INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTATIONSDAG 2015-06-04

EXAMENSARBETE Optimizing clients with API gateways
STUDENT Anton Fagerberg
HANDLEDARE Roger Henriksson (LTH), Nils-Olof Bankell (Jayway)
EXAMINATOR Görel Hedin (LTH)

Optimera klienter med API-gateways

POPULÄRVETENSKAPLIG SAMMANFATTNING **Anton Fagerberg**

APIer över HTTP anpassas sällan för olika klienters behov vilket medför krånglig kommunikation och reducerad prestanda. En API-gateway kan placeras mellan klienter och APIer för att åtgärda detta.

När datan som en server erbjuder inte överensstämmer med vad en klient vill ha leder detta ofta till att flera anrop måste göras och att datan sedan måste sammanfogas på klienten. Detta medför ökad komplexitet i klienten och reducerad prestanda. Vanligtvis sker kommunikationen via HTTP/1.1 där begränsningar i protokollet kan ge upphov till stora problem. Genom att placera en API-gateway mellan en klient och de APIer som används kan klientens kod förenklas och trafiken optimeras.

Ett ramverk för att konstruera API-gateways utvecklades och användes sedan i tre fallstudier för att undersöka vilken eventuell förbättringspotential som kunde uppnås. I den mest signifikanta fallstudien, vilken undersökte ett system för incidentrapportering i några av Sveriges kommuner, kunde den överförda datamängden minskas med hela 99%. Detta uppnåddes genom att slå ihop flera rekursiva anrop till ett enda anrop, filtrera bort onödig data och komprimera svaret innan det skickades till klienten.

En annan fallstudie, vilken analyserade ett system vilket används av en revisionsfirma, visade hur kommunikationen kunde förbättras ur utvecklingssynvinkel. Ett svårarbetat XML API kunde med hjälp av en API-gateway översättas till ett enklare JSON API vilket medförde en förenkling av klientutvecklingen.

Att introducera en API-gateway leder dock inte alltid till någon förbättring. En fallstudie av ett mobilt banksystem byggt med hybrid-teknik visade just detta vilket understryker behovet av att kunna genomföra en korrekt API analys innan en API-gateway implementeras.

I studien läggs fokus på de prestandaproblem som finns i det underliggande protokollet HTTP/1.1 och hur dessa problem kan undvikas eller reduceras med hjälp av en API-gateway. Att HTTP/1.1 undersöks beror på att det är ett väldigt etablerat protokoll, vilket används i samtliga webbläsare, samtidigt som protokollet lider av flertalet intressanta prestandaproblem. Tidigare har andra liknande problem angripits med hjälp av t.ex. “spriting”, att slå ihop flera bilder till en för att minimera antalet HTTP-anrop, och “sharding”, att använda olika sub-domäner för samma server med målet att öka antalet TCP-uppkopplingar i webbläsaren. Med hjälp av en API-gateway kan liknande tekniker även appliceras på API-anrop och inte bara på statiska resurser.

I studien ingår även en teoretisk översikt över hur en API-gateway fungerar samt en analys av dess potential ur flera olika synvinklar – från en enkel “proxy” till mer avancerade tekniker som att slå ihop anrop och att transformera datan mellan olika format.

Ett ramverk för att skapa API-gateways konstruerades vilket ledde till att reflektioner ur teknisk synvinkel kunde ges. Att utveckla en API-gateway ställer höga krav på tillgänglighet och prestanda med ett stort fokus på “concurrency”. Som en följd av dessa krav har BEAM, Erlangs VM, tillsammans med språket Elixir använts vilket gav insikter i vilken teknik som är lämplig att använda vid konstruktionen av en API-gateway i praktiken.