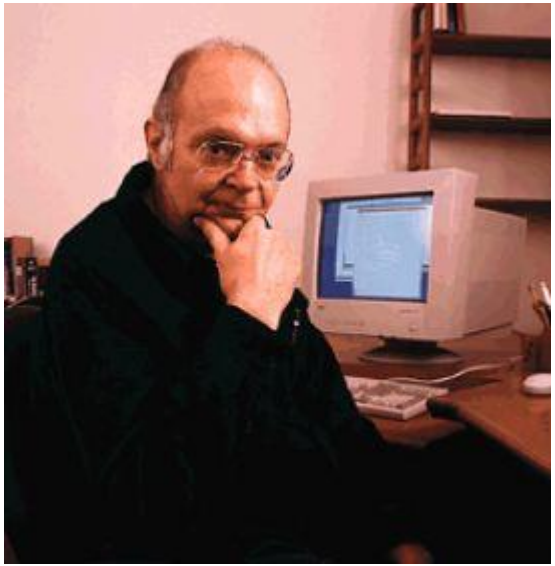
IT/Dev CONNECTIONS

MYTHBUSTING WEB PERFORMANCE

Christian Wenz
CEO
Actition GmbH

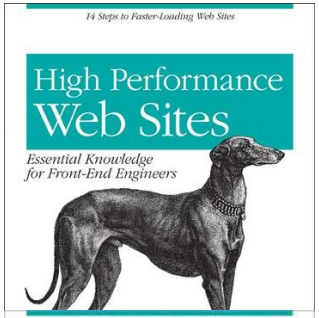
Delivered by KNect365 TMT an Informa business



- Donald E. Knuth: "premature optimization is the root of all evil (or at least most of it) in programming"
- Do not optimize before it is necessary
- Find out as early as possible *whether* it is necessary

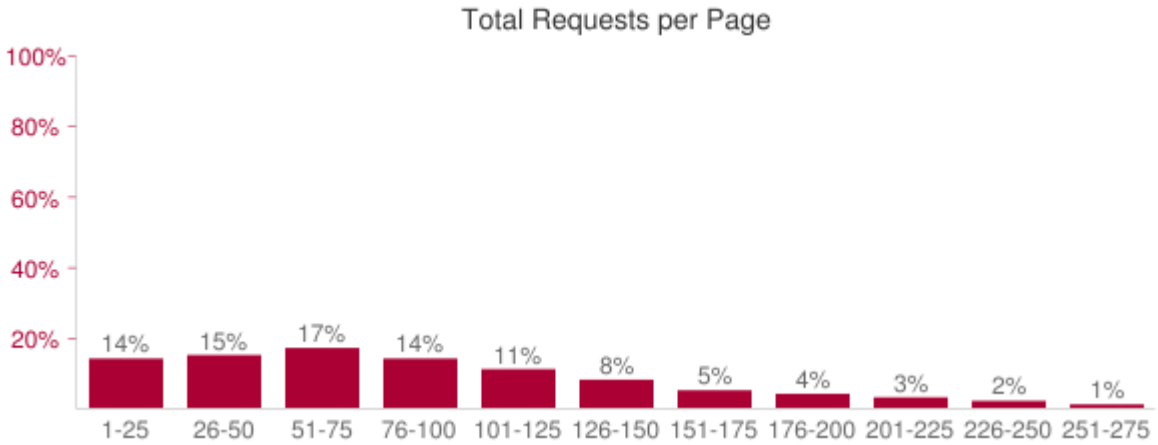
IT/Dev CONNECTIONS

- Steve Souders: 14 Rules for Faster-Loading Web Sites (2007)
- Steve Souders et al.: Even Faster Web Sites (2009)



- Some advice is still valid
- Some advice does not hold anymore
- Some advice will be outdated soon

- "Make fewer HTTP Requests"
- The less HTTP requests, the better
  - Combine files
  - Use CSS sprites



http://httparchive.org/interesting.php (Dec 2, 2016)

- As of now, the less HTTP requests, the better
- This will change with HTTP/2, though.
- „Remove unnecessary HTTP/1.x workarounds […] such as concatenated files, image sprites“ (http://chimera.labs.oreilly.com/books/1230000000545/ch12.html)

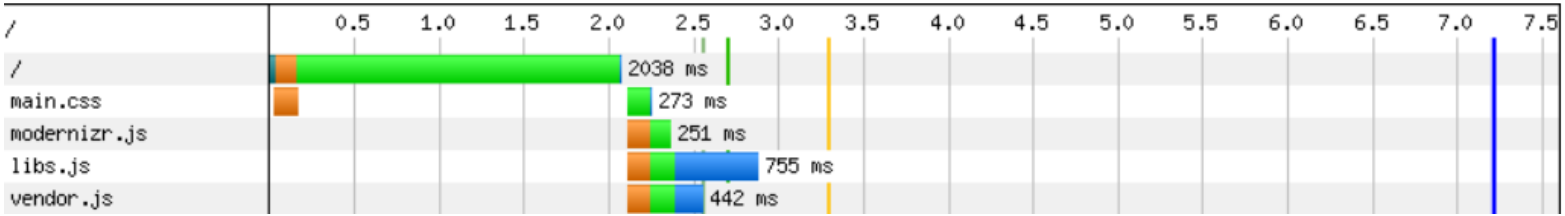
- "Use a Content Delivery Network"
- CDN = Computer network to distribute the load
- Microsoft, Google etc. also provide a CDN for some libraries

IT/Dev CONNECTIONS

- CDN are still a very effective way to speed up an application

- Nowadays we use RUM (Real User Monitoring) to pick the CDN with the best performance

- Old train of thought: use cookieless domain for static content
- With modern browsers: Consider serving CSS from the root domain
- Use a CDN for the root domain (allowing caching)
- Even better with HTTP/2 (header compression)



- http://www.jonathanklein.net/2014/02/revisiting-cookieless-domain.html

- "Add an Expires or a Cache-Control Header"
- Expires header: expiry date
  - For static content, set to a date in the distant future
  - Less data sent over the wire if files are in the cache
- Cache-Control header: Cache or not to cache, depending on content

- Use Caching.

- "Gzip Components"
- If a client sends "Accept-Encoding: gzip, deflate", the server may send gzipped content
- Compress anything that is not compressed
  - Almost anything but images and PDF files
- HTTP Content Negotiation ensures that only compatible browsers are supported

- Even better than dynamic gzipping: compress data as part of building/CI
- When using nginx: use gzip_static

- Average page has 1.6MB image data (http://httparchive.org/interesting.php) – ~2/3 of page
- Try to optimize load size (https://tinypng.com/, https://tinyjpg.com/, https://github.com/rflynn/imgmin, …)
- Evaluate new formats like WebP – which GPUs are not optimized for, though

- Resize images for devices (https://speakerdeck.com/tkadlec/mobile-image-processing-at-velocity-sc-2015)
- From a user experience point of view, do not use progressive image rendering (http://www.webperformancetoday.com/2014/09/17/progressive-image-rendering-good-evil/). Consider using Chroma subsampling.
- Lazy-load images below the fold

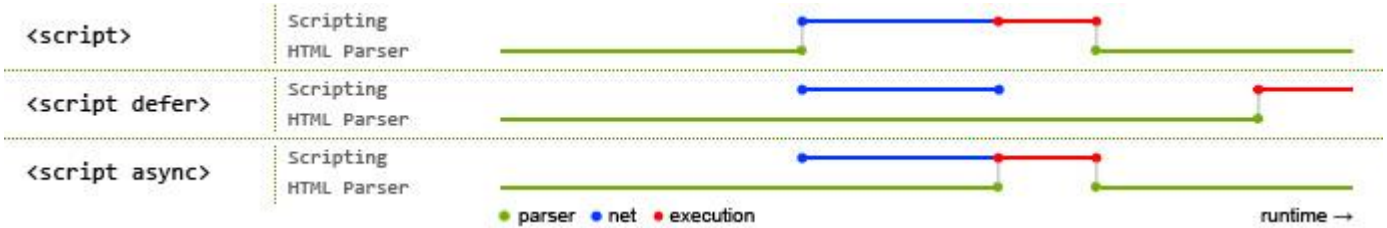
- "Put Stylesheets at the Top"
- If there are style sheets in the <head> section of the page, rendering is faster
- Browser (usually) wait with rendering until CSS files are loaded

IT/Dev CONNECTIONS

- Still true
- Use flushing (not just for CSS)

- "Put Scripts at the Bottom"
- Depending on browser/OS, clients limit the number of concurrent connections per HTTP host
- Some browsers do not load additional resources while JavaScript code is loaded
- Mind potential side effects with JavaScript code!

- Categorize JavaScript code: what you need now, and what you need later
- Important JavaScript code goes into the <head>
- Other JavaScript code is loaded asynchronously (<script async> <script defer>)



  - http://peter.sh/wp-content/uploads/2010/09/execution2.jpg
- This will eventually change with HTTP/2

IT/Dev CONNECTIONS

- "Avoid CSS Expressions"
- Only works in IE.
- Next!

- "Make JavaScript and CSS External"
- External resources may be cached
- Possible exception: homepages that are only loaded once
  - Here: consider integrating JS/CSS
  - Load external files later to cache them

- Google homepage is avoiding external JavaScript
- That‘s basically the only major page to do so
- Still true!

- "Reduce DNS Lookups"
- Some numbers:
  - Internet Explorer caches DNS lookups for 30 minutes
  - Firefox caches DNS lookups for one minute
- Use multiple domain names (but not too many)
  - Mind limitations for concurrent connections

- Lookup time vs better throughput due to parallel connections
- Rule of thump: shard on two domains, not four as previously recommended (YMMV)
- Better bandwith does not guarantee better performance – „Increasing bandwith up to 1233% makes pages just 55% faster“ (https://www.soasta.com/blog/more-bandwidth-isnt-a-magic-bullet-for-web-performance/)
- With HTTP/2, no need to shard domains anymore

- Preload ressources
  - <link rel="preload" href="/Content/styles.css" as="style">
  - http://w3c.github.io/preload/

- Provide ressource hints
  - <link rel="dns-prefetch">
  - <link rel="preconnect">
  - <link rel="prerender">
  - https://w3c.github.io/resource-hints/

- "Minify JavaScript and CSS"
- Minify files by removing whitespace and comments
  - Microsoft Ajax Minifier
  - Google Closure Compiler
  - Yahoo! YUI Compressor
  - …
- Both for external and internal code

- Absolutely.
- UglifyJS seems to give the best results (YMMV)
- Comparison of a few other tools: http://compressorrater.thruhere.net/ (if you want to upload your code there)

- "Avoid Redirects"
- Including redirects that are easy to overlook
  - http://www.xy.tld/dir -> http://www.xy.tld/dir/

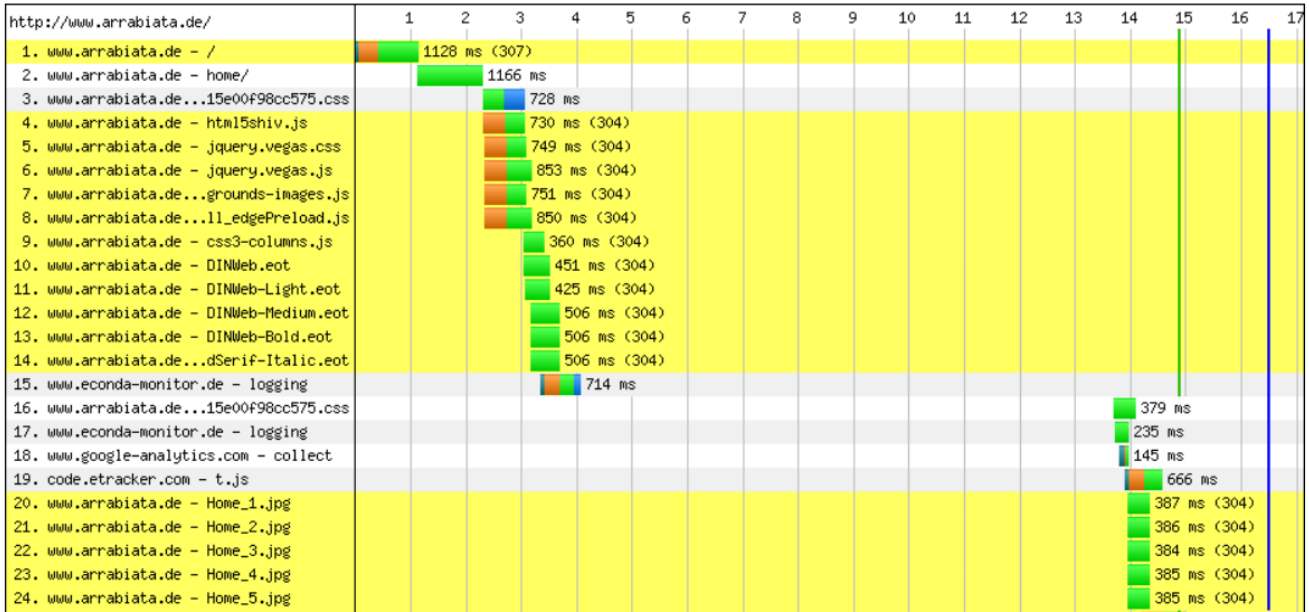
IT/Dev CONNECTIONS

- "Remove Duplicate Scripts"
- Two of the Top 10 US sites load at least one script twice
- Firefox is caching automatically, Internet Explorer is not
- Both browsers execute duplicate scripts twice, though

- "Configure ETags"
- ETag HTTP header contains a kind of fingerprint for a ressource
  - ETag: "abcd123-456-789abcde"
- On subsequent requests the browser determines with an ETag whether the resource has been changed
- Different ETags when using multiple web servers!

IT/Dev CONNECTIONS

- Do not use Last-Modified
- Do not use ETags
- Avoid zombie 304 requests



- "Make Ajax Cacheable"
- Use the preceding rules for Ajax, as well
  - "Gzip Components"
  - "Reduce DNS Lookups"
  - "Minify JavaScript"
  - "Avoid Redirects"
  - "Configure Etags"

- Optimize JavaScript
  - Specifically declare local variables with *var*
  - Cache variables
  - Cache functions
  - Amend string concatenations (not always recommended!)
  - Minimize ~~DOM~~ layout operations (first read, then batch-write)

- Newer browsers require updated strategies

IT/Dev CONNECTIONS

- Thank you!
- info@christianwenz.de
- @chwenz

Delivered by KNect365 TMT an Informa business