Use WebSphere z/OS messages & settings to debug timeout. Hands On Lab

August 2, 2010

12:15PM – 1:30PM
ROOM 312 (Hynes Convention Center)
SHARE Summer Session Boston

Michael Stephen
WebSphere App Server z/OS
L2 Support Teamlead
IBM Poughkeepsie, NY
msteff@us.ibm.com

http://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Application_Server_for_z~OS

A little background / reference information

Application Server Timeouts have been the cause of many issues over the years. Diagnosing the timeouts have come a long way since WebSphere Application Server 4.1.

WebSphere Application Server z/OS has the Control Region (CR) / Servant Region(s) (SR) configuration,where application code runs in the SR(s). Depending on the work (application) code running in the SR(s), there can be Message Beans, Servlets, and other types of application workloads.

There are different timer settings for the various types of work running in the server (HTTP(S), SIP(S), IIOP, MDB, and so on).

WebSphere still has to ABEND the servant, when an application timeout occurs, as there is no fail-safe way to tear down a particular Java thread in an address space, and ensure it's data integrity.

Improvements have been made throughout the years / releases, in how you can diagnose the cause of the timeout.

__ To see the highlights of what was done for V6.1, open a browser (Firefox or Internet Explorer) and direct the browser to url:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101233

“ **WebSphere Application Server for z/OS V6.1 Configuration Options for Handling Application Dispatch Timeouts** ”

__ Or open the following file on the lab machine:

c:\wastimeout\docs\WP101233.pdf

A little background / reference information (cont)

In V7, there have been improvements made in the Dispatch Timeout areas, which include:

- Ability to "nudge" a timed-out request to completion without abending the servant region
- Stalled thread threshold to avoid abending a servant until a specific number of problem requests is reached
- CPU timeout option to prevent a runaway request from consuming excessive resources
- DISPLAY,THREADS command for information about server activity and for identifying long running requests
- Dispatch Progress Monitor (DPM) to gather documentation about a request that runs longer than expected

__ To see more on the above, open a browser (Firefox or Internet Explorer) and direct the browser to url:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101374

“ Dispatch Timeout Improvements in WebSphere Application Server for z/OS Version 7 ”

__ Or open the following file on the lab machine:

c:\wastimeout\docs\WP101374.pdf

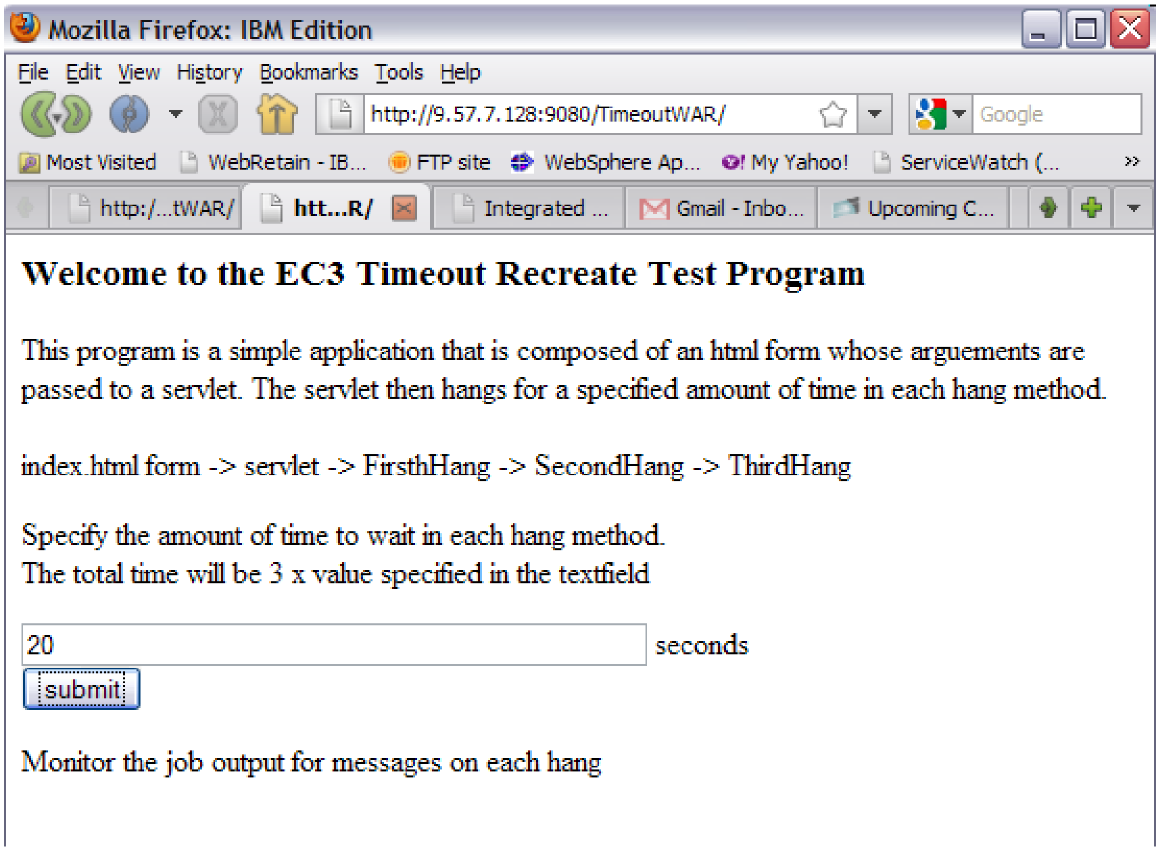
Link to the WebSphere Application Server Info Center:
http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/welcome_zseries.html

Lab Overview:

LAB will show examples using the DPM (Dispatch Progress Monitor) to diagnose where a long running application is hanging.

Different types of Diagnostic Data can be gathered using the DPM:
traceback / javacore / svcdump / heapdump / javatdump / none

Here is a simple HTTP Application, which has 3 nested timers. The timer is set on invocation, and the application will loop 3 times.



Welcome to the EC3 Timeout Recreate Test Program

This program is a simple application that is composed of an html form whose arguements are passed to a servlet. The servlet then hangs for a specified amount of time in each hang method.

index.html form -> servlet -> FirsthHang -> SecondHang -> ThirdHang

Specify the amount of time to wait in each hang method.
The total time will be 3 x value specified in the textfield

20 seconds
submit

Monitor the job output for messages on each hang

LAB: Scenario 1, Two invocations of the application with two different timers

Scenario 1 setup:

- Set DPM
- Started the TimeoutWAR application
  - from 2 different browsers timers set to 8 and 15 sec

Scenario 1 ouput:

files in c:\wastimeout\scenario1\

- sr.output1.txt
- cr.output1.txt

Find the following information:

What DPM Monitor was set ?? (MDB, HTTP, HTTPS,etc)

____

What is the 'dump_action' for the DPM monitor ?

____

(hint: it's in the cr.output1.txt; line 831; timestamp 17.18.33)

Using the c:\wastimeout\scenario1\sr.output1.txt file

Find the third instance of string 22:21:17.588
or
Go to line 6966 in the file:

(Line should look like: Trace: 2010/07/25 22:21:17.588)

This is the start of the output of the DPM.

What 2 threads (TCB addresses) are the 2 instances of the TimeoutWAR application running on ??

____

____

(hint: it's in the BBOJ0117I message)

Pick one of the threads, and follow it through the life of the application invocation.
(hint: you can use the t=######## information to follow a particular thread t=tcb address)

Traceback for thread WebSphere:ORB.thread.pool t=006c77a0

You will know if you picked the 15 second invocation, as it will be active longer in the SR output.

Note that the applications keep 'progressing', so although one is taking longer than the other, they are both progressing.

Think of how this could be used in your shop, to see the flow of applications that are running on your WebSphere Application Servers.

For Example:
- If the application goes outbound to a database, and you suspect that there are issues when the app goes to the database, you could use this tracing to see if there are delays.

If you want to get a snapshot of the entire JVM while the application is running, you can use the DPM and set the 'action' to be a javacore dump.

That will be the next scenario, Scenario 2

****
****

LAB: Scenario 2, DUMP_ACTION=JAVACORE

Scenario 2 setup:

- set servant_region_custom_thread_count to 2
  - Workload Profile set to CUSTOM
- set DPM HTTP=30, DUMP_ACTION=JAVACORE
- invoked app 3 times
  - timers 16, 14, 12

Scenario 2 ouput:

files in c:\wastimeout\scenario2\

- cr.output2.txt
- sr.output2.txt
- javacore.20100726.005056.65657.0001.txt
- javacore.20100726.005058.65657.0002.txt
- javacore.20100726.005138.65657.0003.txt

What are the 2 SR TCB addresses which are running the application code ?

____

____

(hint: BBOJ0018I message in the sr output)

Where were the javacore files written to ?

____

(hint: JVMDUMP034I message in the sr output)

Open one of the javacore files, and find the threads (TCB's) that are running the application code. Note that you can see the entire stack trace at the time of the javacore.

Compare the stacktraces of the application threads between javacore '0001' and '0003' to see the stack changes.

Thought provoking question:
Do you see where having the entire javacore may be helpful when diagnosing a timeout problem?

Example:

- If there is a java sync lock issue, you can use the javacore to see what thread is holding the lock, and what threads are waiting on the lock.

LAB: Scenario 3: MVS MODIFY DISPLAY,THREADS Command

Scenario 3 setup:

- set servant_region_custom_thread_count to 5
  - Workload Profile set to CUSTOM
- set protocol_http_timeout_output_recovery=session
- did NOT set DPM
- invoked app 7 times
  - timers 150, 45, 40, 35, 33, 30, 20
- Issued MVS Modify commands
  - F,BBOS001,DISPLAY,THREADS,age=30
  - F,BBOS001,DISPLAY,THREADS,age=30,details
  - F,BBOS001,STACKTRACE

Scenario 3 ouput:

files in c:\wastimeout\scenario3\
- cr.output3.txt
- sr.output3.txt
- syslog.output3.txt (good to use to get when (and what) modfiy commands were issued

From the syslog file, find the modify commands that were issued during the test run.

Time issued	Command issued
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________
____________	____________________________________

NOTE that the time stamps are 5 hours off, between the syslog and the cr/sr outputs.

- Syslog is 'local time', cr/sr is GMT time.
- WebSphere Application Server Custom Property for z/OS 'ras_time_local' set to 'true' if you want the time to be local for the cr/sr.

What was the request ID of the 150 second invocation of the application ? __________________

(hint: use the series of outputs from the display,threads and display threads,age=30 commands in the cr and/or syslog to see which ran the longest)

What was the TCB address of the 150 second invocation of the application ? ___________

(hint: BBOJ0106I from display,threads,age=30 command)

What happened to the 150 second invocation ?

____

(hint: at the bottom of the cr and syslog output files)

How can you tell ?

____

(hint: BBOO0327I message, your 'friend' when looking at timeout issues)

Why didn't the Servant Region ABEND when the 150 second invocation timed out ?

____

(hint: in the cr and sr output, look for the server custom property: protocol_http_timeout_output_recovery)

Use the output from the modify stacktrace command (from the sr.output3.txt file) to see the application progressing.

What are the top 6 entries of the stacktrace, for the 150 second invocation thread, for the 4 issuances of the modify stacktrace command ?

- Timestamp below is from syslog.
- The timestamp in sr output is +5 hours +a few mili-sec
- Remember to make sure you are on the correct tcb address

18:09:09.33 ______________________________

____

____

____

____

____

18:09:57.59 ______________________________

____

____

____

____

____

18:11:24.28 ______________________________

____

____

____

____

____

18:11:50.14 ______________________________

____

____

____

____

____

WebSphere Application Server Sessions at SHARE in Boston 2010

Room	Day	Time	Title	Speaker
312	Monday	12:15	Lab	Stephen
203	Monday	4:30	WebSphere: What's New ?	Follis
203	Wednesday	9:30	WebSphere 101	Houde/Stephen
201	Wednesday	1:30	Introduction to IBM Support Assistant (ISA)	Hutchinson
200	Wednesday	3:00	WebSphere Process Manager and Business Process Manager Configuration	Hutchinson
310	Wednesday	4:30	OSGi/JPA/Batch Feature Packs	Follis/Bagwell
203	Wednesday	6:00	WebSphere for z/OS: I'm no longer a dummy but...	Bagwell
310	Thursday	8:00	WOLA Application Designs	Bagwell
310	Thursday	9:30	Security Architecture: How does WebSphere Play ?	O'Donnell
310	Thursday	11:00	WAS on z/OS High Availability Considerations	Bagwell
200	Thursday	12:15	Staged Application Development in a WebSphere ND Cluster	Loos
310	Thursday	1:30	WAS on z/OS and WLM Interactions	Follis