Performance Evaluation of Websites Emulating Referenced Resources

Khushboo Singh  
M. Tech Scholar, CSE  
Jodhpur National University  
Jodhpur, India

Murlidhar Verma  
Lecturer, CSE  
Govt. Polytechnic College, Jodhpur  
Rajasthan, India

ABSTRACT
The objective of this paper is to investigate and optimize the performance of web sites and take steps to improve the end-user experience. Taking site’s loading speed into consideration is still something we should really look for. Extensive literature is there on website popularity, content-based, collaborative or knowledge-based filtering etc but there has been a little work regarding performance optimization of individual website. Web site performance optimization checks for various referenced resources such as JavaScript, CSS, HTML status code, HTTP request, response time, DNS lookups etc. A key indicator of the performance of any web site is its referenced resources. Web applications need to be monitored in an objective manner to ensure optimal performance translating web performance to dollars and cents. As users are in hurry and don’t feel happy if their web site is responding slowly in several seconds. Hence the site is contingent to lose a lot of visitors.

This paper purely illustrates the parameters by which web ecosystem can be improved. One website was created to measure parameters and impact of execution on the responsiveness of the website. The performance can be measured by sampling site on the basis of milliseconds a site takes to load and can also view time-based response that gives a fair idea of the performance of websites over a interval of time. If a constant degradation of performance is noticed, then necessary actions can be taken to improve it. By spiriting images, minifying JavaScript, CSS, zipped files and html tags the response and loading time of website has reduced to a great extent. The performance can be evaluated by using tools which give details of resources and components on that web site. They give the option to understand how the site is loading with different global perspective, visitors, page size and Internet connection speeds.

General Terms
Measurement, performance, metrics, response time

Keywords
Website optimization, JavaScript, CSS, page speed

1. INTRODUCTION
The number of masses using internet as their working platform has increased at an exponential rate in the past decade. Hence web sites on internet need to be monitored to ensure optimal performance. This study discusses the challenges faced in monitoring the performance of websites referenced resources, the need for an effective web site monitoring strategy and the feature.

In this study, a methodology is proposed to improve the success of web sites, based on the discovery in exploitation of navigation pattern. As the success of a web site affects and reflects directly the success of the company in the electronic market. The site optimization analyzes the website performance by emulating how a web browser would load site and all resources referenced in it. To measure website quality a large number of performance metrics have been proposed to validated empirical studies [7][8][22]. The webpage and referenced resources are loaded and important performance metrics are measured and displayed in a waterfall diagram along with other per-resource attributes such as URL, size, compression ratio and HTTP status code. Each load time bar has parts of different color corresponding to a performance metric:

- **Time in queue**: time a resource is idle in browser queue waiting to be downloaded.
- **DNS lookup time**: the time it takes to resolve the host name into an IP-address.
- **Connection time**: the time it takes to resolve the host name with the server.
- **Redirect time**: the time lost due to a redirect, a response with a HTTP status code
- **Time to first byte**: the time it took to receive the first byte from the server.
- **Time to last byte**: the time it took to receive the last byte from the server.
- **Download time**: the time spent downloading the resource.

2. MOTIVATION
As Internet expands to different geographies and seeks entry into electronic markets everybody is dependent on websites according to necessity. People book tickets, shop and sell online and do money transactions etc. Hence taking site’s loading speed into consideration is significant. The reason is pretty simple as majority of visitors are usually in rush and no one is fond of waiting half a century before website stalls to load.

In another words, site would lose viewers it if is slow [10]. Any delay in the loading of page could result in a reduction of traffic and revenue; something each competitiveness website wants to avoid. For top companies, competitiveness in e-commerce requires a successful presence on the web. For speed tech companies are on a new quest. Visitors are traffic, traffic affects reputation and they can potentially generate revenue. It can be a domino effect if one ignores the consequences of a slow-loading site. Hence it becomes obligatory to reduce page load time of site and make it as fast as possible.
Surveys suggest two thirds of users encounter slow websites every week [1] and that 49% of users will abandon a site or switch to a competitor after experiencing performance degradation. Based on Google and Microsoft engineers experience New York Times elucidated that after waiting 3 seconds for loading web pages 57% of online consumers will abandon a site. Among them 80% will not return and almost half of them go on to tell others about their negative experience [3]. If website is slower than a close competitor by more than 250 milliseconds then people will visit it less often [5]. Even a second delay in page load time produces 7% loss in conversions, 11% fewer page views and 16% decrease in customer satisfaction [6]. According to new research due to slow loading speeds, online retailers lose an estimated £ 1.73 billion in global sales each year. An average user perceives page load time as being about 15% slower than actual page load time. When sharing experience to others, people recall that the page was 35% slower than it actually was.

3. OPTIMIZATION TECHNIQUES
To study how the success of a site is reflected in the user’s behavior, by measuring the success of created site’s components a concrete indications of how the site would be optimized have been obtained.

Now we can turn our attention to the larger evaluation, end-to-end picture of web performance optimization:

- Load time, JavaScript, CSS, rich image etc. minimization
- Features and shortcomings of the HTTP protocol itself
- Web application trends and performance requirements
- Website constraints and optimizations

The designers have accumulated a series of techniques for decreasing load times. Top suggestions to improve site speed:

3.1 Dynamic Optimization with Page Speed
Google Page Speed Optimization Libraries (PSOL) provide an open source implementation of over 40 various "web optimization filters", applied dynamically to any application. Backed by PSOL libraries optimize delivered asset e.g. resource inlining, minification. Each optimization is applied dynamically which means entire optimization process is fully automated.

Emulating Event Source with Custom JavaScript

```javascript
if (!window.Event Source) {
  // load JavaScript poly fill library
}
var source = new Event Source("/event-stream-endpoint");
```

JSON (JavaScript Object Notation) use in coding

```
GET /stream HTTP/1.1 1
Host: example.com
Accept: text/event-stream
Content-Type: text/event-stream
Transfer-Encoding: chunked
```

data: Message of type "foo"

```
id: 42
event: bar
```

data: Multi-line message of type "bar" and id "42"

```
id: 43
data: Last message, id "43"
```

Client connection initiated via Event Source interface

Server response with "text/event-stream" content-type

Server sets client reconnect interval (15s) if the connection drops

Simple text event with no message type

JSON payload with no message type

Simple text event of type "foo"

Multiline event with message ID and type

Simple text event with optional ID

3.2 Lower the Number of HTTP Requests
Whenever visitor visits your website, the related files i.e. all CSS files, JavaScript library references, images and more must be sent and loaded into that person’s browser. This can hinder sites performance. Therefore, it is essential to ensure that your website's coding only lists what is necessary.[11]

3.3 Correctly Optimize and Display Images
Virtual images, depending on their format, increase the file size. Image resizing is neglected by website designers prior to uploading them, hence affecting loading speed of the website image-heavy pages. Inexperienced web designers also tend to upload an image that is bigger than required. Instead, use applications such as Picnik[21] to resize and optimize images.

3.4 Minimize HTML, CSS and JavaScript
Compacting of these codes can save many bytes of data and speed up downloading, parsing, and execution time [10].

```html
<table>
  <tr>
    <td colspan="2">top navigation bar (branding and advertising) </p></td>
  </tr>
  
  <div>
    <p>left navigation bar</p>
  </div>
  
  <div>
    <p>main content area</p>
  </div>
</table>
```

Fig 1: CSS Three-column layout

The spaces, tabs and structure set forth in code is only there to make it readable by humans. Servers, browsers don’t care what your coding looks like as long as it’s valid and error-free. If you’re trying to quicken site, consider removing this whitespace before serving your code.
3.5 Use a Content Delivery Network to Lighten Your Server’s Load
Using a Content Delivery Network (CDN) allows websites to utilize a high-performance network of web- servers that is able to replicate the static assets of website and deliver them to visitors. Therefore, less work is required increasing the speed at which your site loads. One such product is Cloud Flare

3.6 GZIP Files and Compression
Before sending files to the browsers for human viewing, one should compress them by just adding a few simple lines of code.

3.7 Choose Linking over Import
When working with a site’s style sheets, it’s a better idea to connect files using linking instead of through the import reference. Import reference automatically assuming that they are located at the bottom of the document even if it’s not.

3.8 Place Style sheets at the Top
As we always want to display styled content to visitors, files for the appearance of a webpage should be placed in the header section. Ensuring that users see the webpage’s design and content at the same time.

3.9 Place Scripts at the Bottom
Make sure that tools related to functionality are placed at the bottom of a webpage as they are not important for viewer.

3.10 Use Browser Caching
When the frequently accessed file is needed, the browser can easily grab it from its local cache instead of requesting it from the server.

3.11 Consider Utilizing CSS Sprites
CSS Sprite acts like a map of your images for the server. As a picture comprised of other images used by your design, it allows all of the files to be accessed from one place thus minimizing the loading process. Using these simple but important tips can improve a website’s overall performance.

4. IMPLEMENTATION
We are no longer building websites instead we are building dynamic websites. So it is essential to identify what builds up the website before jumping into the codes trying to speed things up because it cannot be done without knowing the culprits that are slowing down the web site. For improving the quality, reliability and speed of web applications using tools and testing products by comparing sites performance can be calculated.

In addition to measuring the page load time (PLT), we are now interested in answering application-specific questions:

- What are the milestones in the loading progress of the page?
- What are the times to first interaction by the user?
- What are the engagement and conversion byte rates for site?
- What is meant by "complex website scripts, style sheets, and markup" found in a modern website?

5. TOOLS
This research will compile a collection of useful web service and tools that will help to diagnose and analyze web site, allowing to understand the site better. By using these tools the need to measure end-user experience from a global perspective can help web site to achieve optimal level of quality end-user experience. Online tools and services ensure growth and revenue. While interaction with these tools, it becomes easy for these sites to gain insight into the end-user experience by their evaluation and services.

For continuously monitoring and analyzing full Webpage Objects Test, the loading time of website using tool, in this study several useful tools are involved i.e. Ping Test’s Site24x7, AOL’s Web Page Test[12], Page Speed Insights[2], Yahoo’s Y Slow[11],GT Metrix[18].

6. EVALUATION
In this section real website contents are evaluated. It has been demonstrated that minimizing response time, html codes etc can substantially improve load times for websites with large amount of CSS and JavaScript. An analysis of how content chunks evolve within the same page and across different pages is also done.

6.1 Preparation
Optimizing the interaction among sites pages is not unlike solving a family of equations, each dependent on the others, but nonetheless yielding many possible solutions. For this paper a website was created named http://www.shriramits.in/. Grades of the website were taken emulating them on tools (i.e. GT Metrix, Site24x7, Y slow). After analyzing and evaluating all referenced resources. Further received and selected the rules to optimize and make website reaching better performance [14][16][19][20]. There is no one fixed set of recommendations or best practices.
Hence, before we dive into enumerating and analyzing individual performance best practices, it is important to step back and define what problem is faced, what a modern web application is, what tools we have at our side, how we measure web-performance, and which parts of the coding are helping and hindering our progress.

6.2 Experimental Setup
Experimental setup consisted of a client machine whose browser fetched content from website, resided on a desktop running Windows 7. The desktop had an Intel Duo Core with 2.59 GHz processors and 1.87GB of RAM. The client communicated with the server over a residential local area network by DSL modem. Across all experiments, the bandwidth varied between 700 – 850 Kbps. In the results presented below, the google chrome browser was used, but the results for IE8 were similar.

6.3 Characterizing Complexity
The analysis of performance is two-pronged. In this section first of all, web pages of created website with respect to various complexity metrics were analyzed. Afterwards, analysis of the impact of metrics was performance. Here is presented a comparison of metrics across different categories of websites (e.g. social networking, Shopping etc). By considering the website complexity, the number and size of objects fetched to load the web page can be captured. Table 1 summarizes key findings for the various websites' complexity metrics.

A wide overview about the performance states of the most visited websites is presented. Here measured statistics of the top most visited websites. The selections and metrics are made in May 2014 and merging categories into one descending list it can be identified that website reached the poor 11th place in the overall table (Table 1) [18].

Table 1: Overall grades before optimization

<table>
<thead>
<tr>
<th>No</th>
<th>Overall before</th>
<th>PageSpeed</th>
<th>Yslow</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>facebook.com</td>
<td>98</td>
<td>97</td>
<td>97.5</td>
</tr>
<tr>
<td>2.</td>
<td>google.com</td>
<td>98</td>
<td>94</td>
<td>96</td>
</tr>
<tr>
<td>3.</td>
<td>myntra.com</td>
<td>96</td>
<td>89</td>
<td>92.5</td>
</tr>
<tr>
<td>4.</td>
<td>youtube.com</td>
<td>95</td>
<td>86</td>
<td>90.5</td>
</tr>
<tr>
<td>5.</td>
<td>amazon.com</td>
<td>94</td>
<td>82</td>
<td>88</td>
</tr>
<tr>
<td>6.</td>
<td>booking.com</td>
<td>93</td>
<td>82</td>
<td>87.5</td>
</tr>
<tr>
<td>7.</td>
<td>ibibo.com</td>
<td>89</td>
<td>84</td>
<td>86.5</td>
</tr>
<tr>
<td>8.</td>
<td>flipkart.com</td>
<td>86</td>
<td>83</td>
<td>84.5</td>
</tr>
<tr>
<td>9.</td>
<td>yahoo.com</td>
<td>89</td>
<td>77</td>
<td>83</td>
</tr>
<tr>
<td>10.</td>
<td>shriramitii.in</td>
<td>73</td>
<td>84</td>
<td>78.5</td>
</tr>
<tr>
<td>11.</td>
<td>expedia.com</td>
<td>84</td>
<td>57</td>
<td>70.5</td>
</tr>
</tbody>
</table>

7. RESULTS
The results are taken after analyzing website for 24 hours before optimizing it (Fig 4) and (Fig 5) . The script, image CSS minification, html code and gzip compression techniques are applied on sample website. For this website 24x7 supported tools are used. After optimizing website and observing it for 24 hours, grades are taken (Fig 6) (Fig 7). Website comparison before and after optimization of referenced resources has been illustrated in (Fig. 8). On Y-axis response time in milli-seconds and on X-axis time in hours has been taken. [4]

Unfortunately, there is no "ideal" size for a image sprite. CSS, gzip or a JavaScript bundle. However, tests performed by the Google Page Speed [2] team indicate that 30–50 KB is a good (compressed) is a good. Your results may vary based on type and number of scripts in use. Note that the percentage improvement is not a fixed performance gain. The parameters are specific to chosen network connection.

![Fig 4: Before optimization](image)

![Fig 5: Waterfall diagram before optimization](image)

Table 2. Parameters of website

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Time</td>
<td>1 sec(s)</td>
<td>358 ms</td>
</tr>
<tr>
<td>Max Resp Time</td>
<td>9.951 ms</td>
<td>940 ms</td>
</tr>
<tr>
<td>Avg Resp Time</td>
<td>152 ms</td>
<td>76 ms</td>
</tr>
<tr>
<td>Min Resp Time</td>
<td>46 ms</td>
<td>57 ms</td>
</tr>
</tbody>
</table>

Table 3. Total Time in milli-seconds

<table>
<thead>
<tr>
<th>DNS Time before</th>
<th>Connecti on Time</th>
<th>First Byte Time</th>
<th>Start Render</th>
<th>Documen t Complete</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2531</td>
<td>3045</td>
<td>4627</td>
<td>5572</td>
<td>6858</td>
</tr>
<tr>
<td>12</td>
<td>227</td>
<td>790</td>
<td>2271</td>
<td>3548</td>
<td>1578</td>
</tr>
</tbody>
</table>

Table 4. Reduction of page size in bytes and percentage

<table>
<thead>
<tr>
<th>Report Analysis</th>
<th>Byte turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Size Reduced By</td>
<td>1,182 Bytes</td>
</tr>
<tr>
<td>Reduced By</td>
<td>13.61 %</td>
</tr>
<tr>
<td>Original Page Size</td>
<td>8686 Bytes</td>
</tr>
<tr>
<td>Optimized Page Size</td>
<td>7504 Bytes</td>
</tr>
</tbody>
</table>
Fig 6: After optimization

Fig 7: Waterfall diagram after optimization

Table 5: Overall grades after optimization

<table>
<thead>
<tr>
<th>No.</th>
<th>Overall after optimization</th>
<th>PageSpeed</th>
<th>Yslow</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>facebook.com</td>
<td>98</td>
<td>97</td>
<td>97.5</td>
</tr>
<tr>
<td>2.</td>
<td>google.com</td>
<td>98</td>
<td>94</td>
<td>96</td>
</tr>
<tr>
<td>3.</td>
<td>myntra.com</td>
<td>96</td>
<td>89</td>
<td>92.5</td>
</tr>
<tr>
<td>4.</td>
<td>youtube.com</td>
<td>95</td>
<td>86</td>
<td>90.5</td>
</tr>
<tr>
<td>5.</td>
<td>shriramiti.in</td>
<td>98</td>
<td>77</td>
<td>87.5</td>
</tr>
<tr>
<td>6.</td>
<td>amazon.com</td>
<td>94</td>
<td>82</td>
<td>88</td>
</tr>
<tr>
<td>7.</td>
<td>booking.com</td>
<td>93</td>
<td>82</td>
<td>87.5</td>
</tr>
<tr>
<td>8.</td>
<td>ibibo.com</td>
<td>89</td>
<td>84</td>
<td>86.5</td>
</tr>
<tr>
<td>9.</td>
<td>flipkart.com</td>
<td>86</td>
<td>83</td>
<td>84.5</td>
</tr>
<tr>
<td>10.</td>
<td>yahoo.com</td>
<td>89</td>
<td>77</td>
<td>83</td>
</tr>
<tr>
<td>11.</td>
<td>expedia.com</td>
<td>84</td>
<td>57</td>
<td>70.5</td>
</tr>
</tbody>
</table>

*Note that emphasis is on capturing the complexity of websites as visible to browsers on client devices. The results are only shown for one of the vantage points as the results are quite alike across vantage points.

Fig 8: Comparison

7.1 Comparison Report before Optimization

7.1.1 Page Speed Scores (higher is better)

7.1.2 YSlow Scores (higher is better)

7.1.3 Page Load Times (lower is better)

The comparison of three sites is shown. Here flipkart.com, own created site (i.e. shriramiti.in) and facebook.com are taken to display the quality parameters changing after optimization. The grades are shown before and after optimizing the experimental site. Here other two popular sites are taken just to compare previous and optimized results. They are kept unchanged.

It’s clearly shown that after applying optimization techniques the website has improved its performance regarding different parameters and matched them remarkably.

Experiments show that optimization and analysis can reduce load times by 20%–80% for pages with large amounts of JavaScript and CSS, image size is reduced by 43%, java script by 29%. Additionally, the turnover rate of data in web pages is reduced by 13.61% and 858 white spaces were removed from HTML code.
7.2 Comparison Report after Optimization
7.2.1 Page Speed Scores (higher is better)

7.2.2 YSlow Scores (higher is better)

7.2.3 Page Load Times (lower is better)

8. CONCLUSION AND FUTURE WORK
In this paper, a new framework for reducing load times while preserving optimization of referenced resources is introduced. Slow web pages frustrate users and decrease revenues for performance evaluation. Developers have created various ways to defer or hide fetch loading time, but perhaps the most effective technique is the most straightforward: reducing the number of resources required to build a page. Unfortunately, this strategy presents performance evaluation with a quandary, reducing the number of objects in each page, but this can negatively impact the rich content of the website.

The complexity of websites based on the resources they include is characterized. Also, it is found that significant fraction of objects and bytes fetched from most websites, contribute in page load time in comparison.

Our next task should be to dig deeper into impact of latency and bandwidth on web performance, client and server-side code (AJAX, PHP, ASP), TCP constraints imposed, and SQL database optimization. Completing and combining all known issues could result an effective and valuable web application either on desktop or mobile devices.

9. REFERENCES