Web Performance Optimization: Analytics

Wim Leers
December 3, 2009

Thesis proposal for the degree of
master in computer science/databases

Hasselt University
Academic year 2009-2010

1 Introduction

My bachelor thesis was about making Drupal [1] web sites load faster. 80 to 90% of the response time (as observed by the end user) is spent on downloading the components of a web page [3]. Therefor this is also the part where optimizations have the largest effect.

To be able to prove the positive impact of optimizing the loading of the components of a web site — thereby proving that the work I was going to have done had a positive impact — I researched existing page loading profiling tools. Episodes [4, 5] (which refers to the various *episodes* in the page loading sequence) came out as a clear winner:

- Episodes aims to become an industry standard;
- Episodes is open source;
- Episodes is a piece of JavaScript that runs in the browser on each loaded page, thus for each real visitor, thus it represents the real-world performance (all existing solutions [6, 7, 8, 9] require simulations, which implies they're also only suitable for simulating traffic on a new version of a web site before it goes live);
- Episodes does not require any hardware other than a server to log to.

Also as part of my bachelor thesis, I wrote a simple Drupal module — the Episodes module [10] — that could create simple charts to compare the average page loading time per day per geographic region. For my test case, with two weeks of collecting data, this was the resulting dataset:

> About 100 MB worth of statistics had been logged. These were then imported on June 25, resulting in a database table of 642.4 MB. More than 2.7 million episodes were collected over more than 260,000 page views.

While my test case was a fairly big web site (500,000-1,000,000 page views per month), that's nothing when compared with the top-100 web sites. And even for these mere 2.7 million recorded episodes, it took several minutes to generate simple charts (see figures 1 and 2). And that doesn't include importing the log file into the database.

That is of course largely due to the fact that the database schema used is extremely inefficient: it is in fact a verbatim copy of the log file. The database schema should be optimized for the queries that are necessary to generate the charts. In its current incarnation, multiple full table scans are required, which is a worst case scenario.

Despite its obvious (intended) lack of optimizations, it was sufficient to prove that File Conveyor [2] — the daemon that I wrote to automatically sync files to any CDN, regardlesss of the file transfer protocol used — when integrated with a Drupal web site and thus providing CDN integration for that web site,

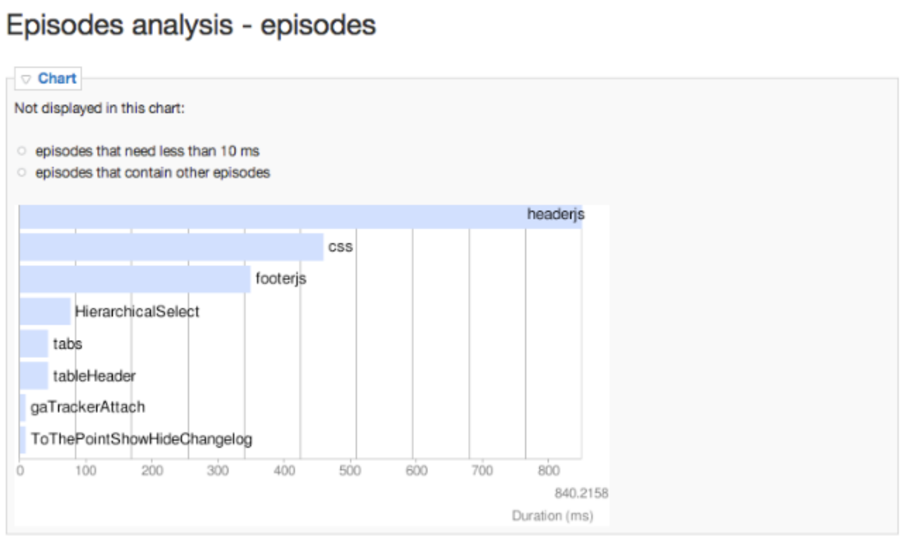

Figure 1: Episodes analysis charts about episodes generated by the Drupal Episodes module.

had a positive impact: the test web site consistently loaded about *twice as fast*, especially for visitors with slower internet connections, such as visitors from Brazil. Without this proof-of-concept implementation, I would never have been able to prove the positive impact on performance.

2 Context

Ever since Steve Souders' *High Performance Web Sites* book [3], interest in making web sites load faster has been increasing. More and more big companies with a strong *web presence* are paying attention to page loading performance: the well-known ones such as Microsoft, Yahoo, Google, but also big companies that are not technology companies such as Amazon, White Pages, Shopzilla, Edmunds, Netflix . . . .

Page loading profiling tools

As a result of this trend, a large number of advanced page loading profiling tools are being developed:

- Deep tracing of the internals of Internet Explorer, by using dynaTrace Ajax [11]
- JavaScript memory heap profiler and sample-based CPU profiler in WebKit/Google Chrome [12]

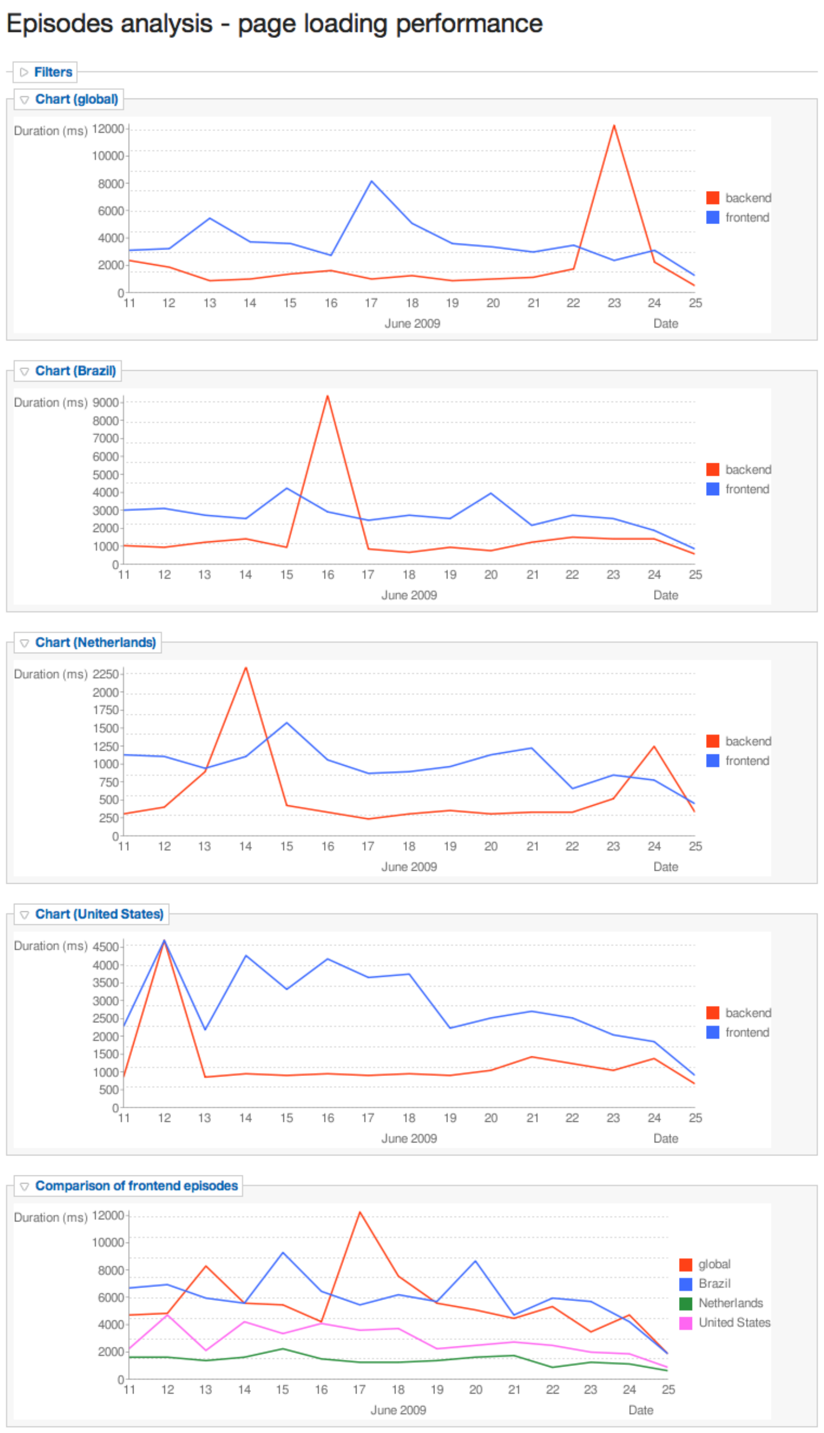

Figure 2: Episodes analysis charts about page loading performance generated by the Drupal Episodes module.

- Firefox has been leading the way with the development of the Firebug extension and the Yahoo! YSlow [13] & Google Page Speed [14] Firebug plug-ins

Proposals

Recent proposals (in the last three months at the time of writing) for web performance optimization include:

- SPDY [15], a new application-level protocol that learns from the mistakes of HTTP (which is ten years old). This protocol specification is currently in draft state, but tests of the researchers (at Google) show that pages of the top 25 web sites loaded up to 55% faster.
- Resource Packages [16, 17]. A resource package is a zip file that bundles multiple resources into a single file and therefor requires only a single HTTP response and avoids multiple round trip delays. Browsers typically only take advantage of about 30% of their bandwidth capacity because of the overhead of HTTP and TCP and the various blocking behaviors in browsers. This proposal would result in less bandwidth being consumed by overhead. Plus, it's backwards compatible: browsers that don't support it load the page the same way as today.
- Web Timing [18]. This is a proposal presented to the W3C and welcomes feedback from browser vendors. It effectively means that Episodes is being moved into the actual browser partially, to get rid of the latency of loading Episodes' JavaScript and the relatively inaccurate time measurements of JavaScript. It would also allow us to get a complete picture of the end-to-end latency, which is impossible to do with Episodes (which can only rely upon what JavaScript itself can do). This proposal is only a working draft and requires interacting with browser vendors to ensure all current major browsers will implement this. Even in the best case scenario, it will take years until the majority of the installed browsers will support this. Until then, we will be limited in what we can measure. Hence this proposal should move forward as fast as possible.

Both would severely affect browser implementations, which indicates the willingness and likeliness to change the way data is transferred over the internet to make web sites load faster.

Search engine ranking

The importance of page loading performance is lifted to an even higher level by the fact that Google announced that it will likely let page speed (i.e. page loading performance) influence the ranking of web pages [19]. This effectively means that all companies whom have been paying for SEO (search engine optimization) will also have to consider page loading performance.

I said "likely", but you can probably interpret that as "definitely", because Google has already added page load time tracking to its Webmaster Tools [20] and even announced its own DNS service (Google Public DNS [21]) "to make the web faster" just one day later!

3 Problem

The main problem is that sites are too slow. In my bachelor thesis, I implemented a daemon to synchronize files to a CDN, which is one of the most important ways to speed up the loading of a web site.

However, simply implementing all known tricks is not enough, because using a CDN might speed up your web site for half your visitors and slow it down for the other half — although that's an extremely unlikely scenario. That's why you need to be able to do Continuous Profiling (cfr. Continuous Integration).

Continuous Profiling means that you are continuously monitoring your real-world page loading performance: you must track the page loading characteristics of each *loaded page!* That by itself is easy: all it requires is to integrate Episodes with your web site. The actual problem lies in analyzing the collected data. To be able to draw meaningful conclusions from the collected data, we need to apply data mining techniques as well as visualizing the conclusions that are found. E.g. pages may be loading slower from South-Africa because the CDN's server there (a *PoP*) is offline, or your shopping cart checkout page may be loading slow in Firefox because of a JavaScript issue, or a particular page may be loading slow in all web browsers because of bad CSS on that page, or maybe your site is loading very slow for all users of a certain ISP because their DNS server has poor performance. All of these problems (and more) could be pinpointed (albeit partially) automatically.

So what I think is needed, is something like Google Analytics, but for *page loading performance* instead of just *page loads.*

4 Proposal: Analytics Suite

So that is exactly what my proposal is: an analytics suite for tracking page loading performance. An application that can automatically extract conclusions out of Episodes logs and visualize them. This application should be very scalable (as the number of recorded episodes is typically an order of magnitude higher than the number of page views) and possibly also distributed. You should also be able to go back to any point in the past and view the page loading performance at that time. Thus, efficient storage is also a requirement. Finally, it should be an open source application that can be developed further by others after I finish my master thesis.

I told Steve Souders about my idea for my master thesis — he is the most prominent speaker, researcher and evangelizer in the page loading performance scene and on Google's payroll to push this forward — and asked him for feedback. His response:

> I did a mini performance conference in LA last month and heard three big companies (Shopzilla, Edmunds, and Google PicasaWeb) get up and say they had regressed in their web site performance because they weren't tracking latency. I realized that most companies aren't even at the point where they have good metrics. I think the first idea — Google Analytics for latency — is the best idea. [. . .] It would be great if this lived on Google AppEngine. Users could take the code and spin up their own instance — for free! You could also host a shared instance. I will say that the work [. . .] on AppEngine has been hard because of the datastore - my officemate does the programming and it's taken him months to do what I did in a few days on the LAMP stack.

He agrees on the necessity for such an application and immediately proposes to make it run on Google AppEngine [23], which is a free platform for web applications with its own, apparently complicated, datastore that is schemaless. The idea is that anybody can create a free AppEngine account, install this application and get a Continuous Profiling application for free!

Whether it would run or Google AppEngine or not, it's certain that an open source Continuous page loading performance profiling would be very valuable. I hope I'll be able to build it!

References

[1] *Drupal*, http://drupal.org/

[2] *File Conveyor*, http://fileconveyor.org/

[3] *High Performance Web Sites*, Steve Souders, 2007, O'Reilly, http://stevesouders.com/hpws/

[4] *Episodes: a Framework for Measuring Web Page Load Times*, Steve Souders, July 2008, http://stevesouders.com/episodes/paper.php

[5] *Episodes: a shared approach for timing web pages*, Steve Souders, 2008, http://stevesouders.com/docs/episodes-tae-20080930.ppt

[6] *Gomez*, http://www.gomez.com/

[7] *Keynote*, http://www.keynote.com/

[8] *WebMetrics*, http://www.webmetrics.com/

[9] *Pingdom*, http://pingdom.com/

[10] *Episodes*, http://drupal.org/project/episodes

[11] *Deep Tracing of Internet Explorer*, John Resig, Mozilla, November 17, 2009, http://ejohn.org/blog/deep-tracing-of-internet-explorer/

[12] *An Update for Google Chrome's Developer Tools*, Pavel Feldman, Google, November 30, 2009, http://code.google.com/events/io/2009/sessions/MeasureMillisecondsPerformanceTipsWebToolkit.html

[13] *Yahoo! YSlow*, http://developer.yahoo.com/yslow/

[14] *Google Page Speed*, http://code.google.com/speed/page-speed/

[15] *A 2x Faster Web*, The Chromium Blog, Mike Belshe, November 11, 2009, http://blog.chromium.org/2009/11/2x-faster-web.html

[16] *Making browsers faster: Resource Packages*, Alexander Limi, November 17, 2009, http://limi.net/articles/resource-packages/

[17] *Fewer requests through resource packages*, Steve Souders, November 18, 2009, http://www.stevesouders.com/blog/2009/11/18/fewer-requests-through-resource-packages/

[18] *Web Timing (Working Draft)*, Zhiheng Wang, Google Inc., September 26, 2009, http://dev.w3.org/2006/webapi/WebTiming/

[19] *Google: Page Speed May Become a Ranking Factor in 2010*, WebProNews, November 19, 2009, http://www.webpronews.com/topnews/2009/11/13/google-page-speed-may-be-a-ranking-factor-in-2010

[20] *How fast is your site?*, Webmaster Central Blog, Sreeram Ramachandra & Arvind Jain, December 2, 2009, http://googlewebmastercentral.blogspot.com/2009/12/how-fast-is-your-site.html

[21] *Introducing Google Public DNS: A new DNS resolver from Google*, Google Code Blog, Prem Ramaswami, December 3, 2009, http://googlecode.blogspot.com/2009/12/introducing-google-public-dns-new-dns.html

[22] *Google Analytics*, http://google.com/analytics

[23] *Google AppEngine*, http://code.google.com/appengine