International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

Packyard- Data Compression Proxy for Mobile Web

Arpit Agarwal[1], Gargi Gupta[2], Harshta Mishra[3], Manoj Kumar[4]

[1] *UG Scholar, Dept. of IT, GCET, Greater Noida, UP, India*
[2] *UG Scholar, Dept. of IT, GCET, Greater Noida, UP, India*
[3] *UG Scholar, Dept. of IT, GCET, Greater Noida, UP, India*
[4] *Assistant Professor, Dept. of IT, GCET, Greater Noida, UP, India*

Abstract - *Data Compression Proxy for the Mobile Web is an important field for research because Mobile devices in the present world form a major part in the usage of internet access technology .Feature-rich devices with faster speeds, smaller sizes and multiple amount of applications continue to impel explosive incessant growth for mobile Internet traffic. But still, high rated data plans, excessive loading time of the web page and the bandwidth gap are the reasons of widespread concern and disappointment among the users. This paper presents PACKYARD which is a proxy service. Its main objective is to extend the usage of data plans. Its working involves compression among two parties i.e. origin servers and client browsers. Image compression, minification, reducing the size of 404 error web page and Accelerated Mobile Pages i.e. AMP are the few techniques that we are using for the compression of web page. The AMP is an open-source initiative. It compresses the web page to a great extent thus making it smoother and faster. It aims to provide mobile optimized content that can load instantly everywhere and thus enables the creation of websites and ads that are consistently fast, beautiful and high-performing across devices and distribution platforms. The end of this paper suggests some future advances on to how we can minimize the size of the web page for the mobile phones.*

Key Words: Data compression, proxy, image compression, minification, amp

1. INTRODUCTION

Our main focus is on mobile devices because they are fast becoming the dominant mode of Internet access. This paper gives a detail account of our experience building and running a mobile web proxy for the mobile web users supporting billions of requests per day. This process of developing and deploying this system, gave us a deep understanding of modern mobile web traffic and the challenges faced in delivering good performance. Mobile devices are becoming the dominant mode of internet access is clear from the trends: In many markets around the world, mobile traffic volume already exceeds desktop [2]. Though the trends shows mobile devices dominant in near future still web content is mainly designed for the desktop browsers. High mobile data rates make the conditions even more badly.

Mobile Internet usage is growing at an exponential rate. The massive growth of mobile Internet traffic has created a tremendous opportunity for automatic optimization. In Asia and America, 40% of web page views are performed on mobile devices as of May 2016, a year over-year increase exceeding 10% [3]. In North America, mobile page loads are 19% of total traffic volume with 8% growth yearly. In February 2014, research firm comScore reported that time spent using the Internet on mobile devices exceeded desktop PCs for the first time in the United States. Mobile is increasingly dominant. In the process of developing and deploying this system, we gained a deep understanding of modern mobile web traffic.

Growth in emerging markets is hampered by cost. Emerging markets are growing faster than developed markets. Year-over-year growth in mobile subscriptions is 26% in developing countries compared to 11.5% in developed countries. In Africa, growth exceeds 40% annually [4]. Despite the never ending rise in popularity, the high cost of mobile access frustrates the mobile user. One survey of 16 countries in Saharan region of Africa reports that mobile phone spending was 15-26% of individual income in the lower 20-30% income bracket [5].

For example, when researched on a wide population of United States analyzing their trends of usage of internet on Mobile devices, it accounted for 35% in January 2016. It was figured out that Apps made up to a large section of 45% of Internet traffic and 8% of traffic was bagged by mobile browsers, according to data from comScore, cited by research firm.
This was one of the remarkable times of history that the total Internet usage on mobile devices has outweighed that on PCs. In emerging markets, data access is often priced per-byte at more than an average cost, consuming up to 25% of a user's total income. In the midst of these costs, our main aim is to support the continued exponential growth of the mobile Internet and web browsing in particular.

Although there is a huge increase in a number of sites that are made for mobile device, there is still a huge chance of saving users money by compressing the web content by using proxy and compression techniques [6]. Packyard uses the WebP image compression format which improves

efficiency, yet is rarely used by website operators because they require cumbersome, browser-specific configuration.

This paper gives us some important contributions such as our experience with Packyard has given us a deep understanding of the performance issues with proxying the modern mobile web. Although proxy optimization delivers crystal clear benefits for data reduction, its impact is mixed. It has made us to explore the concept of AMP – Accelerated Mobile Pages.

2. EXISTING METHODOLOGY

Packyard was built in response to the practical obstructing blocks of today's mobile web. Ideally, Packyard would not be needed if mobile data would be cheap, and content providers would be quick to adopt new technologies. But neither is true today.

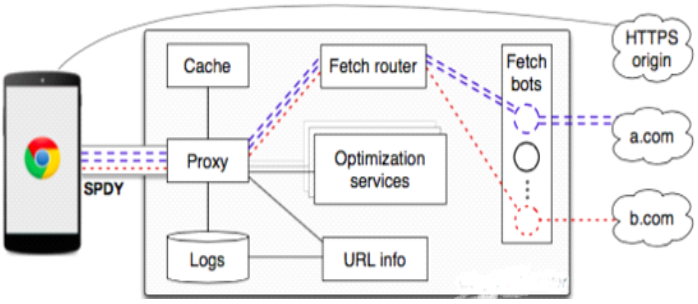
2.1 Optimisation Techniques

These technologies are the ones used in flywheel and we have also used it for our Packyard. The optimization techniques reduce mobile data usage for web traffic.

1. Image Transcoding – WebP follows lossless and lossy compression for images on the web. WebP helps in getting smaller and good quality richer images that makes the web smoother and faster. WebP lossy as well as lossless images are 35% smaller than comparable formats such as JPEG and Png images [7]. Packyard transcodes image responses to the WebP format, which provides roughly 35% better compression than JPEG. Animated GIF images are transcoded to the animated WebP format. It occurs very rarely that the transcoded WebP image is larger than the original, in which case we serve the original image instead.

2. Minification - Readability of code is not an area of major concern for web browsers when it comes to generating a page or running a script. All the data that is considered irrelevant code files is removed by the minification. Unlike old ways of compression techniques, minified files don't need to be decompressed before they can be read or executed [8]. It is performed after the code for a web application is written, but prior the application is implemented. The minified version is sent instead of the full version. Hence, minification results in faster response times and lower data cap costs. It removes unnecessary whitespaces and comments.

3. Preconnect and Prefetch - Preconnect and prefetch are also performance optimizations that reduce the 2 way round between the client and origin server. The important aspect is that proxy has the ability to make sure what the client soon will be requesting while it receives a response back from the client. For example, image, CSS links embedded in HTML and JavaScript will likely to be requested after the HTML is delivered. Packyard resolves HTML and CSS responses as they are served in order to discover subresource requests.



3. PROPOSED TECHNIQUES

The optimization techniques proposed in this paper focuses on two major techniques which can help reduce data to much larger extent than the previous optimization techniques.

3.1 Lightweight error pages

While browsing on the web many requests from clients result in a 404 error response from the origin, for example, due to a broken link. But, in many cases the 404 error page is not shown to the user. Like, Chrome automatically requests a small preview image called as favicon when navigating to a new site, which results in a 404. These error pages can be quite large, averaging 3.2KB—a fair number of "invisible" bytes for each page load lacking a favicon. Packyard returns a small (68 byte) response body for favicon.

3.2 AMP

In this paper we have made use of Amp short for Accelerated Mobile Pages. If you browse the web today on mobile, in today's world you expect rich graphics, smooth scrolling, fast animations and the good news is that on a platform and browser level, we are making a lot of great progress towards faster and smoother websites. The bad news however is that even with these improvements, turn into speedy experiences. This required of a simple and elegant solution, a new way to implement and ensure beautiful, streamlined fast content web pages without all the extra clutter. And it's built on the openness of the web and doesn't try to replace it which allows everyone to participate and collaborate. The Amp project to a large extent improves the performance of websites on the mobile, often to the point where their load appears to be instant and quick. In fact an html page is just a normal html website with a couple of restrictions and extras. In fact, if your website doesn't use custom JavaScript and is mostly static, you probably don't even have to create a second version of it for AMP. AMP JS that is AMP Javascript library is included in each and every AMP document and it

delivers optimal performance with the addition and validation of a few important rules in the markup. Looking at an AMP document the biggest difference one can see is that some elements, like the image tag are replaced with custom elements. For example, <amp-img>……..<amp-img>. Replacement of HTML tags with custom elements is done for two reasons. First it allows AMP JS to control the entire load chain and prioritize certain elements and requests over others. In practice, this means that most third party content and elements below the fold are loaded after the main content arrives, so the users can start reading as soon as possible. Second AMP'S custom properties strictly require the dimensions like width, height, or other aspect ratio defining attributes to be set. This way AMP JS knows exactly how your page looks like before any assets are loaded, and thus is able to lay out the page in advance. This prevents the famous flash of on – site content, the ugliness of half loaded website that then starts to jump around while loading more stuff, as well as need to re-render and do additional layout calculations, a browser task that can be very slow.
Implementation-Packyard checks for if a particular site's AMP version is available, if available it loads it and no other optimizations are performed but if not available other optimizations like lightweight error pages, minification, prefetch and preconnect and image transcoding are performed.

4. RESULTS

Our evaluation of Packyard focuses on data reduction and performance. All the techniques resulted in reducing data usage.

1. Webp quality - As images dominate our workload, the optimization through WebP encoding has a significant influence on our overall data reduction. Our main goal is to achieve as much reduction as possible without affecting the perceived quality of the image. Conversion of JPEG, GIF, PNG formats to Webp format saves about 20-30 percentage of data.

JPEG file size: 251.03 KB

WebP file size: 172.82 KB

2. Lightweight error responses - Packyard sends a small (68 byte) response body for 404 errors returned for favicon. Despite the fact that 404 responses for these images constitute only 0.07% of requests, the full error pages would account for 2% of the total data consumed by Packyard users. The average 404page for an apple-touch-icon is 3.3KB, and two such requests are made for every page load. Our lightweight 404 responses eliminate nearly all of this overhead.

3. AMP - It helps save data as AMP pages on Search use 10 times less data than the equivalent non-AMP page.AMP is designed to offer fast browsing on mobile devices. The average AMP page currently loads four times faster than non-AMP pages. An AMP page websites load time is more than 60 percent than the traditional mobile web. For example Scoopwhoop site has its AMP version too and thus the AMP page site loads in much faster time than non-AMP site.

5. CONCLUSION AND FUTURE SCOPE

We have presented Packyard, a data reduction proxy service that provides an average 58% byte size reduction of HTTP content. We found that data reduction is the easy part. The practical realities of operating with geodiverse users, transient failures, and unpredictable middleboxes consume most of our effort, and we report these tradeoffs in the hope of informing future designs.

6. REFERENCES

[1] Victor Agababov, Michael Buettner, Victor Chudnovsky, Mark Cogan, Ben Greenstein, Shane McDaniel, Michael Piatek, Colin Scott, Matt Welsh, Bolian Yin, "Flywheel: Google's Data Compression Proxy for the Mobile Web", Google, Inc. †UC Berkeley

[2] CNN. Mobile Apps Overtake PC Internet Usage in U.S., http://money.cnn.com/2014/02/28/Technology/mobile/mobile-apps-internet/

[3] M. Meeker. Internet Trends 2014, http://pathwaypr.com/must-read-mark-meekers-2014-internet-trends

[4] International Telecommunications Union. The World in 2014: ICT Facts and Figures. http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf

[5] A. Chabossou, C. Stork, M. Stork, and P. Zahonogo, "Mobile Telephony Access and Usage In Africa", African journal of Information and Communication '08

[6] WebP: A New Image Format For The Web, http: //developers.google.com/speed/webp/

[7] WebP: A New Image Format For The Web, http:// developers.google.com/speed/webp/

[8] Robert Gibb. What is minification? https://blog.stackpath.com/glossary/minification