



Debugging a Performance Issue with a WebSphere Application Server for z/OS

Magda Matysiewicz mmaty@us.ibm.com

Koelli Mungie koellib@us.ibm.com

January 19, 2006



Agenda



- Introduction
- WebSphere specific settings and how it works
- Timeouts and Performance
- RMF Example
- JVM Considerations
- Miscellaneous
- References
- Q&A
- Appendix – Documentation for L2 support



WebSphere Settings

How they work and influence
performance



Application reload = hfs access = expensive



- How often, if at all, do you want the application to be reloaded when application files are updated?
 - Reload Enabled: true or false
 - Reload Interval: 0 (reloading disabled) or 1 to 2147483647 (in secs)
- Settings in two places to consider:
 - ibm-web-ext.xmi – for Web Modules (ex. Servlets, JSPs)
 - admin console (in deployment.xml) – for EJB Modules
 - Note: If 'Reload Enabled = false' specified in admin console, the settings in ibm-web-ext.xmi are still honored.
- By default, reloading option in WSAD (what's in ibm-web-ext.xmi) is enabled and interval set to 3 seconds.
 - With this setting, the Web Module files in hfs are accessed every 3 seconds. **HFS access is expensive!**

More info on reloading in the InfoCenter:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//topic/com.ibm.websphere.zseries.doc/info/zseries/ae/urun_rapp_inst.html

Application reload - settings



- In admin console: Applications > Enterprise Applications > ApplicationName > Configuration Tab

Reload Enabled	<input type="checkbox"/>	<small>i Specifies if class reloading is enabled for application files when they get updated.</small>
Reload Interval	<input type="text" value="3"/>	<small>i The timeperiod (in seconds) in which the application's filesystem will be scanned for updated files.</small>
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>		

- In ibm-web-ext.xmi - /AppServer/installedApps/NodeName/EARName/WARName/WEB-INF

```
xmlns:webappext="webappext.xmi" xmlns:webapplication="webapplication.xmi"  
xmi:id="WebAppExtension_1" reloadInterval="5" reloadingEnabled="false" default  
additionalClassPath="" fileServingEnabled="true" directoryBrowsingEnabled="false"
```

Application reload – 4 Examples



- Turn off application reloading (note, this disables hot deployment):
 - Admin console: Reload Enabled = true; Reload Interval = 0
 - ibm-web-ext.xmi: settings do not matter; They are overwritten by the admin console setting of 'Reload Enabled = true'
- Both EJB and Web Modules reload set to 30 mins
 - Admin console: Reload Enabled = true; Reload Interval = 1800
 - Ibm-web-ext.xmi: settings do not matter
- EJB modules reload disabled and Web Modules reload set to 3 seconds (if WAS and WSAD defaults taken)
 - Admin console: Reload Enabled = false; Reload Interval = 0 (or any #)
 - Ibm-web-ext.xmi: Reload Enabled = true; Reload Interval = 3
- EJB modules reload enabled and Web Modules reload disabled: Cannot do it. Do you really want to?

Alarm Manager and 'quiet' mode



- **com.ibm.ejs.am.mode.workbased=true**
- When above variable is set to 'true' and application server is *idle*, the alarm manager will not be active (eg. application reloading will not occur regardless of reload option settings)
- Available only for servant, not controller. You will see some CPU usage in the controller when server is idle.
- There is really no side effect of having this option set to 'true'
- App Reloading disabled when server is idle is default in V6.

To set com.ibm.ejs.am.mode.workbased variable, go to admin console:
Servers > Application Servers > *ServerName* > Process Definition > Servant > Java Virtual Machine > Custom Properties > add or modify the variable

Once set, it appears in servant.jvm.options (located in the hfs under the following symbolic link: /WASRoot/Cell.Node.Server) as -D option:

-Dcom.ibm.ejs.am.mode.workbased=true

InfoCenter Link for com.ibm.ejs.am.mode.workbased variable:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1//topic/com.ibm.websphere.zseries.doc/info/zseries/ae/xrun_jvm.html

Decrease CPU overhead in Application Server Controller



- Increase the Monitoring Policy ping interval to reduce the number of 'isAlive' SOAP messages in the controller (default 60 sec)
 - How often NodeAgent attempts to contact the server
 - Application Servers > *server* > Process Definition > Control > MonitoringPolicy > Ping interval
- Increase (or disable if logstream is not used) logstream compression interval (default = 30 sec)
 - How frequently the recovery log service attempts to compress any logstreams that application components are using.
 - **RLS_LOGSTREAM_COMPRESS_INTERVAL** – variable shipped in PK08016 (W510217) / PK08027 V6.01
- Increase the synchronization interval (default 10 mins)
 - How frequently configuration data is synchronized across a cell.
 - System Administration > Node Agents > *node_agent_name* > File Synchronization Service > Synchronization Interval

More about “Monitoring policy settings”:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/urun_rmonitorpol.html

Tet RLS_LOGSTREAM_COMPRESS_INTERVAL variable, go to admin console:
Environment > Manage WebSphere Variables > choose the scope > Add (or modify if already there) the variable.

Affinity work and WLM (with multiple servants running)



- How does WLM work?
 - WLM is responsible for dispatching requests to the servants in a way so the goal that was set is met.
 - One servant may get more work (within its capacity) as WLM will try to bring down any additional servants if they are not necessary for handling the given workload.
 - This can cause a problem when one servant region receives majority of the requests that require affinity (the “matching” request has to be handled by the same servant when it comes back) – **work might be waiting on WLM queue even though there are servants available.**
- Solution – use **wlm_stateful_session_placement_on=1**
 - WLM will attempt to balance affinities within the servants associated with the same service class.
 - WLM will dispatch a new request to a servant with the least existing affinities. If the request creates affinity, the WLM affinity count for that servant will be increased. Otherwise, nothing changes.
- You need WLM apar **OA04699** – see apar for more details

To set wlm_stateful_session_placement_on variable, go to admin console: Environment > Manage WebSphere Variables > choose the scope > Add (or modify if already there) the variable.

From InfoCenter:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/trun_wlm_sessionplacement.html

Why and when to perform this task

Use this task if your application server is experiencing problems with the default workload distribution strategy. The default workload distribution strategy uses a hot servant region for running requests that create HTTP session objects. Consider configuring WebSphere Application Server and the z/OS Workload Manager to distribute your HTTP session objects in a round-robin manner in the following conditions:

- HTTP session objects in memory are used, causing dispatching affinities.
- The HTTP sessions in memory last for many hours or days.
- A large number of clients with HTTP session objects must be kept in memory.
- The loss of a session object is disruptive to the client or server.
- There is a large amount of time between requests that create HTTP sessions.

Number of worker threads per servant



- Workload Profile: How many threads per servant/JVM your environment needs?
 - ISOLATE – 1 thread
 - IOBOUND – $\text{MIN}(30, \text{MAX}(5, (\text{Number of CPUs} * 3)))$
 - CPUBOUND – $\text{MAX}((\text{Number of CPUs} - 1), 3)$
 - LONGWAIT – 40
- What is the majority of work being performed. CPU intensive? DB2 work?
- Keep in mind, the larger number of threads, the longer it takes for JVM to quiescent the threads for Garbage Collection and restart. It also means, that all threads are not available to continue with work during the GC cycle.
- Consider increasing number of servants as well.

Workload Profile can be set in Admin console: Servers > Application Servers > *server name* > ORB service > Advanced Settings > choose option from Workload Profile.

Detailed explanation of the settings:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/uorb_rorb_service.html



WebSphere Settings cont.

How they work and influence
performance



Suggested settings for JDBC Connection Pool (1 of 2)



- If the ConnectionWaitTimeout associated with the data source > server HTTP/MDB/WLM timeout that could lead to potential EC3 timeout abends.
- Set ConnectionWaitTimeout < HTTP/MDB/WLM timeout to prevent abends and in case there are no more connections a ConnectionWaitTimeoutException is thrown.
- maxConnection = number of worker threads in servant to prevent delays for a connection to free up when all threads require a connection to DB2.

EC3/0413000x timeouts can occur in servers that have database intensive applications deployed and if the ConnectionWaitTimeout > HTTP, MDB or WLM timeout value. Instead of just giving a ConnectionWaitTimeout exception the servant will abend with an EC3 and recycle. This causes an unnecessary delay in getting a connection and the application request to complete. For this reason the ConnectionWaitTimeout on the datasource should be set less than any of the server timeout values.


A good rule of thumb is to have the maxConnection setting equal to at least the number of worker threads in the servant. This way if there are x requests processing simultaneously with x number of connections open to DB2, no thread has to wait for another connection to free up in the pool. This is assuming maxConnection=x=Number of threads in the servant. The maxConnection is a per servant setting.

Suggested settings for JDBC Connection Pool (2 of 2)



[JDBC Providers](#) > [DB2 Universal JDBC Driver Provider \(XA\)](#) > [Data Sources](#) > [shareTest](#) >

Connection Pools

Connection pool properties that can be modified to change the behavior of the J2C connection pool manager. Reviewing and possible modification of these configuration values is recommended. 

Configuration

General Properties

Scope	cells:PLEX1Network:nodes:PLEX1Manager	
Connection Timeout	<input type="text" value="1800"/>	seconds
Max Connections	<input type="text" value="10"/>	connections

MDB Throttle Considerations (1 of 3)



- Message Driven Beans (or MDBs) use a throttle value to control the number of incoming MDB requests to the servant.
- This value is used to prevent the servant from getting inundated with an unmanageable number of MDB work requests.
- There are certain values to consider before determining a suitable throttle value for the server.
 - Number of worker threads in given servant
 - maxConnection setting under ConnectionFactory -> SessionPool used by the MDB Listener Port.
 - MDB ListenerPort maxSessions

The discussion assumes one ConnectionFactory is dedicated to providing a single Connection for a single MDB and its ListenerPort.

The WebSphere MDB ListenerPort infrastructure maintains a pool of ServerSession objects, for each MDB, in each servant, each of which requires its own worker thread. The ServerSessions are used to consume messages and to dispatch the application's onMessage() upon. The ServerSession pool size in a given servant is set by the WebSphere runtime to be equal to the minimum of the three user-defined settings:

- Number of worker threads in given servant
- maxConnection setting under ConnectionFactory -> SessionPool used by the MDB Listener Port.
- MDB ListenerPort maxSessions

AdminConsole Path to Settings mentioned:

1. Listener Port maxSessions (throttle)

Servers-> Application Servers-> *application_server*-> Message Listener Service-> Listener Ports-> *listener_port* -> maxSessions

2. Session Pool maxConnections

Navigate to ConnectionFactory -> Session Pool -> Max Connections

3. Connection Pool maxConnections

Navigate to ConnectionFactory -> Connection Pool -> Max Connections

MDB Throttle Considerations (2 of 3)



- Choose the number of servants configured for application server.
- Choose the number of worker threads per servant.
- To use every worker thread in the servant for MDB work, set maxConnections for the SessionPool to the number of threads per servant
- Message Listener Port maxSessions setting could be set to at least (number of threads per servant * number of servants). This would be the high threshold value for server. This is the **throttle** value.
- Backlog factor may be desired so when current work requests complete there are new requests ready to be dispatched.

maxConnection setting can be used to “limit” the number of worker threads available for ServerSessions. This can be set to something lower than the number of worker threads. So if you want 20 out of 40 of your available worker threads to do MDB work you would set maxConnection=20 in this case.

MDB Throttle Considerations (3 of 3)



- No backlog factor when the number of messages coming in are higher than the throttle value would result in throttle blocking message browsing and no new work will be queued up.
- Example if backlog factor = 2 and user is expecting 20 worker threads for a given MDB with 5 servants, Listener Port's maxSession = $20 * 5 * 2 = 200$.
- Desirable throughput to take advantage of available worker thread resources available. Please note that 2 is not an IBM recommendation, but just an example.

See documentation APAR PK08712 which address MDB Throttle article changes in InfoCenter.



Timeouts and Performance



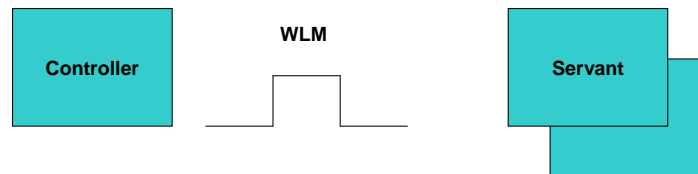
Timeouts and performance



- Timeout – Abend EC3 with RC = 0413000x
 - Analyze the thread that timed out and the reason why.
 - See technote # 1215659 for some hints on analyzing timeouts. Search for technote at the following site:
http://www.ibm.com/software/webservers/appserv/zos_os390/support/
- After analysis of the timeout if all worker threads show delay in processing (no one thread at fault) consider WLM setting problem
 - Use “ip verbx cbdata ‘asid(xxxx) tcb(xxxxxxxx)’”
 - Is WLM goal set high enough?
 - Do we meet the goal set in WLM? RMF records can help to verify that.
 - Consider WLM Critical attribute for CPU when defining a service class for WebSphere tasks.

More info on WLM Critical attribute: “z/OS MVS Planning: Workload Management”
Ch. 12 -> “Long-Term CPU Protection” (SA22-7602-09)

Server initialization delays causing EC3 timeouts (1 of 3)



- **Case I** Requests come into the server before the servant is fully initialized during initial server startup.
- **Case II** Requests come into the server during servant recycle which is normal recovery after an EC3/0413000x abend.
- Both of the above result in the delay of the work request getting processed.

Background:

- In the WebSphere for z/OS application server, work requests are initially queued to the controller.
- From the controller, they are then sent to the WLM queue.
- The requests are then picked up by the servant where they are actually processed.

There are cases where during an EC3 timeout abend or the initial start of the server client requests are accepted by the controller but cannot be processed before the servant is fully initialized. This causes the request to ultimately timeout when the servant does come up causing another EC3 abend and recycle. This causes an unnecessary delay in the request to eventually get processed.

Server initialization delays causing EC3 timeouts (2 of 3)



Case I – Initial server startup

- **protocol_accept_http_work_after_min_srs = 1**
- If above variable set to 1 (TRUE), the application server does not accept any http work requests until the minimum number of servants are open for business.
- wlm_minimumSRCount determines minimum number of servants.
- Prevents EC3/0413000x timeouts during server initialization since requests do not time out waiting for servant to become available.

The environment variable is protocol_accept_http_work_after_min_srs

This parameter has only OFF and ON values, 0 = OFF 1= ON

To set protocol_accept_http_work_after_min_srs variable, go to admin console:
Environment > Manage WebSphere Variables > choose the scope > Add (or modify if already there) the variable.

Server initialization delays causing EC3 timeouts (3 of 3)



Case II – Recycle of servant due to EC3/0413000x or other cause.

- Server has one servant that becomes unavailable due to the EC3/0413000x or other cause.
- Application requests continue to come into the controller and with no servant available to process them they eventually time out when the server initializes.
- You can set this variable:
control_region_confirm_recovery_on_no_srs=1
to tell the controller to stop the http listeners until minimum number of servants are back up and running.
- Variable shipped in PK09553(V5.1) and PK09554(V6.0)
 - This function is not available in V5.0.

To set control_region_confirm_recovery_on_no_srs variable, go to admin console:
Environment > Manage WebSphere Variables > choose the scope > Add (or modify if already there) the variable.



RMF Example



RMF Workload activity Report (1 of 3)



POLICY ACTIVATION DATE/TIME 04/13/2005 15.24.30

----- SERVICE CLASS PERIODS

REPORT BY: POLICY=CBPTILE **WORKLOAD=ONLINE** SERVICE CLASS=BOSSCLAS RESOURCE GROUP="NONE"
PERIOD=1 IMPORTANCE=2 CRITICAL =NONE

TRANSACTIONS	TRANS. TIME	HHH.MM.SS.TTT	--DASD I/O--	---SERVICE---	--SERVICE TIMES--
AVG 6.51	ACTUAL	6	SSCHRT 0.0	IOC 0	TCB 455
MPL 6.51	EXECUTION	4	RESP 0.8	CPU 9422K	SRB 0.0
ENDED 400627	QUEUED	2	CONN 0.5	MSO 0	RCT 0.0
END/S 1335.43	R/S AFFINITY	0	DISC 0.0	SRB 0	IIT 0.0
#SWAPS 0	INELIGIBLE	0	Q+PEND 0.3	TOT 9422K	HST 0.0
EXCTD 0	CONVERSION	0	IOSQ 0.0	/SEC 31474	IFA N/A
AVG ENC 6.51	STD DEV	6			APPL% CP 151.7
REM ENC 0.00					
MS ENC 0.00					

GOAL: RESPONSE TIME 000.00.00.250 FOR 80%

PERF	-----RESPONSE TIME-----		-EX VEL%-	TOTAL	-EXE--
INDX	-----GOAL-----	---ACTUAL---	TOTAL	GOAL ACT	USING% DELAY%
0.5	00.00.00.250	80% 100%	100%	15.5 14.4	78.6

RMF Workload activity Report (2 of 3)



- **Service Rates**
 - APPL% = the percent of a single processor needed to drive the workload in the interval. In the previous example the number is 151.7 and this means that $151.7/100 = 1.517$ processors were needed to drive this workload.
- **Transactions**
 - Every transaction in WebSphere for z/OS causes an enclave to be created.
 - $AVG = MPL =$ Number of transactions in the interval
 - $AVG\ ENC =$ Number of enclaves in the interval
 - Therefore, $AVG = MPL = AVG\ ENC$

RMF Workload activity Report (3 of 3)



- Transaction Time
 - Actual = Execution + Queued
- Response Time
 - Goal = the percentage of work that was to be completed and this is configured in the WLM service class.
 - Performance Index < 1 indicates that we are doing better than goal
- Execution Delays – not displayed if no delay
 - Check if Actual < Goal
 - Queuing (QMPL)
 - Paging (AUX XMEM, SERV PRIV, AUX COMM)
 - Operator (UNKN)
 - Crypto (CRYPTO)
- See Ch. 5 “Resource Measurement Facility Report Analysis” (SC33-7991) for more information.

QMPL - Time spent in WLM queue waiting to be dispatched to servant thread.



JVM Considerations



Improve server startup time?



- **-Xquickstart:** JVM option that was recommended in different places to speed up server startup
 - 15-20% improvement
 - This speed up is achieved by compiling methods at a lower degree of optimization.
- **Tradeoff of quickstart :** pprofManager – polling profiler thread is created to determine if methods need to be compiled => *Causes higher CPU usage when server is idle.*
 - Set NPPOLLING.
- **JITC_COMPILEOPT=NPPOLLING:** can be used to deactivate pooling thread.
 - It disables some method recompilation that can occur.
 - Still some higher CPU usage during the life of the server due to the fact JIT compiles for some methods are disabled.

To set -Xquickstart JVM option, go to admin console:

Servers > Application Servers > server_name > Process Definition > Control OR Servant > Java Virtual Machine > add the variable to “Generic JVM Arguments”

To set JITC_COMPILEOPT=NPPOLLING, go to admin console:

Servers > Application Servers > server_name > Process Definition > Control OR Servant > Environment Entries > Add new variable.

JVM Heap - Fragmented Heap



- What does it mean to have fragmented heap?
 - It occurs when available memory is broken into many non-contiguous blocks.
 - Compaction will occur when there is not enough contiguous heap space to satisfy a request for object
 - Note: The objects that are 'pinned' cannot be moved during compaction.
 - Garbage Collection with compaction will run more often when there is a high heap fragmentation. **This causes performance degradation due to extra paging and CPU utilization.**
- IBM Java Diagnostics Guide, Chapter 2 – “Avoiding Fragmentation”

Additional Info:

-- IBM Java Diagnostics Guide - SC34-6358-00

Garbage Collection Example – Heap fragmentation leading to OutOfMemory (1 of 4)



③
<AF[234]: Allocation Failure. need 17855992 bytes, 2604 ms since last AF>
<AF[234]: managing allocation failure, action=2 (214935912/671021568)>
<GC(234): GC cycle started Fri May 27 17:41:50 2005
<GC(234): freed 92512688 bytes, 45% free (307448600/671021568), in 6071 ms>
<GC(234): mark: 275 ms, sweep: 22 ms, compact: 5774 ms> ②
<GC(234): refs: soft 93 (age >= 32), weak 2, final 1780, phantom 0>
<GC(234): moved 3827975 objects, 246366576 bytes, reason=1, used 40 more bytes>
<AF[234]: managing allocation failure, action=3 (307448600/671021568)>
<AF[234]: managing allocation failure, action=4 (307448600/671021568)>
<AF[234]: managing allocation failure, action=6 (307448600/671021568)>
JVMDG217: Dump Handler is Processing **OutOfMemory** - Please Wait.

①
④

This is an example of a verbose GC trace entry. You can enable verbose GC trace through WAS Admin Console: Servers > Application Servers > ServerName > Process Definition > Servant > Java Virtual Machine > Check checkbox for 'Verbose garbage collection' > Save and recycle the server.

Many customers run with verbose GC tracing enabled in production environment.

Additional Info:

-- White Paper WP100292 "Understanding the IBM Java Garbage Collector":

<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100292>

Garbage Collection Example - JVM Heap compaction (2 of 4)



1. <AF[234]: Allocation Failure. need 17855992 bytes, 2604 ms since last AF>
 1. Size of the requested object - 17855992 = 17M
 2. Question: Do you want your application to make requests for large object like this? The larger the object, the larger consecutive JVM heap space is needed => **compaction needs to run at higher rate**
2. <GC(234): mark: 275 ms, sweep: 22 ms, compact: 5774 ms>
 1. Compaction took 5774 ms. By looking at different trace AF trace entries, you can see how often compaction runs and how long it takes

Garbage Collection Example – Heap size and fragmentation (3 of 4)



3. <GC(234): freed 92512688 bytes, 45% free (307448600/671021568), in 6071 ms >
 1. Heap is 45 % free and 55% full
 2. Free space = 307448600 = 293M
 3. Total Heap space = 671021568 = 640M
 1. If the heap is larger than needed, unnecessary time is spent on maintaining the heap;
 2. If the heap is too small, OutOfMemory errors can occur
 3. ***You need to tune the JVM heap size based on what your applications require.***
 4. Object of 17K < Free space of 393KB, but still receiving OutOfMemory exception => ***Heap is fragmented***

Garbage Collection Example – Compaction (4 of 4)



4. <AF[234]: Allocation Failure. need 17855992 bytes, 2604 ms since last AF>
<GC(234): freed 92512688 bytes, 45% free (307448600/671021568), in 6071 ms>
1. Time elapsed since last Allocation Failure = 2604 ms
 2. This GC cycle took 6071 ms to complete.
 3. Use the formula below to calculate the GC overhead.
 4. Take an average of multiple GC trace entries – *The average overhead should be < 5%:*
 5. *Omit first GC entries when JVM heap is stabilizing.*

$$\text{GC Overhead (\%)} = \frac{100 \times \text{"ms used to complete GC"}}{\text{"ms since last AF"} + \text{"ms used to complete GC"}}$$

Based, on this example, the overhead is 70%:

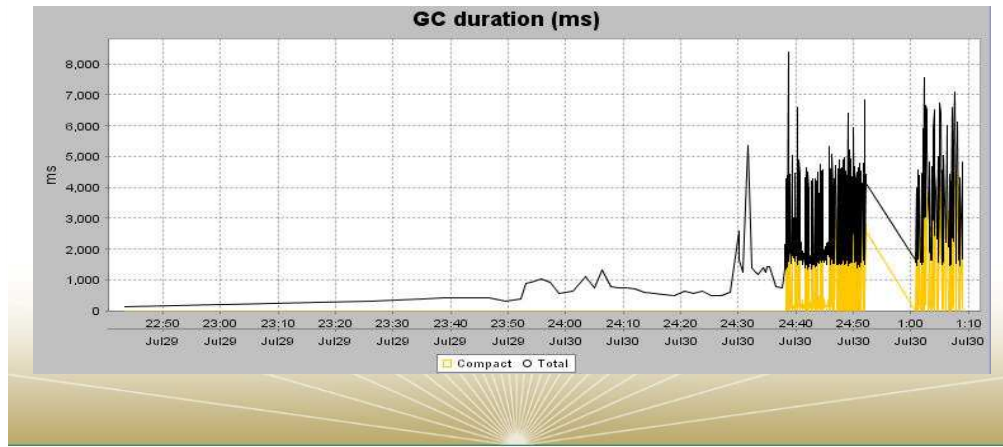
$$70\% = \frac{100 \times 6071}{2604 + 6071}$$

However, note, this is the last GC entry where OutOfMemory was detected so is not a good tracepoint to take under consideration. Also, you want to average multiple trace entries to get a more realistic value.

JVM Heap Analyzer – GCCollector



- Available free: <http://www.alphaworks.ibm.com/tech/gcdiag>
- Does calculations for you





Miscellaneous



Performance Monitoring Tools



- IBM Tivoli Performance Viewer (shipped with WebSphere V5.x)
- IBM WSAM - www.ibm.com/software/tivoli/
- IBM Tivoli OMEGAMON www.ibm.com/software/tivoli/
- BMC - MainView www.bmc.com
- Wily Technology - Introscope www.wilytech.com



Other Known Issues – APARs



- Performance impact for LTPA token users
 - V5.0 - Not applicable
 - V5.1 - PK10963 - W510223
 - V601- PK13985 - open
- Cache failures
 - **HIPER**
 - V5.0 - PK06997 - W502033
 - V5.1 - PK06996 - W510218
 - V601- PK03351 – 6.0.1
 - The authentication cache fails to find users that have already been authenticated with the z/OS Local-OS Registry
 - Local Fix: Specify IP name in UPPER case under Administration > z/OS Location Service in admin console
- MutualAuthCbindCheck (CBIND) causes storage leak and performance degradation
 - V5.1 - PK03536 - W510212
 - V601- PK03538 - 6.0.1.2

See APAR description for more details

Other Known Issues – APARs

- CPU spike in the Controller Region after a timeout
 - V5.0 - Not applicable
 - V5.1 - PK01151 - W510208
 - V601- Fixed in 6.0.2
- High CPU in the Deployment Manager
 - V5.0 - PQ92052 - W502016
 - V5.1 - PQ96630 - W510201
 - V601- Not applicable
- HA Manager threads use High CPU in CRs for ND
 - Not applicable to V5.0 and V5.1
 - Major improvement in 6.0.2
 - Additional improvement - PK13520 – open
- Storage Leak caused by JIT:
 - PK04164 => shipped in JDK 1.4.2 SR2 – PTF UK04987
 - PK07321 => shipped in JDK 1.4.2 SR3 – PTF UK07860
 - PK13368 => open

See APAR description for more details



References



References & Useful Sites



WhitePapers: <http://www.ibm.com/support/techdocs>

- [WP100292](#) "Understanding the IBM Java Garbage Collector":
- [WP100678](#) "Diagnosing Performance Problems with WebSphere Application Server on z/OS – extended checklist"
- InfoCenter:
 - V5.0: <http://publib.boulder.ibm.com/infocenter/wasinfo/v5r0>
 - V5.1: <http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1>
 - V6.0: <http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0>
- WAS for z/OS Support page
 - http://www.ibm.com/software/webservers/appserv/zos_os390/support/
 - Consider signing up for email notification for 'hot' issues => MySupport
- IBM Java Diagnostics Guide - SC34-6358-00

Questions?



Appendix

What will L2 Support need from you



Documentation requested by L2



- Problem Type
 - WebSphere Server using high CPU
 - Minimal
 - Maximize systtrace: TRACE ST,999K
 - Console Dump of Address space experiencing high CPU
 - Optimal
 - Maximize systtrace: TRACE ST,999K
 - Take 2 console dumps of all WebSphere address spaces experiencing the high CPU for comparison (taken some time apart)
 - RMF Workload Activity Report & SMF type 72/79 records

TRACE ST,999K – command issued from MVS console; issue few minutes before taking a dump.

Documentation requested by L2



- Problem Type
 - Server Timeout (EC3/0413000x)
 - SVC Dump generated with the EC3/0413000x timeout abend
 - Server hang with no response
 - Console dump of the controller and servant address space for the server with no response.
 - Slow response from server
 - RMF Workload Activity Report & SMF type 72/79 records

TRACE ST,999K – command issued from MVS console; issue few minutes before taking a dump.



Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM IBM (logo) e! (logo)/business AIX	CICS Cloudscape DB2 DB2 Universal Database	IMS Informix iSeries Lotus	MQSeries OS/390 OS/400 pSeries	Tivoli WebSphere xSeries zSeries
--	---	-------------------------------------	---	---

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.