

Konzeption und Entwicklung eines Auswertungstools für die Logdateien des JRockit Garbage Collectors

Raffael Schmid

Hochschule für Technik Zürich, Studiengang Informatik (1. Dezember 2011)

Garbage Collection bezeichnet eine Methode in der Programmierung zur automatischen Speicherbereinigung von nicht mehr verwendeten Objekten. Sie findet Anwendung in allen modernen Programmiersprachen, ist in der Regel ressourcenintensiv und hat Einfluss auf das Laufzeitverhalten der Applikation. Ob Performanceprobleme im Zusammenhang mit der Garbage Collection stehen, kann beispielsweise mit der Suche nach dem Dominating Consumer (dominierender Ressourcenverbraucher) [1] eruiert werden. Die Detailanalyse wird anschliessend auf der Basis von Garbage Collection Logdateien oder mit Werkzeugen zur Analyse am laufenden System gemacht. Logdatei-Analysewerkzeuge sind bereits für einige virtuelle Maschinen (Laufzeitumgebungen) verfügbar. Diese Arbeit konzipiert und implementiert eine Software zur automatisierten Analyse der Garbage Collection Logdateien der JRockit virtual Machine (Release R28).

Die Evaluation des Basis-Frameworks, der Bibliothek zum Erstellen von Diagrammen sowie die Konzeption der funktionalen Aspekte werden getrieben durch eine methodenbasierte Anforderungsanalyse. Es standen drei Methoden zur Auswahl: Use Cases, Requirements Engineering nach IEEE 830 und User Stories. Auf Ebene der Customer-Requirements wird die Methode der Use Cases eingesetzt. Sie zeichnet sich dadurch aus, dass man die Anforderungen nicht nur textuell, sondern auch modellbasiert mit UML (Modellierungssprache) definieren kann. Für die Dokumentation der Development-Requirements (auch die Qualitätsanforderungen) wird der Standard IEEE 830 verwendet. Die damit definierten Anforderungen lassen sich gut in einzelne Entwicklungspakete (Features) integrieren. Auch die Struktur der Anforderungsanalyse wurde anhand dieser Definition aufgebaut.

Die Analysesoftware basiert auf dem Rich Client Framework Eclipse 3.x. Für die Anzeige der Diagramme wird JFreeChart verwendet. Die Software wird als Plugin für die Entwicklungsumgebung implementiert. Die Installation kann über das Netzwerk, durch Angabe einer Software-Seite, gemacht werden. Auch Updates können im laufenden Betrieb durchgeführt werden. Der Anwender startet eine Analyse, indem er eine Logdatei aus dem Dateisystem importiert. Er ist nun in der Lage, das Analysefenster für die importierte Logdatei zu öffnen. Die Applikation lädt den Inhalt und wandelt ihn mittels regulären Ausdrücken in eine Reihe von Tokens um (lexikalischer Scanner), anschliessend werden die Daten analysiert und in einen Objektgraphen transformiert. Die Verarbeitung der Daten ist in eine separate Erweiterung ausgelagert, damit können später auch Parser für andere Formate entwickelt werden. Das geöffnete Analysefenster zeigt die Daten in unterschiedlichen Ansichten. Der Benutzer kann in der Standardanalyse die statistische Übersicht, den genutzten Speicher über die Zeit und die Dauer jeder einzelnen Garbage Collection anschauen. Es besteht aber auch die Möglichkeit, aus den verfügbaren Daten eigene Diagramme zu erstellen. Verschiedene Diagramme können zusammen als ein Benutzerprofil gruppiert werden. Es besteht die Möglichkeit, diese Profile zu speichern, exportieren und importieren. Somit können sie auch an andere Benutzer verteilt werden. Zur Benutzerführung stehen ein Dashboard (Übersicht vieler verfügbarer Aktionen und Hilfethemen) sowie verschiedene Hilfeseiten zur Verfügung. Die Texte (Labels) innerhalb der Software und Hilfeseiten sind grösstenteils in Deutsch und Englisch verfügbar. Sie werden abhängig von der Eclipse-Umgebungssprache geladen.

Mit dem Prototyp der Analysesoftware können die Garbage Collection Logs der JRockit virtual Machine (Release 28) hinsichtlich verschiedener Kenndaten analysiert werden. Der Anteil an manueller Arbeit kann somit verringert werden. Aktuell können für Release 27, der noch vielerorts im Einsatz ist, keine Auswertung gemacht werden. Dies wäre für den realen Einsatz wünschenswert. Zudem kann der Logger der virtual Machine noch feiner eingestellt werden, so dass er weitere Informationen liefert. Diese wären aber in einer Analyse mit dieser Software noch nicht ersichtlich.

[1] Kirk Pepperdine. Concurrent and performance reloaded.

<http://www.jfokus.se/jfokus/page.jsp?id=recordings#page=page-1>, 2011.