Bottlenecks and performance optimization in web application

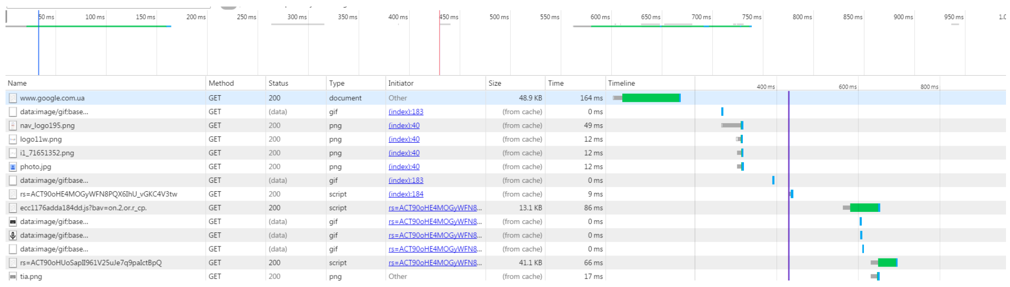
Bottlenecks

In current time, 80% problems of the causes slow performance occurs on the client side.

To reduction performance problems use this rules

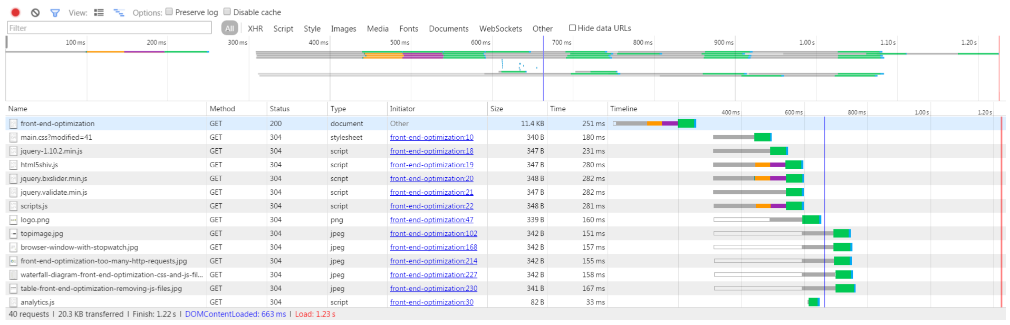
1. Minimize number of request to server.

   Good:



Name	Method	Status	Type	Initiator	Size	Time	Timeline
www.google.com.ua	GET	200	document	Other	48.9 KB	164 ms	
data:image/gif;base...	GET	(data)	gif	(index):183	(from cache)	0 ms	
nav_logo195.png	GET	200	png	(index):40	(from cache)	49 ms	
logo11w.png	GET	200	png	(index):40	(from cache)	12 ms	
i1_71651352.png	GET	200	png	(index):40	(from cache)	12 ms	
photo.jpg	GET	200	png	(index):40	(from cache)	12 ms	
data:image/gif;base...	GET	(data)	gif	(index):183	(from cache)	0 ms	
rs=ACT90oHE4MOGyWFN8PQX6IhU_vGKC4V3tw	GET	200	script	(index):184	(from cache)	9 ms	
ecc1176adda184dd.js?bav=on.2,or.r_cp.	GET	200	script	rs=ACT90oHE4MOGyWFN8...	13.1 KB	86 ms	
data:image/gif;base...	GET	(data)	gif	rs=ACT90oHE4MOGyWFN8...	(from cache)	0 ms	
data:image/gif;base...	GET	(data)	gif	rs=ACT90oHE4MOGyWFN8...	(from cache)	0 ms	
data:image/gif;base...	GET	(data)	gif	rs=ACT90oHE4MOGyWFN8...	(from cache)	0 ms	
rs=ACT90oHUoSapII961V25uJe7q9paIctBpQ	GET	200	script	rs=ACT90oHE4MOGyWFN8...	41.1 KB	66 ms	
tia.png	GET	200	png	Other	(from cache)	17 ms	

   Bad:



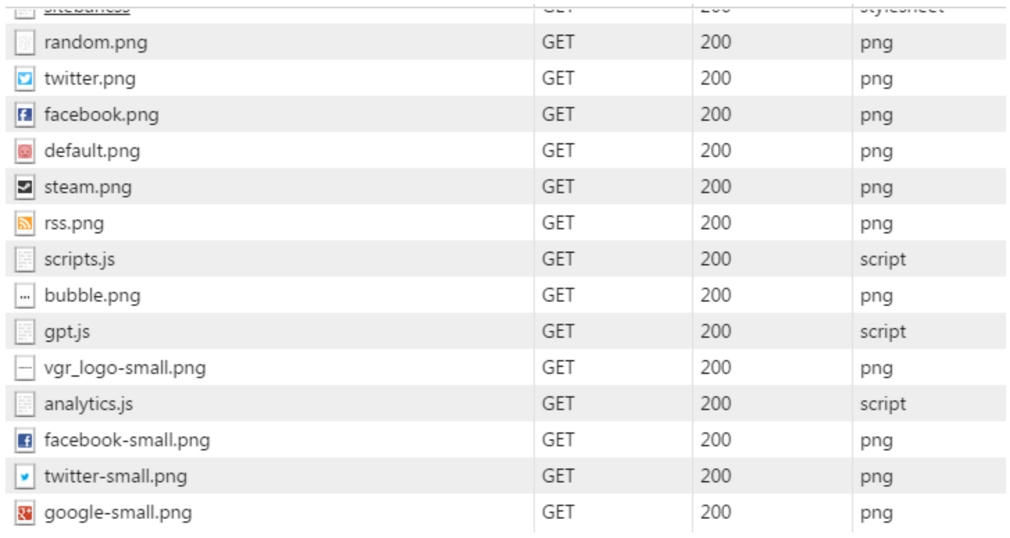
Name	Method	Status	Type	Initiator	Size	Time	Timeline
front-end-optimization	GET	200	document	Other	11.4 KB	251 ms	
main.css?modified=41	GET	304	stylesheet	front-end-optimization:10	340 B	180 ms	
jquery-1.10.2.min.js	GET	304	script	front-end-optimization:18	347 B	231 ms	
html5shiv.js	GET	304	script	front-end-optimization:19	347 B	280 ms	
jquery.bxslider.min.js	GET	304	script	front-end-optimization:20	348 B	282 ms	
jquery.validate.min.js	GET	304	script	front-end-optimization:21	347 B	282 ms	
scripts.js	GET	304	script	front-end-optimization:22	348 B	281 ms	
logo.png	GET	304	png	front-end-optimization:47	339 B	160 ms	
topimage.jpg	GET	304	jpeg	front-end-optimization:102	342 B	151 ms	
browser-window-with-stopwatch.jpg	GET	304	jpeg	front-end-optimization:168	342 B	157 ms	
front-end-optimization-too-many-http-requests.jpg	GET	304	jpeg	front-end-optimization:214	342 B	155 ms	
waterfall-diagram-front-end-optimization-css-and-js-fil...	GET	304	jpeg	front-end-optimization:227	342 B	158 ms	
table-front-end-optimization-removing-js-files.jpg	GET	304	jpeg	front-end-optimization:230	341 B	167 ms	
analytics.js	GET	304	script	front-end-optimization:30	82 B	33 ms	

40 requests | 20.3 KB transferred | Finish: 1.22 s | DOMContentLoaded: 663 ms | Load: 1.23 s

   Reduction requests leads to increased productivity, because browsers limit the number of concurrent requests a site.

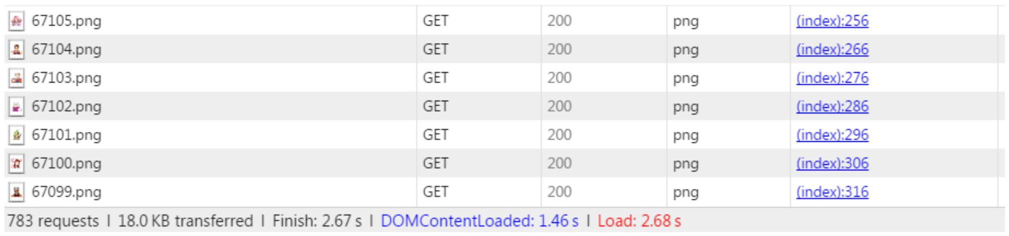
2. Compress images.

   Good:



random.png	GET	200	png
twitter.png	GET	200	png
facebook.png	GET	200	png
default.png	GET	200	png
steam.png	GET	200	png
rss.png	GET	200	png
scripts.js	GET	200	script
bubble.png	GET	200	png
gpt.js	GET	200	script
vgr_logo-small.png	GET	200	png
analytics.js	GET	200	script
facebook-small.png	GET	200	png
twitter-small.png	GET	200	png
google-small.png	GET	200	png

   Bad:



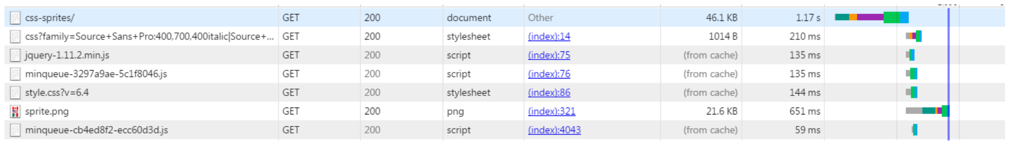
67105.png	GET	200	png	(index):256
67104.png	GET	200	png	(index):266
67103.png	GET	200	png	(index):276
67102.png	GET	200	png	(index):286
67101.png	GET	200	png	(index):296
67100.png	GET	200	png	(index):306
67099.png	GET	200	png	(index):316

783 requests | 18.0 KB transferred | Finish: 2.67 s | DOMContentLoaded: 1.46 s | Load: 2.68 s

Compress images to resolution, in which it will be displayed. If you need to display images on mobile and desktop, create two version of image.

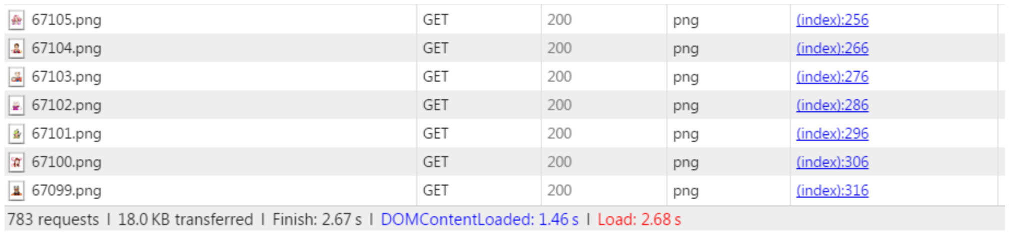
3. use less pictures.

Good:



css-sprites/	GET	200	document	Other	46.1 KB	1.17 s
css?family=Source+Sans+Pro:400,700,400italic|Source+...	GET	200	stylesheet	(index):14	1014 B	210 ms
jquery-1.11.2.min.js	GET	200	script	(index):75	(from cache)	135 ms
minqueue-3297a9ae-5c1f8046.js	GET	200	script	(index):76	(from cache)	135 ms
style.css?v=6.4	GET	200	stylesheet	(index):86	(from cache)	144 ms
sprite.png	GET	200	png	(index):321	21.6 KB	651 ms
minqueue-cb4ed8f2-ecc60d3d.js	GET	200	script	(index):4043	(from cache)	59 ms

Bad:



67105.png	GET	200	png	(index):256
67104.png	GET	200	png	(index):266
67103.png	GET	200	png	(index):276
67102.png	GET	200	png	(index):286
67101.png	GET	200	png	(index):296
67100.png	GET	200	png	(index):306
67099.png	GET	200	png	(index):316

783 requests | 18.0 KB transferred | Finish: 2.67 s | DOMContentLoaded: 1.46 s | Load: 2.68 s

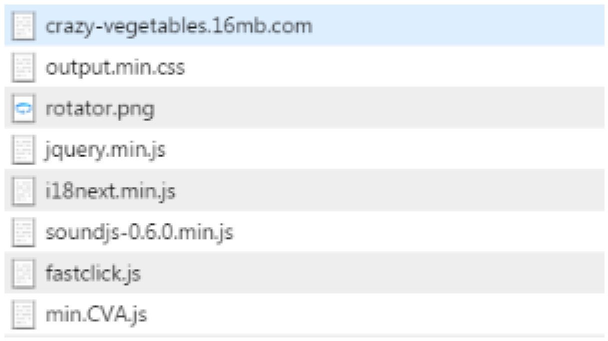
To reducing number of pictures use sprite.

4. Minimize weight JS and CSS files, combine all files(JS and CSS) in one.

Example:

Example of Gruntfile

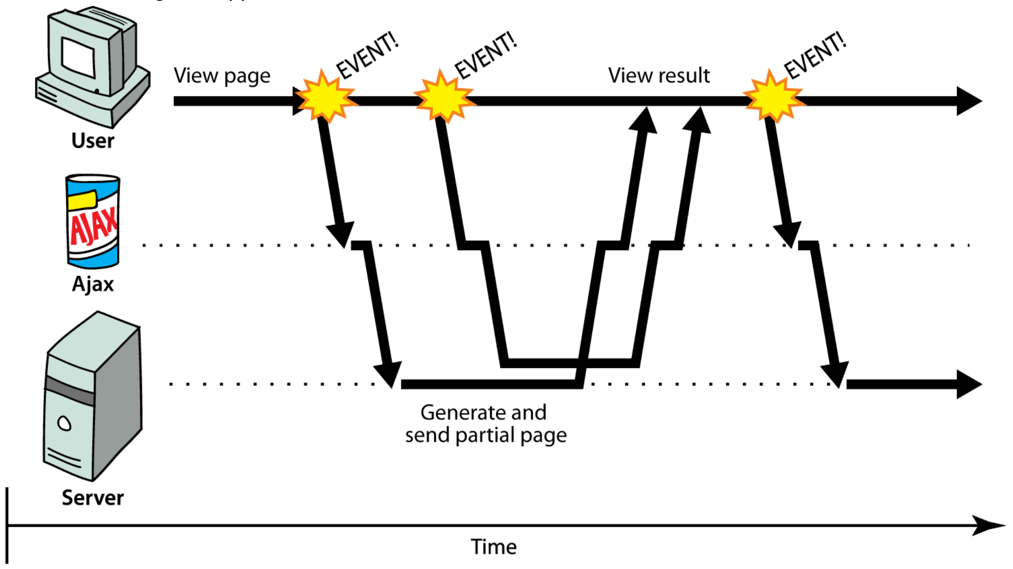
```
cssmin: {
      options: {
        shorthandCompacting: true,
        roundingPrecision: -1
      },
      target: {
        files: {
          '../build/css/output.min.css': ['../src/css/backgroundKeyframes.css',
'../src/css/style.css', '../src/css/sprites.css']
        }
      }
    },
```



crazy-vegetables.16mb.com
output.min.css
rotator.png
jquery.min.js
i18next.min.js
soundjs-0.6.0.min.js
fastclick.js
min.CVA.js

For minify CSS, you can try YUI Compressor and cssmin.js. For minify JavaScript, try Closure Compiler, JSMin or the YUI Compressor.

5. Use asynchronous requests.
   It`s make loading web application faster.



6. Do not use css @import.

Don't do like that

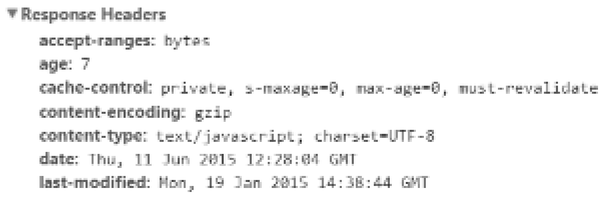
```
@import url(style1.css);
@import url(main.css);
@import url(reset.css)
```

   This leads to additional request in different css files.
7. Use data uri for small images.

Data uri

```
<img
src="data:image/gif;base64,R0lGODdhMAAwAPAAAAAAAP///ywAAAAAMAAw
AAAC8IyPqcvt3wCcDkiLc7C0qwyGHhSWpjQu5yqmCYsapyuvUUlvONmOZtfzgFz
ByTB10QgxOR0TqBQejhRNzOfkVJ+5YiUqrXF5Y5lKh/DeuNcP5yLWGsEbtLiOSp
a/TPg7JpJHxyendzWTBfX0cxOnKPjgBzi4diinWGdkF8kjdfnycQZXZeYGejmJl
ZeGl9i2icVqaNVailT6F5iJ90m6mvuTS4OK05M0vDk0Q4XUtwvKOzrcd3iq9uis
F81M1OIcR7lEewwcLp7tuNNkM3uNna3F2JQFo97Vriy/Xl4/f1cf5VWzXyym7PH
hhx4dbgYKAAA7"
alt="Larry" />
```

   Data uri include pictures on the page in form base64 code
8. Less DOM change action.
   DOM originally not intended for dynamic changes, so if you can, avoid DOM changing.
9. Enable gzip compression.



```
▼Response Headers
    accept-ranges: bytes
    age: 7
    cache-control: private, s-maxage=0, max-age=0, must-revalidate
    content-encoding: gzip
    content-type: text/javascript; charset=UTF-8
    date: Thu, 11 Jun 2015 12:28:04 GMT
    last-modified: Mon, 19 Jan 2015 14:38:44 GMT
```

Browsers read compressed pages like normal, but weight of it page less on 60-80% than uncompressed.

10. Less dependencies from frameworks and libraries.
    If you take frameworks for the one feature, you did something wrong.
    Bad:

Registration in project after professional team

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.15/angular.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/angular_material/0.9.4/angular-material.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquerymobile/1.4.5/jquery.mobile.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/spf/2.2.0/spf.js"></script>
```

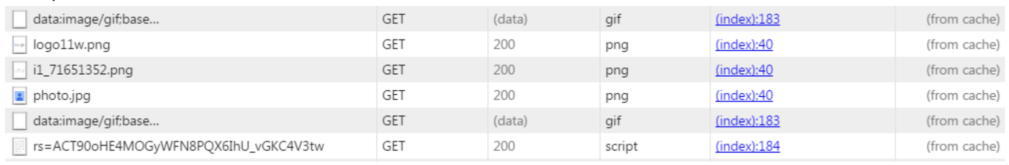
11. Use CDN and JS fallback.

CDN and fallback

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
<script>window.jQuery || document.write('<script
src="js/lib/jquery-2.1.3.min.js"><\/script>')</script>
```

CDN are reduces load on server, and JS fallback insures against falling server with CDN.

12. Use browser cache.
    Example:



data:image/gif;base...	GET	(data)	gif	(index):183	(from cache)
logo11w.png	GET	200	png	(index):40	(from cache)
i1_71651352.png	GET	200	png	(index):40	(from cache)
photo.jpg	GET	200	png	(index):40	(from cache)
data:image/gif;base...	GET	(data)	gif	(index):183	(from cache)
rs=ACT90oHE4MOGyWFN8PQX6IhU_vGKC4V3tw	GET	200	script	(index):184	(from cache)

It will not load the same files after each reloading.

13. Use less animation.
    Animation is quite costly process so if you can, do not waste CPU power to render cyclical animation . Also use smaller images, it is reduce number of pixel to paint.
14. Do not use few canvas.
    Often in articles recommended use 2 canvas to reduce loading. But on devices with older equipment it's doesn't works well.

Performance optimization

For optimization you code, at first, you should found the problems. This list of useful programs will help you with finding:

1. Site speed.
   Site speed of google analyzes you site for the presence most popular bottlenecks.
2. Chrome developer tools.
   It`s hard to explained how useful this tools(Chrome DevTools Overview). You clicked F12 and got almost all you need for testing(profiles which find memory leaks and show CPU loading, timeline which show how browser displays, step by step, how your code transformation in picture, and many other tools).
3. Xcode instruments.
   Instrument is one of parts of Xcode, what designed for testing and debugging your application on any Apple deviceInstrument include tools for CPU, GPU, leak and memory test.
4. IE6 css fixer.
   Destination this tool is simplify debugging for ie6.
5. IE tester
   It allows you to install several versions IE for testing.
6. Browser shots
   Shots you application view in different browsers.
7. Charles web proxy
   With Charles you can imitate connection, requests to test different inputs, view XML and JSON requests and responses and many other thinks.