



IBM Software Group

OutOfMemory (OOM) Issues in WebSphere Application Server Open Mic

21 April 2010



WebSphere® Support Technical Exchange



Agenda

- Introduce the panel of experts
- Introduce OutOfMemory (OOM) Issues in WebSphere Application Server
- Answer questions submitted by email
- Open telephone lines for questions
- Summarize highlights

Panel of Experts

Panelist	Role at IBM®
Giri Paramkusham	WebSphere Application Server Support Engineer
James Fox	WebSphere Application Server Support Engineer
Jinwoo Hwang	WebSphere Application Server Support Engineer
Thomas Ireton	WebSphere Application Server Support Engineer

Introduction

- Discuss the symptoms of Java™ heap OOM and native heap OOM and how to verify if it's a heap sizing issue or a true memory leak.
- Depending on the type of OOM condition, what documentation to collect.
- How to debug Java heap exhaustion issues and tips on finding the leak suspects using Heap Analyzer.

Introduction [continued]

- How to debug native memory issues and the various steps involved in isolating the issue.
- Discuss some of the potential memory suspects in WebSphere env you should be aware of, when dealing with Java heap and native memory issues
- When to engage WebSphere support and setting the right expectations.

Question 1

I'm seeing OOM errors in the WebSphere Application Server logs, where do I start? How do I determine if the OOM errors are due to Java heap exhaustion or native memory exhaustion?

Answer to Question 1

Verbose garbage collection trace (verbosegc):
We need to look at the cycle just prior to the OOM errors in the native_stderr.log:

```
<af type="tenured" id="780" timestamp="Dec 12 11:01:30 2008" intervalms="338.539">  
  <minimum requested_bytes="57732968" />  
  <time exclusiveaccessms="0.062" />  
  <tenured freebytes="6979664" totalbytes="536870912" percent="1" >  
  ...  
  <timesms mark="212.284" sweep="3.550" compact="785.650" total="1004.557" />  
  <tenured freebytes="44760920" totalbytes="536870912" percent="8" >  
    <soa freebytes="44760920" totalbytes="536870912" percent="8" />  
    <loa freebytes="0" totalbytes="0" percent="0" />  
  </tenured>  
</gc>  
<tenured freebytes="44760920" totalbytes="536870912" percent="8" >  
  <soa freebytes="44760920" totalbytes="536870912" percent="8" />  
  <loa freebytes="0" totalbytes="0" percent="0" />  
</tenured>  
<time totalms="1456
```

Answer to Question 1

Sometimes you may get a message from a native OOM:

- native_stderr.log:

JVMCI015:OutOfMemoryError, cannot create anymore threads due to
memory or resource constraints

JVMDBG001: malloc failed to allocate n bytes

JVMDBG004: calloc failed to allocate an array of ...

JVMCL052: Cannot allocate memory in initializeHeap for heap segment

- javacore:

1TIS!GINFO Dump Event "systhrow" (00040000) Detail

"java/lang/OutOfMemoryError":

"Failed to fork OS thread" received

Or "Failed to create a thread: retVal

Question 2

Now that I have identified the type of OOM condition, what documentation should I collect?

Answer to Question 2

- verboseGC trace information provides crucial information for debugging OOM issues.

"Enabling verbose garbage collection (verbosegc) in WebSphere Application Server"

<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21114927>

For Java heap issues -

- In addition to verboseGC trace, heapdumps and javacores are critical. These are automatically created on an OOM condition.
- For advance debugging we may ask for core dump. This will be useful if the heapdump shows lot of objects as "root" objects or primitive objects such as "char" & "String".
- "MustGather: Out of Memory errors on AIX, Linux, or Windows - - Heap Leak "
<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21138587>
- "MustGather: Out of Memory errors on Solaris & HP - Heap Leak"
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21145349>

Answer to Question 2

- For **native memory issues**, we have separate scripts for each operating system. These scripts are useful to monitor the total process size ('ps' output) of the process at regular intervals. In addition, we request a system core file.
- "MustGather: Native Memory Issues on AIX"
<http://www-01.ibm.com/support/docview.wss?uid=swg21405353>
- "MustGather: Native Memory Issues on Linux -"
<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21138462>
- "MustGather: Native Out Of Memory on Windows"
<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21313578>
- "MustGather: Out of Memory errors on Solaris - Native Leak"
<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21104470>

Answer to Question 2 contd ...

- To manually generate a heapdump to track the memory growth by comparing 2 heapdumps
 - "How to manually generate a Heapdump in WebSphere on IBM platform"
<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21297060>
 - "Getting Heapdumps on the Solaris platform"
<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21242314>
- To identify the source code responsible for large objects –

"How to identify the Java stack of a thread making a large allocation request
<http://www.ibm.com/support/docview.wss?rs=180&uid=swg21236523>



Question 3

- How do I debug Java heap exhaustion issues and what tools should I use?

Answer to Question 3

Tools to analyze heapdumps –

1) Heap Analyzer

<http://www.alphaworks.ibm.com/tech/heapanalyzer>

2) MAT (Memory Analyzer) as part of ISA workbench

<http://www.ibm.com/developerworks/java/jdk/tools/memoryanalyzer/>



Answer to Question 3

- HeapAnalyzer analyzes Java heap dumps by parsing the Java heap dump, creating directional graphs, transforming them into directional trees, and executing the heuristic search engine.

- HeapAnalyzer provides –
 - List of Java heap leak suspects
 - Tree view of Java heap dump
 - List of objects/classes/arrays by frequency
 - List of objects/classes/arrays by size and number of children

Question 4

How do I debug native memory exhaustion issues and what tools should I use?

Answer to Question 4

- Debugging native memory issues can be a lengthy process
- There is no way of finding which code allocated the native memory, so we have to use trial and error going through each of the potential native memory users, until we find the one causing the issue.
- Please see the following page to start:
“Troubleshooting native memory issues”
<http://www-01.ibm.com/support/docview.wss?uid=swg21373312>

Answer to Question 4 (4 steps)

1. Check the maximum heap size:
For 32 bit, try to keep the max java heap size at or under 1536m.
2. DirectByteBuffer use, from WebSphere Application Server 6.0.2 on:
Change the WebContainer to send data synchronously.
3. DirectByteBuffer use, part II. WSAS V6.1 & V7.0:
Disable the AIO (Asynchronous Input/Output) native libraries.
4. Thread pool issue. Used fixed size thread pools to avoid dangling ThreadLocal objects.

Question 5

What are some of the potential memory suspects in WebSphere env I should be aware of, when dealing with Java heap and native memory issues?

Answer to Question 5

From a Java heap perspective, based on the heapdump analysis:

- Session data usage - both size and number of sessions
- Dynacache - reduce the maximum number of items that are cached
- Check the application code for the identified collections such as HashMap, to see if the heap usage is expected or if there is a potential leak.

Answer to Question 5

From a native memory usage perspective:

- Maximum Java heap settings
- AIO native libraries (WSAS V6.1 & above)
- Async/Sync for WebContainer response data
- JDBC driver
- JNI code or Native libraries
- garbage collection of unused classes. Ensure that `-Xnoclassgc` is not set.
- Thread Pool settings (fixed size thread pools)
- Too many classloaders etc, but these are not very common. Number of classes/classloaders from javacores.

Question 6

We have had success turning off AIO on AIX and observed no performance issues. What types of performance issues should we be looking for?

Answer to Question 6

It depends. Different applications with different loads are going to have different impacts. Applications that write very small response data and have high volume do well with AIO. Applications that write large response data tend to not do as well with AIO. Most real world applications are in the middle, and you will probably notice no performance difference at all with the AIO libraries disabled.

AIO can also have a larger native memory footprint. For systems where native memory is constrained, the performance benefit of AIO may be offset by the amount of memory management overhead.

Question 7

I am developing a solution for monitoring WebSphere Application Servers. I use an AdminClient program to monitor for things like WebContainer hung threads, DataSource waiting threads, etc. Is there any way to detect if there has been an OutOfMemory Exception thrown. I realize a script could check for heap dumps or check native_stderr.log but I'm looking for something that can be checked from a client program using something like the JMX/PMI interface.

Answer to Question 7

You can use the JVMTI (JVM™ Tool Interface) to capture OutOfMemoryError errors:

```
JVMTI_RESOURCE_EXHAUSTED_JAVA_HEAP  
JVMTI_RESOURCE_EXHAUSTED_OOM_ERROR
```

But this will require you to write a native library.

“JVM™ Tool Interface”

<http://java.sun.com/javase/6/docs/platform/jvmti/jvmti.html>



Open Lines for Questions



We Want to Hear From You!

Tell us about what you want to learn

Suggestions for future topics
Improvements and comments about our webcasts
We want to hear everything you have to say!

Please send your suggestions and comments to:
wsehelp@us.ibm.com

Summary

- Differentiate between Java heap and native memory errors.
- Documentation to collect to debug Java heap and native memory issues.
- How to debug Java heap issues
- How to debug native memory issues
- Potential memory suspects in WebSphere env

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:
http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere User Group Community:
<http://www.websphere.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:
<http://www.ibm.com/software/support/einfo.html>