

Verteilte Systeme

Übung D2

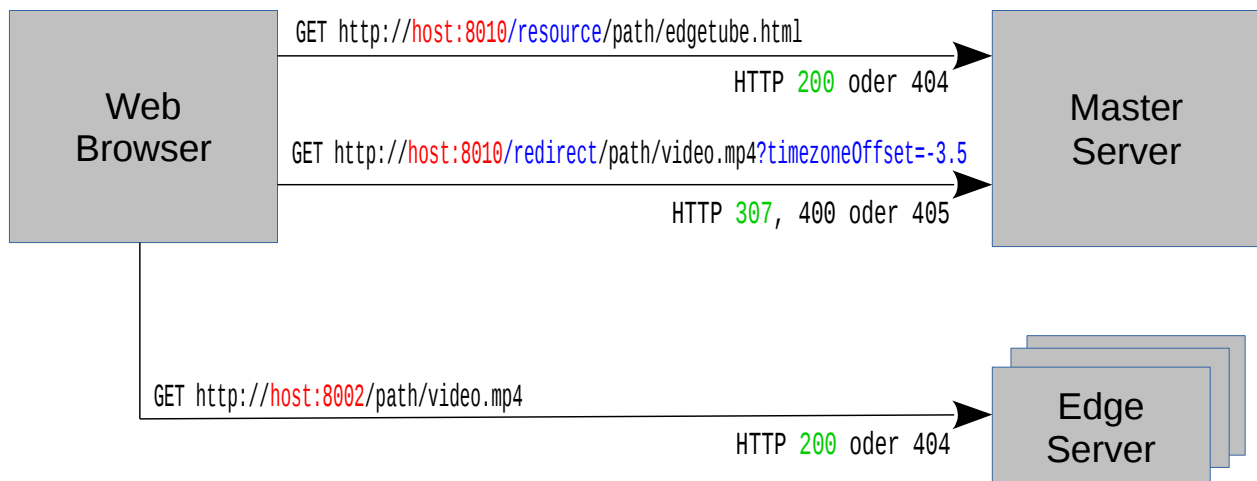
Bearbeitungszeit 2 Wochen

Die Übungen adressieren Techniken zum Multi-Programming mittels HTTP-Redirect und zu Edge-Servern.

Aufgabe: Edge-Server

Es soll ein Edge-Server System erstellt werden. Die Idee dabei ist dass Anfragen gegen Medien zwar an einen zentralen Medienserver geschickt, aber für jeweils einem Abschnitt von Client-Zeitzone transparent von einem dafür konfigurierten und weltweit verteilten Replikat-Knoten (die Edge-Server) beantwortet werden sollen. Dies löst u.a. das „Last-Mile“-Problem, i.e. große Videodateien können transparent relativ nah beim Web-Browser platziert werden, müssen daher nur noch eine kleine Strecke im Internet übertragen werden, und verstopfen so letzteres nicht.

Eine der am einfachsten umsetzbaren Realisierungen basiert dabei auf Zeitzone: Auf einem speziell präparierten Master-Server liegt eine HTML-Seite mit der sich Videos abspielen lassen. Jeder Link dieser Seite auf eines der verfügbaren Videos wird dabei mittels JavaScript mit einem versteckten Query-Parameter „timezoneOffset“ (der Offset des Web-Clients gegenüber GMT in Stunden) erweitert. Wird nun eines der Videos ausgewählt, dann antwortet der Master-Server nicht mit dem Video-Inhalt selbst, sondern mit einer HTTP-Redirect (Code 307) Response die den Web-Browser anweist den Video-Inhalt automatisch in einem für den Benutzer transparenten zweiten HTTP Request von einem anderen Web-Server (dem ausgewählten Edge-Server) zu laden.



Aufsetzen des Edge-Server Netzwerks

Die Konfiguration der Abbildung von Zeitzonen auf zur Verfügung stehende Replikat-Knoten (die Edge-Server) findet sich in der Datei `edgetube.properties`. Normalerweise würden sich dort weltweit verteilte Maschinen referenziert, doch zum Testen behelfen wir uns mit drei Web-Servern mit im Java-Package eingebetteten Inhalten (Klasse **HttpEdgeServer**) welche auf den Ports 8001, 8002 und 8003 zu starten sind.

Die Klasse **HttpMasterServer** (der Master-Server des Netzwerks) ist dagegen einmalig auf Port 8010 zu starten. Die obige Konfigurationsdatei wird automatisch von der Fassade **HttpRedirectHandlers** ausgewertet, und dieser Server nutzt (wenn komplett implementiert) diese Information um die Abbildung von Zeitzonen auf Edge-Server zu bewerkstelligen.

Die im Java-Package eingebettete „EdgeTube“-Demoseite kann in einem Web-Browser mittels der URL <http://hostname:8010/resource/de/htw/ds/edge/edgetube.html> vom Master-Server abgefragt werden. Alle URL-Pfade welche mit „/resource/“ beginnen sollen dabei vom Master-Server selbst beantwortet werden, während alle Pfade welche mit „/redirect/“ beginnen in einem HTTP-Redirect resultieren sollen.

Implementierung

Benennt die Klasse **HttpRedirectHandlerSkeleton** nach **HttpRedirectHandler** um, und erweitert diese an den dort unter TODO vermerkten Stellen; die Klasse **HttpRedirectHandlers** stellt dazu nützliche Methoden bereit; zudem dient der Quellcode der in der `java-util` Bibliothek enthaltenen Klasse **HttpModuleHandler** als Beispiel für den Umgang mit **HttpExchange**-Objekten.

Für einen erfolgreichen HTTP-Redirect müsst ihr das Response-Header Feld „Location“ mit einer aus dem Zeitzonen-Offset und der Request-URI ermittelten Redirect-URI belegen, und dann Code 307 zurück liefern. Code 405 soll dagegen zurück gegeben werden falls die Anfrage nicht die HTTP-Zugriffsmethode GET verwendet, und 400 falls sich aus den Ausgangsdaten keine syntaktisch korrekte Redirect-URI bilden lässt. Ihr könnt Eure Implementierung testen indem ihr

- zuerst die drei Edge-Server und den Master-Server startet
- dann den Netzwerk-Monitor Eures Web-Browsers aktiviert, z.B. in *Firefox* mittels der Tastenkombination `Ctrl-Shift-Q`, oder in *Chrome* mittels `Ctrl-Shift-I` und nachfolgender Auswahl des Network-Tabs
- dann in der Navigationszeile des Browsers die oben genannte URL eingibt
- und schließlich auf der geladenen Seite eines der Videos auswählt und überprüft ob der HTTP-Redirect korrekt erfolgt

Beachtet dass mit dieser Technik statt des „src“-Attributs eines Video-Elements auch Verweise aus anderen HTML-Elementen leicht mit Zeitzonen-Information angereichert werden können. Einzig das Anker-Element (`<a>`) bildet eine Ausnahme, da es JavaScript aus Sicherheitsgründen nicht gestattet ist das „href“-Attribut eines Ankers nach dem Laden einer Seite noch zu verändern. Genauso ist diese Vorgehensweise natürlich dazu geeignet eine sehr effiziente session-basierte Weiche (siehe Aufgabe **C1**) für einen HTTP Container-Cluster (z.B. für REST-basierte Web-Services) zu realisieren. Diese ist dann allerdings im Gegensatz zum dort realisierten Design nur für HTTP (und nicht mehr für TCP generell) geeignet.