto Update Statistics Asynchronously setting for the database. If settings do not meet the specified standard, an alert needs to be generated. Expected setting for this monitor is `IS_AUTO_UPDATE_STATS_ASYNC_ON = OFF`.

Probable Cause(s) : A warning alert is sent if the option does not match the required setting. Out of the box, the monitor is configured to alert when this setting is set to ON.

The reason this database option is OFF by default is for backward compatibility with the existing applications. Since queries do not wait or block for statistic updates, there is a chance for temporary changes in performance if a query compiles and runs with old statistics. After the statistics are updated, the same query would recompile against the updated statistics before the next run. But while the stats are updating, the query might have momentary performance problems since it was compiled against old statistics.

ASYNC update is probably the best choice for most applications since it minimizes blocking and waiting overall. Consider turning it ON, and if it works for your application, then disable this unit monitor.

Potential Impact: Query performance issues.

Suggested Action(s) : This issue may be resolved by:

- Changing the configuration setting for this database to match the expected value.
- Overriding the expected value for this unit monitor for this specific database or all databases.

Alternatively, if this monitor is not required for this database:

- Disable the monitor using overrides for this specific database or all databases.
- Disable the top-level aggregate configuration monitor using overrides for this specific database or all databases.

DatabaseReplicaDataMovement

Description: This monitor checks the state of data movement of the database replica (M3513_DataMovement).

Collection interval: HIGH

Policy: MSSQLServer_3513

Aspect: Microsoft SQL Server High Availability (DR)

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBMP_MSS_GRAPH / DBMP_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> availability replica(s) are having unhealthy data movement for <OPTION (dbserv)> [Policy: <NAME>]

Instruction Text:

Metric returns 'operational_state' of database replica. It is healthy when data movement is unsuspended or unhealthy otherwise. (M3513_DataMovement)

Probable Cause(s) : Either a database administrator or the system has suspended data synchronization on this availability database.

- The system might have suspended the data movement due to an error.
- You might have suspended the data movement for a maintenance purpose.

Potential Impact: Availability database may not be synchronised.

Suggested Action(s) : Resume the data movement. If the issue persists, check the event log for the availability group and diagnose the reason why the system suspended the data movement.

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

AvailableReplicaMsgRcvPerSec

Description: This metric monitors the total number of messages received from the replica.

Collection interval: HIGH

Policy: MSSQLServer_3431

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

SqlAdtAdHoc

Description: This metric monitors the number of ad hoc queries in the plan cache (M3091_AdHocQuery).

Collection interval: VERYHIGH

Policy: MSSQLServer_3091

Aspect: Microsoft SQL Server Audit

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Admin

Severity / Threshold: MINOR / 10000

Message Text: <VALUE> ad hoc queries in the plan cache that have run only one time goes above threshold value (<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

This metric measures the total number of ad hoc queries in the plan cache that have only run one time. The value is only accurate for the instant the query is run, and the value can change radically from one capture time to the next.

Having too many one-time use ad hoc queries in an instance’s plan cache may indicate plan cache bloat, which is a condition where memory in the plan cache is wasted by storing the execution plans of queries that will never be run more than one time. This wasted space can be used by the data cache. The more RAM devoted to the data cache, the faster SQL Server can perform. This particular metric measures the number of one-time use ad hoc queries that are currently in the plan cache.

This value changes often and that one-time use ad hoc queries can become multi-use ad hoc queries. For example, if this metric returns 3,500 during one run, it is possible that 3,450 of these one-time use ad hoc queries can be used a second time before the same metric runs again. Since this value changes often, it is important to maintain a baseline of data to establish a “typical” value for this metric for each SQL Server instance.

If the value of this metric is high, then it might be a good idea to turn on the SQL Server option “optimize for ad hoc workloads”. When this option is turned ON, the first time an ad hoc query is executed, the entire execution plan is not stored in the plan cache. Instead, a small stub that consumes less memory is stored. The next time the same ad hoc query runs, SQL Server will notice this and will store the entire execution plan in the plan cache. If your instance has a lot of one-time use ad hoc queries, then turning on “optimize for ad hoc workloads” prevents plan cache bloat, helping to boost the performance of your SQL Server instance.

It is not necessary to create an alert for this metric, as you need to watch this value over a period of time. If this value exceeds 10,000, then there is a good chance your SQL Server may be suffering from plan cache bloat. But this is only an estimated number. The only way to really know if turning on “optimize for ad hoc workloads” will be beneficial is to try it, and then use the custom metric, “Number of Ad Hoc Stubs Created When ‘Optimize for Ad Hoc Workloads’ is Turned On” to determine if turning on “optimize for ad hoc workloads” is effective or not at reducing plan cache bloat.

If you turn ON the optimize for ad hoc workloads, consider both the values of this custom metric, plus the Memory used by ad hoc queries running only once custom metric. If both of these metrics are high, then it is likely that your instance is suffering from plan cache bloat.

Retrieving count of Adhoc object type from the dm_exec_cached_plans table.

Probable Cause(s) : Having too many one-time use ad hoc queries in an instance’s plan cache may indicate plan cache bloat, which is a condition where memory in the plan cache is wasted by storing the execution plans of queries that will never be run more than one time.

Potential Impact: It will lead to memory issues and may impact system performance.

Suggested Action(s) :

Turn on the SQL Server option “optimize for ad hoc workloads”. When this option is turned ON, the entire execution plan is not stored in the plan cache when the ad hoc query is not run for the first time. Instead, a small stub, that consumes very little memory is stored. The next time the same ad hoc query runs again, SQL Server stores the entire execution plan in the plan cache. If your instance has a lot of one-time use ad hoc queries, then turning on “optimize for ad hoc workloads” will prevent plan cache bloat, helping to boost the performance of your SQL Server instance.

DatabaseReplicaRecordRedonBytePerSec

Description: This metric monitors the amount of log records redone in the last second to catch up the database replica.

Collection interval: HIGH

Policy: MSSQLServer_3437

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

PageSplitsRates

Description: This metric monitors the page splits rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3055

Aspect: Microsoft SQL Server Data Access Methods

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 1000

Message Text: Page splits rate (<VALUE>/sec) too high (>=<THRESHOLD>) for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of page splits that occur per second are greater as the result of overflowing index pages.

Heavy table inserts or updates change the positions of rows. If the index pages are full, they will need to be split which produces excessive IO.

Suggested Action(s) : Use fillfactors on the indexes to decrease the number of Page Splits, periodically rebuild indexes to enforce the fillfactors.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Data Access graph to understand about performance over a period of time.

AvailableReplicaMsgSentPerSec

Description: This metric monitors the total number of messages sent over the network to the replica.

Collection interval: HIGH

Policy: MSSQLServer_3429

Aspect: Microsoft SQL Server High Availability (AOAG)

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

Note: This metric is supported on SQL Server version 2012 onwards and is not applicable on standalone environment.

FullTextSearchStatus

Description: This metric monitors the status of Full Text Search Service.

Collection interval: VERYHIGH

Policy: MSSQLServer_3517

Aspect: Microsoft SQL Server Full-Text Search

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MAJOR / 0.5

Message Text: Full text search service is not running on <OPTION(dbserv)> [Policy: <NAME>]

Instruction Text:

The status of SQL full text search service is checked and alert is generated when the service is not running.

Probable Cause(s) :

- The SQL Server Full Text Search Service was stopped by an administrator.
- The SQL Server Full Text Search Service did not start because the user account could not be authenticated.
- The SQL Server Full Text Search Service was not configured correctly which prevented it from starting.

Potential Impact: Full Text Search feature would be unavailable.

Suggested Action(s) :

1. Try to restart the SQL Server Full Text Search service.
2. Check the Windows Event Viewer for any errors related to this service. Also look for errors that occurred at the same time when the service was down.
3. Stop and restart the Microsoft Search service.

TblLckEscalRate

Description: This metric monitors the table lock escalation rate.

Collection interval: MEDIUM

Policy: MSSQLServer_3056

Aspect: Microsoft SQL Server Data Access Methods

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: Table lock escalation rate (<VALUE>/sec) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of times locks on a table were escalated.

Large numbers of locks are placed by a process on a table. According to Microsoft, Lock escalation is the process of converting many fine-grain locks into fewer coarse-grain locks, reducing system overhead. Microsoft SQL Server automatically escalates row locks and page locks into table locks when a transaction exceeds its escalation threshold. The thresholds are determined dynamically by SQL Server and require no configuration. While this is true, the major problem to be addressed is not when the table escalations occur, but whether they would occur at all. In other words, even if you could alter the lock escalation threshold, it would be addressing the symptoms, not the cause. Increased table locks may lead to increased blocking and/or deadlocks.

Suggested Action(s) : Analyze and tune queries, run `UPDATE STATISTICS`, and make sure you have useful indexes on the table.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Data Access graph to understand about performance over a period of time.

AutoUpdateStatisticsSetting

Description: Monitors the Auto Update Statistics setting for the database. If settings do not meet the specified standard, an alert needs to be generated to avoid query performance issues.

Collection interval: HIGH

Policy: MSSQLServer_3505

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> databases with 'is_auto_update_stats_on' setting disabled for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Auto Update Statistics specify whether the database automatically updates out-of-date optimization statistics. Possible values are True and False. When True, any out-of-date statistics needed by a query for optimization are automatically built during optimization. Monitor the auto update statistics setting for the database. If settings do not meet the specified standard, an alert needs to be generated. Expected setting for this monitor is `IS_AUTO_UPDATE_STATS_ON = ON`.

Probable Cause(s) : A warning alert is sent if the option does not match the required setting. Out of the box, the monitor is configured to alert when this setting is set to OFF.

Potential Impact: Query performance issues.

Suggested Action(s) : This issue may be resolved by:

- Changing the configuration setting for this database to match the expected value.
- Overriding the expected value for this unit monitor for this specific database or all databases.

Alternatively, if this monitor is not required for this database:

- Disable the monitor using overrides for this specific database or all databases.
- Disable the top-level aggregate configuration monitor using overrides for this specific database or all databases.

AutoCreateStatisticSetting

Description: Monitors the Auto Create Statistic setting for the databases. If settings do not meet the specified standard, an alert has to be sent to avoid poor query performance.

Collection interval: HIGH

Policy: MSSQLServer_3502

Aspect: Microsoft SQL Server Configuration Settings

CIT: SQL Server

Alarming / Logging: Alarming

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 0.5

Message Text: <VALUE> databases with 'is_auto_create_stats_on' setting disabled for <OPTION (dbserv)>. [Policy: <NAME>]

Instruction Text:

Auto Create Statistics specifies whether the database automatically creates missing optimization statistics. Possible values are `True` and `False`. If the value is `True`, any missing statistics needed by a query for optimization are automatically built during optimization. Monitor the Auto Create Statistic setting for the database. If settings do not meet the specified standard, an alert needs to be generated. The expected setting for this monitor is `IS_AUTO_CREATE_STATS_ON = ON`. The query optimizer needs up-to-date and accurate statistics to generate good plans. In most cases, it is recommended to let SQL Server maintain the statistics. If you turn “Auto Create Stats” and “Auto Update Stats”, then it is up to you to keep the statistics up-to-date somehow. Failure to do so will lead to poor query performance. Most applications should have these options `ON`.

Probable Cause(s) : A warning alert will be raised if the option does not match the required setting. Out of the box, the monitor is configured to alert when this setting is set to `OFF`.

Potential Impact: Query performance issues. And overall SQL Server performance issues.

Suggested Action(s) : This issue may be resolved by:

- Changing the configuration setting for this database to match the expected value.
- Overriding the expected value for this unit monitor for this specific database or all databases.

Alternatively, if this monitor is not required for this database:

- Disable the monitor using overrides for this specific database or all databases.
- Disable the top-level aggregate configuration monitor using overrides for this specific database or all databases.

LogReaderDeliveredCmdsPersec

Description: This metric monitors the number of LogReader commands per second delivered to the Distributor.

Collection interval: HIGH

Policy: MSSQLServer_3409

Aspect: Microsoft SQL Server Replication

CIT: SQL Server

Alarming / Logging: Logging

Data source/ Data class: MSSQLSERVER_DATA / PERF_COUNTER

ReportsFailed

Description: This metric monitors the number of Reports (Reporting Services) that have Failed (drill-down).

Collection interval: NORUN

Policy: MSSQLServer_3280

Aspect: NA

CIT: NA

Alarming / Logging: Alarming

Message Category: MSSQLServer_Admin

Severity / Threshold: MINOR / 1

Message Text: Report <OPTION(report_name)> owned by <OPTION(owner)> failed to execute with error <OPTION(report_error_code)> for instance <OPTION(dbname)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : This could have happened because of one of the following reasons:

- DataSource is not configured properly.
- Report took too long to execute.
- Schedule used to trigger the report has expired.
- The report is undeliverable (it is too big).
- The delivery extension specified in the subscription has been uninstalled or disabled.
- The credential settings are changed from integrated to stored or prompted values.
- Parameter name or datatype changed in the report definition and the report was republished. If the subscription includes a parameter that is no longer valid, the subscription becomes invalid.

Suggested Action(s) : You can do the one of the following steps:

1. Check the Application Log for errors related to Reporting Service.
2. Ensure that the subscription is active.
3. Ensure that the credentials used to run the report are valid.
4. Datasource used to connect to the DB is working.

DBActivTransCnt

Description: This metric monitors the number of active transactions for the entire server.

Collection interval: MEDIUM

Policy: MSSQLServer_3064

Aspect: Microsoft SQL Server Transactions

CIT: SQL Server

Alarming / Logging: Alarming / Logging

Data source/ Data class: DBSPI_MSS_GRAPH / DBSPI_MSS_GRAPH

Message Category: MSSQLServer_Perf

Severity / Threshold: WARNING / 5

Message Text: # of active transactions for the entire server (<VALUE>) too high (>=<THRESHOLD>) for <OPTION(dbserv)>. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Number of active transactions for the entire server.

Increased server workload.

Suggested Action(s) : Optimize queries, upgrade server hardware or migrate part of the data to a separate server.

The automatic action report for this metric shows which users are connected to SQL Server.

By default, a set of graphs are launched from this event. See the Server Status for processes and transactions graph to understand about performance over a period of time.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Reference Guide (OMi Management Pack for Microsoft SQL Server 1.01)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to docfeedback@hpe.com.

We appreciate your feedback!