INSPECTING THE DYNAMICS OF FASTER WEB

Pradeek P Mohandas[1], Rishana Rabiya P[2], Anjali P Jayan[3], Shifa Jahan[4], Shafna Ashraf[5]

[1]Department of computer science and Engineering, College of Engineering, Thalassery
[2]Department of computer science and Engineering, College of Engineering, Thalassery
[3]Department of computer science and Engineering, College of Engineering, Thalassery
[4]Department of computer science and Engineering, College of Engineering, Thalassery
[5]Department of computer science and Engineering, College of Engineering, Thalassery

Abstract

Internet has revolutionized communications, to the extent that it is now our preferred medium of everyday communication. It has also transformed the way companies conduct business. Thus fast internet access has become one of the highest priorities. Every company is in race to provide faster loading web pages. There are many methods in existence to load a web page faster. In this paper we study these different methods and their impact. This paper provides comparative factual collation of these methods which are being used for optimization of web. Different strategies are proposed, each claiming to provide an improvement over other strategies. We will be making a study on different protocols(SPDY,h2) that effect speed, data compression proxy for fast mobile web, revolutionary technologies like POLARIS and study about dev tools of leading browsers like opera mini(data optimization mode),chrome etc. Paper also discuss about the practices that a developer can follow to create faster loading pages. This paper present a study on all the above mentioned techniques which are very much helpful for web optimization.

Keywords— *Browser, Data Compression, Proxy, Web Optimization.*

----------***----------

1. INTRODUCTION

Internet, an inevitable part in this digital age has made our life easy by facilitating so many uses. Internet is a huge source of information and it pave the way for communicating with the world. With progress in technologies, new uses get added to the existing array of Internet uses. As internet technology has changed the way we communicate personally, it has truly revolutionized business communications too. When it comes to internet services, it's all about speed and reliability. Wireless protocols (WiFi, 3G, 4G, etc.) has multiplied the number of internet users [1]. Slow speeds and weak connections can be frustrating and even costly. To keep up with the pace of innovation and growth, businesses completely relay on fast internet access. As more Internet-ready devices make their way into our homes and workplaces, keeping our web speed up is more important than ever. Slow internet thus has got many consequences. Extra delays of just a few milliseconds can result in users abandoning a page early. In fact, Strange Loop has stated that just "a one second delay can cost you 7 percent of sales" [2].

A modern web page contains a lot more objects, and many network round trips are required to fetch these objects. Over 80% of user-perceived load time is spent downloading HTML, JavaScript, images, and other objects, and 40–60% of page visitors arrive with an empty cache. One method adopted to reduce the number of round-trips needed to fetch the page's objects is by minimizing the number of HTTP requests needed to build a page. A page's load time also influences how the page is ranked by search engines—faster pages receive higher ranks . Thus, a variety of research projects and commercial systems have tried to reduce page load times. Website optimization(as shown in Figure.2) is best conducted as an ongoing process that over time makes your website more effective and valuable.

2. BACKGROUND

A web page request could be made by refreshing a page or by typing URL into a browser or a page could also be accessed when its corresponding link is clicked. The requested page and its resources (files) are downloaded. The web browser uses the page resources to build the page which is then rendered to the user. In a conventional rendering, the browser first downloads the page's top-level HTML. For now consider the case of a simple web page where the HTML does not reference any JavaScript, CSS, or multimedia files. The browser generates a data structure called the Document Object Model (DOM) tree while parsing the HTML tags. Each node in the DOM tree, corresponds to a HTML tag . When browser completes a DOM tree, it constructs a render tree.DOM tells how the page will actually be laid out and painted. To calculate geometric properties like sizes and position of objects, the browser traverses the render tree and produces a layout or the reflow, and it is basically where the browser determines how big a screen is and how that will affect the way the page is displayed. This is called the paint. Each node in the render tree is converted to actual pixels on the screen in this final stage. Modern browsers try to pipeline the construction of the various trees, in order to progressively display a page.

However, For a typical page request(as shown in Figure 1), the browser need to retrieve resources like CSS, JavaScript, media files, apart from HTML and use these resources to generate a document object model (DOM). Then the

browser's layout engine renders a layout, and the browser paints the page. In this approach script tag blocks HTML parsing and Java Script execution is blocked by CSS evaluation [3]. These inefficiencies are results of lexical dependencies between HTML tags. Here the browser pessimistically guess dependencies, since HTML does not express all of the true semantic dependencies. The uncertainties in the traditional approach forces to think on a better rendering method.

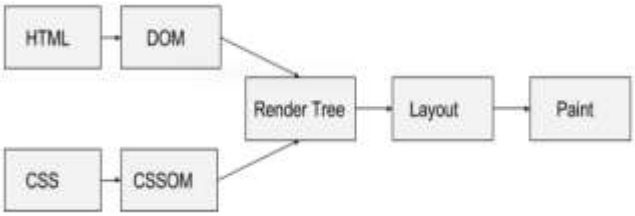

Fig 1: Rendering of a web page

3. STUDY

In this paper we make a study on major underlying techniques on faster web page loading. The key to faster pages is, reducing the front-end load time, reducing delay on network side and appropriate choice among browsers. According to the latest report from IAMAI [4,8], the country is estimated to have 371 million mobile internet users among 462 million total internet users in India. Therefore mobile-optimized websites lead to positive feedback among huge audience. SPDY [5] an open-specification networking protocol and various other protocols, HTTP 1.1, HTTP 2, are discussed under section A. Studies were also made on Google Weblight and Brotli, categorized under compression technology in section B. Instant pages like AMP, FB instant is discussed under section C. Study on browser technology (section D) includes opera turbo and silk. A major breakthrough which uses the concept of dependency graph is discussed under section E.

3.1 Evolution of Protocol

Hyper-Text Transfer Protocol (HTTP) and Transmission Control Protocol (TCP) have been the backbone of web transport for decades. These had not designed to reduce latency, therefore require some modifications according to current internet traffic demands. For this reason, a lot of research performed at application layer and network level with the goal to make the web faster. HTTP/2 is the most recent proposal to provide faster information exchange over web.

3.1.1 HTTP 1.1

HTTP is a transfer protocol used by the WWW to retrieve information from distributed servers. Here, the client establishes a connection to the remote server, and issues a request. Then the server returns response after processing the request. HTTP relies entirely on the TCP/IP infrastructure. To function efficiently, it must take advantage of TCP/IP's strengths and avoid its weaknesses. HTTP/1.0[16] interacts badly with TCP. Due to connection establishment HTTP/1.0 incurs frequent round-trip delays, for short duration connections performs slow start in both directions, and due to the mismatch of the typical access profiles it incurs heavy latency penalties (with the single request per transaction model). It also requires busy servers to dedicate resources for maintaining TIME_WAIT information for large numbers of closed connections.

The HTTP/1.1[17] protocol is the result of four years of debate and discussion among a broad group of Web researchers and developers. It has an open door policy known as persistent connections. Here, once a TCP connection has been opened, as much data should be sent through it as possible. it reduces network traffic and perceived download times by generating fewer but larger IP packets. Improved caching mechanisms in HTTP/1.1 significantly reduce the number of packets necessary to verify whether to refresh a cached resource on a proxy server. It supports Pipelining which reduces the total elapsed time interval of the initial request and the final response without any loss in the serial nature of the requests. It allows ISP to assign a host name to a company without using up a single IP address by means of it's host headers. Also, portions of a resource can be requested by a client using HTTP/1.1 range requests . To avoid the wastage of bandwidth in transmitting the request body, it includes a new status code, 100 (Continue), to inform the client that the request body should be transmitted(not all servers use this mechanism). It includes the Content-Encoding header, which indicates the end-to- end content-coding used for a message and Transfer-Encoding header, which indicates the hop-by-hop transfer-coding used for a message.

3.1.2 SPDY

It is an application-layer protocol for transporting content over the web, designed concretely for minimal latency. Today, HTTP and TCP are the protocols of the web. HTTP runs on Application Layer whereas TCP runs on Transport Layer. SPDY adds a session layer at the top of SSL which allows for multiple concurrent, interleaved streams over a single TCP connection. The normal HTTP GET and POST message formats does not have any change; however, SPDY specifies a new framing format for encoding and transferring the data over the wire. Streams are bi-directional, i.e. they can be initiated by the client and server.

Major improvements were summarized as :
* Multiplexed requests: There may be any number of requests that can be issued concurrently over a single SPDY connection. The efficiency of TCP is much higher since the requests are interleaved on a single channel.

* Prioritized requests: Request can be made by clients for certain resources to be delivered first. This avoids the problem of non-critical resources congesting the network channel when a high-priority request is pending.
* Compressed headers: A significant amount of redundant data is sent by the clients today in the form of HTTP headers. This data is significant, since 50 or 100 sub requests may be required by a single web page . Compressing the headers saves a significant amount of latency and bandwidth compared to HTTP.
* Server push: SPDY experiments with an option for servers to push data to clients via the X-Associated-Content header. The client is informed by the header, that the server is pushing a resource to the client before the client has asked for it. In the case of initial-page downloads (e.g. the first time a user visits a site), this can vastly enhance the user experience.
* Server hint: Instead of pushing resources automatically to the client, the X-Sub resources header is used by the server to suggest to the client that it should ask for specific resources, in cases where in advance of the client the server knows that those resources will be needed. However, before sending the content ,the server will still wait for the client request. Over slow links, this option can reduce the time for a client to discover it needs a resource by hundreds of milliseconds, and may be better for non-initial page loads.

3.1.3 HTTP 2

HTTP/2[18] was developed by the Hypertext Transfer Protocol working group of the Internet Engineering Task Force from the experimental SPDY protocol. HTTP/2 supports all of the core features of HTTP/1.1 (such as status codes, methods, URI, header fields etc), but aims to be more efficient in several ways. It decreases the latency to improve the load speed in web browser by considering, Data compression of HTTP headers, Server Push, Pipelining of requests, Fixing the head-off-line blocking problem in HTTP 1.x(even after using pipelining). Multiplexing multiple requests over a single TCP connection, prioritization of requests.

Server push allows the server to send additional cache-able information to the client which are not requested but is anticipated in future requests. Each HTTP request/response exchange is associated with its own stream to achieve multiplexing of resources. Streams are largely independent of each other; therefore a blocked or stalled request or response will not prevent progress on other streams. This avoids head of line blocking problem. As web pages have grown to require dozens to hundreds of requests, the redundant header fields in these requests consumes bandwidth unnecessarily which measurably increases latency. To address this issue, instead of SPDY's dynamic stream-based compression , HTTP2 uses a fixed Huffman code-based header compression algorithm. This in turn helps to reduce the possibility for compression oracle attacks on the protocol, such as the crime attack. The standardization result was supported by Chrome, Opera, Firefox, Internet Explorer 11, Safari, Amazon Silk and Edge browsers. HTTP/2 support was added by major browsers by the end of 2015. Table 1. summarizes the browser support on the protocols.

Table 1 Various browser support on HTTP/2 and SPDY protocol

IE	Edge	firefox	chrome	safari	opera	Ios safari	Opera mini	Andriod browser	Blackberry browser	Opera mobile	Chrome android	Firefox android	IE mob	UC android	Samsung
11	12 13 14 -not supported	13 to 14	4 to 50	8 9 9.1 10 TP	12.1 15 to 41	8 8.4 9.2 9.3	Not supported	3,4,4.1,4.3,4.4	7,10-not supported	12.1 37	51-not supported	47	11	9.9-not supported	4
6 to 10 - no 11 - partial	12 13 14	36 to 51	41 to 55	9 9.1 10 TP-partial	28 to 41	9.2 9.3	Not supported	51	7, 10-not supported	37	51	47	10,11-notsupported	9.9-notsupported	

3.2 Compression & Web Optimization

Compression is one of the most important tools, used to accelerate website performance. Compressed content takes less time to transfer and consequently reduces load times. Compression even saves money for consumers, on expensive mobile data plans. It is one of the most expensive operations our servers perform. The better the compression rate we want, more the effort we have to spend. Popular compression format on web, in use is gzip. Brotli a new compression algorithm, introduced by Google is discussed in this section. Weblight technology which makes use of compression is also discussed here.

3.2.1 Brotli

Brotli[15], released on september 2015, is a new compression algorithm, introduced by google. Google says, Brotli is a "whole new data format" that can optimize file sizes up to 26 percent over existing solutions. It squeezes in more data than other compression formats. Brotli is an open source loss less compression algorithm that compresses data using a combination of LZ77 algorithm and Huffman Coding with great efficiency comparable to the currently available general-purpose compression methods. Google claims that Brotli is 20 to 26 percent more efficient than Zopfli and improves by roughly 17-25% on gzip's compression ratio . Smaller the compressed size, better the space utilization. Brotli also points out, less bandwidth utilization, less power consumption and minimizes the CPU cycles . All these are achieved by " re-use of entropy codes, larger memory window of past data and joint distribution codes". Brotli also supports context modeling, a feature that allows multiple Huffman trees for the same alphabet in the same block. It can use any window size from 1KB to 16MB, in powers of 2, for back references. Specification also includes a static dictionary, which can be referenced from anywhere in the stream, increasing its efficiency for larger files. Chrome will support this across all major desktop platforms and Android. Mozilla has also added Brotli support to Firefox. A study compared six compression methods, and based on the results proposes that brotli could be used as a replacement of the common deflate algorithm. Brotli is fundamental and superior to other compression. Introduction of brotli to web can speed up load times significantly.

3.2.2 Google Weblight

Google Web Light, is google's own compression proxy service and is aimed for mobile devices having slow internet connection. This service promises us the optimization of websites for 4x faster loading and 80% (to be exact) less data consumption. Google even claims a 50% increase in traffic to these pages, that have been optimized. This feature is not enabled for faster connections like 3G/4G and works only on android devices and chrome browser. This facility is currently provided in Indonesia, India and Brazil.

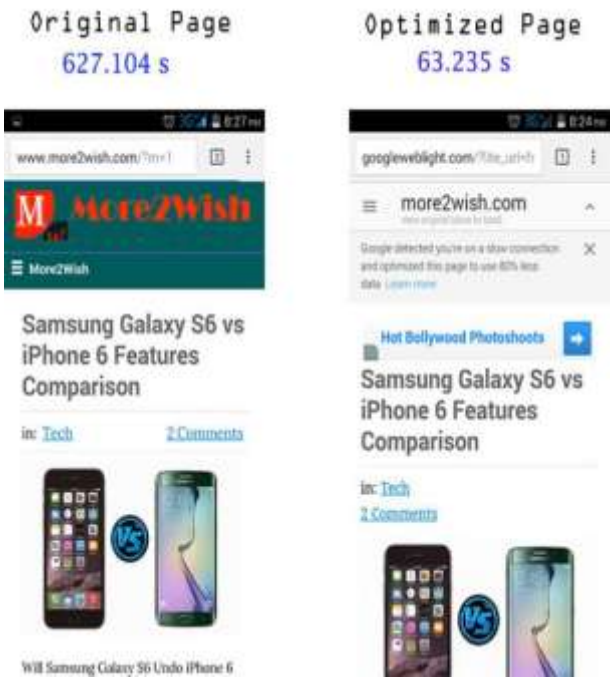

Fig 2 Optimized page comparison

Google transcodes web pages into a version optimized for slow networks, thereby rendering pages fastly and provides a better web browsing experience with saving the bandwidth consumption. It actually transcode the page and any pages that the user clicks to from within that page, unless the page is non-transcodable or opted out from transcoding. Optimized page care for majority of the relevant content and provide a link for users to view the original page. The pages are only transcoded if the detected connection speed is below a certain threshold. What Google actually does is, it compress the various styling codes used in a webpage and combine all the resources in one file. Now Google's robust CDN server serves this single compressed file. This saves the multiple round-trip to the original server to load content. The only problem is, transcoded page doesn't look attractive and is only useful for reading text content. Sites which requires user login to access contents, and those having high graphics and videos only websites can't be loaded via this method.

3.3 Instant Pages

Mobile web revolution is on the way. Big players Facebook, Google, and Apple are delivering content to mobile users in new and innovative ways. Having a fully-responsive site isn't the only goal, but to have an instant loading page is the one. We have Facebook instant articles, google's AMP (accelerated mobile page), Apple news. Aim of all, is to have a monopoly in content publishing and provide faster content. It's the battle for mobile world.

3.3.1 AMP

The Accelerated Mobile Pages (AMP) is used to build web pages that render fast on mobile devices. AMP consists of three different parts: AMP HTML, AMP JS and Google AMP cache.

AMP HTML: AMP- specific tags replaces some regular HTML tags inorder for building rich content than basic HTML.

AMP JS library[5]: It gives AMP- specific tags. It manages loading of resources and implements all practices required for faster page rendering.

Google AMP cache[6]: A content delivery network, which is proxy- based and delivers all valid AMP documents. Page performance is improved automatically by caching the HTML pages which has been fetched.

Practices in AMP such as allowing asynchronous scripts only, static sizing of all resources, preventing extension mechanism in blocking of page rendering, keeping all third-party JS out of critical path, allowing only inline and size-bound CSS etc makes faster rendering of pages.

3.3.2 FB Instant

Facebook Instant Articles[19] (FBIA), a format for mobile publishing, is used by news publishers for distributing articles to Facebook's app in order to make loading and displaying 10 times faster as compared to standard mobile web. A mobile optimized version of article is created by FBIA and it is preloaded partially with its appearance in the user's news feed.

Instant Article actually is an HTML5 document which has been optimized for fast mobile performance. It has capabilities for rich story telling with customized visual display and branded design. Similar to XML, Instant Articles use a standardized markup language for adding styles and interactive functionality to the story. This markup can be applied either automatically or manually. Any type of articles can work as Instant Articles. Instant Articles are available for readers with iPhone Facebook or Android Facebook.

3.4 Browser Technologies

A web browser is a software application which enables us to browse the web. Various technologies have been integrated into browsers over time for faster and smoother surfing. Today's web is a shaped technologies, like HTML5, CSS3 and WebGL etc. New technologies are focused on speed since it is the most important aspect. Under this section we will learn about latest technologies at browser hand that provide faster web pages.

3.4.1 Silk

Web browser designed by amazon for kindle fire tablets and fire phone. It uses split architecture [10] (as shown in figure 3) that divides processing between client and amazon cloud. It makes heavy use of SPDY [9], Google's protocol running on top of SSL, and Silk [11] gives SPDY performance improvements for non-SPDY optimized websites, if the pages are sent through Amazon's servers. In order to improve the web page loading process, the web page requests are routed by Amazon Silk through remote proxy servers powered by Amazon Elastic Compute Cloud(AEC2). Amazon will load the web page, downloading all the components in parallel on the server side. After downloading the content, Amazon will send the compiled page including JavaScript, HTML,CSS, and images, back to the device as a single stream of data. A single persistent connection is maintained by the Silk browser to Amazon's cloud, through which requests are sent and content is received. Amazon servers provide speedy connection, high computing power, reduced bandwidth, expansive memory, fate network pipes.

By amalgamating machine learning with the resources available in the Amazon cloud, Amazon Silk realizes new possibilities in optimization. Page rendering in AEC2 allows the size of the browser cache on the Fire to be reduced. It has a positive impact on 8GB internal memory of Kindle Fire. According to Jon Jenkins, Director of Software Development, Amazon Silk, "it doesn't take a single byte of storage on the actual device". To determine fastest way to deliver the content to the device, Amazon silk uses the vast Aws behind it and analyzes aggregate web traffic patterns, pre-processes pages, and applies predictive algorithms. Silk allows you to turn off it's cloud features and use it like a typical browser. Also the content can be tailored to Fire before it reaches the device. For example, images are automatically trimmed down to a resolution corresponding to the screen they are displayed upon, thereby making them smaller in size for faster delivery and loading.

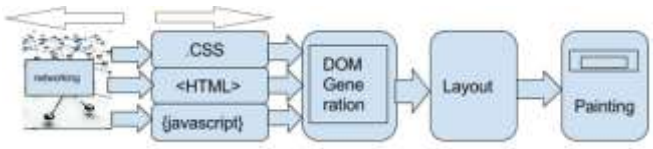

Fig 3 Split Architecture

3.4.2 Opera Turbo

Opera has a long history as an innovator when it comes to Web browsers. Opera Turbo Mode, a new element in Opera 10 Browser permits a page to load much quicker by compressing the picture. Opera's servers will compress the pictures and other components from a page, when this mode is empowered, and send them to your program, bringing about a speedier load time. It actually shrink the web content your device receives to a fraction of its original size. Assume you have a 100 KB picture on the website page with some content on it. As content is fundamental HTML, they scarcely take any data transmission however the 100 KB picture is compressed to a littler size by Opera servers and subsequently it stacks significantly quicker on your moderate web association. Opera is changing them into WebP format. The information won't be diverted through Opera servers because of protection issues,if you are going by HTTPS (secure) site pages, and consequently you won't have the capacity to utilize the turbo mode in such cases to load website pages speedier. It is accessible to Opera clients since 2009.

3.5 Revolution

Till now we discussed about the existing technologies that helps to load a web page faster. Now lets see some of the game changing ideas in the field . This could revolutionize the internet by providing speed with our existing hardware and software's.

3.5.1 Polaris

Most studies focused on compression mechanism, but at MIT and Harvard researchers have tried a different approach to the problem. Dependencies can't be seen before hand for a browser, which means that once one object is downloaded it might reveal that object is dependent on another, so the browser has to go back out to the web and grab that next object and so on and so forth. In order to evaluate an object, the system has to grab more objects and evaluate them, which are 'dependencies' of the originals.

The way these dependencies are displayed in HTML makes it difficult for the browser to locate them, which implies it must be watchful about the order they stack objects. These increased back-and-forth trips. This is what slows down the page load and increase the number of cross-network trips. Polaris works by pre-mapping various connections between different objects on a page in order to figure out the most efficient order in which to load the objects. Thus a browser can know exactly what objects are dependent on each other and download those concurrently. It uses scout -a dependency tracker to track dependencies. It track the fine-grained data flows across the JavaScript heap and the DOM that arise during a page's load process.Polaris, a JavaScript client-side scheduler which leverages Scout graphs to assemble a page. Polaris prioritizes the fetching and evaluation of objects along the dynamic critical path, trying to make parallel use of the client's CPU and network, and trying to keep the client's network pipe full, given browser constraints on the maximum number of simultaneous network requests per origin.

The researchers behind Polaris tested their system under a range of network conditions, "with latencies ranging from 25ms to 500ms, and bandwidths ranging from 1Mbps to 25Mbps", and on 200 popular websites. This showed, they say, a decrease by up to 34 percent at the median and 50% at the 95th percentile. Performance varied significantly across sites, being higher with complex pages and lower with pages making aggressive use of caching.

4. OBSERVATIONS

Having a fast website and developing one, in this modern era of web design is simply indispensable. We have various tools available, to test the website speed and its performance. Website speed test helps us to analyze the load speed of websites and learn how to make them faster. It simply lets you to identify how speed your website is, how big the data is, what best practices you're not following, and so on.

Pingdom, WebPagetest, Neustar, WebsiteTest, iWebTool, Vertain, PageSpeedOnline, GtMetrix, WebsiteOptimization, and ShowSlow are some trustworthy tools for testing your website speed. Here, in this observation we have used various tools for checking the website performance. Results were noted down in the observation table as shown below.

Table 2.1 summarizes the performance of three website facebook.com, tumblr.com and www.dekulk.nl. Here, first one supports all the three protocols, where as second one supports HTTP/2 only and third one does not support SPDY and HTTP/2.

Table 2.1

Website url	Protocols supported	Page size	requests	Image requests	Script requests	Css requests	Plain text requests	Html requests	Other requests	Load time
Facebook.com	HTTP/2 SPDY HTTP/1.1	365kb	35	11	8	7	0	3	2	1.25s
Tumblr.com	HTTP/2	6.6mb	85	45	15	9	2	6	2	2.64s
www.dekulk,.nl	Does not support HTTP/2 and SPDY	2mb	43	27	9	4	0	2	1	6.10s

Table 2.2 gives the result of the study conducted on browser technology taking www.cusat.ac.in as example. Here the specified site does not support brotli compression. And also it does not support SPDY protocol.

Table 2.2

BROWSER TECHNOLOGY	PAGE SIZE	NO`OF REQUESTS	LOAD TIME
Normal chrome	1.3MB	88	8.98s
Google weblight	119.9KB	32	375ms
Opera turbo	120KB	30	5s

Table 2.3 gives the result of study conducted on the website with url http://www.wsj.com/articles/from-mars-to-earth-these-landscapes-traverse-time-1473931801,With and without amp.

Table 2.3

Page size	No of requests	Image request	Script request	Css requset	Html request	Plain text requset	Other requset	Load time
302.8 KB	32	8	13	3	3	3	2	912ms
2.3 MB	229	58	82	4	15	2	20	4.97 s

AMP
NORMAL

Table 2.4 summarizes the performance of two websites that use and does not use brotli compression.

Table 2.4

URL	Brotli support	page size	requets	Image reuets	Script requets	Css requets	Html requests	Other requets	Load time
www.cusat.ac.in	✗	1.2MB	85	47	23	10	1	1	12.45S
articomic.com	✓	698.8KB	64	28	16	7	2	0	2.76S

5. CONCLUSION

Optimizing page load time is a smart thing to do, to help users get faster to where they are searching for. More time is spent in downloading all the components in the page like images, style sheets, scripts, Flash, etc. Reducing the number of components in turn reduces the number of HTTP requests required to render the page. This is the key to faster pages.

In this paper, we analyzed various techniques such as SPDY, HTTP/2, instant pages, Google weblight, brotli, Polaris, and different browser technologies. And also studied how they reduce the latency of web pages. Prior approach to load web page faster was mostly based on data compression, now things have taken a different turn. New researches like Polaris focus on aggressive page load using dependency graphs, HTTP/2 allows unlimited concurrent streams over a single TCP connection etc. For a new research for faster web pages, this paper will help to understand the existing techniques to a greater extend.

REFERENCES

[1]. Chan, m. c., And Woo, T. Cache-based Compaction: A New Technique for Optimizing Web Transfer. In Proceedings of INFOCOM (New York, NY, March 1999), pp. 117–125.

[2]. Albert Costill. .https://www.searchenginejournal.com/seo-101-jbkbjbjbiohhiohimportant-site-speed-2014/111924 / (2014)

[3]. RaviI Netravali, Ameesh Goyal, James Mickens, and Hari Balakrishnan. Polaris: Faster Page Loads Using Fine-grained Dependency Tracking.(2016).

[4]. James Mickens. Silo: Exploiting JavaScript and DOM Storage for Faster Page Loads. In WebApps'10 Proceedings of the USENIX conference on Web application development(2010) .

[5]. https://github.com/ampproject/amphtml/tree/master/src

[6]. https://developers.google.com/amp/cache/.

[7]. Flanagan, D. JavaScript: The Definitive Guide, 5 ed.O'Reilly Media, Inc.(2006).

[8]. http://www.iamai.in/media/details/4620 (New Delhi,2015).

[9]. Spdy White Paper. https://www.chromium.org/spdy/spdy-whitepaper (2012).

[10]. Angeliki Zavou, Elias Athanasopoulos, Georgios Portokalidis, and Angelos D. Keromytis. Exploiting Split Browsers for Efficiently Protecting User Data. In Proceedings of the ACM Workshop on Cloud computing security workshop (New York,2012), pp. 37-42.

[11]. Ryan Paul. Amazon's Silk Web browser adds new twist to old idea. http://arstechnica.com/gadgets/2011/09/amazons-silk-web-browser-adds-new-twist-to-old-idea/ (28,october,2011).

[12]. Dev Shea. http://alistapart.com/article/sprites. (05,March,2004).

[13]. Objects, Images, and Applets. http://www.w3.org/TR/html401/struct/objects.html

[14]. Errata Exist. https://tools.ietf.org/html/rfc2397 (August,1998).

[15]. Jyrki Alakuijala, Evgenii Kliuchnikov, Zoltan Szabadka, and Lode Vandevenne. Comparison of Brotli, D https://cran.r-project.org/web/packages/brotli/vignettes/brotli-2015-09-22.pdf.
[16]. Balachander Krishnamurthy, Jeffrey C. Mogul, David M. Kristol.KeyDifferences between HTTP/1.0 and HTTP/1.1.http://www8.org/w8papers/5cprotocols/key/key.html .
[17]. Rohit Khare, Ian Jacob. W3C Recommendations Reduce 'World Wide Wait'.https://www.w3.org/Protocols/NL-PerfNote.html .
[18]. M. Belshe, R. Peon, M. Thomson. Hypertext Transfer Protocol Version(HTTP/2). https://http2.github.io/http2-spec/ . (2015).
[19]. Instant Articles, https://developers.facebook.com/docs/instant-articles