



IBM Software Group

SW5706 JVM™ Tools



© 2007 IBM Corporation

4.0

This presentation will act as an introduction to JVM tools.

JVM tools for tuning and problem detection

After completing this topic, you should be able to:

- Describe the main tools used for tuning and JVM problem detection:
 - ▶ Tivoli Performance Viewer
 - ▶ Thread Dumps
 - ▶ VerboseGC
 - ▶ Memory Dump Diagnostic Tool for Java (MDD4J)



After completing this topic, you should be able to describe the main tools used for JVM tuning and problem detection.

Key tuning and debugging tools

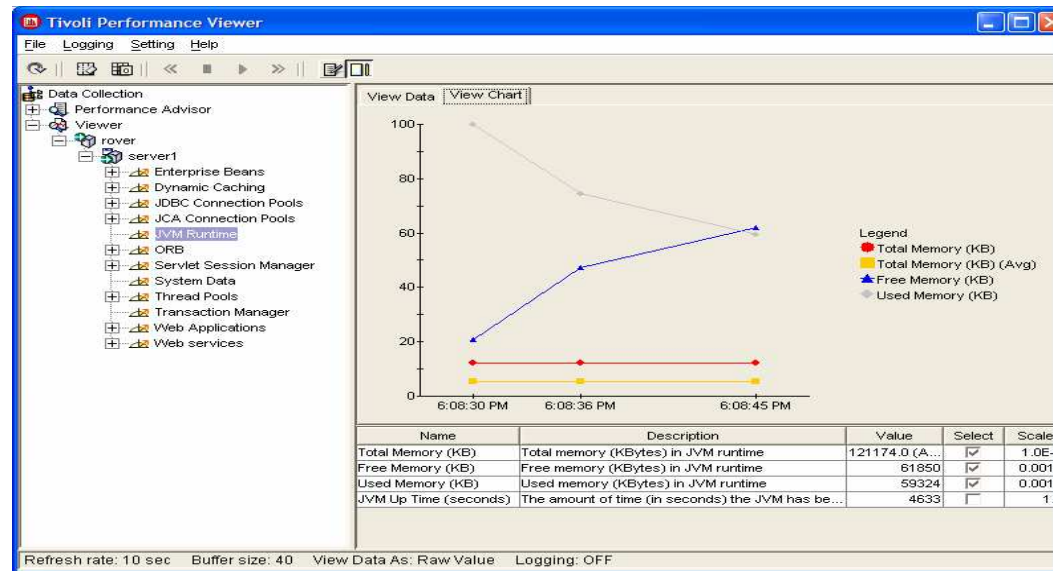
- WebSphere internal thread pools and heap statistics
 - ▶ Tivoli Performance Viewer (TPV)
- Thread activity snapshot
 - ▶ Thread dumps/ThreadAnalyzer
- Memory and garbage collection
 - ▶ verbosegc
- Memory utilization by object
 - ▶ Memory Dump Diagnostic Tool for Java (MDD4J)



Most tuning and debugging can be accomplished with free tools that many customers have on-hand, including using ThreadAnalyzer for thread dumps, using verbosegc and MDD4J for heap analysis, and using Tivoli Performance Viewer for overall WebSphere® Application Server health. High-end tools, such as IBM Tivoli Composite Application Manager for WebSphere or other vendor products, are also useful.

Tivoli Performance Viewer

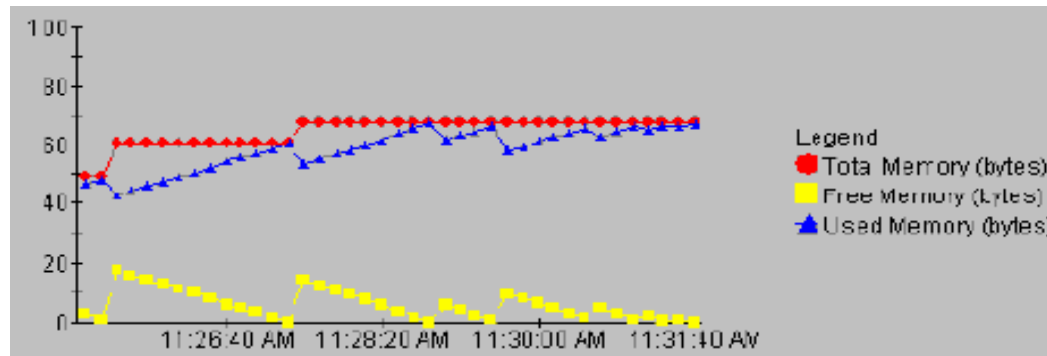
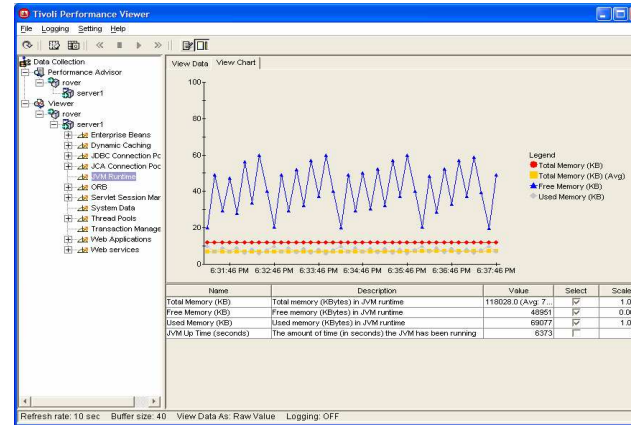
- Tivoli Performance Viewer (TPV)
 - ▶ internal thread pools and heap statistics
- Know your application's expected behavior – I/O intensive, JMS, number of EJBs, expected load requirements



The Tivoli Performance Viewer, or TPV, can be used to monitor overall heap usage and the internal threads. If needed, even more specific information can be collected on garbage collections, but this option adds a large overhead to the process. See the Information Center for details. TPV is an integrated tool provided with WebSphere Application Server. There are more sophisticated monitoring tools available from Tivoli and other vendors.

Debug memory leaks

- Use TPV JVM runtime monitor to identify memory leak pattern.
- Use profiler for deep dive - Rational® PurifyPlus™



Use TPV to monitor the heap usage. If the used heap continues to grow overtime with no corresponding increase in user load there may be a memory leak. Use a profiler dig into the specific heap to determine where the problem may exist.

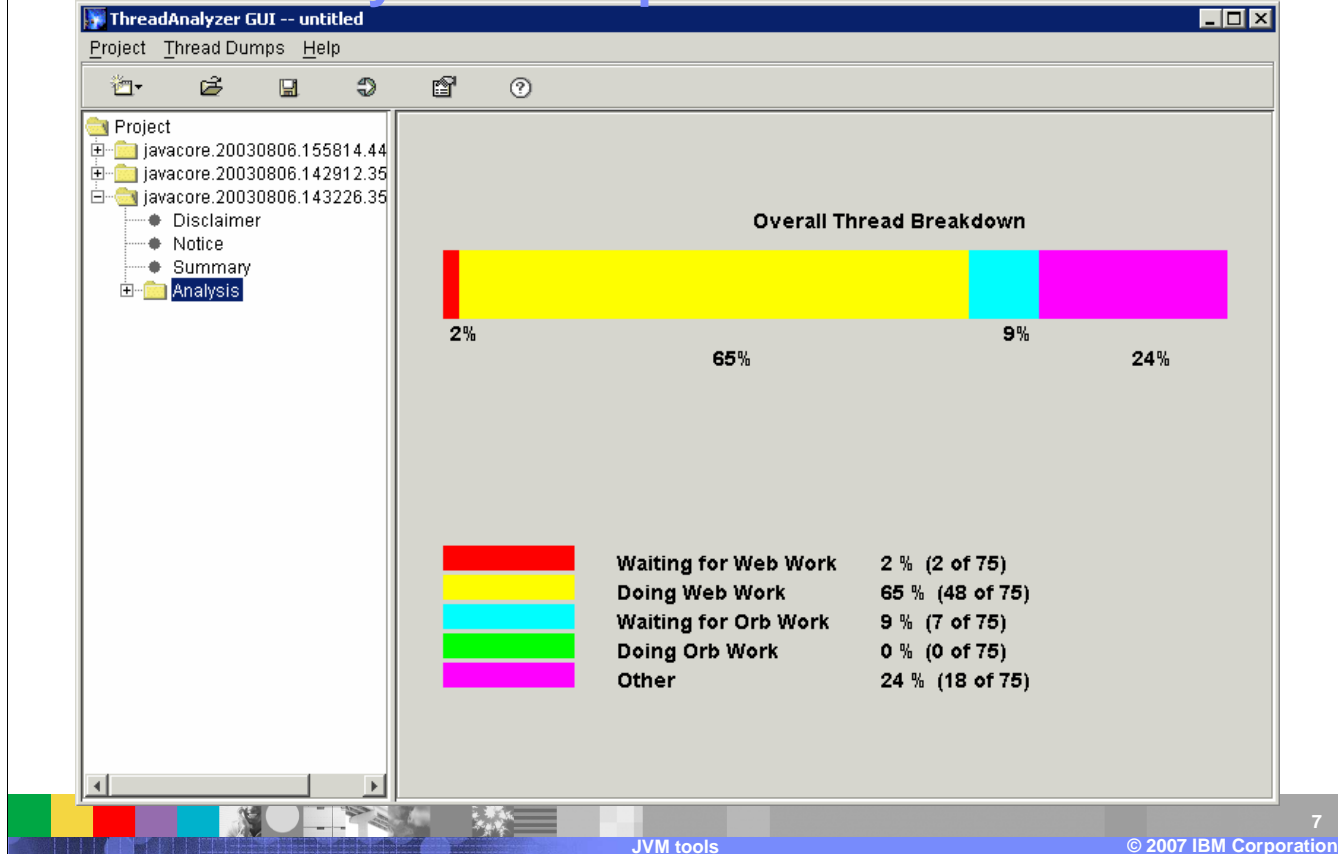
ThreadAnalyzer tool

- A tech-preview tool to analyze thread dumps
 - ▶ Also useful for initiating a thread dump on Windows
 - ▶ Available as a plug-in to IBM Support Assistant
 - <http://www-306.ibm.com/software/support/isa/>
- Useful for analyzing synch points in large dumps
- Provides graphical interface to view contents of the thread dump
- Gathers and analyzes thread dumps from a WebSphere Application Server
- Use to analyze threads for the following
 - ▶ Performance bottlenecks due to either WebSphere configuration or application problems
 - ▶ Determining if deadlocks are being created
 - ▶ Determining if threads are being blocked on monitors (may not be a deadlock)



The thread analyzer is a tool that can take a thread dump or javacore from the server JVM and run an analysis to help determine the cause of the problem. This was done manually before, but now is an automated process with a GUI interface. By using the top of stacks (TOS) analysis found in the Overall Thread Analysis section, you can use Thread Analyzer to pin-point areas of high contention in your code. Thread Analyzer is supported on Solaris, AIX, Windows, and Linux platforms, and is available as a plug-in to the IBM Support Assistant.

Thread Analyzer example



Here is an example screen of the Thread Analyzer providing an overall thread work breakdown for a particular javacore file.

Thread dump analysis

- Look at the big picture
- What are the trends in the thread dump?
 - ▶ Lots of threads waiting in the same method for some resource
 - Probably a synchronization issue
 - Could be a remote outage
 - ▶ No activity
 - WebSphere is not receiving traffic for some reason
 - Check front-end resources, networks, test clients
 - Also check timing of the thread dump
 - ▶ Hundreds of threads
 - Shared resource not available
 - Customer needs to control Web Container threads better
 - May need more capacity

By taking several thread dumps over short, three-minute intervals, trends may be identified in the thread work. Lots of threads waiting in the same method could be a synchronization issue. Hundreds of threads could suggest that a shared resource is not available, or that more server capacity is needed.

Verbosegc tools

- Various tools available for GC analysis
 - ▶ GC Collector on alphaworks
 - ▶ PMAT
 - ▶ <http://www.alphaworks.ibm.com/tech/gcdiag>

- Keep the JVM edition in mind
 - ▶ Not all tools support all JVMs
 - ▶ GC Collector is one example
 - ▶ Tools not rated for the JVM may provide misleading/wrong results



Various tools are available for garbage collection analysis. For example, PMAT (Pattern Mapping Analysis Tool) is a common tool used to analyze Verbose:gc output for IBM JVMs. It is packaged within the IBM Support Assistant and can be launched from within IGAA. PMAT will display all GC events in text, tabulated, and graphical views. It will let you know if you are dealing with insufficient heap space or large object allocations.

Memory Dump Diagnostic Tool for Java (MDD4J)

- Memory Dump Diagnostic Tool for Java
 - ▶ Available from the “Tools” tab in ISA
- Analyzes Java memory (heap) dumps
 - ▶ Identifies data structures that are likely causes of memory growth (leaks) and their relationships
- Handles several dump formats
 - ▶ IBM portable heap dump (.phd)
 - ▶ IBM text heap dump (.txt)
 - ▶ HPROF dump (hprof.txt)
 - ▶ z/OS® SVC dump (dump.bin)

IBM Support Assistant includes a tool for inspecting Java memory dumps, called the Memory Dump Diagnostic Tool for Java, or MDD4J. You can launch it, and any diagnostic plug-ins that you might download through the “Updater” component from the “Tools” tab. This tool can analyze several Java heap dump formats, including z/OS SVC dumps and standard dump formats from both the IBM and Sun Java Runtime Environments. It displays and analyzes the data structures in the heap and their relationships, helping you identify the structures that are most likely responsible for memory leakage.

Using memory dump diagnostics

- Single dump analysis
 - ▶ Inspects a single heap dump file for container objects that have very large reach size compared to their largest child object
 - ▶ Most commonly used to analyze automatically generated heap dumps after an `OutOfMemoryException`
- Comparative analysis
 - ▶ Analyzes heap growth between two dumps taken over time
 - ▶ Identifies the objects with the largest size difference and their ownership relationships
- Analysis results
 - ▶ Summary of analysis results showing heap contents, size and growth
 - ▶ Lists suspected data structures, data types and packages contributing to the growth in heap usage
 - ▶ Tabular views of all the objects and data-types in the memory dump with filters and sorted columns



The Memory Dump Diagnostic Tool for Java can either perform analysis on a single heap dump file or perform comparative analysis on two heap dumps taken over time. You will most often use the first option to analyze a heap dump that was automatically generated at the time of an `OutOfMemoryException`. This method looks for objects in the heap that have very large reach sizes compared to their largest child objects. The comparative analysis feature examines the changes in the heap between two heap dumps taken over time to analyze which objects have grown the most. If you believe an application is leaking memory, a comparative analysis of two dumps separated by several minutes will often highlight the objects that are the most likely source of leaking memory, helping you to quickly locate parts of the application's code that should be examined. The analysis results panel displays the results of memory analysis in several ways. It displays first a list of suspected data structures, which are the objects in your heap that are most likely the cause of memory growth. The data in the analysis results screen can be easily sorted and filtered to make it easy to locate objects that interest you.

Some useful Web addresses

- IBM JVM Diagnosis documentation:
 - ▶ <http://www.ibm.com/developerworks/java/jdk/diagnosis/>
- IBM JRE and JDK™ forum:
 - ▶ http://www.ibm.com/developerworks/forums/dw_forum.jsp?forum=367&start=0&thRange=15&cat=10
- Memory leak detection and analysis in WebSphere Application Server
 - ▶ http://www.ibm.com/developerworks/websphere/library/techarticles/0606_poddar/0606_poddar.html



Here are some useful sites regarding IBM JDK problem determination.

Related courses and redbooks

- WF881 course
 - ▶ *IBM WebSphere Application Server V6 Performance Monitoring and Tuning for Administrators*

- IBM Redbooks
 - ▶ *WebSphere Application Server V6 Problem Determination for Distributed Platforms, SG24-6798-00*



Here are some related courses and IBM RedBooks.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

<mailto:iea@us.ibm.com?subject= Feedback about SW5706G14C JVM Tools.ppt>

You can help improve the quality of IBM Education Assistant content by providing feedback.

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 PurifyPlus Rational Redbooks WebSphere z/OS

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

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

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

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 or changes in the products or programs described herein at any time without notice.

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 (for example, 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.

© Copyright International Business Machines Corporation 2007. 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.