International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-10 Issue-2, July 2021

Web Browsing with Edge Computing

Abhishek Rastogi, Shashank Vats, Shivam Pundir, Ramender Singh

Abstract: ***Webpages have become increasingly complex in recent years, with longer loading times to match. This paper uses tailored edge computing to address this issue. As is customary, A grip server interacts with cloud web servers in edge computing. In A footing server, on the other hand, is a personalised edge computing system. referred to as a foothold The Server in the Middle (ESM) collaborates with other servers. users' cell phonesThis research focuses on two strategies based on personalised edge computing: edge aided caching and edge aided reprioritizing. Edge-assisted caching decreases the time it takes for a page to load. Because an ESM saves the cached data on mobile devices, So far, we've got components. Edge helps in the reprioritization of forces on the internet. browser to show visual components earlier and lowers the amount of white space Time spent in front of a screen.In addition, the ESM uses HTTP/2 rather than HTTP/1.1. This decreases the number of interactions between a mobile device and, as a result, the ESM, allowing advanced functionalities to be used. such as priority and server push Edge-assisted caching has been implemented. built in a high-end PC for Google's web browser Chrome for Android is a mobile web browser. Edge aided in an experiment, according to the results. The time it took for a popular website to load was cut in half because to caching. 59 percent in a network that is extremely congested.Another experiment found that edge-assisted reprioritization cut the white screen time of a webpage with a lot of photo photos by 21%. edge computing, reprioritization, mobile device, index terms browsing the web, caching***

Keywords: ***This decreases the number of interactions between a mobile device and, as a result, the ESM, allowing advanced functionalities to be used.***

I. INTRODUCTION

Because the front-end of the internet has become more sophisticated in recent years, it has taken longer for web pages to load. According to information from HTTP Archive, the median data transfer size rose by 738 percent from May 2011 to May 2019 [1].

Manuscript received on June 27, 2021.
Revised Manuscript received on July 09, 2021.
Manuscript published on July 30, 2021.
* Correspondence Author
Abhishek Rastogi, Pursuing, B.Tech final year, Department of Information Technology, Meerut Institute of Engineering and Technology, Meerut, Uttar Pradesh 250005, India. Email: abhishek.rastogi.it.2017@miet.ac.in
Shashank Vats*, Pursuing, B.Tech Final year, Department of Information Technology, Meerut Institute of Engineering and Technology, Meerut, Uttar Pradesh 250005, India. Email: shashank.vats.it.2017@miet.ac.in
Shivam Pundir, Pursuing, B.Tech Final year, Department of Information Technology, Meerut Institute of Engineering and Technology, Meerut, Uttar Pradesh 250005, India. Email: shivam.pundir.it.2017@miet.ac.in
Ramender Singh, Associate Professor, Department of Information Technology, Meerut Institute of Engineering and Technology, Meerut, Uttar Pradesh 250005, India. Email: ramender.singh@miet.ac.in

Web browsing on mobile devices such as smartphones and tablets, on the other hand, accounts for more than half of all traffic [2]. Because of their lower performance, mobile devices take longer to load pages than desktop environments It's also worth noting that the current website transmission procedure isn't always optimal for mobile devices, resulting in slower page loading times. Researchers have recently been working to improve the online loading process using cloud computing technology by offloading heavy processes from mobile devices to the cloud [3]. A cloud server, for example, runs JavaScript code on behalf of mobile devices in cloud-based browsers like Opera Mini. A cloud server reprioritizes webpage components in Paper [6] so that key components appear first on mobile devices.However, due to the significant delay between a cloud server and a mobile device, many cloud-based solutions provide a very limited benefit [7]. We address concerns with edge computing technology in this study. Edge computing offers the advantage of reduced latency between mobile devices and processing nodes, which makes it superior than cloud computing. In addition, we rely on the idea of tailored edge computing. Figure 1 shows the differences between cloud computing, traditional edge computing, and personalised edge computing. Service providers run servers in cloud data centres in cloud computing. In traditional edge computing, the servers run on the edge nodes and collaborate with service providers.

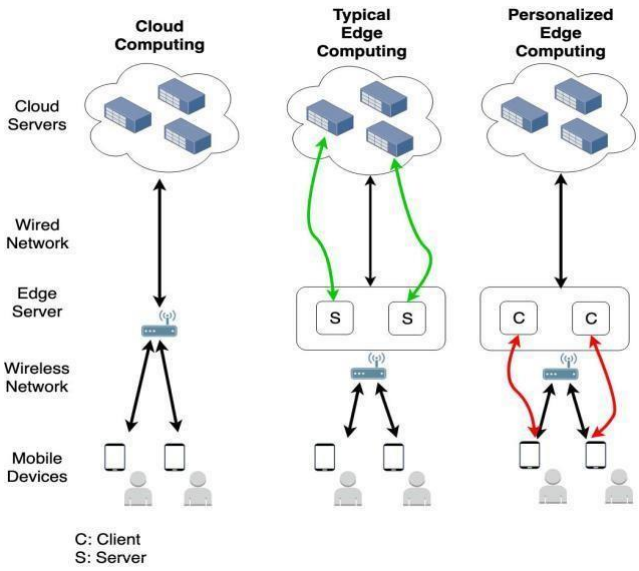


Fig. 1. Comparisons among cloud computing, typical edge computing and personalized edge computing.

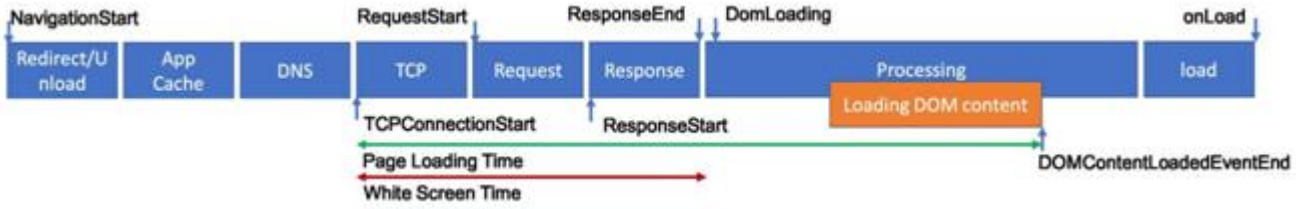


Fig. 2.Webpage loading timeline in modern browsers

There are various advantages to using this strategy. For starters, we'll employ ESMs that are both powerful and always connected to enhance the online browsing experience. Second, we'll use our strategies on any web servers without relying on them. Finally, we can employ idle resources such as personal computers (PCs) and broadband routers to receive data. We concentrate on the following two strategies based on personalised edge computing: Edge made caching easier (EAC). Because an ESM keeps the cached components up to date, the page loading time on mobile devices is reduced

Re-prioritization was made easier with the help of Edge (EAR). This forces a web browser to display visual components sooner and decreases the amount of time spent on the white screen. Within the ESM, we also use HTTP/2 rather than HTTP/1.1. This decreases the number of exchanges between a mobile device and, as a result, the ESM, allowing advanced capabilities like server push and priority to be used. We implemented edge-assisted caching in the real world using an Android phone and a PC. We employ cache partitioning to protect consumers' privacy, unlike earlier experiments. We demonstrated the effectiveness of edge-assisted caching for a well-known website using four different network scenarios. We proved the utility of edge-assisted repriori-tizing on a webpage with multiple photo images. We've enabled HTTP/2 capability on legacy web pages that only support HTTP/1.1. II and IV, respectively. Section V illustrates and assesses the implementation of our proposed approaches as a position node in a PC. Section VII brings the report to a close by displaying the onger-term projects.

II. GOALS OF RESEARCH AND PROPOSED METHOD

- This section clarifies our study objectives and provides an outline of our proposed technique. In the browser, the time it takes for a page to load and the time it takes for a white screen to appear When a web browser displays a webpage to a user, it goes through a series of processes. It establishes TCP connections and monitors them. User partitions in an ESM (Fig. 3). Using Hyper Text Transfer Protocol (HTTP), it requests data from web servers, receives replies from web servers, builds a Document Object Model (DOM) tree and a Cascading Style Sheets (CSS) DOM tree, runs JavaScript files, and so on. Many events can be triggered in these steps by modern web browsers. as depicted in Fig.2. The following are significant events: Start: The browser loads the page

We made edge using an Android phone and a computer. The remainder of the paper is laid ou as follows. Section II outlines our objectives as well as the definitions of key variables used in our study. Edge aided caching and edge aided reprioritizing are discussed in depth in Sections

Request Start: When the browser starts making requests to web servers, it is known as RequestStart.

Response Start: The first byte of data sent by the web servers is received by the browser.

End Response: The browser collects all of the information sent by the web servers. The contents of the screen are visible at this moment.

Dom Loading: The browser completes the HTML file parsing and begins creating the DOM tree.

DOM Content Loaded indicates that the HTML document has been fully loaded and parsed by the browser.

- Two variables are defined in our study:
- The time between TCPCon-nectionStart and DOMContentLoadedEventEnd is the Page Loading Time.
- The time between TCPConnec-tionStart and ResponseEnd is known as the White Screen Time. This is the amount of time it takes for consumers to view the first thing that appears on their screen after typing in the URL.

With personalised edge computing, we suggest approaches for reducing these delays.

B. Personalized Edge Computing Server Nodes

Typical edge computing employs server nodes at the wired network's sting, as defined in Section I. Network operators can add server nodes in 5G base stations, for example, in 5G communication. These server nodes, like those in cloud computing, can host several tenants.

A user of a mobile device uses a partition in such a shared server node in personalised edge computing, as shown in Fig.3. Each user has his or her own storage space as well as an access token for his or her partition. This will be viewed as an increase in the storage capacity of his or her mobile device. During this time, the user can save sensitive information such as cookies and logged-in pages. We believe we can assist such a user in obtaining a partition in a heavily shared server node.

In personalised edge computing, unlike traditional edge computing, a user can additionally use the subsequent private resources.

A personal computer that uses Ethernet to connect to a home network.

A router for your home.

A partition of a shared server node or a private resource is referred to as an ESM in this work. In Section V, we use a personal computer to test the effects of an ESM.

C. Target Web Pages

On the Internet, there are many different types of webpages. In this work, we aim to improve surfing experience by focusing on the following types of web pages.

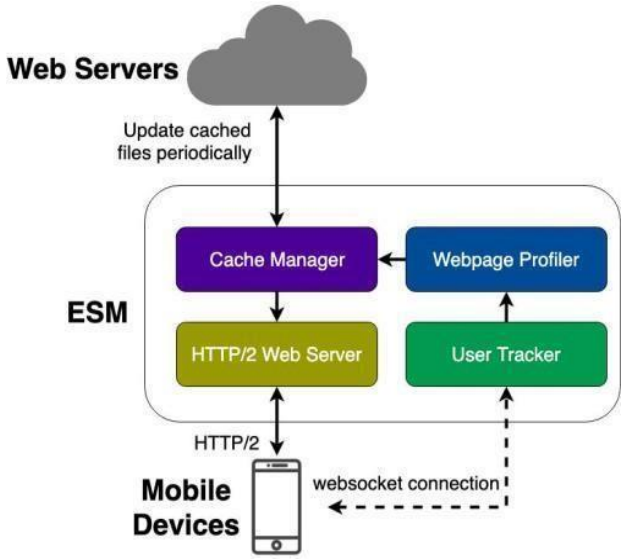


Fig. 4. Edge aided caching architecture.

Small data transfers result in higher energy usage and a longer time for a page to load [4], [9].

HTTP Version 2 (HTTP/2) was introduced in 2015 to address the issues with HTTP/1. HTTP/2 decreases the number of interactions and speeds up page loading . However, as of June 2019, just 38% of all websites on the internet had used HTTP/2 [10]

An HTML file, stylesheets, and picture components make up a single webpage. Web servers can generate these files dynamically.

A webpage can include JavaScript files that allow users to interact with it, such as displaying menus.

Complex webpages, such as single page applications (SPAs) and progressive web applications, are not discussed in this study (PWAs). We allow users to browse these webpages, but we don't improve their browsing experience.

D. Edge Aided Caching and Edge Aided Reprioritizing

We propose two strategies in this paper: edge helped caching and edge aided reprioritizing. Edge-assisted caching eagerly caches files at the webpage level, reducing the time it takes for a page to load. Edge-assisted reprioritization rearranges transferred files in order of priority, reducing white screen time. We employ HTTP/2 between mobile nodes and the ESM to reduce the number of connections required to download files.

E. Delivering Files to Mobile Devices using HTTP/2

Nowadays, the vast majority of webpages are transferred via HTTP/1.1, which is a stateless protocol. When a single webpage is made up of several small components, loading it results in an excessive amount of HTTP exchanges. a rush of data in an extremely mobile device environment with Wi-Fi or 4G

III. EDGE AIDED CACHING

We used the following four pieces to develop our first strategy, edge aided caching, in an ESM (Fig.4).

The user tracking system.

The profiler for web pages.

The cache administrator.

The web server that supports HTTP/2.

A. User Tracker

The user tracker communicates with a mobile device and tracks the mobile device's webpage accesses. The mobile device provides access information to the user tracker whenever a user opens a webpage. This data comprises the following:

A URL that has been visited.

A date and time mark. This keeps track of when the activity takes place.

A user token is used to verify a user's identity.

The user tracker keeps track of how many times each URL and each user of the mobile device is accessed. The user tracker adds the URL to a watch list if the quantity surpasses a threshold for a webpage in a long period of time. The user tracker removes the URL from the watch list if the amount falls below the threshold within the time frame.

We create a Google Chrome plugin that uses Web Socket to communicate with the user tracker. Using Google Chrome's

Web Request API, the extension keeps track of user activity.

B. Webpage Profiler

As described in paper [11], the webpage profiler scans webpages and creates attributes for them. It gives these attributes to the cache management so that the cache can update itself. The following information is contained in the attributes:

The update gap is the duration between two cached file update attempts.

The number of unaltered update attempts of a cached file is the unmodified times. The MD5 digest of the cached files is the cache digest. This is used to determine whether or not the cache has been changed.

C. Cache Manager

Cache files are kept at the website level at the cache manager. If a webpage includes components, such as style sheets and images,

They are cached together with the webpage's HTML file. Components of single webpage are frequently provided from many sources. In this scenario, the cache manager additionally keeps track of all the components of a single webpage, as well as the html file.

Until now, the cache manager has kept the cached components of web pages on the watch list. It eagerly caches webpages and stores the caches in the database. Unlike traditional edge computing, the ESM continues to function on behalf of a mobile device even while it is offline. Web servers can't tell the difference between an ESM and a mobile device.

Because the cache manager sends requests to web servers on a regular basis, it's critical to figure out how long it takes between queries. It's obvious that different types of websites evolve at different speeds. A news website, for example, indicates that the webpage is updated frequently. The cache manager closes the gap in this situation.c

For privacy and security reasons, we don't allow users to share their cache. A malicious web server can track how long it takes the user's browser to accomplish specific tasks and deduce whether the user has recently visited unrelated websites. [12], [13], [14], [15], [16]. Similarly, if many users share web caches, a user might infer the activity of other users by monitoring the time it takes for a page to load. Because memory page sharing is banned in multi-tenant cloud servers, we deactivate cache sharing to prevent this type of attack [16] [17] [18].

D. HTTP/2 Web Server

The ESM's web server connects with the mobile device via HTTP/2, and cached components are sent to the device.

HTTP/1-based websites can benefit from HTTP/2 since the ESM always connects to mobile devices using HTTP/2. We utilise HTTP/2's severe push to allow mobile devices to load components before parsing HTML pages.

IV. EDGE AIDED REPRIORITIZING

We discussed caching within the ESM in the previous section. The ESM also has the ability to reprioritize components on a webpage. We want to reduce the amount of time spent on the white screen during this optimization. Users perceive that webpages run faster when the white screen duration is reduced, even though the page loading time remains unchanged.

An example of edge-assisted reprioritization may be seen in Figure 5. An HTML file, two JavaScript files, one layout stylesheet, and two pictures make up a basic webpage. The time it takes for the jobs that process these files to complete is shown in Table I.

The webpage is processed as indicated in Fig.5 when we don't use edge assisted reprioritizing (EAR off) (a). The following are the page loading time (PLT) and white screen time (WST) for this example.

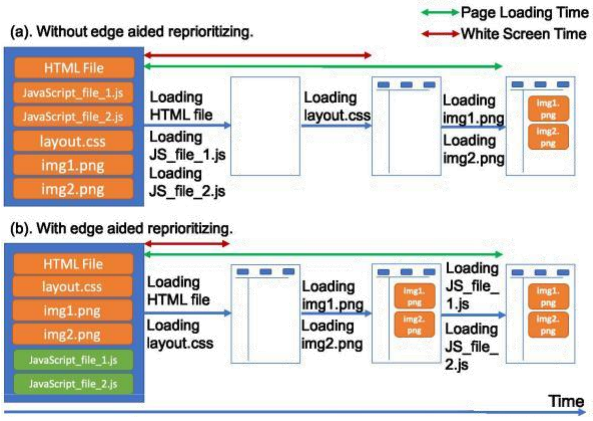
$$PLT_{ear\ of\ f} = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 \quad (1)$$

$$WST_{ear\ of\ f} = t_1 + t_2 + t_3 + t_4 \quad (2)$$

$$PLT_{ear\ on} = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 \quad (3)$$

Table I Task Time Consumptions

Task Name	Time Consumptions
HTML File	t_1
JS___file __1.js	t_2
JS file_2.js	t_3
layout.css	t_4
img1.png	t_5
img2.png	t_6



- File types.
- File sizes.
- Dependencies. For example, showing an image can depend on analyzing a stylesheet.
- Critical rendering path. This is the essential path to render HTML, stylesheets and JavaScript files into the webpage shown on the screen.

Using these attributes, the reprioritizer gives components priorities as follows.

1) Stylesheets for page layout have the highest priority

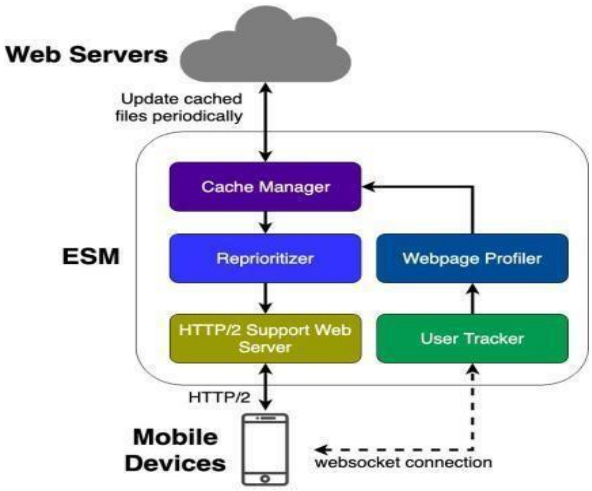


Fig. 5. Shortening the white screen time with edge aided reprioritizing.

1) The webpage is processed as indicated in Fig.5 when we don't use edge assisted reprioritizing (EAR off) (a). The following are the page loading time (PLT) and white screen time (WST) for this example.
2) The second greatest importance is given to images and icons at the top of a webpage. The amount of pictures displayed is determined by the screen sizes of mobile phones and websites.
3) JS files are given the lowest priority since they frequently obstruct the display of the DOM tree.

Take a look at the HTML file below, which contains references to one of the stylesheets, two JS files, and two pictures.

```
<head>
    ...
    <script
    src="JS_file_1.js"></script>
    <script
    src="JS_file_2.js"></script>  <link
    rel="stylesheet"
    href="layout.css"> ...

</head>
<body>
    <img src="img1.png">
    ...
    <img src="img2.png">
    ...
</body>
```

The prioritizer rewrites this HTML into the following one.

```
<head>
    <link rel="stylesheet" href="layout.css">
    ...
    <script  defer  src="JS_file_1.js"></script>
    <script  defer  src="JS_file_2.js"></script>
    ...

</head>
<body>
    <img src="img1.png">
    ...
    <img src="img2.png">
    ...
</body>
```

The reprioritizer in this case puts the CSS line to the top of the HTML head> element. The stylesheet is given top priority as a result of this. The property ”defer” is added to the script> tag by the reprioritizer. Fig. 6 shows the result of this. Reprioritizing architecture was made easier with the help of Edge.

Files containing JavaScript are given a lower priority. The image-related sections, on the other hand,

The reprioritizer gives the pictures a higher priority than the HTML.

Despite the fact that the PLTs are the same in both situations, the user feels that the second case's homepage loads quicker. WST is a shortened version of WST.

W STear on = t1 + t4 (4) is a high-priority task. The HTTP/2 web server will utilise this later.

As an extension, we created edge-assisted reprioritization.

We introduce a to the edge assisted caching as illustrated in Fig.6. between the cache manager and the reprioritizer

The website profiler is extended by an HTTP/2 web server.

We are choosing to rewrite HTML files because some priority features in the HTTP/2 specification did not work in the targeted browser, Google Chrome. When the target browser adopts a better priority feature, we will use it.

The reprioritizer uses information received from the webpage profiler and gives priorities to components. The extended webpage profiler calculates the following attributes.

Using HTTP/2's server push capability, the HTTP/2 web server delivers modified HTML files to mobile devices together with high-priority components. Before the client sends requests for the stylesheet and photos, the server pushes the stylesheet and two images combined with the HTML file.

V. IMPLEMENTATION AND PERFORMANCE

A. Environment and Implementation

Within the environment described in Table II and Fig.7, we developed sting-assisted caching. As an HTTP/2 web server, we utilised Node.js. We utilised a Nexus 5X smartphone as a mobile device and the Google Chrome browser with our user tracker plugin. Within the ESM, we utilised MongoDB to store cache components in a database. Between the trial area and the Internet, we placed a congested network emulator. This emulator was used.

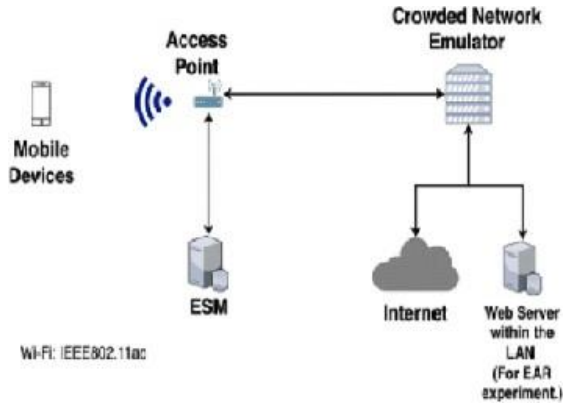


Fig. 7. Test environment.

Table II Experiment Environment

Computer	Configurations ESM
Mobile device	Intel Core i7/32GB/Ubuntu 18.04 LTS/node.JS v8.11.04 Nexus 5X/Android 6.0.1/Google Chrome v73.0

Linux, Ubuntu 18.04, and it worked like a bridge, with a limited-bandwidth and reduced to the packet transmission time. We are in three networking tools, like the one shown in Table III. I love all of these parameters are dependent on the gas settings the Google Chrome developer tools as a flap [19]. We are using the Linux traffic control commands to manage throughput, and latency.

- Performance of Edge Aided Caching Previous studies have shown that the average frequency of the user's browsing on the web is required, with the Zipf law [20]. For the purpose of determining the optimal web caching is the threshold, and we took to the user (the author of this article, web page views as of May 1, 2018. up to May 17, 2019. Over 381 days in, and the user is allowed to access the web-5817 times 2470 the unique Url.. We have selected for 2, 3, 4, and 5, as well as the thresholds and evaluated, and the cache hit rate and the use of the storage space. The contact rate is calculated as follows:

VI. RELATED WORK

On the basis of the final computation, several web caching and preloading methods have been developed. EdgeBuffer [21], for example, pre-determines the content of fragments and has the capacity to forecast the network to which the device is travelling for a range of mobile devices. [22] proposes a paradigm for cache management that is both in-memory and based on Bayesian reasoning. It is to be considered in [23] caching on many nodes, as well as sharing the cache between various devices. The research on edge caching were based on the model, and the goal was to enhance the cache hit rate. We'll show you how the performance looks in real life in this post.combining an Android phone and a computer We're attempting to decrease the amount of time it takes for a website to load as a blank screen. Furthermore, current caching strategies must consider the potential for a breach of your privacy and security is in the process of sharing the cache. In accordance with Section III. We don't enable the cache memory to be shared between user privacy and security in C. Google's Accelerated Mobile Pages (AMP) [24] is a web server that provides a restricted alternative to HTML and JavaScript on a mobile device.This service needs the developer to rebuild the site based on their own constraints, and therefore does not improve the performance of outdated web servers. Our current technologies are designed to function without the need of web servers and do not require any changes to the web server.

VII. CONCLUSION

In this post, we'll look at two approaches: the cache's edge and the national review. Edge-backed caching, such as the European Stability Mechanism, to automatically cached content, to speed up page loading on mobile devices. Reprioritizing, which is enabled by Edge, enables the browser to show the visual components while reducing the screen size from time to time. It COULD potentially be used instead of HTTP / 1.1 in HTTP/2. This will decrease the amount of contact between the mobile device and the European Stability Mechanism, and will give precedence to features such as push-based servers.We're setting up for each individual user to address privacy issues. Each user has access to the whole path of the company's class, borderline, which can be used to store personal data such as cookies and log in to the site.

We also have edge-backed caching, the free PC is a boundary node, and we're now utilising the border version, which has assisted with reprioritization.

The testing findings reveal that edge-assisted caching reduces the load time of sites and the most popular pages, and that 59 percent of the time, the network was congested. Another experimental result shows that the structural changes made with the help of the rim to reduce the screen time of a web page with a large number of images to reduce the screen time of a web page with a large number of images to reduce the screen time of a web page with a large number of images to reduce the screen time of a web page with a large (21 percent).

We'll keep evaluating edge-assisted caching in the future, including the impact of the relocation and the usage of ESM money. Automatic cross-references must also be implemented.

REFERENCES

1. [Online] HTTP Archive. Reports may be found at https://httparchive.org/reports/. [Accessed: 06 August 2019]
2. The percentage of global
3. mobile phone website traffic.
4. [Online]. http://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share [Accessed: 15 April2019]
5. “Clonecloud: elastic execution between mobile device and cloud,” by B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, in Proceedings of the sixth conference on computer systems. ACM, pp. 301–314, 2011.
6. “PARCEL: Proxy aided browsing on cellular networks for energy and latency reduction,” in Proceedings of the 10th ACM International Conference on Emerging Networking Experiments and Technologies, A. Sivakumar, S. Puzhavakath Narayanan,
7. V. Gopalakrishnan, S. Lee, S. Rao, and S. Sen. 325–336 in ACM, 2014.
8. Opera Mini is a mobile-friendly browser. [Online]. [Accessed: 2019-08-01] Available at: https://www.opera.com/mobile/mini/android
9. “Klotski: Reprioritizing web content to improve user experience on mobile devices,” in 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15), pp. 439–453. M. Butkiewicz, D. Wang, Z. Wu, H. V. Madhyastha, and V. Sekar, “Klotski: Reprioritizing web content to improve user experience on mobile devices,” in 12th USENIX Symposium
10. “Cloud is not a silver bullet: A case study of cloud-based mobile browsing,” in Proceedings of the 15th Workshop on Mobile Computing Systems and Applications, A. Sivakumar, V. Gopalakrishnan, S. Lee, S. Rao, S. Sen, and O. Spatscheck. ACM, pp. 21–26, 2014.
11. The time it takes for current browsers to navigate. [Online]. Visit https://www.w3.org/TR/navigation-timing/ for further information. [Accessed: 03 August 2019]
12. “Characterizing resource use for mobile web browsing,” in Proceedings of the 12th annual international conference on Mobile systems, applications, and services, F. Qian, S. Sen, and O. Spatscheck. Pages. 218–231, ACM, 2014.

13. HTTP/2 has a significant Percentage of global Internet traffic. [Online]. https://w3techs.com/technologies/details/ce-http2/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all/all [Accessed: 14 June 2019]
14. “Demystifying page load performance with wprof,” in Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), pp. 473–485. X. S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall, “Demystifying page load performance with wprof,” in Presented as part of the 10th USENIX Symposium on Network

AUTHOR PROFILE

Abhishek Rastogi, I am currently pursuing B.Tech final year in Information technology from MIET, Meerut. I have completed my 10 and 12 from Genius al-Falah inter college from Meerut. I have done many courses from my reputed universities in my graduation like Salesforce, SAP, core java, and web development. I have Salesforce admin certification issued by salesforce.com. I have certification in core java. I got many certifications in my college time from college. I have done many projects in salesforce technology and also created an Education learning website in web development. Shivam and Shashank helped me in this project. and the main aim of our project is to build every student can learn new technologies easily. Coming to my interest I love travelling and cooking. I got many certification in my school.

Shashank Vats, I am currently pursuing B.Tech Final year in Information Technology from MIET, Meerut. I have done various online courses in my graduation- Programming In java from NPTEL. Machine Learning from Coursera. I have developed my final year project called "Web Browsing with Edge Computing' with my friends Abhishek Rastogi & Shivam Pundir. Currently, I am doing internship as Software Developer at Astrea It Services, Noida. We build this project as a motive to learn and experience browsing with Edge Computing. Coming to my interest, I have good knowledge of Salesforce and java and also done a project in both of them.

Shivam Pundir, I am currently pursuing in B.Tech Final year in information Technology from Meerut Institute of engineering and Technology , meerut. I have completed my schooling from Divya Public School, Thanabhawan. I have done many courses from the reputed organization in my graduation time which is given below: I have completed one course of python in June 2020 which is “Python Data structure” from coursera. I have completed one python course from cisco in july 2020 . Getting knowledge of server side development from cousera ,in july 2021. I also developed my final year project called e-learning website along with my friend shashank and abhishek. I got one certificate on SAP. And I am always like outdoor games like football, cricket etc. I got many certification in my school.

Mr. Ramander Singh, B.E,M.Tech, My Name is Ramander Singh ,I have done B.E and M.Tech and currently a Associate Professor In Meerut Institute Of Engineering & Technology in Information Technology Department. I have given guidance in Web Browsing with Edge Computing Project to my Students Shashank Vats, Shivam Pundir and Abhishek Rastogi. The main motive of this project is to explore huge advantage of Edge computing and use them for purpose.

Retrieval Number: 100.1/ijrte.B61870710221
DOI: 10.35940/ijrte.B6187.0710221
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication