ISSN:0975-9646

Pramod Gandugade et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 8 (4) , 2017, 502-505

Efficient Web Browsing on Smartphones

Pramod Gandugade[1], Disha Deotale[2]

[1,2] *Department of Computer Network*

GH Raisoni institute of engineering and technology

Abstract— **In a general day of smartphones user utilizes data connection for browsing the various websites. Web browsing is crucial for any smartphone user in day to day life. But this web browsing results in utilizing smartphones battery which is important for other task done on smartphones. Web browsing involves downloading web pages using wireless interface. This wireless interface has special characteristic, optimizing them results in drastic reduction in battery utilization. We rethink the computation sequence of browser occurring which wireless radio interface can be put in power saving mode. In addition of this we predict users reading time of web browser over which smartphone can go in idle state. This overall results in speediness of web browsing and network capacity enhancement.**

Keywords— **Mobile computing; web browsing; system design**

I. INTRODUCTION

Smartphones are becoming a key element of our day-today life. We all are connected to the internet throughout the world. The survey states that there are around 3% of people worldwide who use internet on mobile phones. And some of them use more than one smartphones for their own. Smartphones have tremendous demand in the market because of its amazing applications [1].

Although smartphones are great to use for web browsing, there is one issue which must be focused also, determined. Furthermore, that is, the energy utilization while web browsing. For determining this issue, lot of research has been done on various interfaces like, Wi-Fi, Bluetooth. In any case, these have diverse characteristics than cellular interface like 3G, 4G LTE or the wireless radio interface.

Web browsing is major service provided by smartphones used in daily life. In any case, the nowadays smart phone web browser uses a considerable measure of battery while accessing web pages [2].Prior research focuses on accumulation of power level of smartphone however they were concentrate on falling the power of one segment like cellphone display, they tried to reduce power using various hardware display like AMOLED display And remote interface, for example, WIFI interface of which transmission qualities are unique in relation to 3G and 4G LTE, which has higher utilization rate.

In 4G-LTE network and UMTS 3G network, to control the radio resource multiple timers are used, resources are released after some timeout value which is around more than 15 seconds. Hence the transmission of the data is delivered with downloading duration of the whole webpage and the data rate is low. Since out of many idle times, every idle time is smaller compared to value of timeout resulting in resetting the timer each time. Thus, wireless interface is always active without releasing the wireless resource results in consuming more power decreasing the network capacity.

Here in our paper, we describe the issue of power utilization in the smart phone web browsing into two types of techniques, initially the calculation cycle of the web browser is changed while loading the web page. The computations to load the webpage such as JavaScript code execution, HTML parsing, style formatting, image decoding, page layout etc. in most of cases it belongs to the two categories over which they will create the data transmission a fresh stream from the web server or not. These two types of computations can be separated in such a way that the new data transmission can be generated by running the first computation by the web browser and retrieving the same data. This web browser puts the WIFI or 3G radio interface into power saving mode releasing the wireless radio resource and then running the rest cycle of computation consuming from 40% up to 70% of time to load which saves certain requirement of power and wireless resource

The effects of energy-aware approaches can be demonstrated by applying a prototype which is depends on smartphone phones of 4G or 3G network.

II. ENERGY EFFICIENT METHOD

In this section, we rearrange possibilities of data transmission by recompiling sequence from web browser as soon as possible. After that we explores low overhead method to enhance user browsing experience.

In the end, we show the practical data mining based approach to detect user reading time of pages due to which smartphone able to switch between IDLE and power saving mode if reading time is more than threshold this we achieved by parsing and fetching data in single go.

A. Intermediate Display

In order to improve user browsing experience, original web browsers always draw intermediate display and update it frequently when loading webpages. However, this approach has two drawbacks. First, even if the web browser already has the web content, it has to wait prior displaying the intermediate results. This is due to the browser needs to collaborate DOM nodes and CSS style rules to lay out these data in web browser. However, since the CSS file is large and complex, it takes a lot of processing time to extract the rules [9]. Another disadvantage is that the browser wastes a lot of computation power to frequently redraw and reflow the intermediate display before the last output display [5]. A redraw occurs when updates to DOM nodes will change visibility, be that as it may, won't influence its format. Samples incorporate evolving the background colour or the visibility style. Redraw is costly since it requires the program to seek through all nodes in the DOM tree to give output what should be displayed. A reflow happens at whatever point the DOM tree is changed. Reflow of a node causes the subsequent reflow of its node and sub node in the DOM tree, and the new layout of these nodes will be recalculated. In the end, everything is redrawn. Therefore, a reflow costs more computation resources because it involves changes that affect the layout of the whole page.

By reorganizing the computation sequence, our approach draws the final display at the end of webpage loading, and in this way, it saves the calculation on redrawing or reflowing intermediate display. To improve user experience, we present a low-overhead approach to draw intermediate display with little layout computation during the data transmission time. When the browser begins to open a webpage, it first downloads the webpage which has a lot of text content and the basic html format information. After parsing 1/3 webpage content, we can draw a simplified intermediate display. Because this display does not need CSS rules, style format, or images, it only costs little layout computation [7]. In addition, since it does not need to wait for any information from the CSS files, it can be displayed much earlier than the original web browser. Moreover, because the interval

Between the first intermediate display and the final display is only about 10_20 seconds, we do not update the intermediate display during the loading time to avoid unnecessary computation on redraw and reflow. For webpages with mobile version, since the interval is very short (1_2 seconds) between intermediate display and final display, our approach only draws the final display.

B. *Reorganizing the Computation Sequence*

We must focus on data rearrangement by retrieving all data.

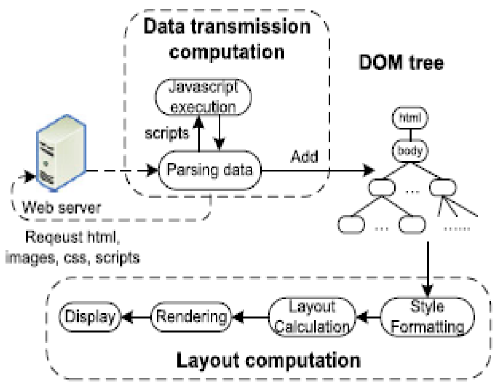

Fig 1 the work flow of webpage processing in smartphone based web browser

In this section, we have explained to separate them to save power

The computation sequence of the energy aware approach with the original web browser for opening the webpage. During time slot 1, the original web browser spends its computation resource on both transmission data as well as layout computation.one object is processed and added to its DOM tree. From our point of view the browser is focused on data transmission and layout is ignored. Hence our approach can be processed by more objects adding them to DOM tree during slot 1

During slot 2, all the objects are processed building the complete DOM tree. During the slot, there is no Data. transmission and our approach is focused on the computing the page layout. However, al each slot the partial web page content is displayed on the screen by the original browser, keeps generating data transmission till slot 3. Both approaches have same DOM tree and displays the same webpage. We described how to make the separations of different types of objects in detail.

Feature	Description
Reading time	The duration from the webpage is completely opened to the time when the user witch to another webpage
Transmission time	Time required for transmitting data
Webpage size	The data size of the webpages without considering the figures
Download objects	The number of total downloaded objects
Download JavaScript	The number of downloaded JavaScript files
Download figures	The number of downloaded figures
JavaScript running time	The time for processing all the JavaScript code
Page height	The height of page
Page width	The width of page
Second URL	The number of secondary URLs

For calculation of DOM tree, we are going to use regression tree boost algorithm state as below:

C. *Regression tree boost*

Get the input training set

Number of iteration M. $(x_i, y_i)_i^n = 1$

Initialize the model with constant value:

$F_0(x) = median\{y_i\}_1^N$

For $m = 1\ to\ M\ do$

$$\tilde{y} = -\left[\frac{\partial L(y_i, F_{m-1}(x_i))}{\delta F_{m-1}(x_i)}\right]_{F(x)=F_{m-1}(x)}, i = 1,\dots,N$$

$$\{R_{jm}\}_1^J = J - terminalnodetree(\{\tilde{y}_i, Xi\}_1^N)$$

$$\gamma_{jm} = arg\,min_\gamma \sum_{Xi \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$$

$$F_m(X) = F_{m-1}(X) + \sum_{j=1}^{J} \gamma_{jm} 1(X \in R_{jm})$$

end for

D. *Energy-Aware Approach*

Begin to open a webpage

Data transmission computation is done

Layout computation is finished

Collect features $x = \{x_1, \dots, x_{10}\}$

Webpage is opened

Wait for α seconds.

Get T_r from the prediction model with x
If $(T_r > T_d)\ OR\ (T_r > T_p\ AND\ mode == power)\ then$
Switch to IDLE state
End if

Parameters	Description
Tr	Predicted reading time
α	Insert threshold (sec)
Td	Time duration threshold(t1+t2) for delay driven mode
Tp	Time duration threshold for power driven mode
Mode	Power driven or delay driven mode

C. Proposed architecture

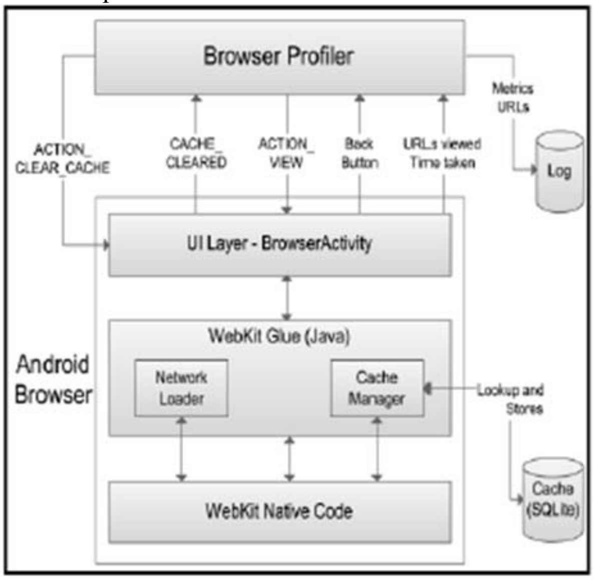

Fig 1. Browser profile used in parsing

Proposed system gives investigation of the energy required to convey renowned sites and the energy required to convey specific web components, for example, pictures, Javas-cript, and Cascade Style Sheets (CSS). Confounded Javas-cript and CSS can be immoderate to convey as pictures in website pages. Besides, ask for element JavaScript by client can amazingly help the expense of conveying the page since it abstains from reserving of substance of website page. Conveyance of JPEG pictures is obviously less costly than other for-mats, for example, GIF and PNG for tantamount size pictures on the Android program. For instance, attempt to interpret all images on the Facebook site to JPEG and check the outcome to get impressive energy funds. Our transformation effectively changes working of JavaScript on the page, without adjusting the client experience. We simply need to adjust the default Android program. Our adjusted program instruments completely stack a URL P in one of two modes

III. Result and Analysis

We have generated WebView on android smartphone using parsing, parsing is done in background and data is displayed in WebView app.

Generated XML tags are shown in tabs, due to onetime parsing data transmission is kept idle in smartphone resulting low battery utilization and low bandwidth capacity requirement from backbone.

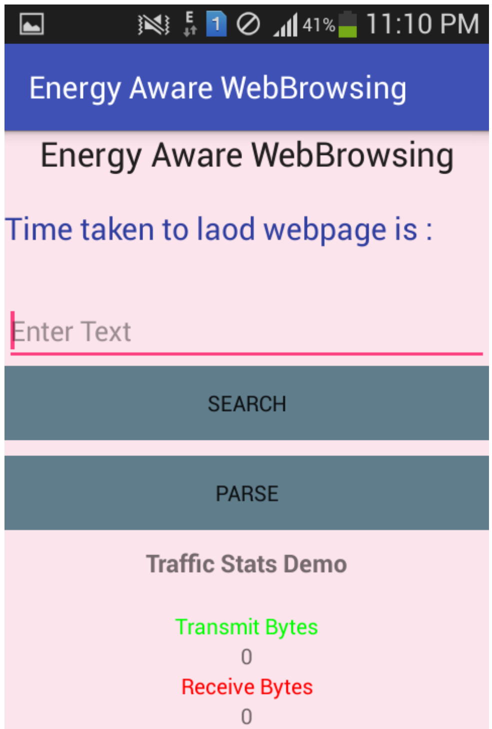

Fig 3. Web view for displaying webpages

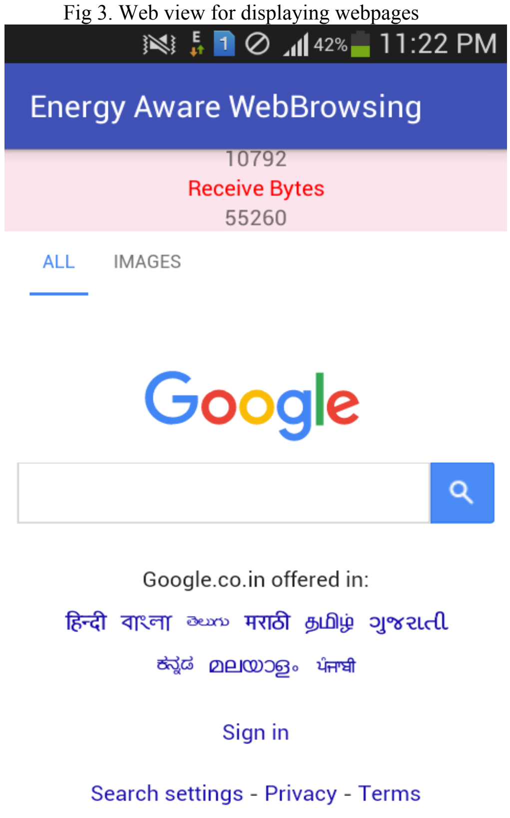

Fig 4. Output of web browsing on WebView

Fig 5 Output of parsed data

IV. Conclusion

In this paper, we put forward an energy-aware route for web browsing smartphones. By rearranging computation sequence while opening pages so that web program can run calculation then after generate data transmission and retrieve the same.

This method not just saves energy and but time period for Handling website page.

References

[1] Bo Zhao, Wenjie Hu, Qiang Zheng, and Guohong Cao “Energy aware web browsing on smartphone” in proc. Of IEEE Transactions on Parallel and Distributed Systems, 2014.J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[2] Dinesh Kumar, Reena Patel, “An Efficient Approach For Optimal Prefetching To Reduce Web Access Latency”, International Journal Of Scientific & Technology Research Volume 3, Issue 7, July 2014.

[3] Mian Dong, Student Member, IEEE, and Lin Zhong, Member, IEEE “Chameleon: A color adaptive web browser for mobile OLED displays”, IEEE Transactions On Mobile Computing, Vol. 11, No. 5, May 2012

[4] Bo Zhao, Byung Chul Tak and Guohong Cao, “Reducing the Delay and Power Consumption of Web Browsing on Smartphones in 3G networks”, 31stInternational Conference on Distributed Computing Systems, 2011

[5] Sandhaya Gawade 1, Hitesh Gupta2,”Review of Algorithms for Web Prefetching and Caching” in proc. Of International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2012

[6] H. Zhu and G. Cao, “On supporting power-efficient streaming applications in wireless environments,” IEEE Transactions on Mobile Computing, 2005.

[7] H. Zhu and G. Cao, “On supporting power-efficient streaming applications in wireless environments,” IEEE Transactions on Mobile Computing, 2005

[8] [8] W. Hu, G. Cao, S. V. K. , and P. Mohapatra, “Mobilityassisted energy- aware use contact detection in mobile social networks,” in IEEE ICDCS, 2013.

[9] [9] J.-H. Yeh, J.-C. Chen, and C.-C. Lee, “Comparative analysis of energy saving techniques in 3GPP and 3GPP2 systems,” in IEEE transactions on vehicular technology, 2009.