

Java Performance Tuning and Optimization

Student Guide

D69518GC10

Edition 1.0

June 2011

D73450

ORACLE®

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Authors

Clarence Tauro, Michael Williams

Technical Contributors and Reviewers

Charlie Hunt, Staffan Friberg

This book was published using: Oracle Tutor

Table of Contents

Introduction	1-1
Introduction.....	1-2
Course Goal	1-3
Course Objectives.....	1-4
Class Introductions	1-5
Audience	1-6
Prerequisites.....	1-7
Course Map.....	1-8
Course Topics	1-9
Course Schedule	1-10
Course Environment	1-11
Additional Resources	1-12
JVM and Performance Overview	2-1
JVM and Performance Overview	2-2
Objectives.....	2-3
JVM Overview	2-4
Java Programming Language.....	2-5
HotSpot JVM: Architecture	2-6
Key HotSpot JVM Components	2-7
What Is Performance?.....	2-8
Memory Footprint.....	2-9
Startup Time.....	2-10
Scalability	2-11
Application Scalability	2-12
Responsiveness	2-13
Throughput.....	2-14
Performance Focus for This Course	2-15
Performance Issues Covered in This Course	2-16
Performance Issues Not Covered in This Course	2-17
Performance Methodology	2-18
Performance Monitoring.....	2-19
Performance Profiling.....	2-20
Performance Tuning.....	2-21
Typical Development Process	2-22
Application Performance Process	2-23
Summary.....	2-24
Java Performance Resources.....	2-25
Additional Resources	2-26
Monitoring Operating System Performance.....	3-1
Monitoring Operating System Performance.....	3-2
Objectives.....	3-3
Why Are We Monitoring?.....	3-4
Monitoring CPU Usage Overview	3-5
CPU Monitoring Performance Indicators	3-6
Voluntary Context Switching (VCX).....	3-7
Involuntary Context Switching (ICX).....	3-8
Tools For Monitoring CPU Usage	3-9

CPU Usage: vmstat.....	3-10
CPU Usage: mpstat	3-11
CPU Usage: prstat	3-12
CPU Usage: prstat -m	3-13
CPU Usage: prstat -Lm	3-14
CPU Monitoring: Solaris - cpubar.....	3-15
Map Lightweight Processes to Java Threads	3-16
Monitoring Network I/O Overview	3-17
Network I/O: Using tcptop.....	3-18
Network I/O: Using nicstat	3-19
Monitoring Disk I/O Overview	3-20
Disk I/O: iotop	3-21
Monitoring Virtual Memory: Overview.....	3-22
Virtual Memory Tools	3-23
Virtual Memory: vmstat.....	3-24
Virtual Memory: Swapping Example	3-25
Monitoring Virtual Memory: cpubar	3-26
Virtual Memory: Fixing the Swapping Problem	3-27
Monitoring Processes Overview.....	3-28
Process Monitoring Tools	3-29
Processes: prstat-Lm	3-30
Processes: mpstat	3-31
Monitoring the Kernel	3-32
Kernel: vmstat.....	3-33
Kernel: mpstat	3-34
Kernel: prstat-Lm	3-35
Summary.....	3-36
Monitoring the JVM.....	4-1
Monitoring the JVM.....	4-2
Objectives.....	4-3
What to Monitor	4-4
HotSpot GC Basics	4-5
The Young GC Process: Part 1	4-6
The Young GC Process: Part 2	4-7
The Young GC Process: Part 3	4-8
The Young GC Process: Part 4	4-9
The Young GC Process: Part 5	4-10
The Young GC Process: Summary	4-11
Young Generation Recap	4-12
Tenured (old) Generation	4-13
Permanent Generation.....	4-14
Tools for Monitoring GC	4-15
Using -verbose:gc.....	4-16
Additional -verbose:gc Print Options	4-17
Using -XX:+PrintGCDetails.....	4-18
Printing Pause Time.....	4-20
Using jps	4-21
Using jstat.....	4-22
Using jconsole	4-23

Using VisualVM	4-24
Using VisualGC	4-25
Using GCHisto.....	4-26
Monitoring JIT Compilation.....	4-27
Using -XX:+PrintCompilation	4-28
What Is the .hotspot_compiler File?	4-29
Using .hotspot_compiler File	4-30
Focusing on Throughput.....	4-31
Focusing on Responsiveness.....	4-32
Summary.....	4-33
Performance Profiling.....	5-1
Performance Profiling.....	5-2
Objectives.....	5-3
Tools for Profiling Java Applications	5-4
NetBeans and NetBeans Profiler	5-5
NetBeans	5-6
Oracle Solaris Studio	5-7
jmap and jhat	5-10
Profiling Tips.....	5-11
CPU Profiling: Why and When.....	5-12
CPU Profiling: Strategies.....	5-13
CPU Profiling Entire Application.....	5-14
CPU Profiling a Portion of the Application	5-15
Heap Profiling: Why and When.....	5-16
Heap Profiling: Strategies.....	5-17
Heap Profiling: jmap/jhat	5-19
Heap Profiling: jmap/jhat Strategies	5-20
Memory Leak Profiling: Why and When	5-21
Memory Leaks Profiling Tips: Tools	5-22
Memory Leaks: NetBeans/VisualVM Strategies.....	5-23
Memory Leaks: jmap/jhat Strategies	5-24
Lock-Contention Profiling: Overview	5-25
How to Reduce Lock Contention.....	5-26
Biased Locking	5-27
Inlining Effect.....	5-28
Identifying Anti-Patterns	5-29
Anti-Patterns in Heap Profile	5-30
Anti-Patterns in Heap Profiles.....	5-31
Anti-Patterns in Method Profiles.....	5-34
Summary.....	5-36
Garbage Collection Schemes	6-1
Garbage Collection Schemes	6-2
Objectives.....	6-3
Garbage Collection Basics	6-4
Normal Deletion.....	6-6
Deletion with Compacting.....	6-7
Generational Garbage Collection.....	6-8
Why Generational GC?	6-9
Generational Garbage Collection: Major Spaces	6-10

Features of Young Generational Space	6-11
Features of Old Generation Space.....	6-12
Generational Garbage Collection.....	6-13
Garbage Collection Notations.....	6-14
Generational Garbage Collection: Young Collection	6-15
GC Performance Metric.....	6-17
Choices of Garbage Collecting Algorithms	6-18
Serial versus Parallel	6-19
Stop the World Versus Concurrent.....	6-20
Compacting Versus Non-Compacting Versus Copying	6-21
Types of GC Collectors	6-22
Serial Collector	6-23
Serial Collector on Young Generation: Before.....	6-24
Serial Collector on Young Generation: After.....	6-25
Serial Collector on Old Generation.....	6-26
Parallel Collector: Throughput Matters!	6-27
Parallel Collector on Young Generation	6-28
Parallel Compacting Collector.....	6-29
Concurrent Mark-Sweep (CMS) Collector	6-31
CMS Collector Phases	6-32
CMS Collector on Old Generation.....	6-33
Garbage Collectors: Comparisons	6-35
Ergonomics: What It Does.....	6-36
Ergonomics	6-38
Summary.....	6-39
Garbage Collection Tuning.....	7-1
Garbage Collection Tuning.....	7-2
Objectives.....	7-3
Garbage Collectors: Recap	7-4
Garbage Collection: Myth.....	7-5
Garbage Collectors: Sizing Java Heap Spaces	7-6
Garbage Collectors: Sizing Spaces.....	7-8
Garbage Collectors: The Choices	7-9
Garbage Collectors: Serial Collector	7-11
Garbage Collectors: Throughput Collector	7-14
Garbage Collectors: Concurrent Collector	7-19
Serial Collector Versus Parallel Collector	7-28
Parallel Collector Versus CMS Collector	7-29
Garbage Collectors: Permanent Generation.....	7-30
GC Output: Using Serial Collector	7-32
GC Output: Using Throughput Collector.....	7-34
GC Output: Using Concurrent Collector	7-36
Concurrent Collector: Losing the Race.....	7-38
Garbage Collectors: PrintGCStats	7-39
Garbage Collectors: GCHisto	7-42
Summary.....	7-43
Language-Level Concerns and Garbage Collection	8-1
Language-Level Concerns and Garbage Collection.....	8-2
Objectives.....	8-3

Object Allocation: Best Practices	8-4
Object Allocation: Working with Large Objects	8-5
Garbage Collectors: Explicit GC	8-6
Data Structure Sizing	8-7
Garbage Collectors: Reference Objects	8-8
Reference Objects, Illustration (1/2)	8-9
Reference Objects: Illustration (2/2)	8-10
Reference Objects: Soft Reference	8-11
Reference Objects: Weak Reference	8-12
Reference Objects: Phantom Reference	8-13
Garbage Collectors: Soft References	8-14
Memory Leaks	8-15
Garbage Collectors: Finalizers	8-17
Finalizers Versus Destructors	8-18
Summary	8-19
Performance Tuning at the Language Level	9-1
Performance Tuning at the Language Level	9-2
Objectives	9-3
Strings: An Introduction	9-4
Compile Time Versus Runtime Strings	9-5
How JVM Works with Strings	9-6
Optimization by Interning Strings	9-7
String Versus StringBuffer Versus StringBuilder	9-8
StringBuffer Versus StringBuilder (in Nanoseconds)	9-9
Execution of "+" Operator in String (in Milliseconds)	9-10
Exception Handling and Performance	9-11
Exception Handling and Performance (in Nanoseconds)	9-12
Primitives Versus Objects	9-13
Primitives Versus Objects [Benchmark Results]	9-14
Prefer Reusing Objects	9-15
Thread Synchronization	9-16
Collections	9-18
Performance of Java Collections	9-20
Benchmark Results	9-21
Copying Entire Array: Use System.arraycopy	9-22
Tuning Java I/O Performance	9-23
Speeding Up I/O	9-24
Speeding Up I/O: Approach 1	9-25
Speeding Up I/O: Approach 2	9-26
Speeding Up I/O: Approach 3	9-27
Speeding Up I/O: Performance Benchmarking	9-28
Buffering	9-29
Reading/Writing Text Files	9-30
Formatting Costs	9-32
Formatting Costs: Use MessageFormat Class	9-34
Formatting Costs: Benchmarking	9-35
Random Access	9-36
Compression	9-37
Tokenization	9-38

Serialization.....	9-39
Summary.....	9-40
Appendix A: Monitoring Linux Performance.....	10-1
Monitoring Linux Performance.....	10-2
Objectives.....	10-3
Tools For Monitoring Linux.....	10-4
CPU Usage: Linux - vmstat.....	10-5
CPU Usage: Linux mpstat.....	10-6
CPU Usage: pidstat.....	10-7
Linux – Gnome System Monitor.....	10-8
Monitoring Network I/O: Using nicstat.....	10-9
Monitoring Disk I/O: pidstat Example.....	10-10
Kernel Monitoring Using: pidstat.....	10-11
Summary.....	10-12
Appendix B.....	11-1
Appendix B.....	11-2
Objectives.....	11-3
Java HotSpot VM Options.....	11-4
Categories of Java HotSpot VM Options.....	11-5
Useful -XX Options.....	11-6