Identifying Web Page Complexity and Recommending Improvement

Chaitali Domde
Department of Computer Science and Engineering,
G. H. Raisoni Institute of Engineering & Technology For Women, Nagpur
Maharashtra, India,
cdomde@gmail.com

Ms.Hemlata Dakhore
Department of Computer Science and Engineering,
G. H. Raisoni Institute of Engineering & Technology For Women, Nagpurs
Maharashtra, India.
hemlata.dakhore@raisoni.net

Abstract-Users being frustrated due to high page load times. Because page load times are directly effects user satisfaction, providers would like to distinguish if and how the complexity of their Web sites affects the user experience. Although there is an extensive literature on measuring Web graphs, Website popularity, and the nature of Web traffic, there has been little work in understanding how difficult individual Web sites are, and how this complexity impacts the client's experience. We proposed a system that can identify a set of metrics to characterize the complexity of Web sites both at a content level and service level. Also the impact of complexity on the user performance and recommending what control measures should take to reduce the complexity.

Keywords—Browsers, Websites, World Wide Web, Internet.

*****

I. Introduction

Since many years, Web pages have become expressively more complex. At first web site used to host text and images, To rich media like Flash and Silverlight now the web pages contain several content type, ranging from video to script performed on the client's device. Additionally, a Web site today fetches content from servers hosted by its providers, and also from a range of third-party services such as advertising agencies, content distribution networks (CDNs), and analytics services. In combination, representation of a single Web page today involves fetching several objects with varying characteristics from multiple servers under different administrative domains.

In contrast, the poor effects of slow Web sites are well known. Users will discard or switch a Web Site due to performance issues. According to recent surveys, out of whole two thirds of users encounter slow Web Site. While abundance of anecdotal proof is that a key factor in slowing down Web Page is the increase in Web page complexity.Official studies on this topic have been limited. Most previous work on Web measurement concentrations on characterizing the Web graph [7], study the network footprint of Web traffic [2]-[5], also studying the rate of change of content on the Web [8]. Although these have contributed to a well understanding of Web usage, they do not examine the Web sites themselves.

In this paper we focuses on two broad questions,First,we count the complexity, called content level complexity of a Web page by means of a broad spectrum of metrics, and we characterize a Web page by the content in rendering like-the number of objects fetched, the sizes of these objects, and the types of content. Also we study the complexity of Web pages relating to the services they build upon. In addition to these we find number of bytes fetched and non-origin content accounts in place of a significant fraction of the number of objects

Our second and the main focusis on the time to download and render a Web page. We find that the total number of bytes fetched to render a Web site is the most dominant indicator of client-perceived load times than the number of objects fetched.

II. Literature Review

Michael Butkiewicz, Harsha V. Madhyastha, and VyasSekar [1] are focuses in these papers on finding the gap in understanding how complex individual Web sites are and how this complexity impacts on the usersperformance. Also characterize the Web site both at content level (like, number and size of images) and service level (like, number of servers/origins). It may happen that some categories are more complex than other such as 'News'. Out of hundred 60% of Web sites fetched content from minimum five non-origin sources, and these give more than 35% of the bytes downloaded. In addition, they examine which metrics are most suitable for predicting page render and load times and catch that the number of objects requested is the most important factor. With respect to variability in load times, however, they alsofind number of servers is the best indicator.

Y. Zhang, H. Zhu, and S. Greenwood [6] discuss about navigability. Navigability has become the axis of website designs. Existing mechanism haveproblem into two types. The major is to assess and measure a website's navigability in contrast to a set of principles. Another is to evaluate usage data of the Website. A metric methodology to Website navigability measurement is studies in this paper. Objectiveness and the probability of using automated tools to assess extensive websites are advantages of navigability metrics as far the existing valuation and analysis techniques.

Axiomatic assessment

Weyuker's axioms of software complexity have been frequently applied in place of a method to authorizing A logically the measurement of software complexity .In this section, they measure the metrics well-defined in the prior section compared to Weyuker's axioms of software complexity. Weyuker's axioms are established on a number of operators and relations on programs. According to the features of websites these operators and relations must be modified.

M. Lee, R. R. Kompella, and S. Singh [3],in this paper discuss on Cloud-based Web applications driven through new knowledge such as Asynchronous JavaScript and XML (Ajax) place an important load on network operators and creativities to effectively manage traffic. Problem happen is that there is no systematic technique to produce their workloads, notice their network performance today and possess track of the varying trends of these applications.they develop a tool, called AJAXTRACKER, that automatically impersonators a human interface with a cloud application and gathers associated network traces.

Ajax tracker

The main workings of AJAXTRACKER contain an event generator, a Web browser, a packet sniffer, and a traffic shaper. The event generator procedures the bulk of the tool.AJAXTRACKER is agnostic to the select of the Web browser and can work through any browser. The aim is to collect representative traces of a client session; packet sniffer captures the packets preceding the client machine.

B. Krishnamurthy, C. E. Willis, and Y. Zhang [2], focuses on Content Distribution Network (CDNs). This is a tool to distribute contents just before and users. In content distribution origin server serves some or all the content of web pages. The technique like DNS resending and URL rewriting are balance the load among their servers. After the observation of results some CDNs provide better results as compare to other. In particular network the dramatic growth in the number of distinct server is gives best performance of one CDN company that can be improved between two testing periods. Either in average or worst case conditions the results shows that the case of DNS in the critical path of resource retrieval is not better than that of server choices related to client response.

F. Schneider, A. Feldmann, B. Krishnamurthy, and W.Willinger[4] focuses in this paper study of Online Social Networks is discussed. Also, they understand which OSN feature inters and which one keep in consideration of poor users. Additionally the topics like friendship graph and sample crawls are studied on surveys. Extracting clickstreams as of inactively observed network traffic these are the techniques using these they study how users are interact with OSN.

III. Proposed Methodology

By using the "Characterizing Web Page Complexity and Its Impact" [1] we are finding the complexity of web page. After the complexity has been calculated we can show the analytical reports in text as well as graphical format using graphs.

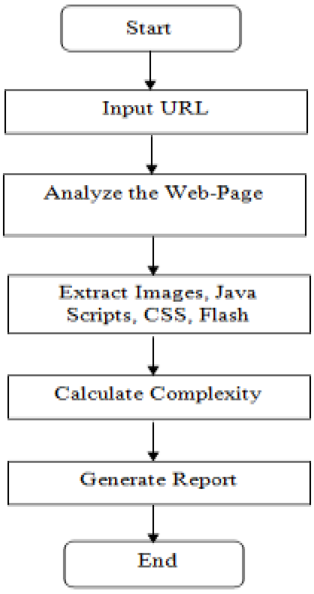

Figure1.Content Flow Architecture

We are start with entering URL.After the loading of page complexity of that web page can be determine using parameter such as content complexity and service complexity.

We parse the data and for parsing we are using PaserDelegator. In this we separate all contents of particular webpage entered by user and this help us to find complexity by counting the contents of that website.

IV. Experimental Setup:

Our analysis is based on the Websites visited and the complexity of that web site depending on the various complexity matrices. Our main focus is to captured complexity of webpage that are visible to the client side browsers (Mozilla Firefox, Chrome,etc). In snapshot we are giving the Login Page of our project.

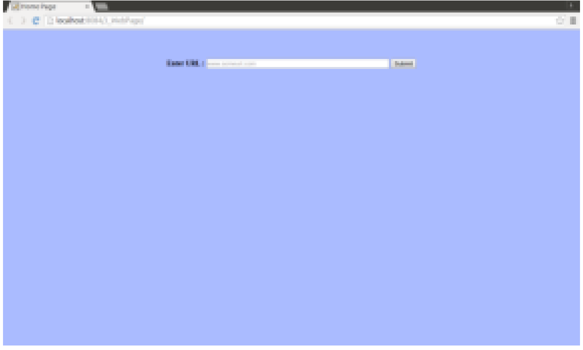

Fig2:Login Page

In the LoginPage user enter any WebSite such as www.google.com or www.w3school.com Internally we capture the size and number objects fetched to load the particular webpage in terms of Content Complexity and also the contents like css,javasripts,images,text that are spread across the objects.

ISSN: 2321-8169
024 – 026

When user type some url to the login page and click on the submit button that site can be open and also the tag tree can be formed that we are use to find the recommendation after finding the complexity of that webpage. We are giving two links on the output page one is to show tag tree and second one is to show complexity recommendation.

For finding the complexity, firstly we stored all website data which user typed in Login page after clicking on submit button in Page Data. That data can be parsed. In parsing process whole website data can be separated into different classes such as urls,meta, images, css,sript and text. This all contents are separated and after that we are using count on these to each content. Using count we get complexity on comparison with threshold value.

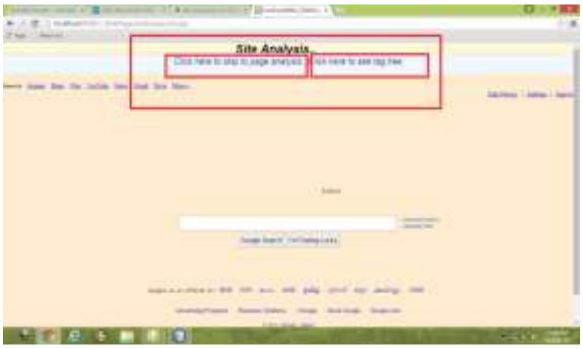
Site Analysis
Click here to skip to page analysis
lick here to see tag tree

Fig3: Site Analysis

Threshold value can be set by our own depending on the observation of visiting number of web sites. After visiting so many number of website we take an average ratio and set that threshold value.

On comparing that count with threshold value we are giving recommendation to reduce complexity and improve page load time.

Fig4: Recommendations

In above diagram we set threshold value (URLs-10,Meta Tags-3,CSS-2 like that),we can change that threshold depending upon the sites visited by user and giving recommendation on comparison with count value that can be found in parsing.

V. Conclusion

We will develop an efficient system that will calculate the complexity of the webpage and identify the components of web pages which requires more time to load. According to the recommendations at last we show offline comparison. In offline comparison we reconstruct one website as far the recommendation that we are given and show comparison. Also loading time calculated required to load webpage.

References

[1] Michael Butkiewicz, Harsha V. Madhyastha, and VyasSekar, "Characterizing Web Page Complexity and Its Impact", VOL. 22, NO. 3, JUNE 2014

[2] B. Krishnamurthy, C. E. Willis, and Y. Zhang, "On the use and performance of content distribution networks" , in Proc. IMW, 2001, pp. 169–182.

[3] M. Lee, R. R. Kompella, and S. Singh, "Active measurement system for high-fidelity characterization of modern cloud applications", in Proc. USENIX Conf. Web Appl., 2010, p. 2.

[4] F. Schneider, A. Feldmann, B. Krishnamurthy, and W.Willinger, "Understanding online social network usage from a network perspective", in Proc. IMC, 2009, pp. 35–48.

[5] F. Schneider, S. Agarwal, T. Alpcan, and A. FeldmanninProc. PAM ,"The new Web:Characterizing AJAX traffic", 2008, pp. 31–40

[6] Y. Zhang, H. Zhu, and S. Greenwood, "Website complexity metrics for measuring navigability," in Proc. Int. Conf. Quality Softw., 2004, pp.172–179

[7] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, Comput. Netw., "Graph structure in the Web," vol. 33, no. 1, pp. 309–320, Jun. 2000.

[8] D. Fetterly, M. Manasse, M. Najork, and J. Wiener, "A large-scale study of the evolution of Web pages," in Proc. WWW, 2003, pp.669–678.