

Studienarbeit

Object Caching

Sitzungsprotokolle

| | | |
|---------------|------------|---------------------------------------------------|
| Projektleiter | 20.02.2012 | Lukas Hofmaier / Raphael Kohler / Timon Brüllmann |
| Betreuer | 20.02.2012 | Prof. Dr. Josef M. Joller / HSR |

Inhaltsverzeichnis

| | |
|-------------------------------------|----|
| Sitzungsprotokoll vom 23.02.12..... | 3 |
| Sitzungsprotokoll vom 08.03.12..... | 6 |
| Sitzungsprotokoll vom 15.03.12..... | 8 |
| Sitzungsprotokoll vom 05.04.12..... | 10 |
| Sitzungsprotokoll vom 19.04.12..... | 12 |
| Sitzungsprotokoll vom 03.05.12..... | 13 |
| Sitzungsprotokoll vom 23.05.12..... | 15 |

Sitzungsprotokoll vom 23.02.12

Kickoff-Meeting

Datum: 23.02.2012
Zeit: 13:30-15:30

Ort: Zürich, Technopark

Anwesend: Raphael Kohler
Lukas Hofmaier
Josef Joller
Georg Troxler
Timon Brüllmann

Entschuldigt: -
Gäste: -

Verteiler: Raphael Kohler
Lukas Hofmaier
Josef Joller
Georg Troxler
Timon Brüllmann

Grund

- Spezifizierung des Auftrages der Studienarbeit
- Diskussion der Fragen, welche sich im Verlauf des Selbststudiums und durch das Studium der Literatur ergeben haben

Themen

- Zukunft Staila / Mercury
- Use Cases
- Nächste Schritte der SA besprochen
- Aufbau der Testumgebung
- Studium RMI
- Fibonacci Problem (verteilt)
- Auswertung der Testdaten, Erstellung der Benchmarks
- Optimistic Concurrency
- Human Intervention
- Ansprechen der Objekte durch Verteilung einer ID
- Generierung der ID
- Umsetzung in Java
- Vorteile durch die Verteilung einer Objekt-ID
- Beschlüsse

- SA wird für Mercury entwickelt, aber losgelöst von der technischen Umgebung von Mercury
- SA wird nicht produktiv eingesetzt, gilt als Prototyp für eventuelle zukünftige Implementationen eines Systems, welches Objekt-Caching benötigt
- Framework für System-Tests entwickeln
- Zeitmessungen
- Instanziierung von n- Clients/Objects via Konfiguration
- Performancemessungen
- Definieren von Referenzwerten(Benchmarks) für das Vergleichen von unterschiedlichen Ansätzen

Weiteres Vorgehen

Zu Beginn unserer SA testen wir zwei Konzepte, wie man in einem verteilten System Methoden gemeinsamer Objekte ausführen kann. Direkte Veränderung der Instanzvariablen wird noch nicht berücksichtigt.

In beiden Konzepten gehen wir davon aus, dass alle Methoden auf dem Server ausgeführt werden, da Seiteneffekte auftreten können.

RMI only

Das erste Konzept bietet den Clients eine RMI Schnittstelle. Objekte in den Clients werden nicht gecached.

Wir implementieren ein einfaches RMI selbst. Dies, da wir eine eigene Version von RMI während der SA weiterhin gebrauchen können, spätestens wenn es darum geht, einen Cache einzubauen und zu managen.

Concurrency wird folgendermassen gemanaged.

Client1 führt Methode M auf Objekt O1 über RMI auf dem Server aus. Die Version des Objektes wird von v1 nach v2 inkrementiert. Clients können die Stubs der Objekte in ihrem Hauptspeicher halten. Das System benachrichtigt den Client nicht, wenn das Original des gehaltenen Objektes auf dem Server geändert wurde. Wenn nun Client2 ebenfalls die Methode M ausführt, wird auf dem Server ein Konflikt detektiert, da Client2 eine veraltete Version des Objektes im Speicher hält. Der Server merkt, dass ein Versionsunterschied vorliegt und benachrichtigt C2, dass die Methode abgebrochen wurde. (optimistic concurrency).

In diesem Fall muss sich C2 die aktuelle Version v2 von O1 beschaffen. Mit der aktuellen Version des Objektes, kann er die Methode M ausführen.

Konflikterkennung in lokalem Cache

Bei einem zweiten optimierten Konzept haben die Clients einen Objektcache.

Client1 führt Methode M auf Objekt O1 über RMI auf dem Server aus. Die Version des Objektes wird von v1 nach v2 inkrementiert. Der Server sorgt dafür, dass alle Objekte, welche durch den Methodenaufruf geändert werden und in einem Cache abgespeichert

wurden, im lokalen Cache der Clients aktualisiert werden (Es ist mir noch nicht klar, ob die Objekte bloss invalidiert werden oder komplett aktualisiert werden sollen). Die Versionsnummer aller betroffenen Objekte wird inkrementiert. Der Client2 führt ebenfalls Methode M aus. Er hat aber noch die alte Version v1 davon in seinem Speicher. Da im Cache bereits eine neuere Version liegt, wird der Konflikt bereits beim Client festgestellt. Es ist kein Netzwerkaufruf nötig.

- Vorteil: Konflikt wird beim Client festgestellt
- Nachteil: Server muss alle Clients updaten.

Fragen:

- Das Objekt O1 könnte mit einem Callback direkt verändert werden. Die Chachelibrary benachrichtigt die Clientapplikation über Callback und verändert das Objekt direkt im Speicher. So würde es nie zu einem Konflikt kommen. Was spricht gegen dieses Vorgehen?

Sitzungsprotokoll vom 08.03.12

Datum: 08.03.2012

Zeit: 14:10-15:30

Ort: HSR, ITA

Anwesend: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Entschuldigt: -

Gäste: -

Verteiler: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Vorbereitete Fragen

- Was passiert, wenn sich ein Client ein Account Object bei RMI beschafft und dann `b.setBalance(lokaleVariable * 1.1)` ausführt. Muss dieser Aufruf auch mit Concurrency Control geprüft werden oder nicht. Der Client hat in diesem Fall nie Instanzvariablen gelesen.
- Anzahl der Objekte werden vom Framework generiert. Das Framework bietet eine Schnittstelle, die es ermöglicht dem System eine Collection von Objekten zu übergeben. Die Typen sind nicht generisch sondern hard-codiert. Ist das in Ordnung?

Themen

- Frage 2: Hard codierte Typen sind in Ordnung.
- Mitgeteilt wann wir an der SA arbeiten: Mittwoch & Donnerstag
- Object Caching
 - writes auf dem Server ausführen
 - reads auf dem clientseitig gecachten Objekt

- zwei Hashmaps für die Konflikterkennung, siehe [[RMI-only-Lösungsansätze]]
 - feinere Granularität kann später eingebaut werden
 - Atomic Package in Java anschauen

Beschlüsse

- Keine Zeiterfassung nötig, ausser für uns selbst.
- Dokumentation sollte vorallem die Entscheidungen während des Projekts dokumentieren.
- SA Projekt Proposal erstellen
- Ein Szenario spezifizieren und diese in das Proposal aufnehmen -> Konflikterkennung ist wichtig
- Hardcodierte Objekttypen sind in Ordnung

Sitzungsprotokoll vom 15.03.12

Datum: 15.03.2012

Zeit: 15:00-16:00

Ort: HSR, ITA

Anwesend: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Entschuldigt: -

Gäste: -

Verteiler: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Vorbereitete Fragen

- Unser RMI unterstützt das Speichern von mehreren Objekten. Wir übertragen Collections von Objekten damit wir für spätere Szenarien mit mehreren Objekten flexibler sind. Die Implementierung von mehreren Objekten ist aber auch aufwändiger.
- Wie sollen die Resultate der Messungen ausgegeben werden? In ein XML-File gespeichert und zusätzlich auf dem Monitor angezeigt werden? Evtl. einen Reader für die XML-Files erstellen, welcher die gewünschten Daten gleich darstellt?
- Wie genau sollen wir messen? Machen wir viele Requests(zB. 1000000), damit wir einen ,möglichst genauen Durchschnittswert bekommen? Mit System.nanoTime()?

Themen

- Frage 1: Herr Joller unterstützt unsere Meinung in diesem Entscheid, trotz Mehraufwand.
- Frage 2: Es ist uns überlassen wie wir die Daten speichern. (txt, csv, xml)
- Frage 3: nanoTime() ist die richtige Wahl
- Um bei allen Testläufen die gleiche JVM Umgebung zu garantieren, könnte es notwendig werden dass die Clients nach jedem Test runtergefahren werden und beim nächsten Test neu gestartet werden.

- Weitere Möglichkeit Szenario Reihenfolge variieren.

Beschlüsse

- Projekt Proposal überarbeiten
 - Logo einbauen
 - Das Proposal sollte für Aussenstehende verständlich sein
 - Korrekturen machen
- Mitte Semester schauen wir, wie weit wir mit Caching gehen

Sitzungsprotokoll vom 05.04.12

Datum: 05.04.2012

Zeit: 16:00

Ort: HSR, Semesterarbeitszimmer

Anwesend: Josef Joller
Raphael Kohler
Lukas Hofmaier

Entschuldigt: Timon Brüllmann (Grund: Krankheit)

Gäste: -

Verteiler: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Vorbereitete Fragen

- -

Themen

- Welche Dokumente werden gefordert? Erfahrungsbericht.
- Project Proposal: Was ist gemeint mit "Das feingranulare Lockung wird protokolliert."
- Wir wollen die einzelnen Klassen in der Doku beschreiben. Ist das i.o? → auf package ebene ist das ok.
- Reicht es wenn die Protokolle im Wiki erfassen sind, oder sollen wir sie als pdf an Sie(Joller) schicken?
- Können wir das Projekt zu einer BA weiterziehen? - ja das wäre interessant. man müsste die ziele konkretisieren.
- Wir wollen noch Object Nachhing machen? Sinnvoll? Remote Write
- Die bisherigen Resultate sind ok.
- wie weiter? das Paper josh bloch beschreibt einen interessanten Ansatz.
- wir arbeiten weiter in Richtung remote-writing Protokoll gemäss Tannenbaum.
- später können wir noch lokale writes zulassen.
- es wird keine Zusammenfassung erwartet. es soll ein Erfahrungsbericht sein. was haben wir gemacht. weshalb?
- Wir können das Raster kontrollieren lassen, ob es ok ist.

- Man muss sehen dass es kein Zufallsergebnis ist. es sollen Resultate aus mehreren Messungen sein. muss in Testbericht ersichtlich sein, Wie viele Messungen durchgelaufen sind.
- was wollen wir verbessern. Zugriffszeit.

Beschlüsse

- -

Sitzungsprotokoll vom 19.04.12

Datum: 19.04.2012

Zeit: 15:00

Ort: HSR, ITA

Anwesend: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Entschuldigt: -

Gäste: -

Verteiler: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Vorbereitete Fragen

- -

Themen

- Unser Caching Konzept aufzeigen
 - Dabei kam das Problem mit der Sicherstellung der Konsistenz auf.

Beschlüsse

- Weiter arbeiten wie bisher.

Sitzungsprotokoll vom 03.05.12

Datum: 03.05.2012

Zeit: 14:00

Ort: HSR, ITA

Anwesend: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Entschuldigt: -

Gäste: -

Verteiler: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Vorbereitete Fragen

- Was für Szenarien erachten Sie als sinnvoll?

Themen

- Software Entwicklung soweit abgeschlossen. Alle Bugs gefixed. :-)
- Aus den bisherigen Tests ist ersichtlich das die Cache- Variante eine bessere Performance bietet.
- Messungsergebnisse / Auswertung
 - Die Resultate sollten Aufschluss geben welches System (Cache vs. RMIOOnly) für welchen Fall am besten geeignet ist. (Immer in Rolle des Leser schlüpfen)
 - Vorhersagen festhalten und mit den Resultaten vergleichen
 - Mittelwerte für die Auswertung nehmen, bei grossen Schwankungen mehrere Durchläufe (3-10 Runs) machen
 - Resultate auch grafisch darstellen (mit Excel I.O.)
 - Können keine Aussagen gemacht werden, weg lassen

Beschlüsse

- 3 Szenarien
 - 1. Szenario: Nur Lesen
 - 2. Szenario: Alle Clients machen Increment() (lesen/schreiben)
 - 3. Szenario: 1 Client macht Increment(), die restlichen lesen nur den Kontostand.
- Aufzeigen wie das System skaliert, also von 2-8 Clients
 - Performancekurve je nach Clientanzahl

Sitzungsprotokoll vom 23.05.12

Datum: 23.05.2012

Zeit: 09:10

Ort: HSR, ITA

Anwesend: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Entschuldigt: -

Gäste: -

Verteiler: Josef Joller
Raphael Kohler
Lukas Hofmaier
Timon Brüllmann

Vorbereitete Fragen

- Poster: Müssen wir das wirklich machen?
- Erfahrungsberichte wirklich nötig?
- Ausblick wirklich nötig?
- Archivierung im Sekretariat, einfach Einleitung abgeben?

Themen

- Feedback von Herr Joller zur SA

Beschlüsse

- Poster erstellen und bis am 29.05.12 an Herr Joller schicken
- Abstract schreiben (Abstract von Heinzmann als Vorlage anschauen)
- Management Summary verfassen
- Persönlicher Bericht (max 1/2 Seite pro Teammitglied)
 - Teambildung
 - Motivation
 - Eigentliche Arbeit
- Projekt Management Dokumente zusammen tragen
- All diese Dokumente können einzeln abgeben werden

- Bis am 25.05.12 neue Version verschicken