

Verteilte Systeme

Übung C2

Bearbeitungszeit 2 Wochen

Die Übung adressiert den Umgang mit Web-Servern, Multi-Programming mittels Verteilung von HTTP/HTTPS-Anfragen in Clustern mittels HTTP-Redirect, und Edge-Servern. Weiterführende Informationen zum HTTP-Protokoll gibt es unter https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol.

Aufgabe A: Redirect-Server

Im Falle eines Clusters von Web-Servern gibt es eine wichtige Alternative zur TCP-Weiche aus Übung C1. Das HTTP-Protokoll definiert einige Status-Codes (3xx) für Anfrage-Umleitungen, welche Web-Browser dazu veranlassen die angefragte Ressource automatisch aus einer Redirect-Location zu laden. Der wichtigste Status-Code für diese Zwecke ist **307** Temporary Redirect.

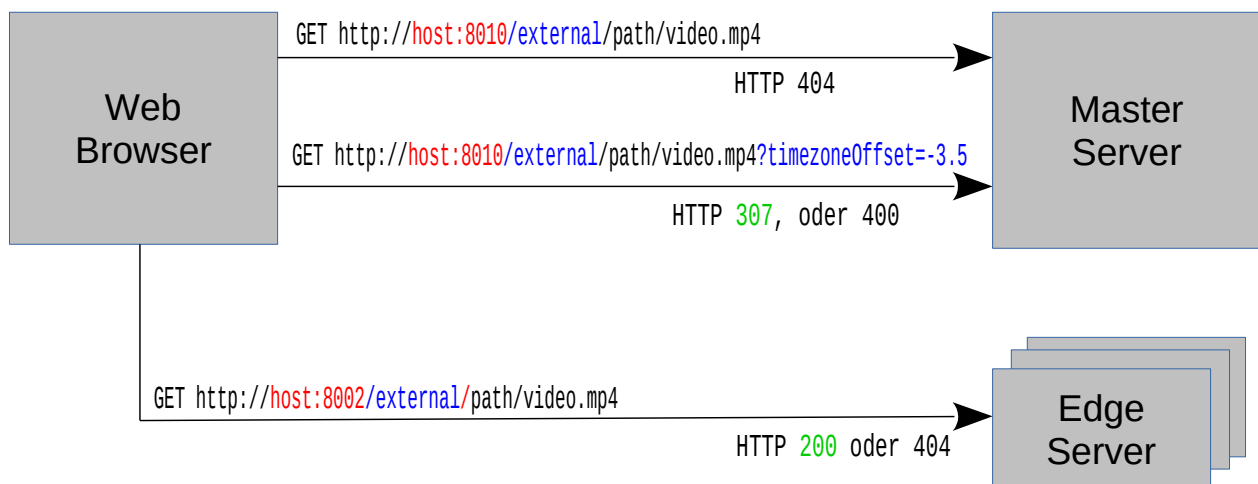
Benennt die Klasse `HttpRedirectHandlerSkeleton` in `HttpRedirectHandler` um, und implementiert die dort mit TODO gekennzeichneten Stellen so dass jeder vom Handler behandelte HTTP Anfrage in einer HTTP 307 temporary redirect Antwort resultiert, und damit die Browser anweist die Anfrage mit derselben HTTP Methode und einer Redirect-URI zu wiederholen.

Testet Eure Implementierung durch Starten der Klasse `HttpRedirectServer`; Die Konfigurationsdatei `redirect-servers.properties` beinhaltet dabei die Socket-Adressen bzw. Port-Nummern der Redirect-Server. Startet diese mittels der Klasse `HttpServer` (in Eclipse mittels einer Run-Configuration pro Port) so dass von den Servern z.B. die Self-HTML Site unter dem „/external“-Kontextpfad angeboten wird.

Aufgabe B: 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 Redirect-Server (die Edge-Server) findet sich in der Datei `edge-servers.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 `HttpServer`) welche auf den Ports 8001, 8002 und 8003 zu starten sind.

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

Implementierung

Benennt die Klasse `HttpEdgeRedirectHandlerSkeleton` nach `HttpEdgeRedirectHandler` um, und erweitert diese an den dort unter `TODO` vermerkten Stellen.

Ihr könnt Eure Implementierung testen indem ihr

- zuerst die drei Edge-Server (mit einigen Videos zum Testen) 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 URL wie `https://myserver:port/external/path/video.mp4?timezoneOffset=3.5` eingibt
- und schließlich prüft ob das Video vom passenden Edge-Server abgespielt wird.