

---

# **Protokoll**

## **Loadbalancer**

**Dezsys**  
**5BHITT 2015/16**

**Niklas Hohenwarter, Selina Brinnich**

**Note:**  
**Betreuer: Thomas Micheler**

**Version 1.0**  
**Begonnen am 26. Februar 2016**  
**Beendet am xx. Februar 2016**

## Inhaltsverzeichnis

1	Einführung.....	3
1.1	Aufgabenstellung .....	3
1.2	Auslastung .....	3
1.3	Tests .....	3
2	Ergebnisse.....	4
2.1	Loadbalancing Software .....	4
2.2	Loadbalancer.....	5
2.3	Server.....	5
2.4	Client .....	5
3	Zeitaufzeichnung .....	6
4	Quellen .....	6
5	Abbildungsverzeichnis .....	6

# 1 Einführung

## 1.1 Aufgabenstellung

Es soll ein Load Balancer mit mindestens 2 unterschiedlichen Load-Balancing Methoden (jeweils 6 Punkte) implementiert werden. 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" implementiert werden.

Vertiefend soll eine Open-Source Applikation aus folgender Liste ausgewählt und installiert werden.

<https://www.inlab.de/articles/free-and-open-source-load-balancing-software-and-projects.html>

## 1.2 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 auftretenden Probleme ausführlich.

## 1.3 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.

## 2 Ergebnisse

### 2.1 Loadbalancing Software

Als Software zum Loadbalancing bzw. für die Webserver wurde nginx gewählt. In diesem Beispiel werden 3 Server verwendet (1 LB, 2 Webserver). Zu Beginn muss nginx auf allen Servern installiert werden.

```
apt-get install nginx
```

Um dannach beim Zugreifen den Unterschied zwischen den beiden Servern zu merken wird ein einfaches index.html File im Webroot erstellt. Auf Server 1 enthält das File

Hallo, ich bin Server 1!

Und auf Server 2

Hallo, ich bin Server 2!

Dazu werden diese Files in /usr/share/nginx/html erstellt. Ruft man nun die IP des Servers auf erhält man folgende Antwort:



*Abbildung 1: Antwort Server1*

Die gleiche Konfiguration wird nun auch für Server 2 vorgenommen. Dannach muss der Loadbalancer konfiguriert werden[1]. Dazu wird in die default site config geändert. Es muss folgendes geändert werden:

```
upstream backend {
    server 10.0.106.210;
    server 10.0.106.220;
}
server {
    location / {
        proxy_pass http://backend;
    }
}
```

Dannach muss Nginx mittels `service nginx restart` neu gestartet werden. Dannach funktioniert das Loadbalancing automatisch mit dem Algorithmus Round Robin. Durch das Aufrufen der IP Adresse des Loadbalancers und mehrmaliges Refreshen wechselt der Text immer von „Hallo, ich bin Server 1!“ und „Hallo, ich bin Server 2!“ hin und her.

## 2.2 Loadbalancer

Als zu implementierende Algorithmen wurden Weighted Distribution und Least Connection gewählt.

## 2.3 Server

## 2.4 Client

Der Client soll mittels Sockets mit dem Server kommunizieren. Um die Anfragen und Antworten besser darstellen zu können wird eine GUI implementiert. Die GUI soll wie folgt aussehen:

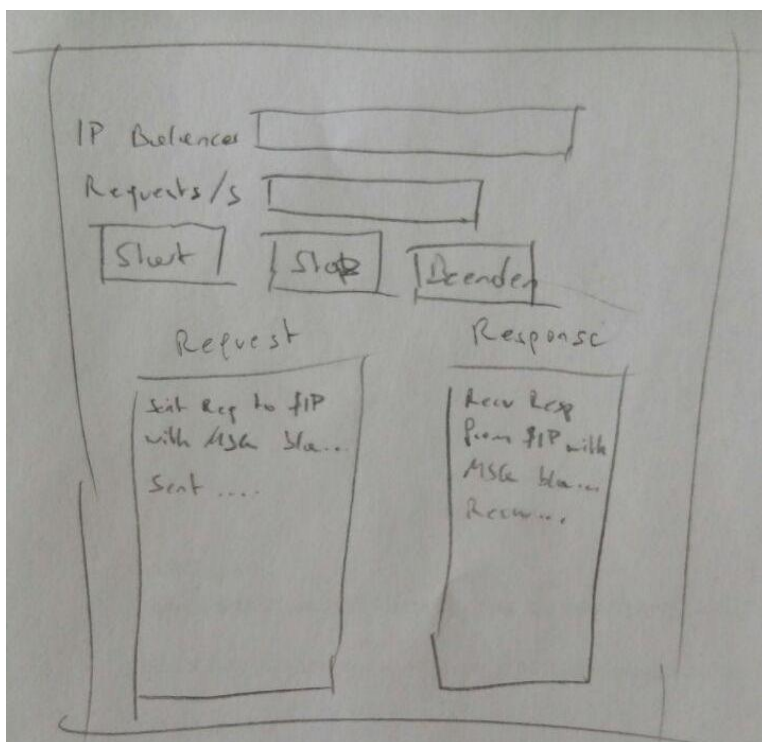


Abbildung 2: Skizze Client GUI

Die Kommunikation zum Loadbalancer wird mittels Java Sockets realisiert. Der Client nimmt die eingegebene IP aus der GUI und erstellt damit einen Socket auf dem Port 4444. Dannach fängt er an folgende Nachrichten an den Loadbalancer zu schicken:

```
[${NACHRICHTENID}] ${CLIENTNAME}
```

Davon werden so viele Nachrichten/s abgeschickt wie in der GUI angegeben. Dabei erhöht sich bei jeder Nachricht die Nachrichtenid um 1. Genauere Dokumentation des Codes ist dem Javadoc zu entnehmen.

### 3 Zeitaufzeichnung

Datum	Dauer	Name	Beschreibung
<b>12.02.2016</b>	2h	Hohenwarter	Protokoll erstellt Loadbalancing mit nginx Skizze Client Dokumentation Client
<b>16.02.2016</b>	2h	Brinnich	Java Client mit GUI & Sockets
<b>20.02.2016</b>	3h		
<b>26.02.2016</b>	2h		

### 4 Quellen

[1] Etel Sverdlov, How To Set Up Nginx Load Balancing,  
<https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-load-balancing>, zuletzt besucht: 26.02.16

### 5 Abbildungsverzeichnis

Abbildung 1: Antwort Server1 .....	4
Abbildung 2: Skizze Client GUI .....	5