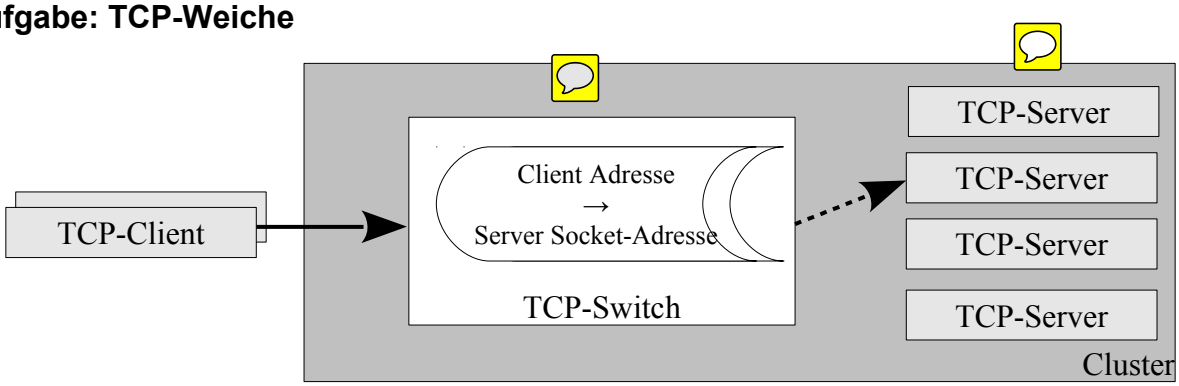
Verteilte Systeme

Übung C1 - Einzelarbeit

Bearbeitungszeit 2 Wochen

Die Übung adressiert den Umgang mit TCP und die Verteilung von Protokoll-Anfragen in Clustern. Weiterführende Informationen zu den TCP- und HTTP-Protokollen gibt es unter http://en.wikipedia.org/wiki/Tcp_protocol, sowie unter http://en.wikipedia.org/wiki/Http_protocol.

Aufgabe: TCP-Weiche



Kopiert die Klasse *TcpSwitchSkeleton* nach *TcpSwitch*. Diese Klasse repräsentiert den Rahmen für einen TCP-Server im Acceptor/Service Multithreading-Entwurfsmuster. Implementiert die run()-Methode des Connection-Handlers unter Beachtung der dort unter TODO hinterlegten Hinweise.

Vor starten der TCP-Weiche müssen zuerst mindestens zwei Web-Server (z.B. zwei *de.htw.ds.http.HttpServer2*, oder zwei Apache-Instanzen) mit identischem Inhalt (ein Webserver-Cluster) auf den Ports 8002 und 8004 gestartet werden. Verwendet die SelfHtml-Seiten für den Inhalt der Server-Instanzen, wie in der Übung.

Startet Euren TCP-Switch dann mit den folgenden Parametern:

- 8010 (Service-Port)
- true/false (Session-Awareness)
- localhost:8002 (IP-Socketadresse des Zielservers 1)
- localhost:8004 (IP-Socketadresse des Zielservers 2)

Bei session-awareness=false sollte die TCP-Weiche alle ankommenden TCP-Anfragen zufällig auf die Zielserver verteilen: Im Falle von HTTP-Servern kommt die HTML-Seite von einem der Zielserver, das CSS-File vom anderen, usw. Verwendet dazu den unter der statischen Variable „RANDOMIZER“ zur Verfügung stehenden Zufallszahlengenerator, sowie die in der Instanzvariable „nodeAddresses“ gespeicherten Socket-Adressen (IP-Adresse & Port) der Zielserver.

Bei session-awareness=true dagegen soll nur bei der ersten Anfrage eines Clients einer der nachgeschalteten Zielserver zufällig ausgewählt, und diese Auswahl dann für alle nachfolgenden Anfragen desselben Clients wiederverwendet werden. Dazu sollt ihr mittels `socket.getInetAddress()` die IP-Adresse des Clients abfragen, und beim ersten Zugriff dieses Clients in der Instanzvariable „sessions“ (eine Instanz der Collection-Klasse *java.util.HashMap*) die Socket-Adresse eines zufällig ausgewählten Zielservers unter dieser Client-Adresse hinterlegen. Dadurch könnt ihr auch herausbekommen ob sich ein Client schon einmal gemeldet hat, denn falls die Instanzvariable „sessions“ keinen Eintrag für die IP-Adresse des Clients beinhaltet, dann handelt es sich um dessen ersten Zugriff!

Beachtet dass das Cluster-Konzept nur funktioniert wenn die Zielserver dieselben Ressourcen unter denselben URL-Pfäden zur Verfügung stellen, achtet also darauf dass dies zutrifft!