

Skalierbarkeit von Webanwendungen mit Ruby on Rails

im Vergleich zu einem bestehenden
Massively Multiplayer Online-Browserspiel
Bachelor Report - B.Sc. Informatik

Universität Bremen / AG Rechnernetze
Thorben Schröder (walski@tzi.de)
Betreuer: Carsten Bormann (cabo@tzi.de)

31. Mai 2009

Inhaltsverzeichnis

1	Lastmodell	2
2	Lastsimulation	2
2.1	Evaluation bestehender Werkzeuge	2
2.1.1	Faban	2
2.1.2	Apache Bench - ab	2
2.1.3	httperf	2
3	Glossar	2

1 Lastmodell

2 Lastsimulation

2.1 Evaluation bestehender Werkzeuge

2.1.1 Faban

<http://faban.sunsource.net/> Eine Java Benchmark Lösung mit der es, anders als mit einfachen Tools wie beispielsweise AB oder httpperf, möglich sein soll auch komplexere Benchmarks durchzuführen. Die Benchmarks werden dabei nicht konfiguriert sondern können in Java programmiert werden. Faban sorgt dann mit einer Infrastruktur aus einem “Master Server” und beliebig vielen, auch auf anderen Hosts als der Master Server liegenden, Agenten dafür, dass dieser Benchmark aufgeführt und die Ergebnisse der einzelnen Agenten wieder zusammengeführt werden.

Die Konfiguration und Benutzung von Faban gestaltet sich schwierig. Nachdem dann auch aus der Dokumentation hervorgeht, dass die Implementierung von zu verfolgenden Klickpfaden per Markovketten ein Erweitern des Faban Systems unumgänglich machen wird das Simulieren von Lasten mit Faban zunächst zu Gunsten der Evaluierung anderer Systeme aufgegeben.

2.1.2 Apache Bench - ab

<http://httpd.apache.org/docs/2.0/programs/ab.html> Apache Bench (*ab*) ist ein Kommandozeilen Programm, dass es auf einfache Weise erlaubt einen Benchmark auf eine einzelne Ressource per HTTP durchzuführen. *ab* misst dabei vor allem die Auslieferungszeit der Ressource und errechnet einige Hilfswerte wie zum Beispiel die bedienten Anfragen pro Sekunde. Auch wenn *ab* die vorgestellte Aufgabe gut erfüllt eignet es sich nur sehr eingeschränkt für das Testen einer Folge von Ressourcen, wie es im Lastmodell dieser Arbeit vorgesehen ist. Der einzige Weg mit *ab* solche Folgen abzuarbeiten wäre es für jede Ressource einen neuen *ab* Prozess zu starten, was untereinander zu Problemen bei der Synchronisierung führen würde, schwer zu kontrollieren wäre und zudem den Nachteil hätte, dass die Ergebnisse schwer zusammenführen sind.

Aus diesen Gründen wird vom Gebrauch von *ab* als Lastsimulator für diese Arbeit abgesehen.

2.1.3 httpperf

<http://www.hpl.hp.com/research/linux/httpperf/> httpperf ist, ähnlich wie *ab*, ein Kommandozeilen Werkzeug zum Durchführen von Benchmarks per HTTP. Auch die dabei gemessenen Werte ähneln denen von *ab*. Im Gegensatz zu *ab* erlaubt httpperf allerdings nicht nur das Benchmarken einer einzelnen Ressource, sondern auch das einer ganzen Folge von Ressourcen. Die Konfiguration dieser Folgen ist sehr detailliert möglich so sind insbesondere auch Wartezeiten zwischen zwei Aufrufen und das Gleichzeitige Aufrufen mehrerer Ressourcen zu einem Zeitpunkt möglich, um so beispielsweise den Aufruf eines HTML Dokuments sowie aller darin referenzierten JavaScript-, Bild- und Cascading Style Sheets (CSS)-Dateien zu simulieren.

Mit dieser, in der httpperf Dokumentation “replay” genannten, Möglichkeit Folgen von Ressourcen zu benchmarken wäre es Möglich httpperf als Lastsimulationswerkzeug im Rahmen dieser Arbeit zu verwenden.

3 Glossar

- Cascading Style Sheets
- HTML
- JavaScript
- ab

- httpperf
- faban
- Markov Kette