

VSDB

Loadbalancing

Backhausen Dominik Dimitrijevic Daniel

6. Dezember 2013

5AHITT

Inhaltsverzeichnis

1 Aufgabenstellung	3
2 Designüberlegungen	4
3 Arbeitsaufteilung	6
4 Arbeitsdurchführung	7
5 Testbericht	8
6 Quellen	9

1 Aufgabenstellung

Aufgabenstellung

Es soll ein Load Balancer mit mindestens 2 unterschiedlichen Load-Balancing Methoden (jeweils 7 Punkte) implementiert werden (ähnlich dem PI Beispiel [1]; Lösung zum Teil veraltet [2]). Eine Kombination von mehreren Methoden ist möglich. Die Berechnung bzw. das Service ist frei wählbar!

Folgende Load Balancing Methoden stehen zur Auswahl:

Least Connection Weighted Distribution Response Time Server Probes

Um die Komplexität zu steigern, soll zusätzlich eine SSession Persistence"(2 Punkte) implementiert werden. Tests

Die Tests sollen so aufgebaut sein, dass in der Gruppe jedes Mitglied mehrere Server fahren und ein Gruppenmitglied mehrere Anfragen an den Load Balancer stellen.

Modalitäten

Gruppenarbeit: 2 Personen Abgabe: Protokoll, Testszenarien, Sourcecode (mit allen notwendigen Bibliotheken), Java-Doc, Jar

Viel Erfolg!

2 Designüberlegungen

Wir haben ein Interface Calculater welches die pi Methode definiert. Nun Implemeniteren wir dies Interface einmal als Server(ClaculatorImpl) und als Balancer somit rufen wir mithilfe von RMI die Pi mehtode von dem Balancer auf welcher sich den Optimalen Server mithilfe der Balancing Methode sucht und die Anfrag somit an diesen Server weiterleitet. Ebenso haben wir ein Main interface welches nur dafür da ist um den User eingaben zu ermöglichen.

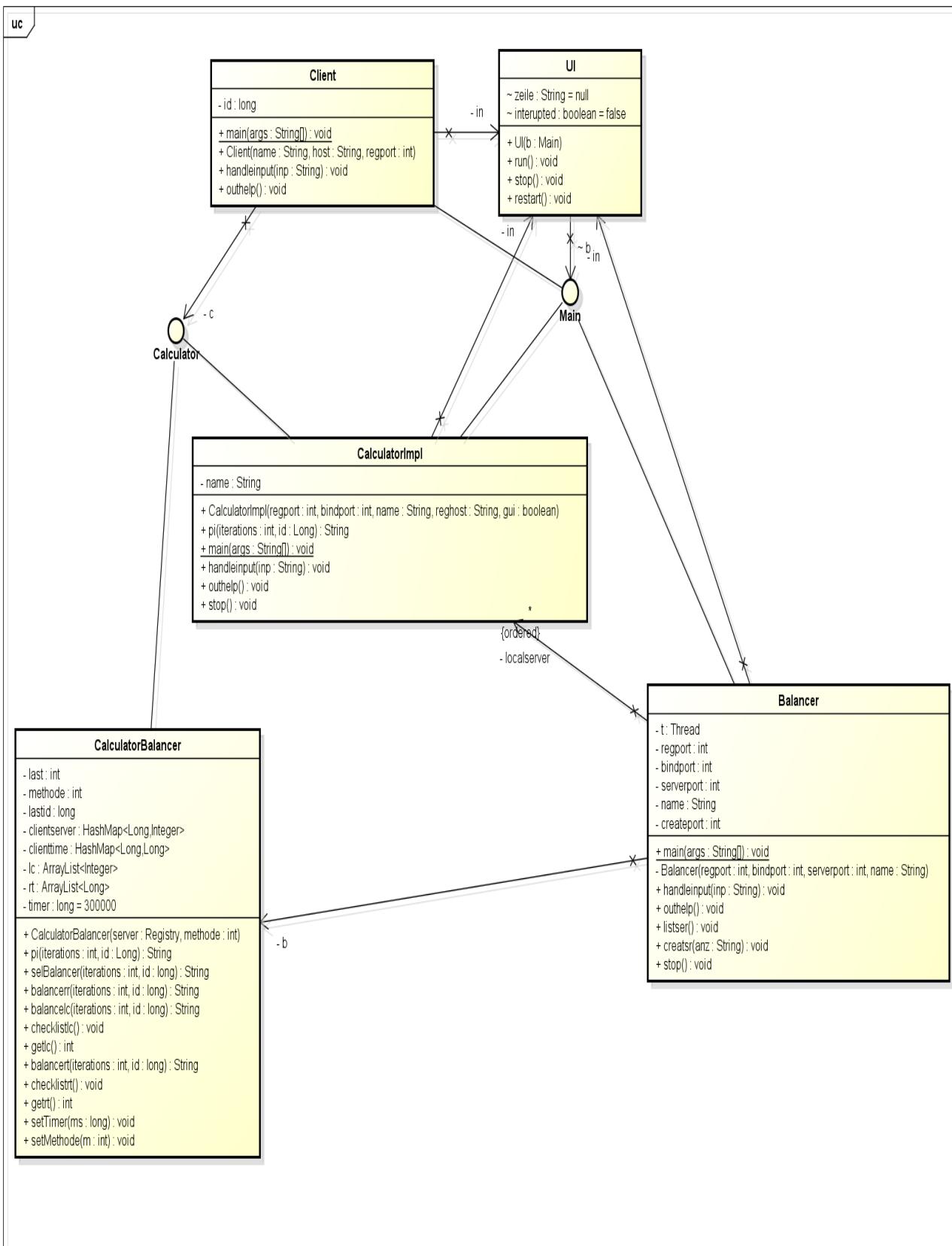


Abbildung 2.1:

3 Arbeitsaufteilung

Name	Arbeitssegment	Time Estimated	Time Spent
Backhausen	Verbindung erstellen und Grundprogramm schreiben	5h	3h
Backhausen	Balancing Methoden schreiben	4h	4h
Backhausen	Balancer schreiben	4h	3h
Backhausen	User Interface	2h	1.5h
Dimitrijevic	Balancer schreiben	4h	2h
Dimitrijevic	Protokoll schreiben	2h	2h
Dimitrijevic	Verbindgun erstellen und Grundprogramm schreiben	4h	4h
Gesamt		25h	19.5h

4 Arbeitsdurchführung

Wir haben damit begonnen uns die uns zur Verfügung gestellten Quellen anzuschauen. Mit Hilfe der Quellen und einem Beispiel aus dem Letzten Jahr haben wir ein Grundprogramm geschrieben um die Verbindung zu testen. Dabei sind wir darauf gekommen das wir die Java Policy nicht überall richtig gesetzt haben was wir auch sofort danach behoben haben. Nun haben wir uns überlegt welche Methoden wir implementieren wollen. Wir sind überein gekommen das wir Least Connection und Response Time nehmen werden.

Nachdem die Festgelegt wurde haben wir dies auch schon in dem Balancer implementiert. Somit haben wir nun 3 Balancing Methoden Round Robin, Least Connection und Response Time. Danach haben wir uns überlegt da man vielleicht auch zur Laufzeit bestimmte Veränderungen vornehmen will und nicht alles über die Startzeit parameter definieren möchte.

Somit haben wir die Startzeitparameter auf ein Minimum reduziert und für alle weiteren Definitionen haben wir Befehle über ein UserInterface bereitgestellt. Somit habe wir die möglichkeit implementiert zur Laufzeit neue virtuelle Server am Balancer zu erstellen. Wir haben ebenfalls implementiert das ein externer Server sich automatisch bei dem Balancer registriert und somit auch in die Serverauswahl kommt. Ebenso kann man diese externen Server auch ohne Probleme wieder beenden ohne den LoadBalancer zu beeinflussen.

Wir haben dann ebenso die Session Persistance mit hilfe des Delayed Bindung gelöst und vergeben dem User somit IDs. Hierbei war es allerdings nötig den Standard Client zu erweitern um mit diesen Ids auch umgehen zu können.

5 Testbericht

Wir haben alle 3 Load Balancing Methoden mit mindestens 6 Servern getestet. Da bei der Session Persistance allerdings der Timer standardmäßig auf 5 min gesetzt ist wurde das esten dieer Funktion mit der Zeit sehr Zeitintensive deswegen haben wir eine zusätzliche möglichkeit implementiert um diesen Timer individuel anzupassen um auch diese Funktion ohne große Probleme zu Testen.

6 Quellen

- [1] "Praktische Arbeit 2 zur Vorlesung 'Verteilte Systeme' ETH Zürich, SS 2002", Prof.Dr.B.Plattner, übernommen von Prof.Dr.F.Mattern (<http://www.tik.ee.ethz.ch/tik/education/lectures/VS/SS02/Praktikum/loesung2.zip>)
- [2] <http://www.tik.ee.ethz.ch/education/lectures/VS/SS02/Praktikum/loesung2.zip>