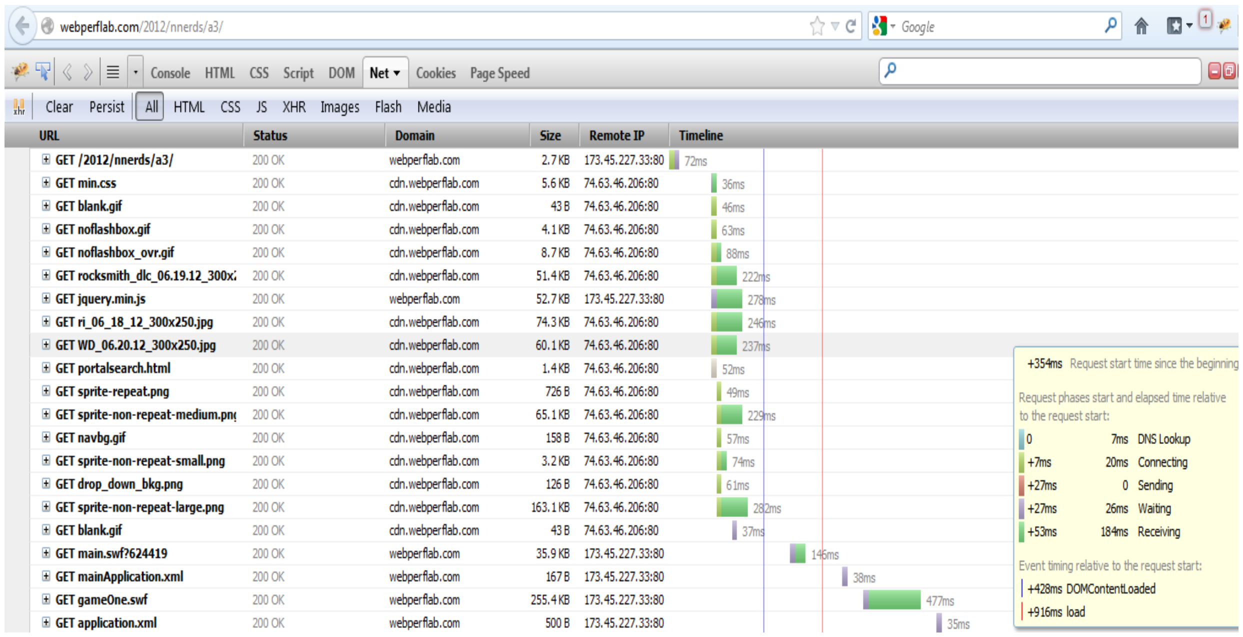
Team Ninja Nerds *webperf3*

Divya Natesan & Shrikant Adhikarla *31 July 2012*

Rules 2, 10, 11

2. Use A CDN

We decided to segregate all the resources of the project in two buckets - static and dynamic. All static resources were retrieved to the client from the cdn while the dynamic pages were maintained on the original webserver itself since it would be easier to maintain the content of such dynamic pages on our webserver itself. With respect to his project, we discovered that all the resources except index.html, Search_alt.aspx and main.swf seemed to be static. The Search_alt.aspx page provides the input to render the flash component (main.swf) that is shown in the center of the page – with images to recent or popular games. index.html, being the home page encapsulating all these resources is also dynamic since it will be updated by the administrator quite frequently. While applying this setting, we also figured out that the website loaded properly only if the javascript was also referenced from the original webperflab server along with main.swf and the related xml-s (application.xml and mainApplication.xml) that are used as parameters to generate the flash file along with gameOne, gameOverlay, screenshots. This setting ideally decreases the latency time for responses from the server since cdn-s are generally a distributed set of servers that cater to the client's request from a geographically near location or a server with the quickest response time. Please find below relevant screenshot:



webperflab.com/2012/nnerds/a3/

URL	Status	Domain	Size	Remote IP	Timeline
GET /2012/nnerds/a3/	200 OK	webperflab.com	2.7 KB	173.45.227.33:80	72ms
GET min.css	200 OK	cdn.webperflab.com	5.6 KB	74.63.46.206:80	36ms
GET blank.gif	200 OK	cdn.webperflab.com	43 B	74.63.46.206:80	46ms
GET noflashbox.gif	200 OK	cdn.webperflab.com	4.1 KB	74.63.46.206:80	63ms
GET noflashbox_ovr.gif	200 OK	cdn.webperflab.com	8.7 KB	74.63.46.206:80	88ms
GET rocksmith_dlc_06.19.12_300x	200 OK	cdn.webperflab.com	51.4 KB	74.63.46.206:80	222ms
GET jquery.min.js	200 OK	webperflab.com	52.7 KB	173.45.227.33:80	278ms
GET ri_06_18_12_300x250.jpg	200 OK	cdn.webperflab.com	74.3 KB	74.63.46.206:80	246ms
GET WD_06.20.12_300x250.jpg	200 OK	cdn.webperflab.com	60.1 KB	74.63.46.206:80	237ms
GET portalsearch.html	200 OK	cdn.webperflab.com	1.4 KB	74.63.46.206:80	52ms
GET sprite-repeat.png	200 OK	cdn.webperflab.com	726 B	74.63.46.206:80	49ms
GET sprite-non-repeat-medium.pn	200 OK	cdn.webperflab.com	65.1 KB	74.63.46.206:80	229ms
GET navbg.gif	200 OK	cdn.webperflab.com	158 B	74.63.46.206:80	57ms
GET sprite-non-repeat-small.png	200 OK	cdn.webperflab.com	3.2 KB	74.63.46.206:80	74ms
GET drop_down_bkg.png	200 OK	cdn.webperflab.com	126 B	74.63.46.206:80	61ms
GET sprite-non-repeat-large.png	200 OK	cdn.webperflab.com	163.1 KB	74.63.46.206:80	282ms
GET blank.gif	200 OK	cdn.webperflab.com	43 B	74.63.46.206:80	37ms
GET main.swf?624419	200 OK	webperflab.com	35.9 KB	173.45.227.33:80	146ms
GET mainApplication.xml	200 OK	webperflab.com	167 B	173.45.227.33:80	38ms
GET gameOne.swf	200 OK	webperflab.com	255.4 KB	173.45.227.33:80	477ms
GET application.xml	200 OK	webperflab.com	500 B	173.45.227.33:80	35ms

+354ms Request start time since the beginning

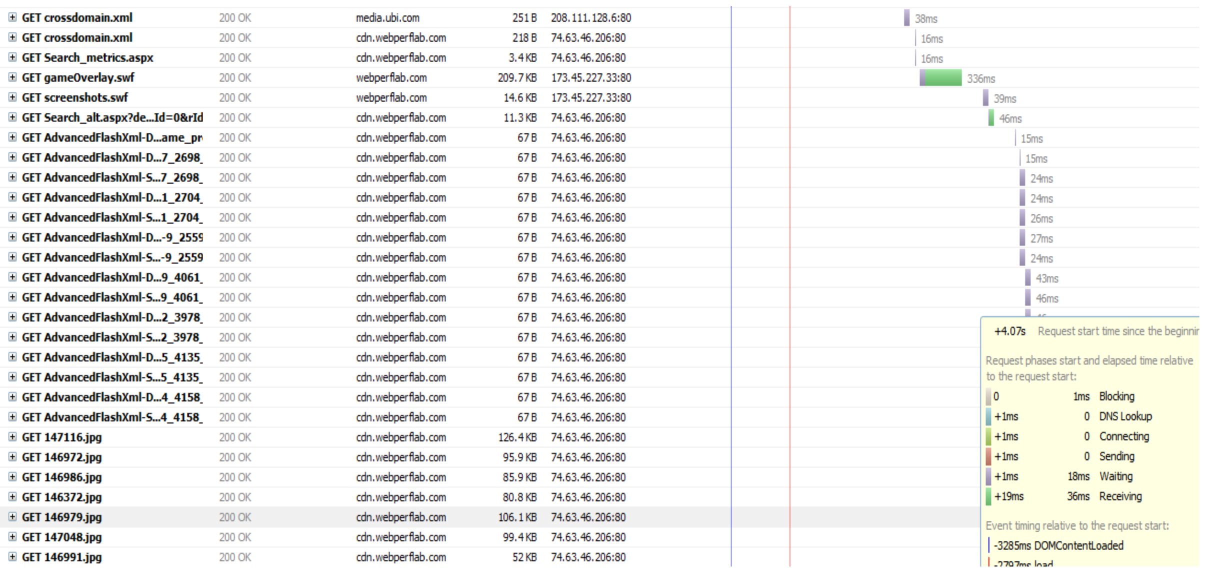
Request phases start and elapsed time relative to the request start:

0	7ms	DNS Lookup
+7ms	20ms	Connecting
+27ms	0	Sending
+27ms	26ms	Waiting
+53ms	184ms	Receiving

Event timing relative to the request start:

+428ms DOMContentLoaded

+916ms load



GET crossdomain.xml	200 OK	media.ubi.com	251 B	208.111.128.6:80	38ms
GET crossdomain.xml	200 OK	cdn.webperflab.com	218 B	74.63.46.206:80	16ms
GET Search_metrics.aspx	200 OK	cdn.webperflab.com	3.4 KB	74.63.46.206:80	16ms
GET gameOverlay.swf	200 OK	webperflab.com	209.7 KB	173.45.227.33:80	336ms
GET screenshots.swf	200 OK	webperflab.com	14.6 KB	173.45.227.33:80	39ms
GET Search_alt.aspx?de...Id=0&rId	200 OK	cdn.webperflab.com	11.3 KB	74.63.46.206:80	46ms
GET AdvancedFlashXml-D...ame_pr	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	15ms
GET AdvancedFlashXml-D...7_2698_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	15ms
GET AdvancedFlashXml-S...7_2698_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	24ms
GET AdvancedFlashXml-D...1_2704_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	24ms
GET AdvancedFlashXml-S...1_2704_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	26ms
GET AdvancedFlashXml-D...-9_2559	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	27ms
GET AdvancedFlashXml-S...-9_2559	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	24ms
GET AdvancedFlashXml-D...9_4061_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	43ms
GET AdvancedFlashXml-S...9_4061_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	46ms
GET AdvancedFlashXml-D...2_3978_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	
GET AdvancedFlashXml-S...2_3978_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	
GET AdvancedFlashXml-D...5_4135_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	
GET AdvancedFlashXml-S...5_4135_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	
GET AdvancedFlashXml-D...4_4158_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	
GET AdvancedFlashXml-S...4_4158_	200 OK	cdn.webperflab.com	67 B	74.63.46.206:80	
GET 147116.jpg	200 OK	cdn.webperflab.com	126.4 KB	74.63.46.206:80	
GET 146972.jpg	200 OK	cdn.webperflab.com	95.9 KB	74.63.46.206:80	
GET 146986.jpg	200 OK	cdn.webperflab.com	85.9 KB	74.63.46.206:80	
GET 146372.jpg	200 OK	cdn.webperflab.com	80.8 KB	74.63.46.206:80	
GET 146979.jpg	200 OK	cdn.webperflab.com	106.1 KB	74.63.46.206:80	
GET 147048.jpg	200 OK	cdn.webperflab.com	99.4 KB	74.63.46.206:80	
GET 146991.jpg	200 OK	cdn.webperflab.com	52 KB	74.63.46.206:80	

+4.07s Request start time since the beginnir

Request phases start and elapsed time relative to the request start:

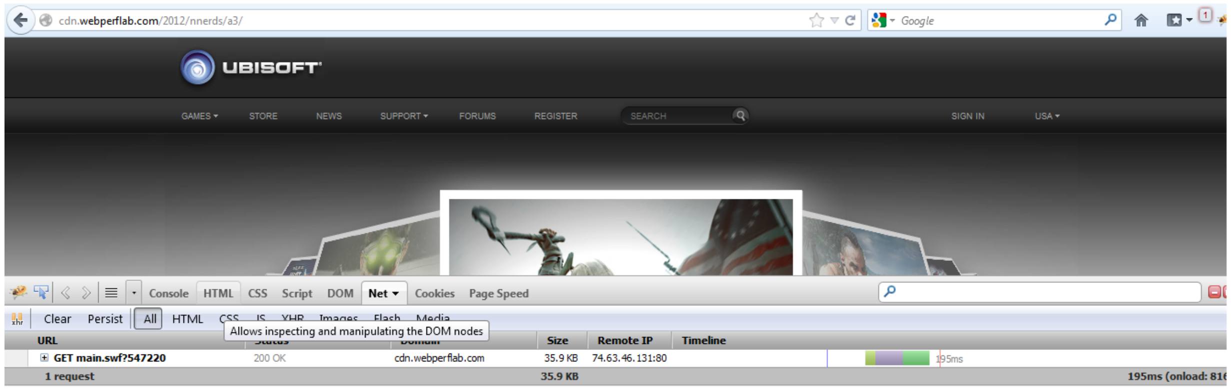
0	1ms	Blocking
+1ms	0	DNS Lookup
+1ms	0	Connecting
+1ms	0	Sending
+1ms	18ms	Waiting
+19ms	36ms	Receiving

Event timing relative to the request start:

-3285ms DOMContentLoaded

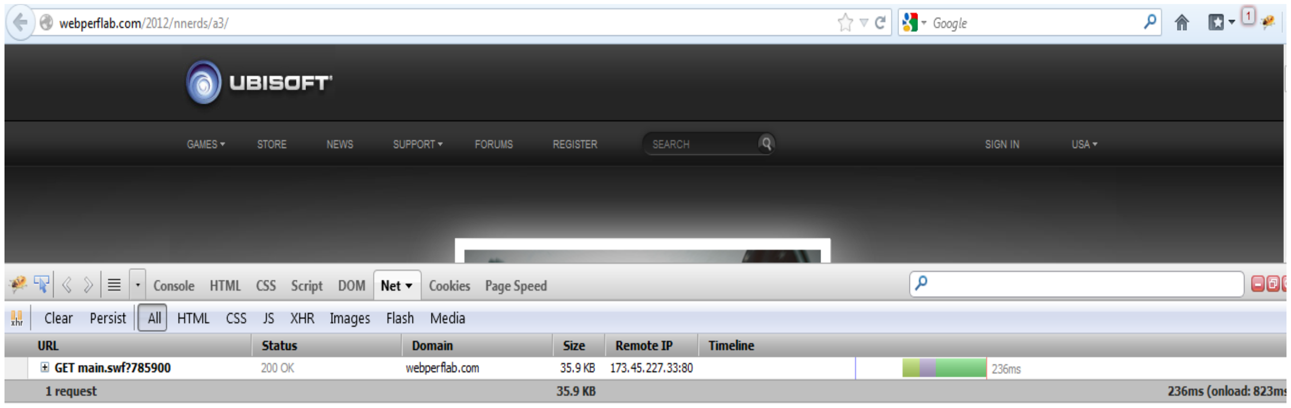
-2797ms load

In addition to the above setting, we made sure that the apache configuration for the website and cdn preserves deflate and expires properties similar to the one that existed on the original webserver during cycle2 (to recap, in cycle2 – we enabled the mod_expires and mod_available components for a whitelisted set of MIME types based on the nature of file and its extension type respectively). Since we did not have root priveleges to the apache configuration file, we incorporated these changes into .htaccess and placed this file under the site. Please find below relevant screenshot that shows caching success in cdn:


cdn.webperflab.com/2012/nnerds/a3/
Google
UBISOFT
GAMES STORE NEWS SUPPORT FORUMS REGISTER SEARCH SIGN IN USA
Console HTML CSS Script DOM Net Cookies Page Speed
Clear Persist All HTML CSS JS XHR Images Flash Media
Allows inspecting and manipulating the DOM nodes

URL	Status	Domain	Size	Remote IP	Timeline
GET main.swf?547220	200 OK	cdn.webperflab.com	35.9 KB	74.63.46.131:80	195ms
1 request			35.9 KB		195ms (onload: 81

Please see relevant screenshot below that shows caching success in webpeflab's a3 version:


webperflab.com/2012/nnerds/a3/
Google
UBISOFT
GAMES STORE NEWS SUPPORT FORUMS REGISTER SEARCH SIGN IN USA
Console HTML CSS Script DOM Net Cookies Page Speed
Clear Persist All HTML CSS JS XHR Images Flash Media

URL	Status	Domain	Size	Remote IP	Timeline
GET main.swf?785900	200 OK	webperflab.com	35.9 KB	173.45.227.33:80	236ms
1 request			35.9 KB		236ms (onload: 823ms

10. Minify Javascript

js minification removes comments, unnecessary characters and unwanted semicolons (like the ones right before }). In addition, we made use of the YUI compressor which also replaces all local symbols with one letter (or essentially symbol with lesser letters) symbols.

The tool makes sure that the local symbols are obfuscated only when appopriate (in cases when eval is used, the browser takes a defective approach by not replacing symbols that belong to relevant scope). This improves the page speed because the file size of the above mentioned components is reduced. The CSS minifier uses a compression algorithm that does the following:

I. strips comments (except special), whitespaces, last semicolon, extra semicolons, empty declarations, extra zeroes, measurement unit for zero values, leading zeroes

II. abbreviates color values (except in filters) and covert them to hex value equivalents, shortens none values

III. maintains the first charset entry

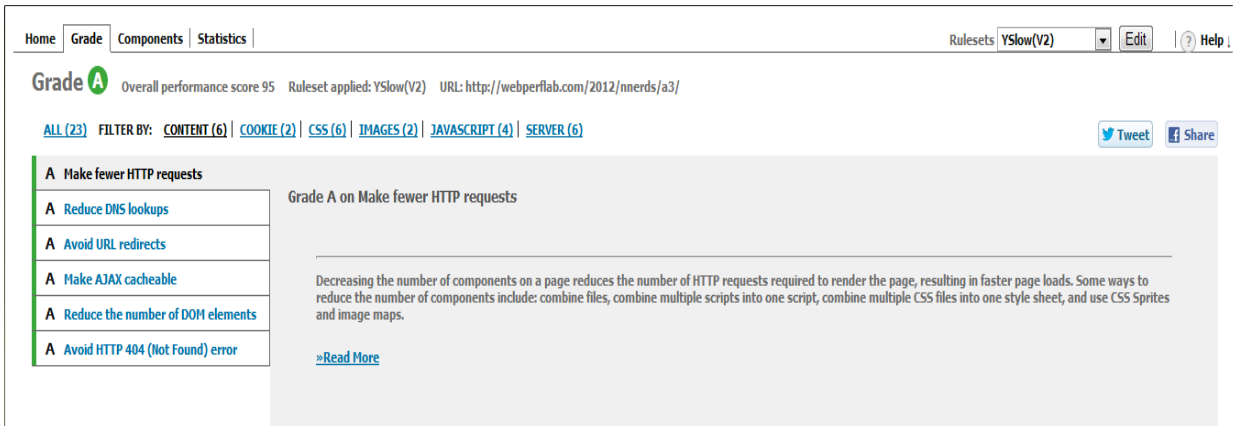
IV. shortens alpha opacity filter

The minifier makes sure that the underscore / star hack, child selector hack, IE5 / Mac hack and box model hack are not overridden.

We also made sure that the compressor did not gzip the final output further because the javascript would be compressed by the apache server's mod_deflate module. We chose to compress the javascript and stylesheets within the server rather than offline because this mechanism allows us to support those browsers that do not accept gzipped files as well since we added the vary header to .js and .css type files (this was done by whitelisting the MIME types related to javascript and stylesheets used by common browsers and their different versions).

11. Avoid Redirects

The external resource (image of flash player) was combined into the sprite image in one of the previous iterations. This got rid of the 301 redirect http response. We also ended up reorganizing the directory structure of the website – to remove deep directory levels. In addition, this also took care of the other redirect case in question – since the index.html was placed on the root (at a3). All files were organized on the basis of their type and put into respective folder (the css is placed directly on root because it is a single file, so is portalsearch.html being the html file other than index.html) so that the web developer also finds it easy to reference these objects. In addition, a shallow directory structure allows better search ranking of the website. The end-users also hence end up generally clicking on shorter urls for websites that appear on the search results page. The server can also navigate through a shallow directory structure to grab files and serve the client much easier than the previous one. After rearranging the files into various folders, all the 404s on the website were removed. Please find below relevant screenshot that shows the YSlow screen for the corresponding rule (You can also see that we fixed all the 404-s):



Grade A Overall performance score 95 Ruleset applied: YSlow(V2) URL: http://webperflab.com/2012/nnerds/a3/

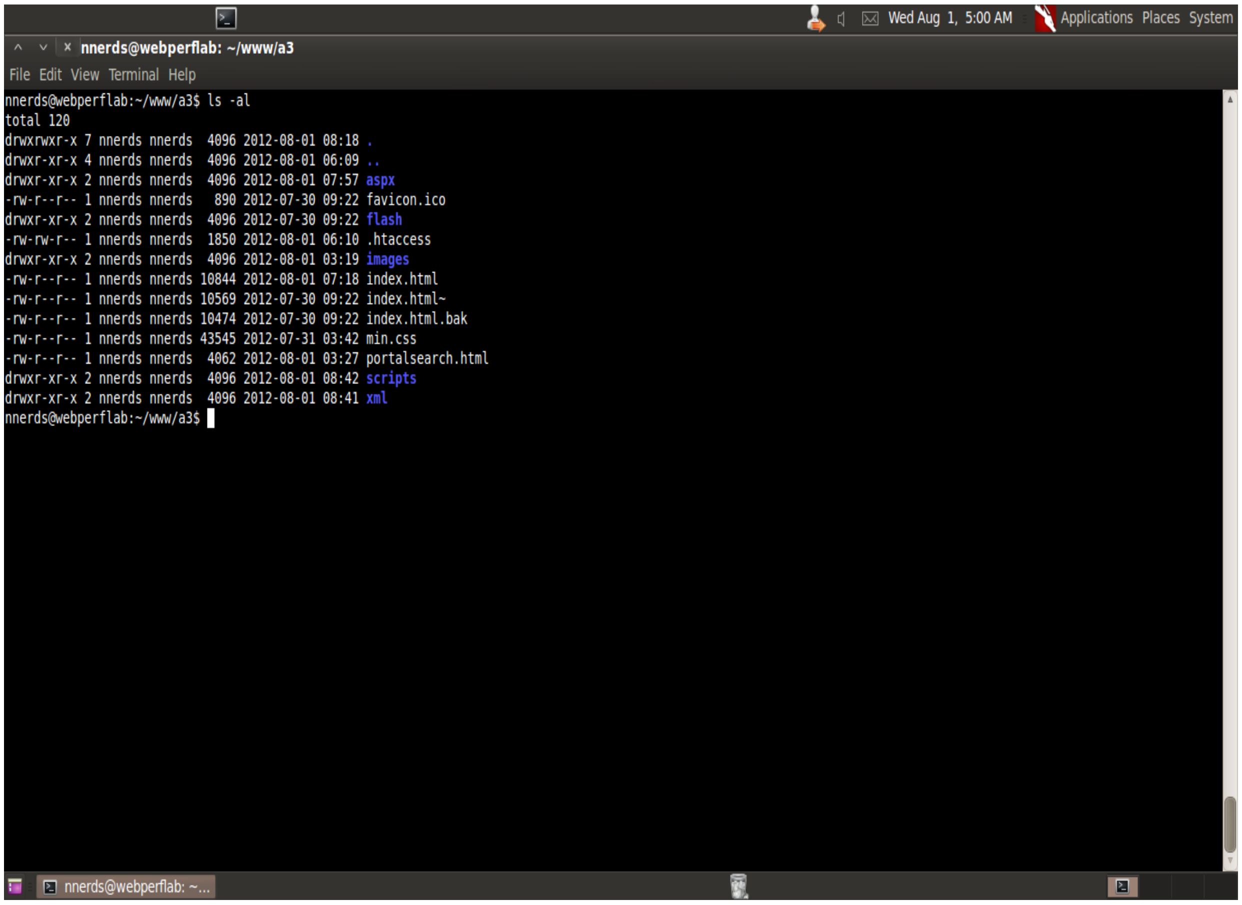
ALL (23) FILTER BY: CONTENT (6) | COOKIE (2) | CSS (6) | IMAGES (2) | JAVASCRIPT (4) | SERVER (6)

A Make fewer HTTP requests
A Reduce DNS lookups
A Avoid URL redirects
A Make AJAX cacheable
A Reduce the number of DOM elements
A Avoid HTTP 404 (Not Found) error

Grade A on Make fewer HTTP requests

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

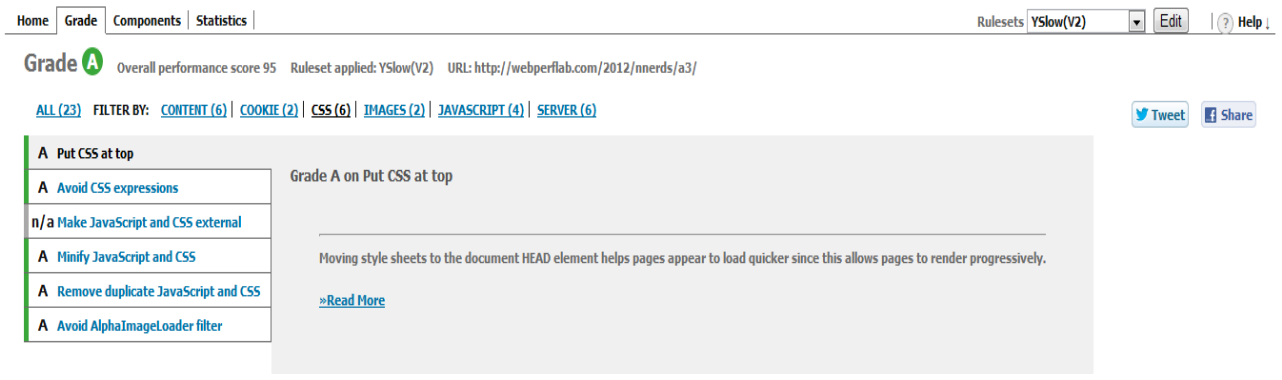
Below is also a description of the website's directory structure:


Wed Aug 1, 5:00 AM
Applications Places System
nnerds@webperflab: ~/www/a3
File Edit View Terminal Help
nnerds@webperflab: ~...

```
nnerds@webperflab:~/www/a3$ ls -al
total 120
drwxrwxr-x 7 nnerds nnerds  4096 2012-08-01 08:18 .
drwxr-xr-x 4 nnerds nnerds  4096 2012-08-01 06:09 ..
drwxr-xr-x 2 nnerds nnerds  4096 2012-08-01 07:57 aspx
-rw-r--r-- 1 nnerds nnerds   890 2012-07-30 09:22 favicon.ico
drwxr-xr-x 2 nnerds nnerds  4096 2012-07-30 09:22 flash
-rw-rw-r-- 1 nnerds nnerds  1850 2012-08-01 06:10 .htaccess
drwxr-xr-x 2 nnerds nnerds  4096 2012-08-01 03:19 images
-rw-r--r-- 1 nnerds nnerds 10844 2012-08-01 07:18 index.html
-rw-r--r-- 1 nnerds nnerds 10569 2012-07-30 09:22 index.html~
-rw-r--r-- 1 nnerds nnerds 10474 2012-07-30 09:22 index.html.bak
-rw-r--r-- 1 nnerds nnerds 43545 2012-07-31 03:42 min.css
-rw-r--r-- 1 nnerds nnerds  4062 2012-08-01 03:27 portalsearch.html
drwxr-xr-x 2 nnerds nnerds  4096 2012-08-01 08:42 scripts
drwxr-xr-x 2 nnerds nnerds  4096 2012-08-01 08:41 xml
nnerds@webperflab:~/www/a3$
```

<u>Extra Optimization</u>

All Alpha Image Loader Filters were removed from the css. This filter blocks rendering as the image is getting downloaded. This option also increases the memory consumption at the client side: additionally because it is applied per element and not per image. This filter was essntially put in place to fix a problem with semi-transparent true color PNGs for IE < 7. The filters were hence deemed unncessarily increasing the size of the stylesheets since the website will generally be run on the latest version of the browsers. In cases where the filter is absolutely essential, the PNG8 alternatives can be used (this change will be dealt with, in the future cycle rule for optimisation of images). Please find below relevant screenshot:



Grade A Overall performance score 95 Ruleset applied: YSlow(V2) URL: http://webperflab.com/2012/nnerds/a3/

ALL (23) FILTER BY: CONTENT (6) | COOKIE (2) | CSS (6) | IMAGES (2) | JAVASCRIPT (4) | SERVER (6)

A	Put CSS at top
A	Avoid CSS expressions
n/a	Make JavaScript and CSS external
A	Minify JavaScript and CSS
A	Remove duplicate JavaScript and CSS
A	Avoid AlphaImageLoader filter

Grade A on Put CSS at top

Moving style sheets to the document HEAD element helps pages appear to load quicker since this allows pages to render progressively.

»Read More