

Isolating Performance Bottlenecks and Memory Leaks

Jaroslav Bachorik
jaroslav.bachorik@sun.com

Agenda

- Introduction
- Case Study: Memory Leak in JMeter
- Case Study: Performance bottleneck in Roller engine
- Summary: Lessons Learned
- Links & Pointers
- Q&A

Introduction

Performance Problems

- slow application startup
- unresponsive application
- application gradually depletes the memory without obvious reason

Performance Bottleneck

- A part of an application preventing the rest of the application to run at full speed
- Possible Causes
 - Threading
 - Inefficiencies in code
 - I/O congestion

Memory Leak

- A bug in a program that causes an object not being garbage collected even if it's no longer needed
- Possible Results
 - Application crashes with OOME
 - Increased GC activity consumes still more and more CPU time

Case Study:
**Memory Leak in
JMeter**

Memory Leak in JMeter

- open source load generator (<http://jmeter.apache.org>)
- version 2.3 (current)
- 693 classes
- cca. 120000 LOC
- randomly experienced severe memory leak

Note: VisualVM (<http://visualvm.dev.java.net>) will be used

Case Study:
Roller
Performance
Problems

Roller Performance Problems

- Open source blog server
(<http://rollerweblogger.org/project/>)
- Version 2.3 (June, 2006)
- Struts based
- >400 Classes, >190 JSP s
- Almost 40,000 Lines of Code
- Problem with slow performance

Note: NetBeans integrated profiler (<http://profiler.netbeans.org>) will be used

Summary

Reproducible Case

- Captures the steps leading to a performance problem
- Makes it possible to recreate problems in a controlled environment
- Automate it if possible

Use Right Tools

- **No Golden Hammer**
- **JDK monitoring**
 - jps, jstat, jinfo, jstack, jmap
- **VisualVM**
 - combines JDK monitoring tools in user friendly frontend
- **Profilers**
 - used to pinpoint “the offending line”

Links & Pointers

- **JDK Troubleshooting Guides**

- <http://java.sun.com/javase/6/webnotes/trouble/TSG-VM/html/docinfo.html>
- http://java.sun.com/j2se/1.5/pdf/jdk50_ts_guide.pdf

- **Java Performance Tuning Site**

- <http://www.javaperformancetuning.com>

- **List of Open Source Profilers**

- <http://www.manageability.org/blog/stuff/open-source-profilers-for-java>

Q&A