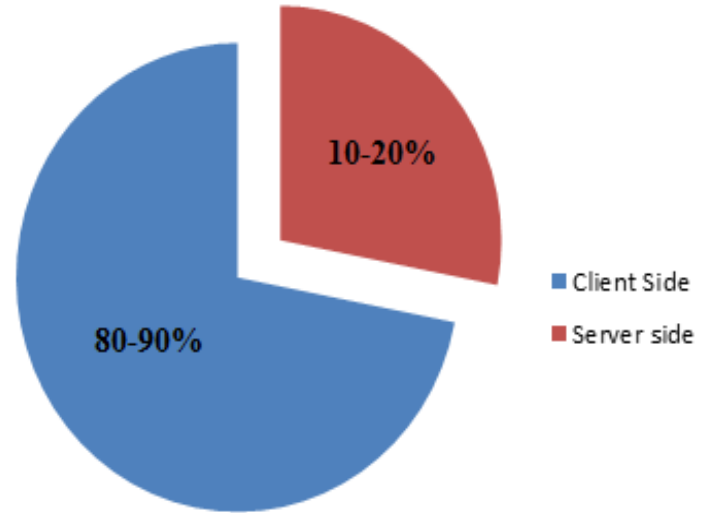


Web Application Optimization

- A web application performance can be improved at two levels.
 - ☐ Back-end/Server side
 - ☐ Front-end/Client side
- Back-end optimization involves upgrading servers and others hardware resources to handle more requests. But it's not the end of world in web application's performance optimization.

Importance of Front End Optimization

- **Client side performance issues are more critical from performance perspectives because they have more impact on user experience.**
- **Front-end performance optimization is quite simple and cost effective as compared to back-end performance optimization where redesigning application architecture, adding or modifying hardware etc. is required.**

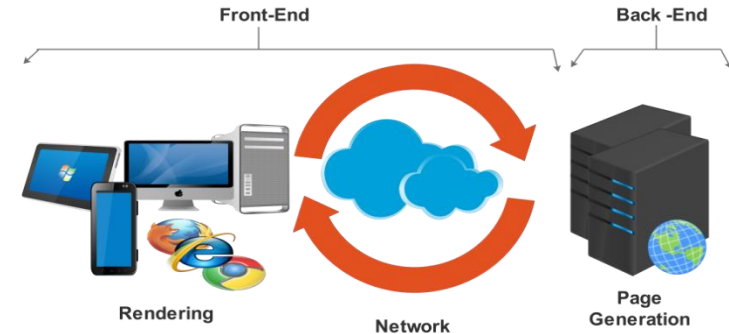


A study at Yahoo found that on average only 10-20% of total page loading time is spent on the back-end and other 80-90% time is spent on the front-end.

Note : "How fast does this page load ?" from a **single user point of view.**

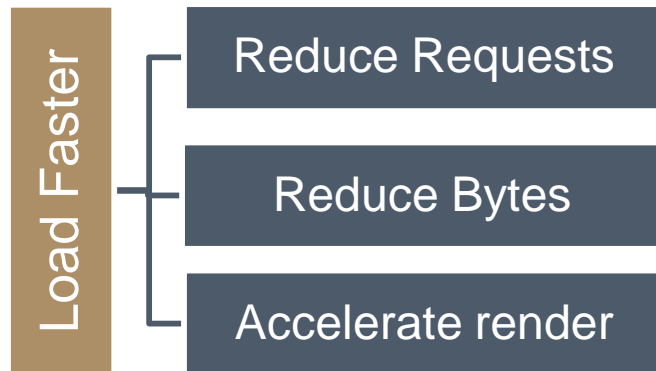
Web Application/Website Front –End

- Front-end or client-side is **user interface** or that particular part of an application (Web-app/Website or software) that user views on his/her screen.
- In web applications/site client-side contains web browser. The browser responsibility is to communicate with the web server over the HTTP protocol, **rendering the UI of the web application and allowing user inputs.**
- In web applications/site the user interface is generally **rendered as an HTML document. Afterward scripts are executed, style sheets are processed** and other contents are rendered to the user.



What is FEO(Front End Optimization)

- Front end optimization (FEO), also known as **content optimization**, is the process of fine-tuning the Webapp/website to make it more browser-friendly and quicker to load.
- FEO focuses on reducing file sizes and minimizing the number of requests needed for a given page to load.



How to Reduce Requests

| CSS | JS | IMAGES |
|--|----|----------------------|
| Bundle | | Sprites |
| Don't use 3rd parties | | Embedded |
| | | Replace with CSS |
| | | Don't use duplicates |
| Use only what is really needed for particular page | | |
| | | |

How to Reduce Bytes

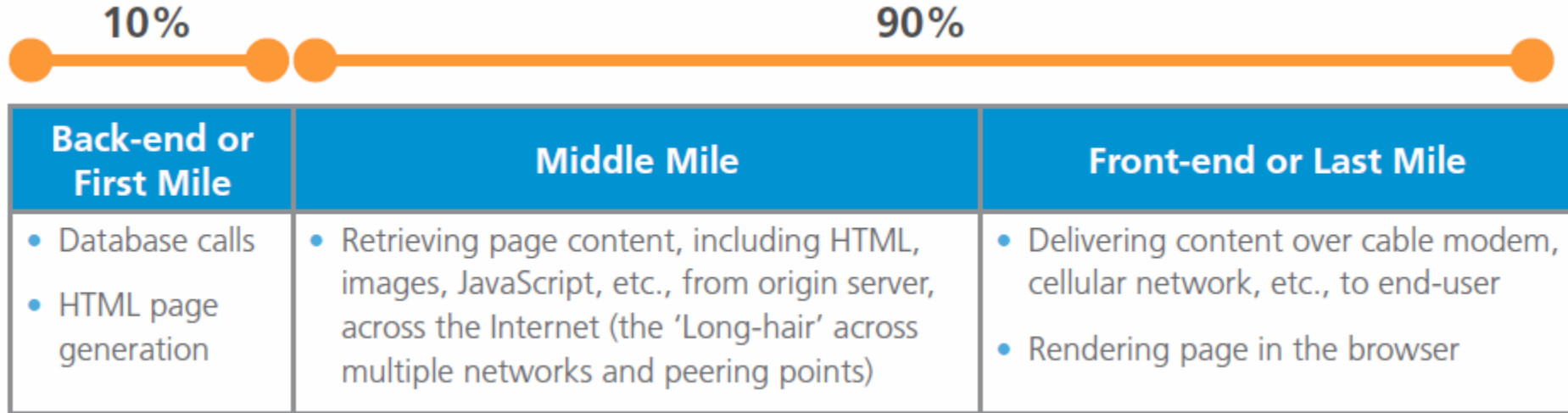
| CSS | JS | HTML | IMAGES |
|-----|--|------|---------------------------------------|
| | Minify | | Format and size |
| | | | Use SVG when possible |
| | | | Minify SVG (Scalable Vector Graphics) |
| | | | Lossless compression |
| | Use only what is really needed for particular page | | |
| | | | |

How to Accelerate Render

| CSS | JS | HTML | IMAGES |
|--|-----------------|----------|----------------------|
| | Minify | | Use appropriate size |
| First view | Less code | Validate | Progressive JPEG |
| | Frontend render | | |
| | Async load | | |
| Use only what is really needed for particular page | | | |
| | | | |

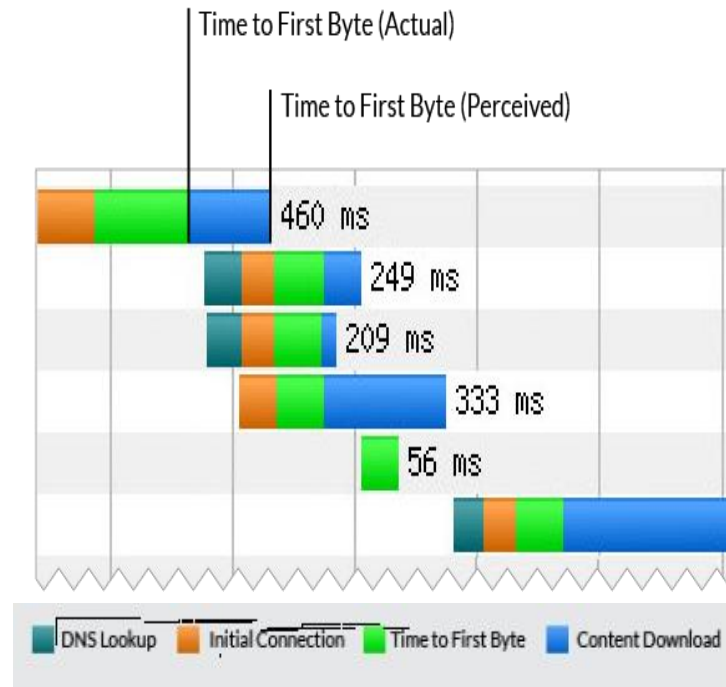
Web-app/Web site page load time

- Web-app /Website page load time breaks down as follows:



Time to First Byte (TTFB)

- TTFB often used to measure a web application/site response time.
- From **an actual load time perspective**, TTFB is the duration it takes for the first data byte to arrive from a server to the requesting browser.
- From **a perceived load time perspective**, TTFB is the duration it takes for the browser to parse the first byte after downloading the HTML file.

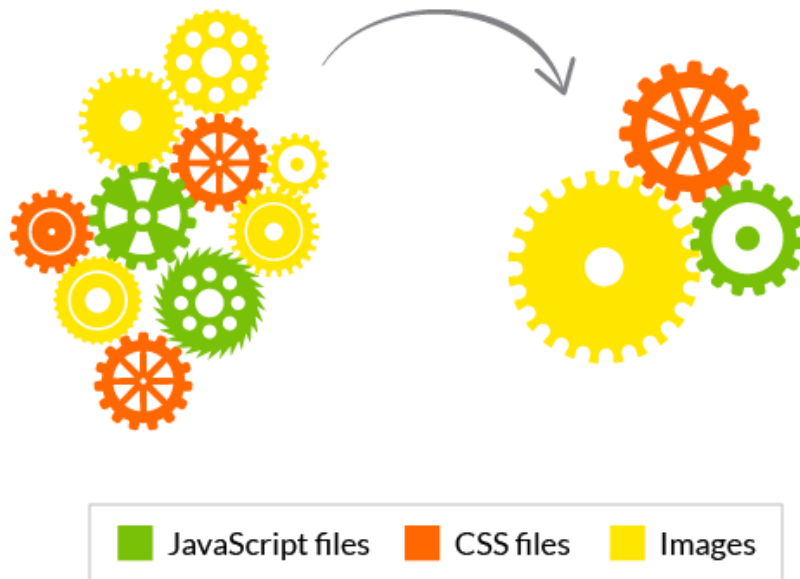


Only perceived TTFB impacts user experience, making it the more valuable of the two metrics.

Front-end Performance Optimization Techniques

Minimize HTTP Requests

- An HTTP request is used to fetch root HTML document that may refer to other page resources like images, scripts and style sheets.
- Each of these resources must be fetched with every HTTP request.
- Every HTTP request adds performance overhead as it creates network traffic between the client and server.



Front-end Performance Optimization Techniques

Minimize HTTP Requests



Front-end Performance Optimization Techniques

FILE COMPRESSION

- Webapp/Website consist of a collection of HTML, JavaScript, CSS and (possibly) other code files.
- With file compression, these files can be shrunk to a fraction of their original size to improve site responsiveness.
- Gzip is the most popular file compression choice. It can shrink a code file by as much as 60 or even 80 percent.



Front-end Performance Optimization Techniques

FILE COMPRESSION

Uncompressed
File



File Size: 500kb

Gzip Compressed
File



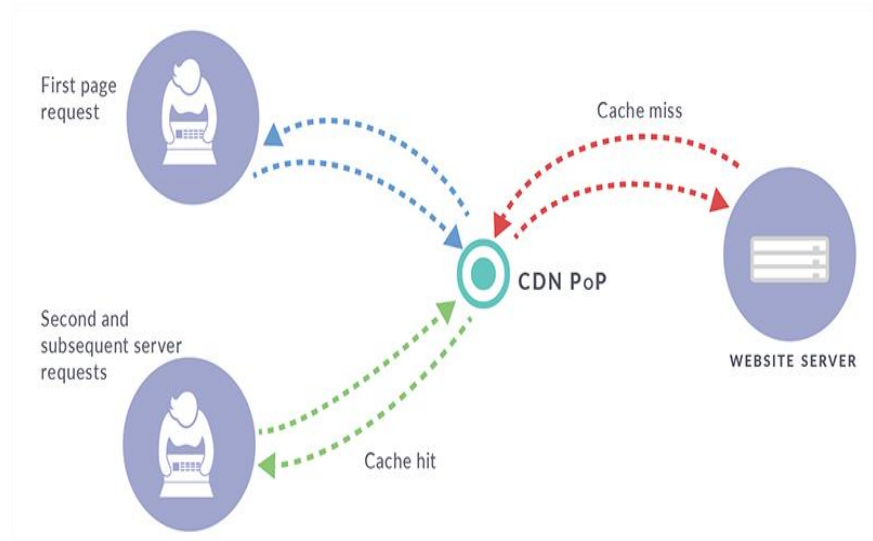
File Size: 200kb

| | Compression | Encode time | Decode time |
|-------|---------------|-------------|-------------|
| GZIP | 0.48 Mb (29%) | 0.79 sec | 1.04 sec |
| LZMA | 0.47 Mb (28%) | 1.26 sec | 1.03 sec |
| LPAQ | 0.35 Mb (21%) | 10.52 sec | 10.86 sec |
| BZIP2 | 0.45 Mb (27%) | 2.1 sec | 1.32 sec |

Front-end Performance Optimization Techniques

CACHE OPTIMIZATION

- Caching is storing the static files, which tend to be your largest ones, outside of your server—either on visitors' local drives or a nearby **CDN(Content Delivery Networks) PoP(Points of Presence)** .
- This can vastly improve the website's load speed.



Front-end Performance Optimization Techniques

CACHE OPTIMIZATION

Without Cached Content



With Cached Content



Front-end Performance Optimization Techniques

CODE MINIFICATION

- Minification techniques refer to process of removing unnecessary characters within a file like comments, new line commands, meta data, white spaces, new line commands etc. from web page source, js and css files without impacting the functionality.
- By removing the additional sources web page size is reduced and its download time as well. The minified file version provides the same functionality while reducing the bandwidth of network requests.
- ✓ **Minifying and gzipping code, combining both methods offers the best results. Thus, minifying files before gzip them will shrink the file size by and additional 5 to 10 percent.**

Front-end Performance Optimization Techniques

CODE MINIFICATION

Before Minification (201 characters)

```
1  /* I'm Just a Code Comment on Minification Example */  
2  Var minifyExample = function () {  
3      fill(0, 0, 0);  
4      text("Minification makes code smaller without changing its behavior", 100, 100);  
5  };  
6  
7  minifyExample();
```

After Minification (137 Characters) = File sized decreased by over 30%

```
1  Var minifyExample=function(){fill(0,0,0);text("Minification makes code smaller without changing its  
2  behavior", 100,100);};minifyExample();
```

Front-end Performance Optimization Techniques

IMAGE OPTIMIZATION

- Caching and compression are the two most common image optimization methods, with caching being the more effective of the two.
- For reduce image file size, image's data has to tamper, either by removing some of the header information or by reducing the original image quality. This is known as **lossy compression**.



Before Compression: 72kb



After Compression: 14.7kb

Front-end Performance Optimization Techniques

IMAGE OPTIMIZATION

- Another image optimization technique is to replace some of your regular (raster) images with their vector counterparts.

Why use vector images:

- They are very small in size, hold data for a set of coordinates—not for each individual pixel.
- Being resolution-independent

Front-end Performance Optimization Techniques

IMAGE OPTIMIZATION

Vector
(5.1kb @ any resolution)



Magnified



Raster
(19.8kb @ 200x200 px)



Magnified



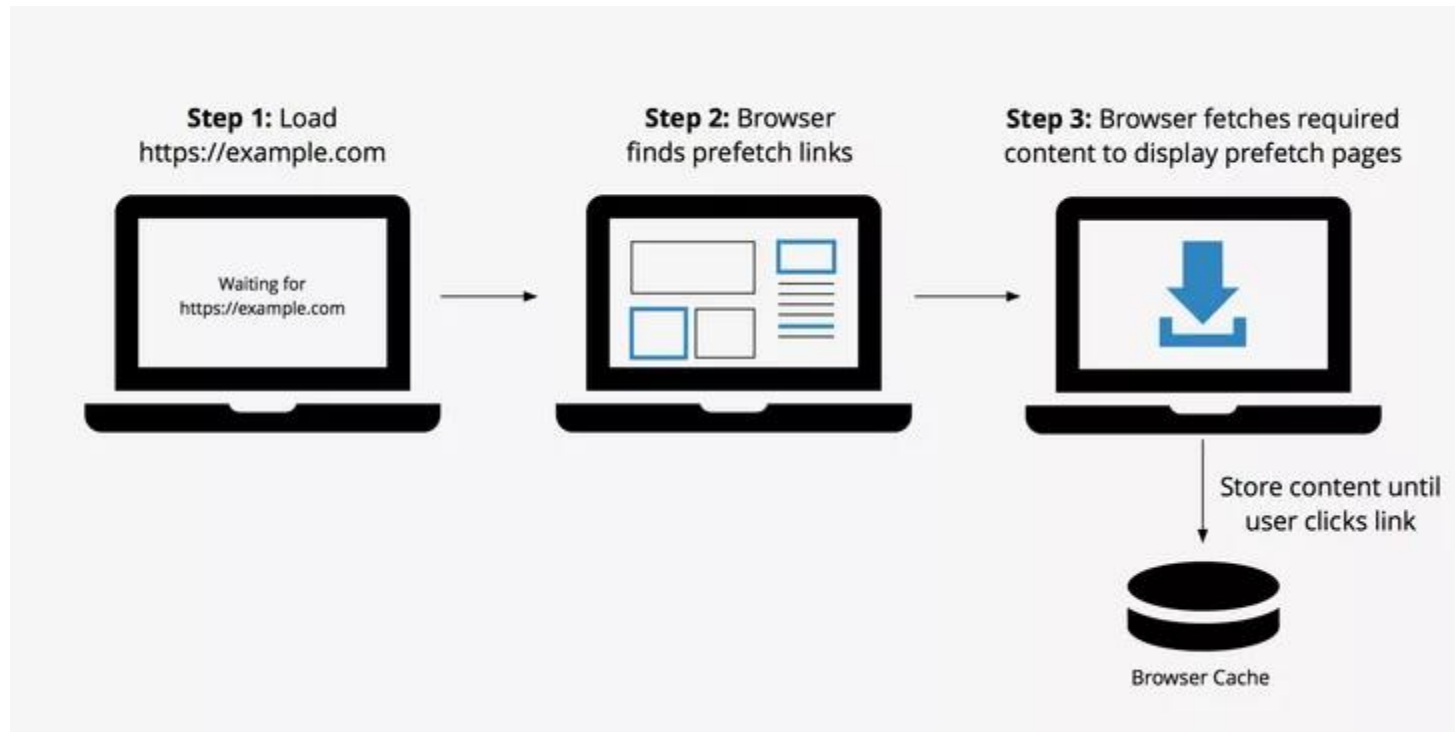
Front-end Performance Optimization Techniques

ENABLE PREFETCHING

- Prefetching can improve web application/site users' experience by fetching necessary resources and related data before they are needed. There are 3 main types of prefetching:
 - **Link Prefetching** used when certain that a specific resource will be required in the future, then we can ask the browser to request that item and store it in the cache for reference later. For example an image or a script
 - **DNS Prefetching** notifies the client that there are assets needed later from a specific URL so the browser can resolve the DNS as quickly as possible.
 - **Prerendering** gives us the ability to preemptively load all of the assets of a certain document
- Depending upon the type of prefetching we want to enable, simply add the ***rel="prefetch"***, ***rel="dns-prefetch"***, or ***rel="prerender"*** tag to the link attributes within your website's HTML

Front-end Performance Optimization Techniques

ENABLE PREFETCHING



Front-end Performance Optimization Techniques

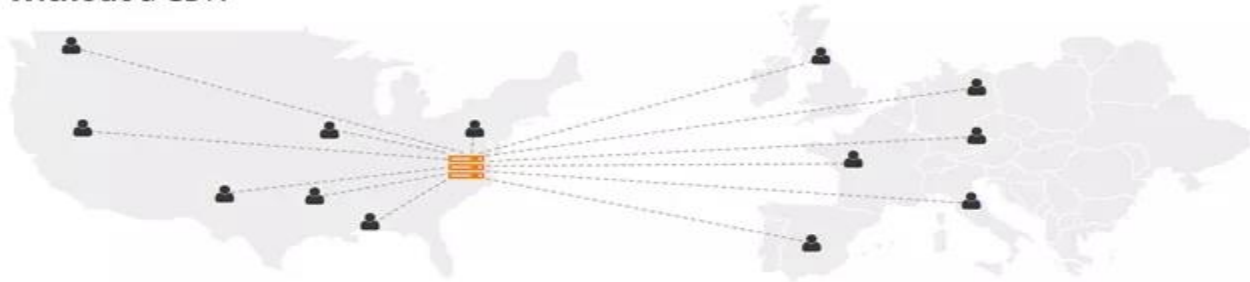
USE A CONTENT DELIVERY NETWORK

- A content delivery network (CDN) is a collection of web servers distributed across various locations to provide web contents in an efficient manner.
- Based on less number of network hop counts, user request should be entertained from the closest web server.
- User response time can be greatly improved by just distributing static web contents on various locations instead of starting from the difficult task of redesigning the application to distribute the dynamic contents.

Front-end Performance Optimization Techniques

USE A CONTENT DELIVERY NETWORK

Without a CDN



With a CDN



Front-end Performance Optimization Techniques

- Put Style Sheets at the Top
- Put Scripts at the Bottom
- Avoid CSS Expression
- Make JavaScript and CSS External
- Avoid Redirects
- Remove Duplicate Scripts
- Turnoff Entity Tag

Front-End Optimization Checklist

1

HTML/CSS/JS

- ☐ Use gzip compression on all HTML, CSS and JS files
- ☐ Create a combined JavaScript file and combined CSS file
- ☐ Minify HTML, CSS and JS
- ☐ Load CSS before Javascript

2

HTML

- ☐ Pre-Fetch Assests
- ☐ Specify a Character set

3

CSS

- ☐ Remove inline style blocks
- ☐ Remove unused CSS
- ☐ Use efficient CSS selectors
- ☐ Avoid CSS @import
- ☐ Avoid CSS expressions

4

JavaScript

- ☐ Load 3rd party assests asynchronously
- ☐ Defer loading JS not executed onload
- ☐ Defer parsing of JS until necessary
- ☐ Use intelligent script loaders for parallel async processing
- ☐ Avoid using document.write()

Front-End Optimization Checklist

5

IMAGES

- ☐ Combine images using CSS sprites
- ☐ Compress images
- ☐ Specify image dimensions
- ☐ Served scaled images
- ☐ Use data URLs for smaller images

6

SVG

- ☐ Use vector images where possible (icon fonts or SVG)
- ☐ Optimize SVGs with an SVG cleaning tool

7

URLs

- ☐ Remove broken links, missing images and bad requests
- ☐ Serve resources from multiple hostnames for parallel processing
- ☐ Serve assets from single URL
- ☐ Serve static content from cookieless domain
- ☐ Minimize redirects

8

CACHING

- ☐ Use a Content Delivery Network(CDN)
- ☐ Use browser Caching
- ☐ Make redirects cacheable
- ☐ Enable public caching in the HTTP headers of static assets

9

GOALS

- ☐ Start render time under 2 seconds
- ☐ Single pages smaller than 500KB

Front-End Optimization Tools



- Web page test is a service which provides the website front-end speed test facility.
- Website speed can be tested on all the famous web and mobile browsers from different geographical locations.
- It provides detailed information on all the application components which can be really helpful in application optimization.

Front-End Optimization Tools



Performance Results (Median Run)

| | Load Time | First Byte | Start Render | Speed Index | First Interactive (beta) | Document Complete | | | Fully Loaded | | | |
|--------------------|-----------|------------|--------------|-------------|--------------------------|-------------------|----------|----------|--------------|----------|----------|--------|
| | | | | | | Time | Requests | Bytes In | Time | Requests | Bytes In | Cost |
| First View (Run 2) | 1.974s | 0.248s | 1.800s | 1.800s | > 1.790s | 1.974s | 20 | 923 KB | 2.043s | 21 | 924 KB | \$5--- |

[Plot Full Results](#)

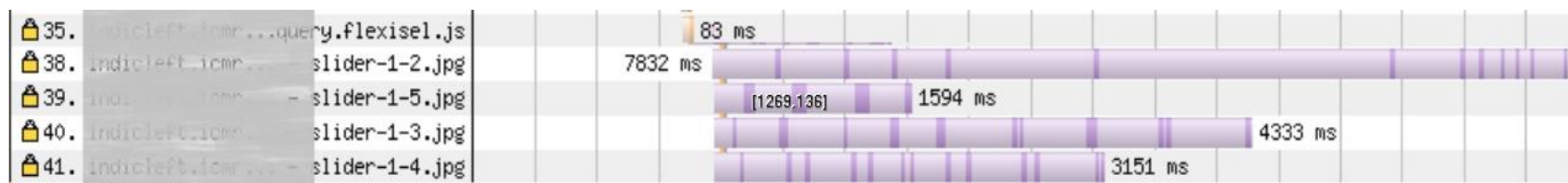
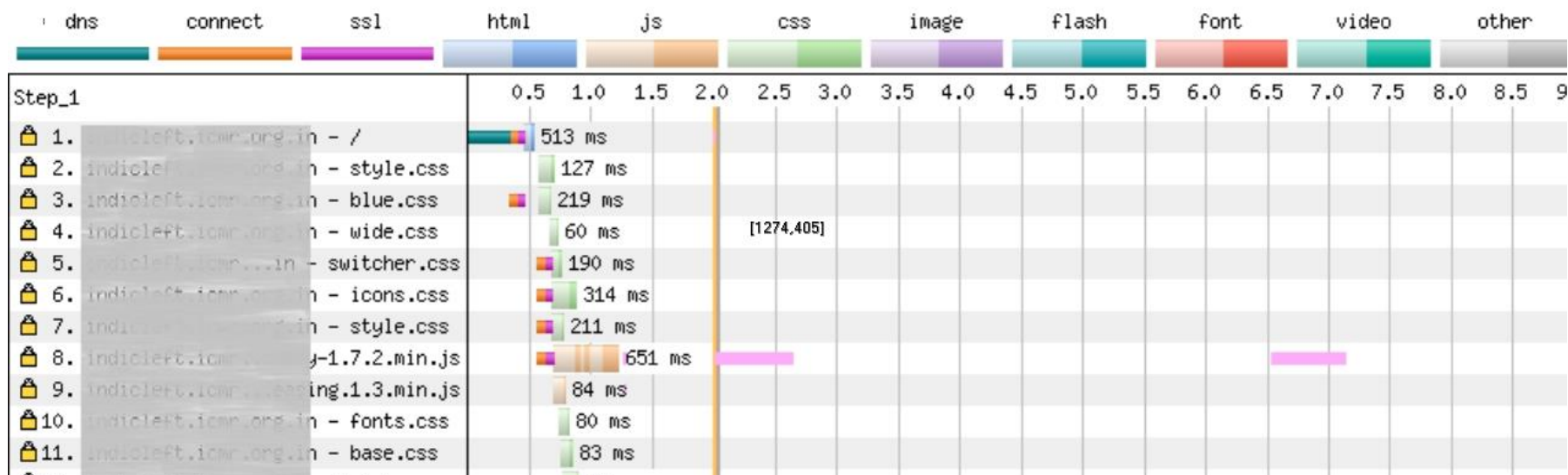
Test Results

Run 1:

| | Waterfall | Screenshot | Video |
|---|-----------|--|-------|
| <p>First View (1.981s)</p> <p>Timeline (view)</p> <p>Processing Breakdown</p> <p>Trace (view)</p> | | <p>Filmstrip View</p> <p>Watch Video</p> | |

Run 2:

| | Waterfall | Screenshot | Video |
|--|-----------|--|-------|
| <p>First View (1.974s)</p> <p>Timeline (view)</p> <p>Processing Breakdown</p> <p>Trace (view)</p> <p>Content Breakdown</p> | | <p>Filmstrip View</p> <p>Watch Video</p> | |



Thank you !

support-sqg@nic.in

IP No: 5294