

Exposé meiner Diplomarbeit „Ein Performance-Benchmark für Java-Web-Frameworks“

Christophe Sokol

Institut für Informatik
Freie Universität Berlin

04. Oktober 2012

Einführung

Motivation

Stand der Dinge

Forschungsfragen

Vorgehen

Literatur

Einführung: Was ist Performance?

- ▶ Durchsatz
- ▶ Latenz / Ladezeit
- ▶ Last

- ▶ *Time to first byte*
- ▶ *Time to last byte*
- ▶ Seitenaufbau

Einführung: Nutzungsprofile

- ▶ Gleichzeitige Verbindungen
- ▶ Parallele Sessions
- ▶ Reihenfolge, Menge und Frequenz der Aufrufe pro Session
- ▶ Komplexität des angeforderten Elemente
- ▶ Statische oder dynamische Inhalte

Einführung: Ressourcen

- ▶ CPU
- ▶ Arbeitsspeicher
- ▶ Disk-I/O

Motivation: Warum ist Performance relevant?

- ▶ Zufriedenheit der Nutzer, Auswirkungen auf den Umsatz
- ▶ Kosten beim Betrieb
- ▶ Eignung für einen bestimmten Zweck
- ▶ *Ein* Kriterium bei der Auswahl eines Frameworks

Motivation: Welchen Einfluss hat das Framework?

- ▶ Erheblichen Anteil bei der Abarbeitung eines Requests
- ▶ Strukturiert die mit dem Framework umgesetzte Anwendung
- ▶ Bringt Werkzeuge mit (Asset-Management, Caching, Profiling, Logging)

Stand der Dinge: Framework-Benchmarks

- ▶ Gibt es wenige
- ▶ Konzentrieren sich auf einen spezifischen Aspekt
- ▶ Einmal erstellt, nicht mehr gepflegt
- ▶ Schnell veraltet, nicht allgemein akzeptiert
- ▶ Methodik unklar

Stand der Dinge: Framework-Benchmarks #2

- ▶ Uribe, Leonardo. 'Understanding JSF 2 and Wicket: Performance Comparison'. JSF and MyFaces Hints and Tips, May 18, 2012. <http://lu4242.blogspot.de/2012/05/understandingjsf-2-and-wicket.html>.
- ▶ Rui, Wang. 'PHP Framework MVC Benchmark - V20111201-4', December 1, 2011. <http://www.ruilog.com/blog/view/b6f0e42cf705.html>.
- ▶ Van den Enden, Stijn. 'WWW: World Wide Wait. A Performance Comparison of Java Web Frameworks' presented at Devovx, Antwerp, Belgium, November 2011. <http://prezi.com/dr3on1qcajzw/www-world-wide-wait-devovx-edition/>.
- ▶ JT Dev. 'Rails, Wicket, Grails, Play, Tapestry, Lift, JSP, Context'. About Programming Stuff, May 28, 2011. <http://www.jtict.com/blog/rails-wicket-grails-play-lift-jsp/>.
- ▶ Thomas, Peter. "'Perfbench" Update: Tapestry 5 and Grails'. Incremental Operations, September 14, 2009. <http://ptrthomas.wordpress.com/2009/09/14/>.
- ▶ Thomas, Peter. 'Seam / JSF Vs Wicket: Performance Comparison'. Incremental Operations, January 14, 2009. <http://ptrthomas.wordpress.com/2009/01/14/>.
- ▶ McCabe, James. 'Webofcontacts - A Comparison of Web Application Frameworks'. Google Project Hosting, 2009. <http://code.google.com/p/webofcontacts/>.

Stand der Dinge: Feedback zu Benchmarks

„I would love to see an update on this for Tapestry 5.3.4 (just about to be released), where I've put in some substantial performance improvements.“

„The 1.5RC versions of Wicket that you tested had at least one major performance bug.“

„Few comments about rails: webrick is slow, for ruby 1.9 use thin (...)“

„There is something wrong with this test. I've scaled rails on sites getting tens to hundreds of millions of requests per day using ruby and jrubby, and the numbers you have above do not reflect reality.“

Stand der Dinge Feedback zu Benchmarks #2

„Hello, I've just added a playapp + rythm template engine into your testing set. [...] Sorry, forgot to mention the pull request is here: (...)“

„The memory problem with play-scala netty is that the server is started with -Xms512m. Try play run -%prod -Xms50m.“

„I'm investigating why Wicket is slow and I see that the app makes 5000 requests to non-existing images. [...] I'll send you Pull Request for all mine findings.“

Forschungsfragen

- ▶ Mit welchen Parametern lässt sich die Performance von Java-Web-Frameworks charakterisieren?
- ▶ Was sind Aufgabenstellungen, die von und mit einem Framework gelöst werden?
- ▶ Wie muss der Benchmark strukturell und organisatorisch aufgebaut sein, damit er Relevanz er- und behält?

Vorgehen: Problem

- ▶ Messaufbau und Beispielaufgaben können beliebig komplex sein
- ▶ Anzahl der betrachteten Frameworks beliebig groß
- ▶ Umfang und Relevanz des Feedbacks von außen nicht vorhersagbar
- ▶ Benötigte Zeit sehr schwer kalkulierbar!

Vorgehen: Lösungsansatz

- ▶ Automation des gesamten Prozesses
- ▶ Iteratives Vorgehen
- ▶ Interaktion mit den Communities

Vorgehen: Szenarien

- ▶ „Hallo Welt“
- ▶ JPetStore
- ▶ TPC-W (E-Commerce)

- ▶ Formulardarstellung und -validierung
- ▶ Templates, Skins
- ▶ I18N, L10N
- ▶ Autorisierung, Authentifizierung
- ▶ Session-Handling
- ▶ Caching
- ▶ Datenbankabstraktion, ORM

Vorgehen: Arbeitspakete

- ▶ Manueller Benchmark mit einem Framework, Hallo-Welt-Anwendung, manuelle Auswertung
- ▶ Automatisierung des Testdurchlaufes, der Auswertung
- ▶ Konzeption der Benchmark-Umgebung (JRE, Bibliotheken)
- ▶ Konzeption von Szenarien bzw. Beispielanwendungen
- ▶ Ermittlung von geeigneten Metriken
- ▶ Implementierung der Anwendungen für die Frameworks
- ▶ Validieren der Umgebung, des Benchmark-Werkzeuges

Vorgehen: Technologien

- ▶ 3x Ubuntu-VMs am FB, später ggf. Amazon EC2
- ▶ Configuration Management mit Puppet
- ▶ Benchmarks mit JMeter
- ▶ Sourcecode bei Github
- ▶ Logging, Monitoring: JMeter, Perfmon, statsd, Greylog
- ▶ Auswertung und Aufbereitung der Ergebnisse? R?

Literatur

- ▶ Binnig, Carsten, Donald Kossmann, Tim Kraska, and Simon Loesing. 'How Is the Weather Tomorrow?: Towards a Benchmark for the Cloud'. In *Proceedings of the Second International Workshop on Testing Database Systems*, 9:1–9:6. DBTest'09. New York, NY, USA: ACM, 2009. <http://www.eecs.berkeley.edu/~kraska/pub/dbtest09-cloudbench.pdf>.
- ▶ Casteleyn, Sven, Florian Daniel, Peter Dolog, and Maristella Matera. 'Performance Testing'. In *Engineering Web Applications*. Data-Centric Systems and Applications. Berlin: Springer-Verlag, 2009. <magnet:?xt=urn:btih:a162db1795910763f447fdc1f19b60325136fc80>.
- ▶ F5 Networks, Inc. 'Creating a Robust Performance Testing Methodology', 2007. <https://devcentral.f5.com/downloads/f5/creating-performance-test-methodology.pdf>.
- ▶ Hashemian, Raoufhsadat, Diwakar Krishnamurthy, and Martin Arlitt. 'Web Workload Generation Challenges – an Empirical Investigation'. *Software: Practice and Experience* 42, no. 5 (2012): 629–647.
- ▶ Nagpurkar, P., W. Horn, U. Gopalakrishnan, N. Dubey, J. Jann, and P. Pattnaik. 'Workload Characterization of Selected JEE-based Web 2.0 Applications'. In *IEEE International Symposium on Workload Characterization*, 2008. IISWC 2008, 109–118, 2008. <http://ldc.usb.gov/~mcuriel/Cursos/WC/11-%20nagpurkar.pdf>.
- ▶ Sakr, Sherif, and Fabio Casati. 'Liquid Benchmarks: Benchmarking-as-a-service'. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, 451–452. JCDL '11. New York, NY, USA: ACM, 2011. <http://www.nicta.com.au/pub?doc=4146>.
- ▶ Sobel, Will, Shanti Subramanyam, Akara Sucharitakul, Jimmy Nguyen, Hubert Wong, Arthur Klepchukov, Sheetal Patil, O Fox, and David Patterson. *Cloudstone: Multi-platform, Multi-language Benchmark and Measurement Tools for Web 2.0*, 2008. <http://cca08.org/papers/Paper33-Armando-Fox.pdf>.
- ▶ Sullivan, Nicole. 'Design Fast Websites', October 14, 2008. <http://de.slideshare.net/stubbornella/designing-fast-websites-presentation>.