JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll
Load Balancing

Systemtechnik Labor
5BHIT 2015/16, Gruppe A

Hagen Aad Fock & Stefan Polydor

Version 0.1

Betreuer: Prof. Micheler
Note:

Begonnen am 20.2.2016
Beendet am 4.03.2016

Inhaltsverzeichnis

Einführung

Dieses Protokoll-Template soll helfen den Laborübungsteil entsprechend dokumentieren zu können.

Ziele

Hier werden die zu erwerbenden Kompetenzen und deren Deskriptoren beschrieben. Diese werden von den unterweisenden Lehrkräften vorgestellt.

Dies kann natürlich auch durch eine Aufzählung erfolgen.

Voraussetzungen

Welche Informationen sind notwendig um die Laborübung reibungslos durchführen zu können? Hier werden alle Requirements der Lehrkraft detailliert beschrieben und mit Quellen untermauert.

Hier zum Beispiel die Architektur der Common Object-Request-Broker Architecture:

Aufgabenstellung

Es soll ein Load Balancer mit mindestens 2 unterschiedlichen Load-Balancing Methoden (jeweils 6 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:

- Weighted Distribution
- Least Connection
- Response Time
- Server Probes

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

Vertiefend soll eine Open-Source Applikation aus folgender Liste ausgewählt und installiert werden. (2 Punkte)
https://www.inlab.de/articles/free-and-open-source-load-balancing-software-and-projects.html

Auslastung

Es sollen die einzelnen Server-Instanzen in folgenden Punkten belastet (Memory, CPU Cycles) werden können.
Bedenken Sie dabei, dass die einzelnen Load Balancing Methoden unterschiedlich auf diese Auslastung reagieren werden. Dokumentieren Sie dabei aufkommenden Probleme ausführlich.

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. Für die Abnahme wird empfohlen, dass jeder Server eine Ausgabe mit entsprechenden Informationen ausgibt, damit die Verteilung der Anfragen demonstriert werden kann.

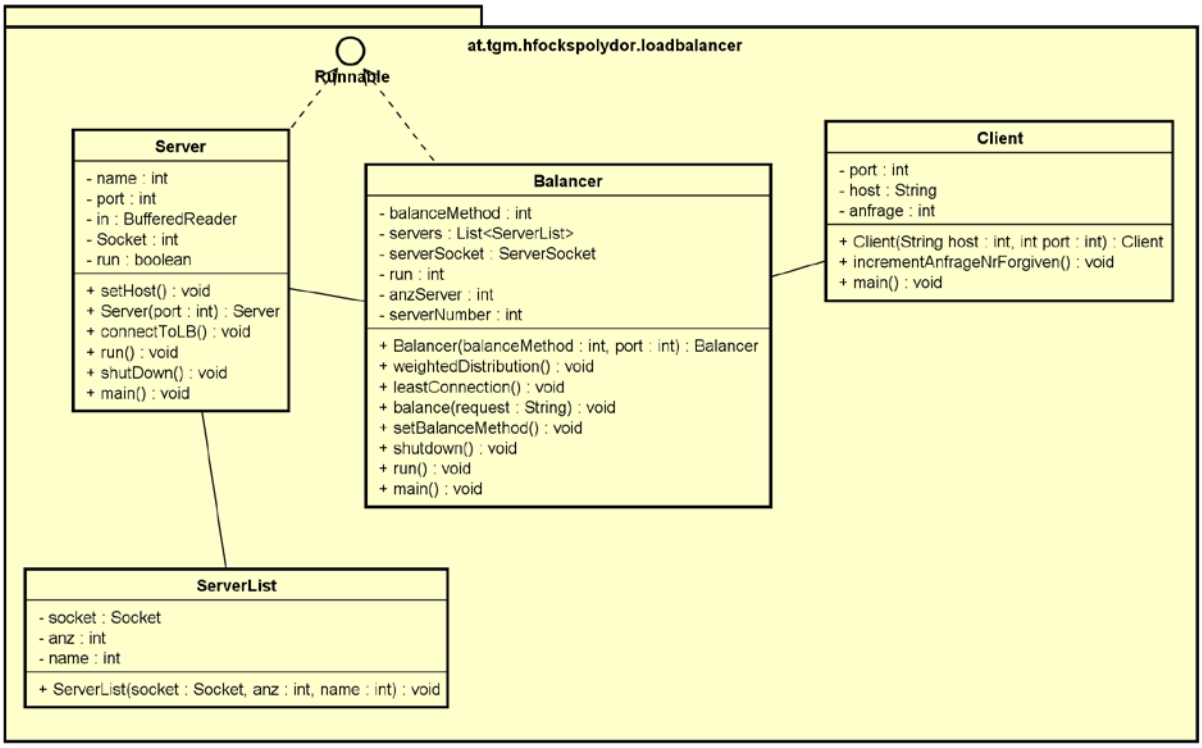
Modalitäten

Gruppenarbeit: 2 Personen
Abgabe: Protokoll mit Designüberlegungen / Umsetzung / Testszenarien, Sourcecode (mit allen notwendigen Bibliotheken), Java-Doc, Build-Management-Tool (ant oder maven), Gepackt als ausführbares JAR

Github

Der Code zu dieser Übung wurde unter Github veröffentlicht. Unter dem folgenden Link kann das Repository gefunden werden.
https://github.com/spolydor-tgm/DezSys10---Load-Balancing.git

Klassendiagramm



Ergebnisse

Bei dieser Aufgabe wurde das Hauptmerkmal auf den Load Balancer und dessen Algorithmen gelegt. Umgesetzt wurde die Aufgabe in Java. Für die Implementierung der Verbindungen zwischen Client, Server und Load Balancer wurden Sockets verwendet. Wie vorgegeben wurden zwei Algorithmen implementiert. Der Weigthed Distribution und der Least Connection Algorithmus.

Weigthed Distribution

Da verschiedene Server hardwaretechnisch verschieden ausgestattet sein können, kann mithilfe dieser Methode eine Angabe der Leistung der einzelnen Server in Relation zueinander erfolgen. Dies geschieht, indem jedem Server eine gewisse Gewichtung zugewiesen wird. Der Load Balancer, kann so beispielsweise bei einer durch einen Serveradministrator festgelegte Gewichtung von 5-1 dem 1. Server 5-mal so viele Anfragen zukommen lassen, wie dem 2. [3]

Least Connection

Jede neue Anfrage wird dem Server mit den geringsten gleichzeitig aktiv vorhandenen Verbindungen zugesandt. Der Load Balancer muss hierbei die Anzahl dieser Verbindungen jedes Servers jederzeit festhalten. Diese Methode ist eine der effektivsten und beliebtesten in verschiedenen Anwendungsbereichen, wie beispielsweise DNS oder dem Web. Der Hauptgrund hierfür ist, das einfache Verstehen und Anwenden der Methode. [3]

Probleme

- Vergabe der Server ID, damit in der Console festgestellt werden kann, wohin die Anfrage weitergeleitet wird
- Verbindung mit Sockets, da wir beim InputStreamReader eine „Endlosschleife“ erzeugt haben. Der Abschnitt hat immer gewartet, bis die nächste Nachricht ankommt, nachdem eine eingetroffen ist
- Kleine Gedankenfehler bei der Implementierung der BalanceMethoden

Lessons Learned

- Bei der Schleife für den InputStreamReader, die Variable die auf != null verglichen wird, nach ankommen einer Nachricht auf null setzen, damit diese beim warten auf die Nächste Nachricht nicht als angekommen gesehen wird, und somit dieser Step einfach übersprungen wurde.
- Den Umgang mit Sockets wieder aufgefrischt
- Funktionsweise zweier Balancingmethoden anhand eines selbst implementierten Beispiels

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/aufgabe2.pdf)

[2] Ähnlich PI – Beispiel, Lösung veraltet, http://www.tik.ee.ethz.ch/education/lectures/VS/SS02/Praktikum/loesung

[3] Chandra Kopparpu. Load Balancing Servers, Firewalls and Chaches. Willey 2002 (Quelle von Load Balancing Ausarbeitung, Thomas Taschner)