

JRokit VM Log Visualizer

Diplomarbeit

Hochschule für Technik in Zürich

Student: Raffael Schmid, raffael.schmid@trivadis.com
Dozent: tbd
Auftraggeber: Trivadis AG
Adrian Hummel, adrian.hummel@trivadis.com
Studiengang: Informatik

1 Ausgangslage

Application Performance Managements (APM) als Disziplin des System Managements kann als einen Prozess angeschaut werden, bei dem es mittels Hilfe von verschiedenen Werkzeugen darum geht, die Performance von Applikationen zu überwachen, um damit die Anforderungen der verschiedenen Stakeholders erfüllen zu können. Aufgeteilt in die Gruppen System und User lassen sich die auftretenden Symptome in folgende Kategorien aufteilen:

- System
 - Threading (Dead Locks, Context Switching, etc.)
 - Network I/O
 - Disk I/O
- User
 - Applikatorisch (ineffiziente Algorithmen, Tasks können nicht parallelisiert werden, etc)
 - Garbage Collection (falsche Strategien, Strategiewechsel, etc.)

Für die Auswertung solcher Probleme gibt es zwei verschiedene Ansätze. Im Bereich von Online-Auswertungen gibt es Werkzeuge wie Profiler, APM-Lösungen die sich nicht kurzfristig installieren lassen (hoher finanzieller Aufwand und lange Integrationszeit) und von den jeweiligen Serverherstellern mitgelieferte Werkzeuge wie JRockit Mission Control. Bei Offline-Analysen geht es meistens um die Auswertung der vom Server geschriebenen Log-Dateien. Wie bereits aus den oben beschriebenen Kategorien ersichtlich kann eine schlechte Applikations-Performance durch falsche Garbage Collection verursacht werden. In diesem Bereich bieten fast alle verfügbaren VMs die Möglichkeit, im Laufenden Betrieb unterschiedliche Log-Dateien zu generieren. Im Nachhinein können diese Dateien toolunterstützt weiterverarbeitet werden. Im Bereich der Sun Hotspot VM gibt es bereits einige Tools (gcviewer, HP JMeter), die einem die Auswertung der Daten zusammenfasst, vereinfacht und in die für Menschen besser zugänglichere grafische Form bringt.

2 Ziel der Arbeit

Wie bereits im Abschnitt "1 Ausgangslage" beschrieben, ist die manuelle Offlineanalyse von Garbage Collection Log-Dateien der JRockit VM zeitintensiv, da die Weblogic spezifische Monitoring Software JRockit Mission

Control diesen Bereich nicht abdeckt. Das Ziel der Diplomarbeit ist es, ein Eclipse-Plugin für Oracle Mission Control oder das Standard-Eclipse zu implementieren, welches die erstellten Log-Dateien grafisch aufbereitet und in eine von Menschen lesbare Form bringt.

3 Aufgabenstellung

Um das angestrebte Ziel dieses JRockit VM Log Visualizers zu erreichen werden folgende Aufgaben umgesetzt:

3.1 Grundlagen

Als Basis für die Diplomarbeit werden folgende Grundlagen erarbeitet:

- JRockit VM und Performance Tuning: Im Bereich der JRockit VM werden die Grundlagen im Bereich der Performance Optimierung erarbeitet. Massgeblich gehört dazu die Thematik der Garbage Collection (verschiedene GC-Strategien und -Algorithmen, Memory Management) und die Unterschiedlichen Konfigurationsmöglichkeiten der Garbage Collection Log-Datei.
- Eclipse RCP: Um die Applikation als Plugin für JRockit Mission Control oder Eclipse zu implementieren, muss ein minimales Wissen im Bereich von Eclipse RCP erarbeitet werden. Ob und wie der Log Visualizer in JRockit Mission Control eingebunden werden kann, wird in dieser Phase anhand eines kleinen Prototypen ermittelt.

3.2 Requirements Engineering

Das Requirements Engineering beinhaltet folgende Aspekte:

- System- und Kontextgrenzen bestimmen
- Ermittlung der Anforderungen aufgrund der zu definierenden Quellen (Stakeholder, Erkenntnisse während der Einarbeitungszeit).
- Dokumentation der Anforderungen (Systemumfeld, Architekturbeschreibung, Funktionalitäten, Nutzer- und Zielgruppen, Randbedingungen, Annahmen) in geeigneter Form

3.3 Konzeption und Implementation

Aufbauend auf der Requirements-Analyse wird das Konzept der Applikation erstellt und Implementiert. Wichtig insbesondere in dieser Phase sind auch die Erkenntnisse aus dem Prototypen in der ersten Phase.

4 Erwartete Resultate

- Dokumentation
 - Grundlagen
 - Requirementsanalyse
 - Konzept
 - Implementationsdetails
- Software
 - Source-Code
 - Installierbare Software (als Plugin oder eigenständige Software)

5 Geplante Termine

| | |
|---------------------|-----|
| Kick-off Meeting | tbd |
| Design Review | tbd |
| Abgabe | tbd |
| Schlusspräsentation | tbd |