

Improving Systems with JBossProfiler

Clebert Suconic
Senior Developer, QA
JBoss Inc.

March 8, 2005

© JBoss Inc. 2005

Agenda

- Profiler
 - ✓ Motivation
 - ✓ Architecture
 - ✓ Features
 - ✓ Modules
 - ✓ New Features Coming
- Practices profiling
- Profiling Production

2

Motivation

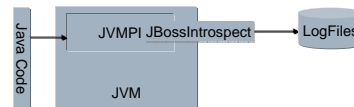
- Intercepting Log files
 - ✓ Started with a simple log in every start/exit method
 - ✓ Before we had to instrument every method manually.
 - Log4J and System.out
 - Manually analyzes the flow
 - ✓ Annalists far from front (support structure)

3

JBossProfiler Architecture

• Introspecting Events

- ✓ LogFiles and disconnection is the key in this architecture
- ✓ Unless you introspect everything the architecture is Light-weight.
 - As the analysis will happens later



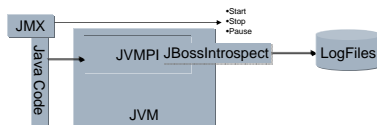
Acronyms: JVMPI – Java Virtual Machine Profile Interface
JVMTI – Java Virtual Machine Tool Interface (New version that replaces JVMPI)

4

JBossProfiler Architecture

• Introspecting Events

- ✓ Capture only when you are ready
- ✓ Automation through JMX, deployment as a SAR

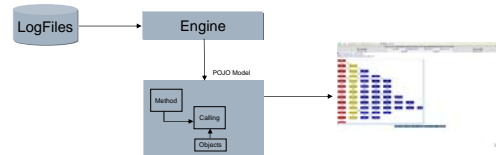


5

JBossProfiler Architecture

• Analyzing events

- ✓ A web application analyzes the application

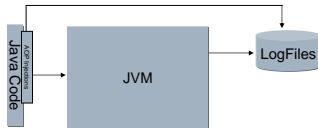


6

JBossProfiler Architecture

- AOP Interceptor

- ✓ Small interceptions in selected points
- Advantages:
 - ✓ Doesn't scare administrators ☺
- Disadvantages:
 - ✓ Not accurate as JMTI/JMPI regarding memory consumption
 - ✓ For example, java.lang is not passive of interception

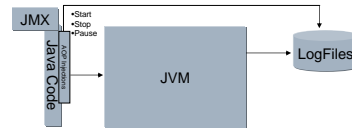


7

JBossProfiler Architecture

- AOP Interceptor

- ✓ There is a *JMX* to control life cycle of AOP interceptor as well.



8

JBossProfiler Features

- Process analysis

- ✓ Inter method relationships
 - Complete graphs everywhere
- ✓ Memory consumption
 - Graphs of allocated objects
 - Releases of objects
- ✓ Connection between Process Analysis and Memory Analysis
- ✓ XML export



9

JBossProfiler Features

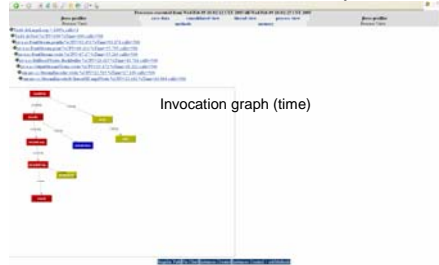
- Inter Method Relationship



10

JBossProfiler Features

- Inter Method Relationship



11

JBossProfiler Features

- Inter Method Relationship



12

JBossProfiler Features

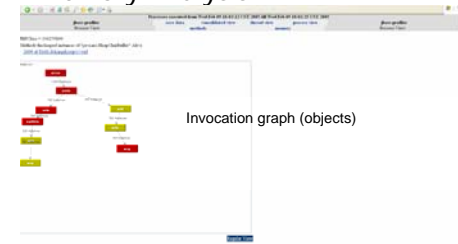
- Memory Analysis
 - ✓ GC Analysis
 - Object Creations vs Where its creations
 - Object Releases vs Where its creations
 - ✓ Memory Leaks
 - Creation vs releases



13

JBossProfiler Features

- Memory Analysis



14

JBossProfiler Features

- New Features Coming
 - ✓ Already under construction:
 - AOPMonitoring – Monitoring any method historically by adding interceptors.
 - JMX Beans for monitoring this stuff.
 - AntProfiler
 - AntJUnitProfiler
 - ✓ For a near future:
 - JVMTI
 - Eclipse plugin



15

Profiling Practices

- This depends on the kind of application you are building. Suppose a J2EE Web application for instance
- During development/QA:
 - ✓ Common Problems:
 - HttpSession abnormal use
 - ✓ Commonly customers intent to use HttpSession as Cache services.
 - ✓ Excessive use of layers
 - ✓ By trying to scale up by adding lots of components you can end up adding lots of garbage in your applicatoins
 - ✓ JDBC Protocol
 - ✓ Database calls



16

Profiling Practices

- During production:
 - ✓ Profile first in development, them:
 - ✓ Network contentions
 - ✓ Connection Pools limits
 - ✓ Thread limits
 - ✓ Synchronizations between callings
 - ✓ Monitoring AOP + other monitoring tools and MBeans
 - ✓ GCs and memory



17

Profiling Practices

- During production:
 - ✓ Using basic stuff also. (JVM 5 has new tools for memory analysis)
 - ✓ Usually those problems are intermitents.
 - Activate



18

Demo

- Live demo
 - ✓ Simple Source Example
 - ✓ Them Live demo



19

Q&A

- Q&A



20