

Tracer for JBoss Profiler

Yoko SEKI
Hitachi, Ltd.

Agenda

- Proposal & Goal
- Design of Tracer
- Screen Shots
- Future Work

Note: Described company name and product name are the registered trademarks or trademarks of each company.

Preface (1)

Japan OSS Promotion Forum

- <http://www.ipa.go.jp/software/open/forum/index-e.html>
- Formed by outstanding system-integrators, vendors, Linux distributors.
- Evaluation and Development of Tools
 - ✓ Linux (RedHat, SUSE, Miracle, ...)
 - ✓ Database (PostgreSQL, MySQL, MaxDB, ...)
 - ✓ Application Server (JBoss, Tomcat, ...)
- Partly supported by Software Development Support Programs of IPA (Information-Technology Promotion Agency, JAPAN)

Preface (2)

Activities regarding JBoss

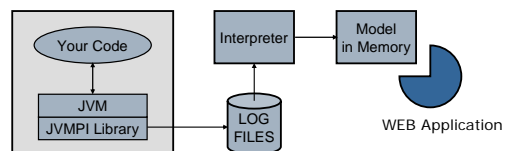
- Evaluation of Performance using SPECjAppServer2004
- Evaluation of Availability of Clustered JBoss
- Development of Microbenchmark Tool (JBento)
- Development of Tracer for JBoss Profiler
- ...

Proposal & Goal

- Profiler
 - ✓ Is a useful tool for runtime analysis.
 - ✓ Visualizes statistics data of total calls on any method
- Proposal
 - ✓ Transaction Tracer
 - "Transaction" corresponds to individual call on any method
 - "Transaction Tracer" traces path of methods executed in any transaction, and visualizes it
 - Based on JBoss Profiler
- Goals
 - ✓ Debug, Test, Failure Analysis

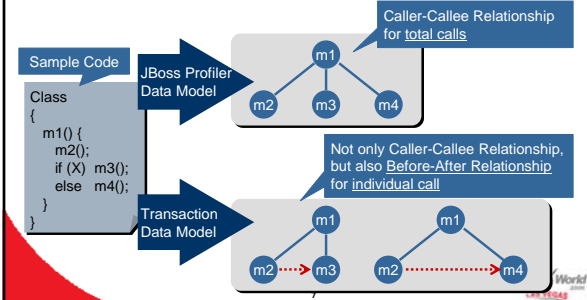
Design of Tracer

- Uses JBoss Profiler's architecture
- Added some modifications to JVMPI Library, Interpreter and WEB Application
- Created new data model (transaction data model)



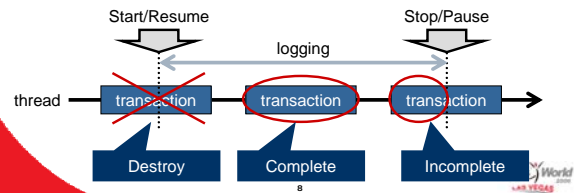
Transaction Data Model

When you choose a method on JBoss Profiler view, transaction data for the method is constructed.



Algorithm

- Picks up MethodEntry & MethodExit events from log files, matches them each other sequentially on each thread, and constructs transaction data
- Stores complete and incomplete transaction data in memory



Failure Detection

- With "tracer" option, JVMPI library logs all Exceptions & Errors
 - Exceptions & Errors are included in transaction path, and visualized
 - Useful to locate failures in transaction
- Usage example
 - JAVA_OPTS="-Xrunjbossinspector:/log_directory include=sample, ignore=*, tracer=true"
- Open Issue : Exceptions & Errors caught in catch-clauses and normally handled are also visualized ...

Tracer View

Tracer View

List View

Start Time	End Time	Total Time	Method	CPU	Locks
08/25/2005 10:34:34.808	NOT EXIT	UNKNOWN	29	UNKNOWN	
08/25/2005 10:34:31.332	08/25/2005 10:34:31.386	54	19	60000000	
08/25/2005 10:34:31.316	08/25/2005 10:34:31.369	53	19	50000000	
08/25/2005 10:34:31.007	08/25/2005 10:34:31.172	165	36	140000000	

Tracer View

The screenshot shows the JBoss Profiler interface with the Tracer View selected. A blue starburst icon with the text "Click 'Tracing'" points to a button. A red box highlights the Search View section at the bottom of the interface.

JBoss Profiler View

Tracer View

List View

Search View

Click "Tracing"

13

Search View

The screenshot shows the Search View interface. A blue box points to the "Method you chose on JBoss Profiler view" field. Another blue box points to the "Complete or incomplete transaction, or both" radio buttons. A third blue box points to the "Various search conditions" section. A blue arrow labeled "Search starts!" points to the Search button.

Method you chose on JBoss Profiler view

Complete or incomplete transaction, or both

Various search conditions

Search starts!

The result of search refreshes "List View".

14

Transaction View

The screenshot shows the Transaction View interface. A blue starburst icon with the text "Click icon!" points to a button. A blue box labeled "Methods executed in transaction" points to a list of methods. Other blue boxes point to various columns in the table: "Start time", "Finish time", "Total time", "Cpu usage time", "Total time of locks", and "The number of locks".

List View

Transaction View

Click icon!

Methods executed in transaction

Start time

Finish time

Total time

Cpu usage time

Total time of locks

The number of locks

15

Examples

The screenshot shows the Examples section of the JBoss Profiler. It displays two panels: "Path of Methods Executed in Transaction" and "Failure Detection". A blue box labeled "Caller-Callee Relationship" points to the first panel. A blue box labeled "Before-After Relationship" points to the second panel. A blue box labeled "RuntimeException occurs." points to a specific entry in the Failure Detection panel. A blue box labeled "For Debug and Test!" points to the top of the first panel. A blue box labeled "For Failure Analysis!" points to the top of the second panel.

Path of Methods Executed in Transaction -

Caller-Callee Relationship

Before-After Relationship

Failure Detection -

RuntimeException occurs.

For Debug and Test!

For Failure Analysis!

16

Present Situation

JBoss Profiler with transaction tracer is available from JBoss web site now !

17

Future Work

To reduce volume of visualized data for efficient analysis

- Grouping
 - ✓ Groups some methods in transaction and visualizes Caller-Callee Relationship among them
- Filtering
 - ✓ Filters methods in transaction and visualizes remained methods only
- Summarizing
 - ✓ Shows statistics data of each method in transaction

18