



OMi Management Pack for Oracle WebLogic

Software Version: 2.00

Operations Manager i for Linux and Windows® operating systems

Reference Guide

Document Release Date: September 2017

Software Release Date: September 2017



Hewlett Packard
Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© 2014-2017 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

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 MP for WebLogic Metrics	5
Appendix: Collection and Metrics	6
WeblgcJMSThrBytPct	6
WeblgcWebAppSsnCnt	7
WeblgcEJBPoolWtRt	7
WeblgcGlbThrPIOvld	9
TranAppErrRbPct	9
WeblgcEJBTranThruRt	10
WeblgcJVMMemUtilPct	11
NumWaitrCurrCnt	13
WeblgcJDBCConPIWtCnt	14
WeblgcReqMaxWaitTime	15
WeblgcJMSUtlBytPct	16
Weblgc_ClusterHealth	16
WeblgcWaitSecHighCnt	17
WeblgcAllProcAvgLd	18
WeblgcJDBCConLkRt	20
WeblgcSrvltReqRate	21
WeblgcCnctrPoolUtil	21
Weblgc_ClsInMsFailRt	22
WeblgcPendingReqPct	24
WeblgcExQueWaitCnt	25
WeblgcTranRIBkPct	26
WeblgcDstroydTICnt	27
WeblgcEJBTranRbRt	28
WeblgcClsOutMsFailRt	29
ConnRejectedTotalCnt	30
Weblogic_ServerStatus	31
WeblgcEJBMssdCntRtSm	32
WeblgcJRktRTTtotGCTme	33
WeblgcTranHeurCnt	35

WebLgcTrnRIBkSysToCt	36
WebLgcTranCapUtil	37
WebLgcJMSThrByMsgPct	38
WebLgcEJBCacheHitPct	39
WebLgcJDBCConFail	40
WebLgcPendingReqCount	41
WebLgcExQThrdUtilPct	42
WebLgcEJBTimeoutRt	43
WebLgcStandbyThrdCnt	44
TranResErrRbPct	45
WebLgcInvlLoginAttCnt	46
WebLgcTranAvgTime	47
WebLgcEJBTimeoutRtSm	48
WebLgcDeferredReqCnt	49
WebLgcEJBPoolWtRtSum	50
WebLgcGCThread	51
WebLgcJDBCConTime	53
WebLgcReqWaitTimThrd	54
TranTimErrRbPct	55
WebLgcEJBMssdCntRt	56
ConnDestByErrTotCnt	57
WebLgcSrvltAvExTime	58
WebLgcJDBCConPoolUtl	59
WebLgcGCCCount	60
Send documentation feedback	63

Chapter 1: OMi MP for WebLogic Metrics

This chapter provides information about the OMi MP for Oracle WebLogic collections, metrics, and data store tables which can be used to configure the data-collection procedure.

Each collection contains a number of metrics which are listed under the related collection.

Field	Description
Metric	Metric label
Description	Metric description
Collection Interval	MEDIUM or HIGH or HIGH NA
Aspect	Contains all the policies required for the collection.
CIT	Instance to which you can deploy Aspect.
Alarming / Logging	Alarming / Logging
Data source / data class	Database where the data is logged.
Message Category	Weblogic
Severity	WARNING / CRITICAL / MAJOR / MINOR
Threshold	1-1000

Appendix: Collection and Metrics

WeblogicJMSThrByBytPct

Description: The amount of time at which the server threshold condition was satisfied, based on total bytes

Collection interval: MEDIUM

Policy: Weblogic_JMSBytesThresholdTimePercentage

Aspect: Weblogic JMS Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0036.Weblogic_M0086 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: Number of times queue threshold condition was satisfied by byte count (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The amount of time this JMS queue has spent in the threshold condition has exceeded the threshold value.

The JMS Server message queue threshold condition for the number of bytes stored, as configured in the administration console, has been satisfied for a significant amount of time.

Potential Impact: Once the queue reaches one hundred percent capacity, users will not be able to deliver messages to this queue.

Suggested Action(s) : If possible, the administrator may want to increase the size of the queue via the Administration Server console.

The administrator can also inspect the individual destinations within this JMS Server via the console to determine which destination queues are having problems."

WeblogicWebAppSsnCnt

Description: Number of open sessions for a Web application

Collection interval: MEDIUM

Policy: Weblogic_WebApplicationSessionsCount

Aspect: Weblogic Web Application Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0033.Weblogic_M0066 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 100

Message Text: # of open sessions for web application <OPTION(applicationname)> (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Sudden increase in client transactions.

Potential Impact:

1. JVM Heap space may reduce rapidly.
2. Frequent JVM garbage collection calls.

Suggested Action(s) : Verify number of client connections created."

WeblogicEJBPoolWtRt

Description: Number of times no EJB beans were available from the free pool (drill down) per minute

Collection interval: HIGH

Policy: Weblogic_EJBPoolWaitCount

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0021.Weblogic_M0038 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of times per minute no EJBs were available from the free pool (<VALUE>/min) for application (<OPTION(applicationname)>) too high (>= <THRESHOLD>/min) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of times per minute no EJBs were available from the free pool has exceeded the threshold value.

The max-beans-in-free-pool element may have been set too low, or all instances of an EJB class may be active.

Potential Impact: New clients requesting an EJB class will be blocked until an active EJB completes a method call.

Suggested Action(s) : When EJBs are created, the session bean instance is created and given an identity.

When the client removes a bean, the bean instance is placed in the free pool.

When you create a subsequent bean, you can avoid object allocation by reusing the previous instance that is in the free pool.

The max-beans-in-free-pool element can improve performance if EJBs are frequently created and removed.

The container creates new instances of message beans as needed for concurrent message processing.

The max-beans-in-pool element puts an absolute limit on how many of these instances will be created.

The container may override this setting according to the runtime resources that are available.

For the best performance for stateless session and message beans, use the default setting max-beans-in-free-pool element. (The default is no limit.)

This way, you can run as many beans in parallel, using as many threads as possible.

The only reason to change the setting would be to limit the number of beans running in parallel or to limit access to an underlying resource.

For example, if you use stateless session EJBs to implement a legacy connection pool, you do not want to allocate more bean instance than the number of connections that can be supported by your legacy system.

For information on tuning EJB parameters, see the 'Performance and Tuning' documentation for your Weblogic Server version."

WeblogicGlbThrPIOvld

Description: Indicates an Overload Condition on General Thread pool

Collection interval: HIGH

Policy: Weblogic_ThreadPoolOverloadCondition

Aspect: Weblogic Thread Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0012.Weblogic_M0026 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 1.0

Message Text: Global thread pool overload condition has occurred. Further incoming requests will get rejected. See the annotated reports for details. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Global thread pool overload condition has occurred.

The total number of requests that can be present in the server (enqueued and those under execution) is exceeded.

Potential Impact: Further incoming requests will get rejected.

Suggested Action(s) : For information on tuning the execute thread pool, see the 'Performance and Tuning' documentation for your Weblogic Server version."

TranAppErrRbPct

Description: Percentage of transactions rolled back due to an application error

Collection interval: MEDIUM

Policy: Weblogic_TransactionAppErrorRollbackPercentage

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0139 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: % of transactions rolled back due to application error (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percent of transactions rolled back due to application errors has exceeded the threshold value.

Transactions are not successfully completing due to application errors.

Potential Impact: Fewer user requests are being successfully completed.

Suggested Action(s) : The administrator can monitor individual transactions from the Administration Console.

In addition to displaying statistics, the following information can also be displayed:

- 1) Transactions by name, including rollback and time active information.
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicEJBTranThruRt

Description: EJB Transaction Throughput Rate

Collection interval: MEDIUM

Policy: Weblogic_EJBTransactionThroughputRate

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0023.Weblogic_M0046 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10000

Message Text: # of EJB transactions per second (<VALUE>/sec) too high (>= <THRESHOLD>/sec)
[Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

1. Not sufficient Entity Beans in pooled state.
2. Entity Bean pool size is set to small value.

Potential Impact:

1. JVM Heap space may reduce rapidly.
2. Frequent JVM garbage collection calls.

Suggested Action(s) :

1. Verify Entity Bean pool size set.

"

WeblogicJVMMemUtilPct

Description: Percentage of heap space used in the JVM

Collection interval: HIGH NA

Policy: Weblogic_JVMHeapUsage

Aspect: Weblogic JVM Heap Memory

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0002.Weblogic_M0003 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 98, MAJOR / 95

Message Text: Rule1: % of heap space used (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Message Text: Rule2: % of heap space used (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The JVM is running out of available heap space. The JVM heap size may be set too low for the client load.

Potential Impact: The JVM heap size determines how often and how long the VM spends collecting garbage (de-allocating unused Java objects).

The Java heap is where the objects of a Java program live. When an object can no longer be reached from any pointer in the running program, the object is garbage.

Garbage collection affects performance because JVM work cannot proceed during full garbage collection.

An acceptable rate for garbage collection is application specific and should be adjusted after analysing the actual time and frequency of garbage collections.

The goal of tuning your heap size is to minimize the time that you spend doing garbage collection while maximizing the number of clients that you can handle at a given time.

If you set a large heap size, full garbage collection is slower, but it occurs less frequently. For a smaller heap size, full garbage collection is faster, but occurs more frequently.

Suggested Action(s) : For additional information on tuning your heap size, see the 'Performance and Tuning' documentation for your Weblogic Server version.

You can set the heap size using the options -Xms and -Xmx on the Java command line in the script used to start the server.

Use the -Xms option to set the minimum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

Use the -Xmx option to set the maximum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

As a general rule, set minimum heap size equal to the maximum heap size. If you are using 1.3 Java HotSpot JVM, also set generation sizes.

Make sure that the heap size is not larger than the available free RAM on your system.

Use as large a heap size as possible without causing your system to swap pages to disk.

The amount of free RAM on your system depends on your hardware configuration and the memory requirements of running processes on your machine.

See your system administrator for help in determining the amount of free RAM on your system.

Typically, you should use 80% of the available RAM (not taken by the operating system or other processes) for your JVM.

If you find that you have large amount of RAM remaining, run more Weblogic Servers on your machine."

NumWaitrCurrCnt

Description: Weblogic_NumWaitersCurrentCount

Collection interval: MEDIUM

Policy: Weblogic_NumWaitersCurrentCount

Aspect: Weblogic JCA Statistics

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0052.Weblogic_M0113 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: Number of waiters held up for a resource (<VALUE>) too high (>= <THRESHOLD>)
[Policy: <NAME>]

Instruction Text:

Probable Cause(s) : number of waiters held up for a resource

Potential Impact: Fewer user requests are being successfully completed.

Suggested Action(s) : The Weblogic administrator should check the connected systems and ensure they are functioning correctly.

In addition, the administrator can monitor transactions from the Administration Console.

This includes:

1) Transactions by name, including rollback and time active information.

- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicJDBCConPIWtCnt

Description: Number of clients waiting for a connection from connection pools

Collection interval: MEDIUM

Policy: Weblogic_RequestsWaitingForConnection

Aspect: Weblogic JCA Statistics

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0045.Weblogic_M0097 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of clients waiting for a connection from connection pools (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of clients waiting for a connection has exceeded the threshold value.

The size of the connection pool is too small relative to the number of current client sessions that require JDBC Connections.

Potential Impact: Client connection requests will be forced to wait for an available connection from the connection pool.

Suggested Action(s) : Increase the maximum size of the connection pool.

A good rule of thumb is that the maximum size of the connection pool should be equal to the number of Execute Threads configured in the Weblogic Server.

This assumes that each thread uses one transaction to service a request and therefore needs just one connection.

If this is not the case, then a slightly larger connection pool may be more efficient.

The connection pool minimum size should be equal to the maximum size.

This ensures that all database connections are acquired during server start-up and not when the server is under load."

WeblogicReqMaxWaitTime

Description: Maximum time a request had to wait for a thread

Collection interval: MEDIUM

Policy: Weblogic_RequestMaxWaitTime

Aspect: Weblogic Cache Usage

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0058.Weblogic_M0129 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 1000

Message Text: Maximum time a request had to wait for a thread (<VALUE> ms) too high (>= <THRESHOLD> ms) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : This metric is used to monitor the maximum time a request had to wait for a thread.

Potential Impact:

1. Applications may respond slow.
2. Applications may block on threads to be available.
3. Application deadlock situation may arise.

Suggested Action(s) :

1. Increase Thread pool size or mark them growable.

This metric is used to monitor the maximum time a request had to wait for a thread."

WeblogicJMSUtilByBytPct

Description:

Collection interval: MEDIUM NA

Policy: Weblogic_JMSUtilizationByBytesPercentage

Aspect: Weblogic JMS Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0035.Weblogic_M0077 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 98, MAJOR / 95

Message Text: Rule1: % of JMS queue filled by byte count (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Message Text: Rule2: % of JMS queue filled by byte count (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The JMS Server queue utilization is greater than the threshold value.

The size of the queue may be set too low.

Potential Impact: Once the queue reaches one hundred percent capacity, users will not be able to deliver messages to this queue.

Suggested Action(s) : If possible, the administrator may want to increase the size of the queue via the Administration Server console.

The administrator can also inspect the individual destinations within this JMS Server via the console to determine which destination queues are having problems."

Weblogic_ClusterHealth

Description: Health of the cluster

Collection interval: MEDIUM

Policy: Weblogic_ClusterHealthStatus

Aspect: Weblogic Cluster Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0074.Weblogic_M0160 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 4.500000

Message Text: Cluster Health is poor. [Policy: <NAME>]

Instruction Text:

Probable Cause(s) :

1. Load balancing is not proper.
2. Network latency is more.

Potential Impact:

1. If cluster fails, disaster recovery might be impacted.
2. Availability might come down.

Suggested Action(s) :

1. Check load balancer's behaviour.
2. Choose sticky routing algorithm.

"

WeblogicWaitSecHighCnt

Description: The longest connection reserve wait time in seconds

Collection interval: MEDIUM

Policy: Weblogic_WaitSecondsHighCount

Aspect: Weblogic JCA Statistics

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0045.Weblogic_M0205 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: The longest connection reserve wait time in seconds (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : When a completed getConnection request takes longer to return a connection than any previous request, the reserve wait time will get increased.

Potential Impact: user request throughput will be reduced

Suggested Action(s) : The Weblogic administrator should check the connected systems and ensure they are functioning correctly.

In addition, the administrator can monitor transactions from the Administration Console.

This includes:

- 1) Transactions by name, including rollback and time active information.
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicAllProcAvgLd

Description: All Processors Average Load

Collection interval: HIGH

Policy: Weblogic_ProcessorsAverageLoad

Aspect: Weblogic JVM Heap Memory

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0003.Weblogic_M0010 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MAJOR / 90

Message Text: % of average load on all processors (<VALUE>%) too high (>= <THRESHOLD>%)

[Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Total Garbage Collection Threads is High.

Potential Impact: The JVM heap size determines how often and how long the VM spends collecting garbage (de-allocating unused Java objects).

The Java heap is where the objects of a Java program live. When an object can no longer be reached from any pointer in the running program, the object is garbage.

Garbage collection affects performance because JVM work cannot proceed during full garbage collection.

An acceptable rate for garbage collection is application specific and should be adjusted after analysing the actual time and frequency of garbage collections.

The goal of tuning your heap size is to minimize the time that you spend doing garbage collection while maximizing the number of clients that you can handle at a given time.

If you set a large heap size, full garbage collection is slower, but it occurs less frequently. For a smaller heap size, full garbage collection is faster, but occurs more frequently.

Suggested Action(s) : For additional information on tuning your heap size, see the 'Performance and Tuning' documentation for your Weblogic Server version.

You can set the heap size using the options -Xms and -Xmx on the Java command line in the script used to start the server.

Use the -Xms option to set the minimum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

Use the -Xmx option to set the maximum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

As a general rule, set minimum heap size equal to the maximum heap size. If you are using 1.3 Java HotSpot JVM, also set generation sizes.

Make sure that the heap size is not larger than the available free RAM on your system.

Use as large a heap size as possible without causing your system to swap pages to disk.

The amount of free RAM on your system depends on your hardware configuration and the memory requirements of running processes on your machine.

See your system administrator for help in determining the amount of free RAM on your system.

Typically, you should use 80% of the available RAM (not taken by the operating system or other processes) for your JVM.

If you find that you have large amount of RAM remaining, run more Weblogic Servers on your machine."

WeblogicJDBCConLkRt

Description: Rate of leaked connections for the JDBC connection pool

Collection interval: MEDIUM

Policy: Weblogic_JDBCConnectionLeakRate

Aspect: Weblogic JDBC Connection Pool Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0045.Weblogic_M0101 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 100

Message Text: Rate of leaked connections for the JDBC connection pool(<VALUE>) belongs to application is too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The rate of new leaked JDBC connections has exceeded the threshold value.

JDBC connection leaks represent connections that were checked out of the connection pool but never returned with a close() method.

Leaked connections cannot be used to fulfil later connection requests

Potential Impact: When a connection is closed, the connection is then available for a future connection request.

If the application fails to close the connection, the connection pool can be exhausted of its available connections, and future connection requests can therefore fail.

Suggested Action(s) : Correct the faulty application component.

Connection pools provide ready-to-use pools of connections to a database, therefore eliminating the overhead of creating each connection when as needed by the application.

When finished with a connection, applications must return the connection to the connection pool.

For information on managing JDBC connections, see the 'Programming Weblogic JDBC' documentation for your Weblogic Server version."

WeblogicServletReqRate

Description: Number of requests for a servlet per second

Collection interval: MEDIUM

Policy: Weblogic_ServletRequestRate

Aspect: Weblogic Servlet Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0030.Weblogic_M0065 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10000

Message Text: # of requests for application (<OPTION(applicationname)>) for a servlet (<VALUE>/sec) too high (>= <THRESHOLD>/sec) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Sudden increase in Client transactions.

Potential Impact: Servlet average response time may increase.

Suggested Action(s) : In a clustered environment, keep checking load balancer performance."

WeblogicCnctrPoolUtil

Description: Percentage utilization of available JCA connections in connection pool

Collection interval: MEDIUM NA

Policy: Weblogic_JCAConnectionsUtilizationPct

Aspect: Weblogic JCA Statistics

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0052.Weblogic_M0107 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 98, MAJOR / 95

Message Text: Rule1: % utilization of available JCA connections in connection pool (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Message Text: Rule2: % utilization of available JCA connections in connection pool (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The utilization of a J2EE Connector connection pool (The number of connections in the pool that are being used) has exceeded a threshold value.

The number of requested connections to a resource is approaching or has reached the maximum allowed.

Potential Impact: As ManagedConnections are created over time, the amount of system resources-such as memory and disk space-that each ManagedConnection consumes increases and may affect the performance of the overall system.

If a new ManagedConnection needs to be created during a connection request, Weblogic Server ensures that no more than the maximum number of allowed ManagedConnections are created.

If the maximum number is reached, Weblogic Server attempts to recycle a ManagedConnection from the connection pool.

However, if there are no connections to recycle, a warning is logged indicating that the attempt to recycle failed and that the connection request can only be granted for the amount of connections up to the allowed maximum amount.

Suggested Action(s) : Weblogic Server allows you to configure a setting for the allowed maximum number of allocated connections.

For information on managing J2EE connections, see the 'Connection Management' section of the 'Programming Weblogic J2EE Connectors' documentation for your Weblogic Server version."

Weblogic_ClsInMsFailRt

Description: Number of multicast messages from cluster lost by server per minute

Collection interval: MEDIUM

Policy: Weblogic_ClusterOutMessageFailRate

Aspect: Weblogic Cluster Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0074.Weblogic_M0159 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: # of multicast messages from cluster lost by server (<VALUE>/min) too high (>= <THRESHOLD>/min) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of multicast messages from the cluster that were lost by the server has exceeded the threshold value.

This could be caused by the cluster configuration or the network topology.

Potential Impact: Potential loss of critical data.

Suggested Action(s) : Because multicast controls critical functions related to detecting failures and maintaining the cluster-wide JNDI tree.

It is important that neither the cluster configuration nor the basic network topology interfere with multicast communication.

Always consider the following rules when configuring or planning a Weblogic Server cluster.

For most deployments, limiting clustered servers to a single subnet ensures that multicast messages are reliably transmitted.

In special cases, however, you may want to distribute a Weblogic Server cluster across subnets in a Wide Area Network (WAN).

This may be desirable to increase redundancy in a clustered deployment, or to distribute clustered instances over a larger geographical area.

If you choose to distribute a cluster over a WAN (or across multiple subnets), you must plan and configure your network topology to ensure that multicast messages are reliably transmitted to all servers in the cluster.

Specifically, your network must meet the following requirements:

- 1) The network must fully support IP multicast packet propagation. In other words, all routers and other tunnelling technologies must be configured to propagate multicast messages to clustered instances.
- 2) The network latency must be sufficiently small as to ensure that most multicast messages reach their final destination in 200 to 300 milliseconds.
- 3) The multicast Time-To-Live (TTL) value must be high enough to ensure that routers do not discard multicast packets before they reach their final destination.

Note: Distributing a Weblogic Server cluster over a WAN may require network facilities in addition to the multicast requirements described above.

For example, you may want to configure load balancing hardware to ensure that client requests are directed to servers in the most efficient manner (to avoid unnecessary network hops)."

WeblogicPendingReqPct

Description: Percentage of requests pending

Collection interval: MEDIUM NA

Policy: Weblogic_PendingRequestPercentage

Aspect: Weblogic Cache Usage

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0058.Weblogic_M0124 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MAJOR / 98, WARNING / 95

Message Text: Rule1: Percentage of pending requests (<VALUE>%) too high (>= <THRESHOLD>%)
[Policy: <NAME>]

Message Text: Rule2: Percentage of pending requests (<VALUE>%) too high (>= <THRESHOLD>%)
[Policy: <NAME>]

Instruction Text:

Probable Cause(s): This metric is used to monitor the percentage of the requests that are pending because they are waiting for an available thread.

Potential Impact:

1. Applications may respond slow.
2. Applications may block on threads to be available.
3. Application deadlock situation may arise

Suggested Action(s) :

1. Increase Thread pool size or mark them growable.

This metric is used to monitor the percentage of the requests that are pending because they are waiting for an available thread."

WeblogicExQueWaitCnt

Description: The metric monitors an execute queue and its associated thread pool for each server. This metric particularly monitors the number of client requests waiting to be serviced

Collection interval: HIGH

Policy: Weblogic_ExecuteQMetricMonitors

Aspect: Weblogic Servlet Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0012.Weblogic_M0024 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 10

Message Text: # of requests waiting to be serviced (<VALUE>) too high (>= <THRESHOLD>)

[Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of client requests waiting to be serviced has exceeded a threshold value.

The rate of incoming requests has exceeded the number of threads available to perform the work.

Potential Impact: Degradation in performance from a client perspective.

Suggested Action(s) : Although client requests are waiting for an execute thread to be allocated, it is important to note that adding more threads does not necessarily imply that you can process more work.

Even if you add more threads, you are still limited by the power of your processor.

You can degrade performance by increasing this value unnecessarily.

Because threads are resources that consume memory, a very high execute thread count causes more memory to be used and increases context switching. This degrades your performance.

If this condition persists, you may need to upgrade your processor power. Another solution is to simply add resources.

If your Weblogic server is configured in a cluster, then to increase the load handling capabilities you can add another Weblogic server to the cluster.

Given a well-designed application, adding additional servers should provide linear scalability.

For information on tuning the execute thread pool, see the 'Performance and Tuning' documentation for your Weblogic Server version."

WeblogicTranRIBkPct

Description: Percentage of transactions rolled back, based on the total

Collection interval: MEDIUM

Policy: Weblogic_TransactionRollbackPercentage

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0135 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: % of transactions rolled back (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percentage of transactions rolled back has exceeded the threshold value.

Application design issues or resource issues.

Potential Impact: User requests are not being successfully completed.

Suggested Action(s) : The Weblogic administrator should check the necessary database systems and ensure they are functioning correctly.

In addition, the administrator should check the following configurable transaction attributes:

Timeout Seconds - The time a transaction may be active before the system forces a rollback.

Abandon Timeout Seconds - The maximum time that a transaction coordinator persists in attempting to complete a transaction.

Before Completion Iteration Limit - The number of beforeCompletion callbacks that are processed before a system forces a rollback.

The administrator can monitor individual transactions from the Administration Console. In addition to displaying statistics, the following information can also be displayed:

- 1) Transactions by name, including rollback and time active information
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicDestroyedTICnt

Description: Total number of times a bean instance from the pool was destroyed due to a non-application Exception being thrown from it

Collection interval: HIGH

Policy: Weblogic_EJBDestroyedTotalCount

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0021.Weblogic_M0180 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of times a bean instance from this pool was destroyed due to a non-application Exception being thrown from it (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Provides a count of the total number of times a bean instance from this pool was destroyed due to a non-application Exception being thrown from it.

Potential Impact: The Business logic associated with this EJB will not have completed successfully. This will impact the application response time.

Suggested Action(s) : Check the exceptions thrown and refer the documentation for your Weblogic Server version."

WeblogicEJBTranRbRt

Description: EJB Transaction Rollback Rate

Collection interval: MEDIUM

Policy: Weblogic_NumberEJBTransactionRollBackRate

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0023.Weblogic_M0052 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 1

Message Text: # of EJB transactions rolled back per second (<VALUE>/sec) too high (>= <THRESHOLD>/sec) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of EJB transactions rolled back per second has exceeded the threshold value.

Application design or resource issues.

Potential Impact: Fewer user requests are being successfully completed.

Suggested Action(s) : The Weblogic administrator should check the necessary database systems and ensure they are functioning correctly.

In addition, the administrator can monitor transactions from the Administration Console.

This includes:

1) Transactions by name, including rollback and time active information.

- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicClusterInMessageFailureRate

Description: Number of multicast messages to cluster that were re-sent per minute

Collection interval: MEDIUM

Policy: Weblogic_ClusterInMessageFailureRate

Aspect: Weblogic Cluster Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0074.Weblogic_M0157 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: # of multicast messages to cluster that were resent (<VALUE>/min) too high (>= <THRESHOLD>/min) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of multicast messages to the cluster that were resent has exceeded the threshold value.

This could be caused by the cluster configuration or the network topology.

Potential Impact: Potential loss of multicast packets.

Suggested Action(s) : Because multicast controls critical functions related to detecting failures and maintaining the cluster-wide JNDI tree.

It is important that neither the cluster configuration nor the basic network topology interfere with multicast communication.

Always consider the following rules when configuring or planning a Weblogic Server cluster.

For most deployments, limiting clustered servers to a single subnet ensures that multicast messages are reliably transmitted.

In special cases, however, you may want to distribute a Weblogic Server cluster across subnets in a Wide Area Network (WAN).

This may be desirable to increase redundancy in a clustered deployment, or to distribute clustered instances over a larger geographical area.

If you choose to distribute a cluster over a WAN (or across multiple subnets), you must plan and configure your network topology to ensure that multicast messages are reliably transmitted to all servers in the cluster.

Specifically, your network must meet the following requirements:

- 1) The network must fully support IP multicast packet propagation. In other words, all routers and other tunneling technologies must be configured to propagate multicast messages to clustered instances.
- 2) The network latency must be sufficiently small as to ensure that most multicast messages reach their final destination in 200 to 300 milliseconds.
- 3) The multicast Time-To-Live (TTL) value must be high enough to ensure that routers do not discard multicast packets before they reach their final destination.

Note: Distributing a Weblogic Server cluster over a WAN may require network facilities in addition to the multicast requirements described above.

For example, you may want to configure load balancing hardware to ensure that client requests are directed to servers in the most efficient manner (to avoid unnecessary network hops)."

ConnRejectedTotalCnt

Description: Weblogic_ConnectionsRejectedTotalCount

Collection interval: MEDIUM

Policy: Weblogic_ConnectionsRejectedTotalCount

Aspect: Weblogic JCA Statistics

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0052.Weblogic_M1113 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: Number of rejected requests for a Connector connections in Connector Pool since the pool is instantiated (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : number of rejected requests for a Connector connections in this Connector Pool since the pool is instantiated

Potential Impact: Fewer user requests are being successfully completed.

Suggested Action(s) : The Weblogic administrator should check the connected systems and ensure they are functioning correctly.

In addition, the administrator can monitor transactions from the Administration Console.

This includes:

- 1) Transactions by name, including rollback and time active information.
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

Weblogic_ServerStatus

Description: Status of a server, monitors whether running or not

Collection interval: HIGH NA

Policy: Weblogic_ServerStatus

Aspect: Weblogic Server Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0001.Weblogic_M0001 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 0.5, WARNING / 1.5

Message Text: Rule1: Server status: Unknown (down) [Policy: <NAME>]

Message Text: Rule2: Server status: Suspended [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : For each server, this metric will report the status (running, suspended, or unknown).

If the server is not in a running state, the following events may have occurred:

- 1) The Weblogic Administrator has selected 'Shutdown this server' from the Administration console.
- 2) The Weblogic Administrator has selected 'Suspend this server' from the Administration console.
- 3) The server may have gone down for other reasons.

Potential Impact: If the server is Shutdown or in the process of shutting down, the server will no longer be available.

If the server is Suspended, it will only accept requests from the Administration Server.

Note that suspending the Weblogic Server only suspends server responses to HTTP requests. Java applications and RMI invocations are not suspended.

Suggested Action(s) : If the designated server is not running, the Weblogic Administrator should start the server using the appropriate script.

It is important to note whether this is the Administration Server or a Managed Server, since the startup script will be different for each type.

If the server has been suspended, it may have been placed in this state for a reason.

A typical use of this feature would be in a situation where a Weblogic Server is running as a 'hot' backup for another server.

When it is OK to do so, execute the 'Resume this server' command from the Administration console."

WeblogicEJBMssdCntRtSm

Description: Number of times a failed attempt was made to get an instance from the free pool

Collection interval: HIGH

Policy: Weblogic_SumOfEJBMissedCountRate

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0021.Weblogic_M0183 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of times a failed attempt was made to get an instance from the free pool (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The total number of times a failed attempt was made to get an instance from the free pool.

Potential Impact: An Attempt to get a bean from the pool will fail if there are no available instances in the pool.

This results in delayed execution of the business logic and hence slows the response time of the application.

Suggested Action(s) : Verify that the max-beans-in-free-pool element has not been set too low.

Also, while Weblogic Server will always try to allocate a new bean instance if one is not available, in reality you are limited by the number of executable threads.

In most cases, each thread will need, at most, a single bean instance.

For information on tuning EJB parameters, see the 'Performance and Tuning' documentation for your Weblogic Server version."

WeblogicJRktRTTtotGCTme

Description: Total Garbage Collection Time

Collection interval: MEDIUM

Policy: Weblogic_GarbageCollectionTime

Aspect: Weblogic JVM Heap Memory

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0003.Weblogic_M0006 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 2000

Message Text: Time spent on garbage collection runs (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Total Garbage Collection Time is High. The JVM heap size may be set too high for the client load.

Potential Impact: The JVM heap size determines how often and how long the VM spends collecting garbage (de-allocating unused Java objects).

The Java heap is where the objects of a Java program live. When an object can no longer be reached from any pointer in the running program, the object is garbage.

Garbage collection affects performance because JVM work cannot proceed during full garbage collection.

An acceptable rate for garbage collection is application specific and should be adjusted after analysing the actual time and frequency of garbage collections.

The goal of tuning your heap size is to minimize the time that you spend doing garbage collection while maximizing the number of clients that you can handle at a given time.

If you set a large heap size, full garbage collection is slower, but it occurs less frequently. For a smaller heap size, full garbage collection is faster, but occurs more frequently.

Suggested Action(s) : For additional information on tuning your heap size, see the 'Performance and Tuning' documentation for your Weblogic Server version.

You can set the heap size using the options -Xms and -Xmx on the Java command line in the script used to start the server.

Use the -Xms option to set the minimum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

Use the -Xmx option to set the maximum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

As a general rule, set minimum heap size equal to the maximum heap size. If you are using 1.3 Java HotSpot JVM, also set generation sizes.

Make sure that the heap size is not larger than the available free RAM on your system.

Use as large a heap size as possible without causing your system to swap pages to disk.

The amount of free RAM on your system depends on your hardware configuration and the memory requirements of running processes on your machine.

See your system administrator for help in determining the amount of free RAM on your system.

Typically, you should use 80% of the available RAM (not taken by the operating system or other processes) for your JVM.

If you find that you have large amount of RAM remaining, run more Weblogic Servers on your machine."

WeblogicTransactionHeuristicCount

Description: Number of transactions returning a heuristic decision

Collection interval: MEDIUM

Policy: Weblogic_TransactionHeuristicsTotalCount

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0151 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: Number of transactions returning a heuristic decision (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percentage of transactions returning a heuristic decision has exceeded the threshold value.

A heuristic completion (or heuristic decision) occurs when a resource makes a unilateral decision during the completion stage of a distributed transaction to commit or rollback updates.

Network failures or transaction timeouts are possible causes for heuristic completion.

Potential Impact: A heuristic decision can leave distributed data in an indeterminate state.

Suggested Action(s) : In the event of a heuristic decision, one of the following heuristic outcome exceptions may be thrown:

1) HeuristicRollback - one resource participating in a transaction decided to autonomously rollback its work, even though it agreed to prepare itself and wait for a commit decision.

If the Transaction Manager decided to commit the transaction, the resource's heuristic rollback decision was incorrect, and might lead to an inconsistent outcome since other branches of the transaction were committed.

2) HeuristicCommit - one resource participating in a transaction decided to autonomously commit its work, even though it agreed to prepare itself and wait for a commit decision.

If the Transaction Manager decided to rollback the transaction, the resource's heuristic commit decision was incorrect, and might lead to an inconsistent outcome since other branches of the transaction were rolled back.

3) HeuristicMixed - the Transaction Manager is aware that a transaction resulted in a mixed outcome, where some participating resources committed and some rolled back.

The underlying cause was most likely heuristic rollback or heuristic commit decisions made by one or more of the participating resources.

4) HeuristicHazard - the Transaction Manager is aware that a transaction might have resulted in a mixed outcome, where some participating resources committed and some rolled back.

But system or resource failures make it impossible to know for sure whether a Heuristic Mixed outcome definitely occurred.

The underlying cause was most likely heuristic rollback or heuristic commit decisions made by one or more of the participating resources.

When an heuristic completion occurs, a message is written to the server log.

Refer to your database vendor documentation for instructions on resolving heuristic completions.

Some resource managers save context information for heuristic completions.

This information can be helpful in resolving resource manager data inconsistencies.

If the ForgetHeuristics attribute is selected (set to true) on the JTA panel of the Weblogic Console, this information is removed after an heuristic completion.

When using a resource manager that saves context information, you may want to set the ForgetHeuristics attribute to false."

WeblogicTrnRIBkSysToCt

Description: Percentage of transactions rolled back due to system error

Collection interval: MEDIUM

Policy: Weblogic_TransactionSystemErrorRollbackPercentage

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0149 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: % of transactions rolled back due to system error (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percent of transactions rolled back due to system errors has exceeded the threshold value.

Transactions are not successfully completing due to system errors.

Potential Impact: Fewer user requests are being successfully completed.

Suggested Action(s) : The administrator can monitor individual transactions from the Administration Console.

In addition to displaying statistics, the following information can also be displayed:

- 1) Transactions by name, including rollback and time active information.
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicTranCapUtil

Description: Percentage utilization of transaction capacity

Collection interval: MEDIUM NA

Policy: Weblogic_TransactionCapacityUtilizationPct

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0153 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 98, MAJOR / 95

Message Text: Rule1: % utilization of transaction capacity (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Message Text: Rule2: % utilization of transaction capacity (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of simultaneous in-progress transactions (as a percent of the maximum number of transactions allowed in the server) has exceeded a threshold value.

Potential Impact: Sudden spike in transactions roll back rate may be observed.

Suggested Action(s) :

1. Restrict simultaneous concurrent transactions.
2. Verify ACID properties of simultaneous transactions.

The number of simultaneous in-progress transactions (as a percent of the maximum number of transactions allowed in the server) has exceeded a threshold value."

WeblogicJMSThrByMsgPct

Description: The amount of time at which the server threshold condition was satisfied, based on the number of messages

Collection interval: MEDIUM

Policy: Weblogic_JMSMessagesThresholdTime

Aspect: Weblogic JMS Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0036.Weblogic_M0084 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: Number of times queue threshold condition was satisfied by message count (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The amount of time this JMS queue has spent in the threshold condition has exceeded the threshold value.

The JMS Server message queue threshold condition for the number of messages stored, as configured in the administration console, has been satisfied for a significant amount of time.

Potential Impact: Once the queue reaches one hundred percent capacity, users will not be able to deliver messages to this queue.

Suggested Action(s) : If possible, the administrator may want to increase the size of the queue via the Administration Server console.

The administrator can also inspect the individual destinations within this JMS Server via the console to determine which destination queues are having problems."

WeblogicEJBCacheHitPct

Description: Percentage of EJBs in the cache in use

Collection interval: MEDIUM

Policy: Weblogic_EJBCacheHitPercentage

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0025.Weblogic_M0054 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 90

Message Text: % of EJBs in the cache in use (<VALUE>%) for application (<OPTION (applicationname)>) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The utilization of the EJB cache has exceeded a threshold value.

The cache size may be set too low.

Potential Impact: When the maximum cache size is reached, Weblogic Server passivates (transfer from memory to secondary storage) some EJBs that have not been recently used by a client.

This could result in performance degradation.

Suggested Action(s) : Set the max-beans-in-cache attribute in the weblogic-ejb-jar.xml file to a higher value.

Tuning this value too high could consume memory unnecessarily.

For information on tuning EJB parameters, see the 'Performance and Tuning' documentation for your Weblogic Server version."

WeblogicJDBCConFail

Description: The number of times that the data source attempted to refresh a database connection and failed

Collection interval: MEDIUM

Policy: Weblogic_FailuresToReconnectCount

Aspect: Weblogic JDBC Connection Pool Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0045.Weblogic_M0104 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: JDBC connection pool failures (<VALUE>) for application is too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of times a connection pool attempted to refresh a connection to a database and failed exceeds the threshold.

This failure may happen because of database unavailability or broken connection to the database.

Potential Impact: Client connection requests to the database may fail.

Suggested Action(s) : For information on managing JDBC connections, see the 'Programming Weblogic JDBC' documentation for your Weblogic Server."

WeblogicPendingReqCount

Description: Number of pending requests

Collection interval: MEDIUM

Policy: Weblogic_PendingRequestCount

Aspect: Weblogic Cache Usage

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0058.Weblogic_M0123 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 100

Message Text: Number of pending requests (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Small Thread pool size is chosen

Potential Impact:

1. Applications may respond slow.
2. Applications may block on threads to be available.
3. Application deadlock situation may arise.

Suggested Action(s) :

1. Increase Thread pool size or mark them gorwable.

This metric is used to monitor the number of requests that are pending because they are waiting for an available thread."

WeblogicExQThrdUtilPct

Description: Percentage of threads in use for a servers execute queue. For Weblogic Server version 9.x and 10.x, there is only one execute queue

Collection interval: HIGH NA NA

Policy: Weblogic_ExecuteQThreadsInUse

Aspect: Weblogic Servlet Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0012.Weblogic_M0021 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 90, MAJOR / 85, MINOR / 80

Message Text: Rule1: % of execute queue threads used (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Message Text: Rule2: % of execute queue threads used (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Message Text: Rule3: % of execute queue threads used (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The utilization of the Weblogic server execute threads has exceeded a threshold value.

The number of incoming client requests has resulted in all the execute threads being allocated.

Potential Impact: At 100% utilization, the Weblogic server will not have any threads available to service incoming requests.

Suggested Action(s) : For additional information on tuning the execute thread pool, see the 'Performance and Tuning' documentation for your Weblogic Server version.

Systems administrators can increase the total number of execute threads via the administrator's console.

However, it should be noted that adding more threads does not necessarily imply that you can process more work.

Even if you add more threads, you are still limited by the power of your processor. You can degrade performance by increasing this value unnecessarily.

Because threads are resources that consume memory, a very high execute thread count causes more memory to be used and increases context switching. This degrades your performance.

The value of the Thread Count depends very much on the type of work the application does.

For example, if your client application is thin and does a lot of its work through remote invocation, the time your client application spends connected will be greater than for a client application that does a lot of client-side processing.

So, if you do not need to use the additional threads for your work then you should not change the value of this attribute. The thread will not be held for the client application.

If your application makes database calls that take a long time to return, you need more execute threads than an application that makes calls that are short and turn over very rapidly.

For the latter, you can use a small number of execute threads and improve performance.

The following scenarios can be used as a guideline for setting the ThreadCount:

Thread Count < number of CPUs Increase the thread count

Thread Count = number of CPUs Increase the thread count

Thread Count > number of CPUs Practically ideal, although some by a moderate number of threads, tuning may be necessary

Thread Count > number of CPUs Reduce the number of threads by a significant number

"

WeblogicEJBTimeoutRt

Description: Number of times a client timed out waiting for an EJB bean (drill down) per minute

Collection interval: HIGH

Policy: Weblogic_EJBTimeoutRate

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0021.Weblogic_M0042 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of times per minute a client timed out for application (<OPTION(applicationname)>) waiting for an EJB (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of times per minute a client timed out waiting for an EJB has exceeded the threshold value.

If all instances of an EJB class are active and max-beans-in-free-pool has been reached, new clients requesting the EJB class will be blocked until an active EJB completes a method call.

Potential Impact: If the transaction times out (or, for non-transactional calls, if five minutes elapse), Weblogic Server throws a RemoteException.

Suggested Action(s) : Verify that the max-beans-in-free-pool element has not been set too low.

Also, while Weblogic Server will always try to allocate a new bean instance if one is not available, in reality you are limited by the number of executable threads.

In most cases, each thread will need, at most, a single bean instance.

For information on tuning EJB parameters, see the 'Performance and Tuning' documentation for your Weblogic Server version.

This metric is used to monitor the number of threads in the standby pool.

Surplus threads that are not needed to handle the present work load are designated as standby and added to the standby pool.

These threads are activated when more threads are needed. The value of this count must be in an acceptable range to meet performance criteria."

WeblogicStandbyThrdCnt

Description: Number of threads in the standby pool

Collection interval: HIGH

Policy: Weblogic_StandbyThreadCount

Aspect: Weblogic Cache Usage

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0012.Weblogic_M0034 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: Number of threads in the standby pool (<VALUE>) too low (less than or equal to <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : This metric is used to monitor the number of threads in the standby pool.

Surplus threads that are not needed to handle the present work load are designated as standby and added to the standby pool.

These threads are activated when more threads are needed. The value of this count must be in an acceptable range to meet performance criteria.

Potential Impact:

1. Thread pool saturation condition may occur.
2. CPU utilization may consistently keep shooting up.

Suggested Action(s) :

1. To fix a saturated thread pool, keep changing the thread pool size in steps until CPU utilization reaches between 75 and 85 percent.
2. Tune the application using a code profiling tool.

"

TranResErrRbPct

Description: Percentage of transactions rolled back due to resource error

Collection interval: MEDIUM

Policy: Weblogic_TransactionResErrorRollbackPercentage

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0138 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: % of transactions rolled back due to resource error (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The percent of transactions rolled back due to resource errors has exceeded the threshold value.

Transactions are not successfully completing due to resource errors.

Potential Impact: Fewer user requests are being successfully completed.

Suggested Action(s) : The administrator can monitor individual transactions from the Administration Console.

In addition to displaying statistics, the following information can also be displayed:

- 1) Transactions by name, including rollback and time active information.
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicInvLoginAttCnt

Description: Number of invalid login attempts

Collection interval: HIGH

Policy: Weblogic_InvalidLoginAttemptsCount

Aspect: Weblogic Authentication

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0077.Weblogic_M0162 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 2

Message Text: # of invalid login attempts (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s): The number of invalid login attempts has exceeded the threshold value. This could be an attempted security breach.

Potential Impact: If the security breach is successful, the security of the Weblogic Server environment could be compromised.

Suggested Action(s): If the invalid login attempts is repeated frequently, you may wish to implement the weblogic.security.audit package.

This will allow you to review the audit records to determine if there has been a security breach or an attempted security breach."

WeblogicTranAvgTime

Description: Average commit time for transactions

Collection interval: MEDIUM

Policy: Weblogic_TransactionAverageTime

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0131 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 100

Message Text: Average commit time for transactions (<VALUE>ms) too high (>= <THRESHOLD>ms) [Policy: <NAME>]

Instruction Text:

Probable Cause(s): The average commit time for a transaction has exceeded the threshold value. This may be an indication of system load.

Potential Impact: Degradation in the transaction throughput rate for the Weblogic Server.

Suggested Action(s) : The administrator can monitor individual transactions from the Administration Console.

In addition to displaying statistics, the following information can also be displayed:

- 1) Transactions by name, including rollback and time active information.
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicEJBTimeoutRtSm

Description: Number of times a client timed out waiting for an EJB bean per minute

Collection interval: HIGH

Policy: Weblogic_EJBTimeoutCount

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0021.Weblogic_M0041 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of times per minute a client timed out waiting for an EJB (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of times per minute a client timed out waiting for an EJB has exceeded the threshold value.

If all instances of an EJB class are active and max-beans-in-free-pool has been reached, new clients requesting the EJB class will be blocked until an active EJB completes a method call.

Potential Impact: If the transaction times out (or, for non-transactional calls, if five minutes elapse), Weblogic Server throws a RemoteException.

Suggested Action(s) : Verify that the max-beans-in-free-pool element has not been set too low.

Also, while Weblogic Server will always try to allocate a new bean instance if one is not available, in reality you are limited by the number of executable threads. In most cases, each thread will need, at most, a single bean instance.

For information on tuning EJB parameters, see the 'Performance and Tuning' documentation for your Weblogic Server version."

WeblogicDeferredReqCnt

Description: Number of deferred requests

Collection interval: MEDIUM

Policy: Weblogic_DeferredRequestsCount

Aspect: Weblogic Cache Usage

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0057.Weblogic_M0121 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 100

Message Text: Deferred requests count (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s): This metric is used to monitor the number of requests that were denied a thread for execution because of the max-threads-constraint constraint.

Max-threads-constraint is one of the work manager components that you can use to control the performance of your application by referencing the name of the component in the application's deployment descriptor.

The max-threads-constraint constraint limits the number of concurrent threads executing requests from the constrained work set. The default is unlimited.

Once the constraint is reached the server does not schedule requests of this type until the number of concurrent executions falls below the limit.

Potential Impact:

1. Application response time may increase with further deferrals.

2. Application may be brought into a suspended state.

3. Application deadlock situation may arise.

Suggested Action(s) :

1. If possible, increase the max-threads-constraint for this application.

This metric is used to monitor the number of requests that were denied a thread for execution because of the max-threads-constraint constraint.

Max-threads-constraint is one of the work manager components that you can use to control the performance of your application by referencing the name of the component in the application's deployment descriptor.

The max-threads-constraint constraint limits the number of concurrent threads executing requests from the constrained work set. The default is unlimited.

Once the constraint is reached the server does not schedule requests of this type until the number of concurrent executions falls below the limit."

WeblogicEJBPoolWtRtSum

Description: Number of times no EJB beans were available from the free pool per minute

Collection interval: HIGH

Policy: Weblogic_EJBBeanUnavailableCount

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0021.Weblogic_M0037 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of times per minute no EJBs were avavailable from the free pool (<VALUE>/min) too high (>= <THRESHOLD>/min) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The number of times per minute no EJBs were available from the free pool has exceeded the threshold value.

The max-beans-in-free-pool element may have been set too low, or all instances of an EJB class may be active.

Potential Impact: New clients requesting an EJB class will be blocked until an active EJB completes a method call.

Suggested Action(s) : When EJBs are created, the session bean instance is created and given an identity.

When the client removes a bean, the bean instance is placed in the free pool.

When you create a subsequent bean, you can avoid object allocation by reusing the previous instance that is in the free pool.

The max-beans-in-free-pool element can improve performance if EJBs are frequently created and removed.

The container creates new instances of message beans as needed for concurrent message processing.

The max-beans-in-pool element puts an absolute limit on how many of these instances will be created.

The container may override this setting according to the runtime resources that are available.

For the best performance for stateless session and message beans, use the default setting max-beans-in-free-pool element. (The default is no limit.)

This way, you can run as many beans in parallel, using as many threads as possible.

The only reason to change the setting would be to limit the number of beans running in parallel or to limit access to an underlying resource.

For example, if you use stateless session EJBs to implement a legacy connection pool, you do not want to allocate more bean instance than the number of connections that can be supported by your legacy system.

For information on tuning EJB parameters, see the 'Performance and Tuning' documentation for your Weblogic Server version."

WeblogicGCThread

Description: Total Garbage Collection Threads

Collection interval: MEDIUM

Policy: Weblogic_GarbageCollectionThread

Aspect: Weblogic JVM Heap Memory

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0003.Weblogic_M0008 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 100

Message Text: Number of Java threads (daemon and non-daemon) that are currently running in the Virtual Machine across all processors (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Total Garbage Collection Count is High. The JVM heap size may be set too low for the client load.

Potential Impact: The JVM heap size determines how often and how long the VM spends collecting garbage (de-allocating unused Java objects).

The Java heap is where the objects of a Java program live. When an object can no longer be reached from any pointer in the running program, the object is garbage.

Garbage collection affects performance because JVM work cannot proceed during full garbage collection.

An acceptable rate for garbage collection is application specific and should be adjusted after analysing the actual time and frequency of garbage collections.

The goal of tuning your heap size is to minimize the time that you spend doing garbage collection while maximizing the number of clients that you can handle at a given time.

If you set a large heap size, full garbage collection is slower, but it occurs less frequently. For a smaller heap size, full garbage collection is faster, but occurs more frequently.

Suggested Action(s) : For additional information on tuning your heap size, see the 'Performance and Tuning' documentation for your Weblogic Server version.

You can set the heap size using the options -Xms and -Xmx on the Java command line in the script used to start the server.

Use the -Xms option to set the minimum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

Use the -Xmx option to set the maximum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

As a general rule, set minimum heap size equal to the maximum heap size. If you are using 1.3 Java HotSpot JVM, also set generation sizes.

Make sure that the heap size is not larger than the available free RAM on your system.

Use as large a heap size as possible without causing your system to swap pages to disk.

The amount of free RAM on your system depends on your hardware configuration and the memory requirements of running processes on your machine.

See your system administrator for help in determining the amount of free RAM on your system.

Typically, you should use 80% of the available RAM (not taken by the operating system or other processes) for your JVM.

If you find that you have large amount of RAM remaining, run more Weblogic Servers on your machine."

WeblogicJDBCConTime

Description: JDBC connection pool connection delay, in milliseconds

Collection interval: MEDIUM

Policy: Weblogic_ConnectionDelayTime

Aspect: Weblogic JDBC Connection Pool Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0045.Weblogic_M0105 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: JDBC connection pool connection delay (<VALUE>) for instance (<OPTION (jdbcdatasourcename)>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The average time it takes to get a physical connection from the database has exceeded the threshold.

Potential Impact: Application response time may increase drastically.

Suggested Action(s) : For information on managing JDBC connections, see the 'Programming Weblogic JDBC' documentation for your Weblogic Server version.

The average time it takes to get a physical connection from the database has exceeded the threshold."

WeblogicReqWaitTimeThrd

Description: Request wait time for a thread

Collection interval: MEDIUM

Policy: Weblogic_RequestWaitTimeforThread

Aspect: Weblogic Thread Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0058.Weblogic_M0122 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 1000

Message Text: Request wait time for a thread (<VALUE> ms) too high (>= <THRESHOLD> ms)
[Policy: <NAME>]

Instruction Text:

Probable Cause(s) : This metric is used to monitor the time (in milliseconds) a request had to wait for a thread.

Only requests whose execution is needed to satisfy the min_threads_constraint are considered.

The min-threads-constraint is one of the work manager components that you can use to control the performance of your application by referencing the name of the component in the application's deployment descriptor.

This constraint guarantees a number of threads the server will allocate to affected requests to avoid deadlocks.

The default is zero.

This type of constraint has an effect primarily when the server instance is close to a deadlock condition.

In that case, the constraint will cause Weblogic Server to schedule a request from a even if requests in the service class have gotten more than its fair share recently.

Potential Impact:

1. Applications may respond slow.
2. Applications may block on threads to be available.
3. Application deadlock situation may arise.

Suggested Action(s) :

1. Increase Thread pool size or mark them growable.

This metric is used to monitor the time (in milliseconds) a request had to wait for a thread.

Only requests whose execution is needed to satisfy the min_threads_constraint are considered.

The min-threads-constraint is one of the work manager components that you can use to control the performance of your application by referencing the name of the component in the application's deployment descriptor.

This constraint guarantees a number of threads the server will allocate to affected requests to avoid deadlocks.

The default is zero.

This type of constraint has an effect primarily when the server instance is close to a deadlock condition.

In that case, the constraint will cause Weblogic Server to schedule a request from a even if requests in the service class have gotten more than its fair share recently."

TranTimErrRbPct

Description: Percentage of transactions rolled back due to a timeout error

Collection interval: MEDIUM

Policy: Weblogic_TransactionTimeErrorRollbackPercentage

Aspect: Weblogic Transactions

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0063.Weblogic_M0140 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: % of transactions rolled back due to timeout error (<VALUE>%) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s): The percent of transactions rolled back due to timeout errors has exceeded the threshold value.

Transactions are not successfully completing due to timeout errors.

Potential Impact: Fewer user requests are being successfully completed.

Suggested Action(s): The administrator can monitor individual transactions from the Administration Console.

In addition to displaying statistics, the following information can also be displayed:

- 1) Transactions by name, including rollback and time active information
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicEJBMssdCntRt

Description: Number of times a failed attempt was made to get an instance from the free pool per minute

Collection interval: HIGH

Policy: Weblogic_EJBMissedCountRate

Aspect: Weblogic EJB Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0021.Weblogic_M0184 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 10

Message Text: # of times a failed attempt was made to get an instance from the free pool (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The total number of times a failed attempt was made to get an instance from the free pool.

Potential Impact: An Attempt to get a bean from the pool will fail if there are no available instances in the pool.

This results in delayed execution of the business logic and hence slows the response time of the application.

Suggested Action(s) : Verify that the max-beans-in-free-pool element has not been set too low.

Also, while Weblogic Server will always try to allocate a new bean instance if one is not available, in reality you are limited by the number of executable threads.

In most cases, each thread will need, at most, a single bean instance.

For information on tuning EJB parameters, see the 'Performance and Tuning' documentation for your Weblogic Server version.

This metric is used to monitor the number of threads in the standby pool.

Surplus threads that are not needed to handle the present work load are designated as standby and added to the standby pool.

These threads are activated when more threads are needed. The value of this count must be in an acceptable range to meet performance criteria."

ConnDestByErrTotCnt

Description: Weblogic_ConnectionsDestroyedByErrorTotalCount

Collection interval: MEDIUM

Policy: Weblogic_ConnectionsDestroyedByErrorTotalCount

Aspect: Weblogic JCA Statistics

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0052.Weblogic_M2113 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: MINOR / 1

Message Text: Number of connections that were destroyed because an error event was received (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s): number of connections that were destroyed because an error event was received.

Potential Impact: user request throughput will be reduced

Suggested Action(s): The Weblogic administrator should check the connected systems and ensure they are functioning correctly.

In addition, the administrator can monitor transactions from the Administration Console.

This includes:

- 1) Transactions by name, including rollback and time active information.
- 2) Transactions by resource, including statistics on total, committed, and rolled back transactions.
- 3) All active transactions, including information on status, servers, resources, properties, and the transaction identifier."

WeblogicServletAverageExecutionTime

Description: Average execution time for a servlet in milliseconds

Collection interval: MEDIUM

Policy: Weblogic_ServletAverageExecutionTime

Aspect: Weblogic Servlet Performance

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0030.Weblogic_M0061 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: WARNING / 1000

Message Text: Average execution time for a servlet (<VALUE>ms) belongs to application <OPTION (applicationname)> too high (>= <THRESHOLD>ms) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The average execution time for a servlet has exceeded the threshold value.

Application design issues.

Potential Impact: Slow response time in returning an HTML or XML response to the HTTP request from a client application.

Suggested Action(s) : The cause of high execution time for the servlet could be a resource contention problem, or it could be due to the design of the servlet.

You may also choose to re-evaluate the threshold setting for this metric if values consistently exceed the threshold value.

If JSPs are used extensively in the Web-based application, there could be a performance impact due to having to compile the corresponding .jsp files into Java servlet code, and then compiling the Java code to a Java class file.

In this situation, performance can be significantly improved by setting the server's java compiler to sj or jikes instead of javac."

WeblogicJDBCConPoolUtl

Description: Percentage utilization of available JDBC connections in connection pool

Collection interval: MEDIUM NA

Policy: Weblogic_JDBCConnectionPoolUtilization

Aspect: Weblogic JDBC Connection Pool Status

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0045.Weblogic_M0092 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 98, MAJOR / 95

Message Text: Rule1: % utilization of available JDBC connections in connection pool (<VALUE>%) for instance (<OPTION(Weblogic_Instance)>) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Message Text: Rule2: % utilization of available JDBC connections in connection pool (<VALUE>%) for instance (<OPTION(jdbcdatasourcename)>) too high (>= <THRESHOLD>%) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : The JDBC connection pool utilization has exceeded the threshold value. The number of available JDBC connections is low.

Potential Impact: Performance degradation caused by having to wait for a JDBC connection to a DBMS.

Suggested Action(s) : If the database system can support additional connections, the Weblogic administrator should increase the number of connections available for this connection pool.

A good rule of thumb is that the maximum size of the connection pool should be equal to the number of Execute Threads configured in the Weblogic Server.

This assumes that each thread uses one transaction to service a request and therefore needs just one connection.

If this is not the case, then a slightly larger connection pool may be more efficient.

The connection pool minimum size should be equal to the maximum size.

This ensures that all database connections are acquired during server start-up and not when the server is under load."

WeblogicGCCount

Description: Total Garbage Collection Count

Collection interval: MEDIUM

Policy: Weblogic_GarbageCollectionCount

Aspect: Weblogic JVM Heap Memory

CIT: Weblogic AS

Alarming / Logging: Alarming / Logging

Data source/ Data class: Weblogic_C0003.Weblogic_M0005 / WEBLOGIC_DATA

Message Category: Weblogic

Severity / Threshold: CRITICAL / 30

Message Text: Number of garbage collection runs that have occurred (<VALUE>) too high (>= <THRESHOLD>) [Policy: <NAME>]

Instruction Text:

Probable Cause(s) : Total Garbage Collection Count is High. The JVM heap size may be set too low for the client load.

Potential Impact: The JVM heap size determines how often and how long the VM spends collecting garbage (de-allocating unused Java objects).

The Java heap is where the objects of a Java program live. When an object can no longer be reached from any pointer in the running program, the object is garbage.

Garbage collection affects performance because JVM work cannot proceed during full garbage collection.

An acceptable rate for garbage collection is application specific and should be adjusted after analysing the actual time and frequency of garbage collections.

The goal of tuning your heap size is to minimize the time that you spend doing garbage collection while maximizing the number of clients that you can handle at a given time.

If you set a large heap size, full garbage collection is slower, but it occurs less frequently. For a smaller heap size, full garbage collection is faster, but occurs more frequently.

Suggested Action(s) : For additional information on tuning your heap size, see the 'Performance and Tuning' documentation for your Weblogic Server version.

You can set the heap size using the options -Xms and -Xmx on the Java command line in the script used to start the server.

Use the -Xms option to set the minimum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

Use the -Xmx option to set the maximum Java heap size. Set this value to a multiple of 1024 that is greater than 1MB.

As a general rule, set minimum heap size equal to the maximum heap size. If you are using 1.3 Java HotSpot JVM, also set generation sizes.

Make sure that the heap size is not larger than the available free RAM on your system.

Use as large a heap size as possible without causing your system to swap pages to disk.

The amount of free RAM on your system depends on your hardware configuration and the memory requirements of running processes on your machine.

See your system administrator for help in determining the amount of free RAM on your system.

Typically, you should use 80% of the available RAM (not taken by the operating system or other processes) for your JVM.

If you find that you have large amount of RAM remaining, run more Weblogic Servers on your machine."

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 Oracle WebLogic 2.00)

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!