



Ansökan om Självständigt arbete, 1DV42E  
*Praktisk uppgift, Datavetenskap*

## Studentinformation

<b>Namn:</b> Josef Ottosson	<b>Användarnamn:</b> jo222ev
<b>Personnummer:</b> 19900123-3890	<b>Program och inriktning:</b> Webbprogrammerare

## Information om arbetet

<b>Arbetsnamn:</b> Prestandajämförelse mellan ramverk och webbservrar
<b>Eventuell uppdragsgivare (namn och ort):</b> Isotop - Stockholm
<b>Övriga upplysningar:</b>  Kontaktpersoner på Isotop: Christer Svensson - <a href="mailto:christer.svensson@isotop.se">christer.svensson@isotop.se</a> Bob Dahlberg - <a href="mailto:bob.dahlberg@isotop.se">bob.dahlberg@isotop.se</a>



## Vision

### **Bakgrund och problembeskrivning**

Det finns idag många olika språk/ramverk att tillgå när det gäller webbutveckling. Men vilket språk/ramverk är egentligen snabbast? Skiljer det något i prestanda beroende på vad för slags server man använder? Det kan också skilja i prestanda beroende på vad man utför för operationer, ett språk kan vara snabbare på rena läs-operationer medans ett annat språk är överlägset på beräkningsoperationer.

En enklare applikation skall tas fram som skall mäta prestandan på diverse saker som till exempel genomsnittlig tid för läsning/skrivning till/från fil/databas och hur lång tid det tar att utföra diverse komplexaoperationer(strängparsning etc).

Applikationen skall mäta prestanda enbart på *dynamiska operationer*, detta på grund utav att det redan finns gott om mätningar på statiskt innehåll.

Applikationen skall utvecklas i flera språk/ramverk; PHP, Ruby/Ruby on Rails, Python/Django och NodeJS och köras på två olika opensourceservrar, Apache och Nginx.

Gällande NodeJS så är det i grunden ett ramverk för att kunna köra Javascript på servern. Oftast brukar man också använda sig utav NodeJS som själva servern, man brukar alltså inte köra NodeJS på till exempel en Apache-installation utan den körs som en helt fristående server.

I det här fallet kommer NodeJS användas för att jämföra prestanda jämfört med en "traditionell" server, till exempel hur står sig en NodeJS backend mot dynamiska språk/ramverk som levereras via Apache eller Nginx?

Tanken med applikationen är att framställa data som kan användas som underlag till B-uppsatsen som skapas parallellt med applikationen.

### **Användar- och målgrupper**

Användare av applikationen är i huvudsak skaparen till det här arbetet, tillsammans med några anställda på Isotop.

Målgruppen är i förstahand personer som är intresserade av prestanda i allmänhet och webbservrar/språk i synnerhet.

### **Marknad**

Det finns egentligen inget standardiserat sätt att mäta ett språks prestanda. Det finns ett projekt på <http://benchmarksgame.alioth.debian.org/> som har försökt att sammanställa/mäta skillnader mellan diverse olika programmspråk. En nackdel där är att det är inte samma skapare till varje applikation vilket gör att koden kanske inte är lika bra/dålig som om samma programmerare hade skapat alla



applikationer.

Det finns desto fler verktyg för att mäta en webbservers prestanda.

De två största är Apachebench och httpperf.

Apachebench är en command-line-tool som installeras vid en standard apacheinstallation. Dess främsta styrka är att Apache är såpass populärt så de har väldigt många möjliga användare. Främsta nackdelen är att det bara kan använda en tråd, det har att göra med att flerkärninga processorer inte existerade då och därför är det i dagsläget nästan meningslöst att testa flerkärninga servrar.

httpperf är ett annat verktyg som är utvecklat av Hewlet-Packard(HP). Dess främsta styrka är möjligheten att "overloada" en webbserver, nästan som en DDOS-attack. Det här gör att det lämpar sig särskilt bra för att testa hur servern beter sig med många samtidiga användare. En nackdel är att det inte är lika enkelt att komma igång med som Apachebench.

Gemensamt för båda dessa är att de syftar till att stresstesta webbservrar. Den applikation som jag skall ta fram syftar till att påvisa skillnader i prestanda mellan olika språk/ramverk och om det skiljer något i prestanda beroende på vad för server som används.

### **Baskrav/Egenskaper/Feauters**

*Har problem att komma på relevanta baskrav, några idéer?*

BK 1 - Applikationen skall utföra samma sak oberoende av språk.

BK 2 - Mätningarna skall utföras flera gånger i följd och spara alla värden.

BK 3 - Statistik skall presenteras efter varje körning.

BK 4 - Applikationen skall enbart mäta dynamiska operationer.

BK 5 - Alla mätningar skall ske i samma hårdvarumiljö och med samma förutsättningar

### **Teknik**

#### **Programspråk - Ramverk**

- Ruby - Ruby on Rails
- PHP 5
- Python - Django
- Javascript - NodeJS

#### **Servermjukvara**

Apache och Nginx med Linux som OS.

#### **Utvecklingsprocess och dokumentation**

Utvecklingsprocessen kommer att vara agil med försök till leverans varje vecka. I början av processen



kommer en idé till applikation att tas fram för att sedan börja implementeras i de olika språken. En metod för hur mätningen skall gå till skall också tas fram.

För att få så relevanta resultat som möjligt kommer varje test att köras X antal gånger och vid olika tillfällen.

De dokument som kommer att tas fram är:

Rapporten - Själva huvuddelen av arbetet.

Statistik - Resultaten för alla tester.

Mjukvaruarkitektur - Vilken miljö testerna har utförts i/på.