ISSN:0975-9646
Asha Khilrani et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1061-1064

Implementation of User's Browse Log Monitoring Tool for Effective Web Usage Mining

Ms. Asha Khilrani[#1], Prof. Shishir K. Shandilya[*2]

[#]*Computer Science, RGPV*
NIIST Bhopal, India

***Abstract*— The complex link structures of large websites and the day to day increasing usage had attracted the website administrator's attention towards the understanding of user's requirements related to their website usage. Statistical analysis of data thus has become an important parameter for most of the administrators. Through statistical analysis methods, analyzing page browsing time gives valuable information about usage of website and its users to the website administrators. This paper analyzes and monitors the user's web browsing behavior on web pages with different modes of action on client side and then calculates browsing time of each web page accordingly. This paper overcomes the inaccuracies in browsing time calculation considering server log data.**

***Keywords*— Client side Web Usage Data, Statistical analysis, Page Browsing Time.**

I. INTRODUCTION

Maintaining a website is as important as building it and to maintain website we need to improve its design. To improve the design of website we should find out how it is used by analyzing users' browsing behavior [1]. Statistical Analysis and web usage mining are two ways to analyze users' web browsing behavior. The result of Statistical Analysis contains Page Views, Page Browsing Time, and so on. Among this, browsing time is a good scale to evaluate website and users, for example in e-learning systems where we expect students spend a minimum duration of time on certain page, the value less than this minimum may represent inattention of student or weakness of the webpage. Web usage data can be collected from three sources: server level, client level, and proxy level [5].

Statistical analysis and web usage mining usually use Server Log as main data source which is server level data source. Server log is a file that automatically created by web server and kept on server. It contains some data about requests which are sent to web server. During reconstruction of users' session, server page caching and POST method, data are not recorded in server log [5]. Moreover, because of it only contains server side data, cannot be useful when we are interested in user's activities on client side such as hitting back button of browser, switching between tabs or windows and so on. To cope with such problems we also need to consider client side data [4].

However the previous work is done by Mehdi Heydari [2] in web page browsing time calculation considering client side data. But in Proposed work we calculate browsing time of web pages by monitoring the user's web browsing behavior on web pages with different modes of actions on client side like when the user is not navigating the web page throughout the time, is not attentive on the page during the navigation or navigate to some other application. Here the calculation is done only on the client side by facilitating the user with simultaneous opening of several web pages in new tabs or windows. It helps us to reconstruct user's session as it has occurred including navigation paths and pages browsing time. As we have calculated the browsing time with more precision, it helps us to find web usage patterns with more accuracy and which in turn helps administrators to evaluate its website usage more effectively.

II. PROPOSED METHOD

The majority of Web usage mining algorithms use Web log files as the main data sources in discovering useful information. The common server log includes the following seven fields: Remote-Host, Identification, Auth-User, Date-Time, HTTP-Request, Status-Code, and Transfer-Volume. The extended common server log format has two additional fields, Referrer and User-Agent field. Referrer field can be used to reconstruct user navigation path and Date-Time field can be used to discover page browsing time. Browsing time or duration of stay on a Web page is a key item for Web mining algorithms. For example, in e-learning, browsing time within the user profile is used to measure the value of each Web page or performance of a user's learning. With inaccurate browsing time captured, decision makers become conservative to the reliability of the measurement.

In general most of the users have tendency to open several pages simultaneously and in between, use some non browsing applications such as Ms-word, Excel etc for their own personal work, in such cases data recorded in server log only shows the requested time of the web pages and cannot help us to find out which web page and for how long has been really browsed on client machine.

In Proposed work ,first we monitor and record web browsing behavior of user on client side by checking whether user is continuously working on web page or not and calculates browsing time accordingly and then record it into a log file. To record user's web browsing behavior, we use remote agent technology. For this, we design an AJAX interface. The interface is a customized web application server which is able to monitor user's browsing behaviors on client side.

Definition 1. *Session Timeout* is the time when user doesn't send any request to the web server for a predefined duration of time. It is shown as t.

Definition 2. *Pageout* is the time at which user does not perform any activity on the current open web page.

The following events and functions are used for recording user's web browsing behavior in proposed method:-

1) *sessionCreated:* this is a server side event, it occurs when user sends request to the web server for the first time. when this event raises we set *MaxInactiveInterval* of session to a fixed time limit t. When user doesn't send any request to web server till time t, the session will be automatically destroyed.

2) When a user requests a web page from web server for the first time, a unique id SID is generated using the function *session.getId ()* It helps to separate user sessions from each other.

3) To generate unique pageId for each webpage a variable *sid* is declared at server side which will be incremented every time and attached with the requested webpage name when the webpage is loaded at client side. For example, when a page A is loaded first time on client side, its unique Id will become A1, for second time Id become A2 and so on.

4) *onload:* When the web page is loaded on user's browser, onload event raises and which in turn calls pageLoad () function.

5) pageLoad() : Firstly this function registers onfocus and onblur events (Lost event) to call pageFocus() and pageLostFocus() functions whenever user sets focus or lost focus from the current web page. This function also registers to call a check function repeatedly after every fixed time delay t1.This function also records some data about user's activity such as TID(*PageID* of *Target)*, Current *Time, Event Name='L'*, and *Date* in session using AJAX interface.

```
Algorithm,pageLoad
{
//Register onfocus and onblur events to call pageFocus () and
//pageLostFocus () functions whenever user sets focus or lost
//focus from the loaded web page.

 window.onfocus=pageFocus;
 window.onblur=pageLostFocus;

//it calls check function repeatedly with a fixed time delay t1
 window.setInterval (check, t1);

//it writes information like SID, TID, Event, Date and Current
//Time in session using AJAX interface.
}
```

*6) pageFocus ()***:**This function will be called when user sets focus on a web page. When this function called, some data about user's activity such as TID (PageID of Target),Event Name='F', Date and Current Time are recorded in session using AJAX interface.

```
Algorithm, pageFocus
{
// it writes information like SID, TID, Event, Date and
//Current Time in session using AJAX interface
 }
```

7) ***pageLostFocus ()*****:** This is a client side function that occurs when user lost focus from a web page. When this function called, some data about user's activity such as TID (PageID of Target), Current Time, Event Name='G', and Date are recorded in session using AJAX interface.

```
Algorithm, pageLostFocus
{
//it writes information like SID, TID, Event, Date and
//Current Time in session using AJAX interface
}
```

8) *check ()*: This function calculates the difference between current time (when check function called) and last active time (it may be equals to load time, focus time, keypress time or mouse move time).If the web page is opened by user and calculated difference is greater than some fixed time limit t1 then this function calls the pageLostFocus () function because user is not interested in browsing that web page or user is busy in another work. At this condition Pageout occurs.

```
Algorithm, check
{
//Calculate difference between last active time and current
//time
if (difference>t1)
{//calls pageLostFocus () function
pageLostFocus ();
}
}
```

9) *onkeypress ()*: This is a client side event that occurs when user press any key on the current web page. When this event raises ke () function will be called.

10)*onmousemove ():* This is a client side event that occurs when user moves mouse over the current web page. When this event raises me () function will be called.

11)*ke():*This function records the keypress time of user on current web page. If user comes back to web page for doing certain activity after *Pageout* time then it calls the pageFocus() function.

```
Algorithm, ke ()
{
//records keypress time

//calls pageFocus () function when user comes back after
//Pageout time
 pageFocus ();

}
```

12) *me ():* This function records the mouse move time of user on current web page. If user comes back to web page for doing certain activity after *Pageout* time then it calls the pageFocus () function.

```
Algorithm, me ()
{
//records mouse move time

//calls pageFocus() function when user comes back after
//Pageout time.
pageFocus();

}
```

13) sessionDestroyed: it occurs when user doesn't send any request to the web server for a predefined duration of time t (*Session Timeout*). When this event is raised, the browsing time and RID(PageID of Referrer) of each web page in session is calculated and assigned to the respective page . The content of user session is recorded in a file named *Log File*. *Log file* is like Server Log but is created by web application and has been customized as per our need.

```
Algorithm, sessionDestroyed
{
for each Page P in session
{
//calculate Browsing time and Referrer of each webPage in
//session and assigned to the respective web page
}
//Write session data into log file
for each Record  in Session {
//Transfer each record to log file;
}
}
```

Clarifying the method with example:

Let we have seven web pages in a website A, B, C, D, E, F and G.

The Open [2] function is used to present user's web browsing behavior.

Open (Referrer, Target, Type);

Here *Referrer* is the current webpage which is being browsed by user, for the first request where there is no opened page, it is *Φ*. *Target* is the page which is requested through *Referrer*. *Type* is one of the four web browsing options: *CurrentWindow, NewWindow*, *NewTab*, *Switch* and *Return.*

Suppose user navigates as follows:-

Open (Φ, A1, NewWindow);

Open (A1, B1, NewWindow);

Open (B1, A1, Switch);

Open (A1, C1, NewWindow);

Opens Ms-Word for some time and returns back to Page C.

Open (C1, E1, NewWindow);

Open (E1, C1, Switch);

Open (C1, F1, NewWindow);

Open (F1, E1, Switch);

Open (E1, G1, CurrentWindow);

Open (G1, B1, Switch);

Open (B1, D1, NewWindow);

Open (D1, B1, Switch);

Open (B1, C2, NewWindow);

Open (C2, F2, NewWindow);

Open (F2, C2, Switch);

Open (C2, E2, CurrentWindow);

Open (E2, D1, Switch);

Open (D1, G2, CurrentWindow);

Left Page opened for some time greater than Pageout time but less than Session Timeout. Then returns back to Page G by pressing any key. After idle time of t the session will automatically destroyed.

In proposed work, all browsing behaviors are monitored and kept in the user session until session timeout. After session timeout browsing time of web page A is calculated as follows:-

- Browsing time of A= Gone time of Page A -Load time of page A or Focus time of Page A.

Similarly, the same formula is used for calculating browsing time of other web pages.

After calculation of Browsing time of each web page data is recorded in a *log file* as shown in Table 1. Where SID denotes SessionID, PID denotes PageID, RID denotes ReferrerID, BT denotes Browsing time, E denotes Event, T denotes Time and D denotes Date.

TABLE I
LOG FILE

SID	PAGE	PID	RID	BT	E	T	D
S1	A	A1	NULL	6	L	39884	D1
S1	A	A1	NULL	0	G	39890	D1
S1	B	B1	A1	5	L	39890	D1
S1	B	B1	NULL	0	G	39895	D1
S1	A	A1	B1	5	F	39895	D1
S1	A	A1	NULL	0	G	39900	D1
S1	C	C1	A1	6	L	39901	D1
S1	C	C1	NULL	0	G	39907	D1
S1	C	C1	C1	18	F	39926	D1
S1	C	C1	NULL	0	G	39944	D1
S1	E	E1	C1	7	L	39945	D1
S1	E	E1	NULL	0	G	39952	D1
S1	C	C1	E1	36	F	39954	D1
S1	C	C1	NULL	0	G	39990	D1
S1	F	F1	C1	8	L	39990	D1
S1	F	F1	NULL	0	G	39998	D1
S1	E	E1	F1	13	F	40000	D1
S1	G	G1	E1	5	L	40013	D1
S1	G	G1	NULL	0	G	40018	D1
S1	B	B1	G1	15	F	40019	D1
S1	B	B1	NULL	0	G	40034	D1
S1	D	D1	B1	6	L	40034	D1
S1	D	D1	NULL	0	G	40040	D1
S1	B	B1	D1	24	F	40042	D1
S1	B	B1	NULL	0	G	40066	D1
S1	C	C2	B1	6	L	40066	D1
S1	C	C2	NULL	0	G	40072	D1
S1	F	F2	C2	5	L	40073	D1
S1	F	F2	NULL	0	G	40078	D1
S1	C	C2	F2	22	F	40080	D1
S1	E	E2	C2	4	L	40102	D1
S1	E	E2	NULL	0	G	40106	D1
S1	D	D1	E2	10	F	40108	D1
S1	G	G2	D1	60	L	40118	D1
S1	G	G2	NULL	0	G	40178	D1
S1	G	G2	G2	48	F	40190	D1
S1	G	G2	NULL	0	G	40238	D1

ASSUMPTIONS:

To explain above example, the following assumptions are used:

- We had taken a Session Timeout (t) of 100 seconds
- After every 30(t1) seconds of load time a check function will be called which checks the difference between current time (when check function called) and last active time (Which may be equal to load time or focus time or key press time or mouse move time).
- If this difference is greater than 30 (t1) seconds, we assumed that user is not interested in browsing on that web page or user is busy in another work.

The Log File Web usage data can be converted in a graph structure [2] which can be used in *Seong Dae Lee et al* [3] graph mining method to discover weighted frequent pattern.

III. CONCLUSIONS

In this paper user's browsing behavior on a web page is considered with different modes of actions on client side to calculate web page browsing time with more precision and accuracy. It helps to maintain the accuracy in finding weighted frequent web usage patterns and administrators to evaluate its website usage more effectively

REFERENCES

[1] I-Hsien Ting, Chris Kimble and Daniel Kudenko, Applying Web Usage Mining Techniques to Discover Potential Browsing Problems of Users, Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)

[2] Mehdi Heydari, Raed Ali Helal, Khairil Imran Ghauth,A Graph-Based Web Usage Mining Method Considering Client Side Data , 2009 International Conference on Electrical Engineering and Informatics 5-7 August 2009, Selangor, Malaysia.

[3] Seong Dae Lee, and Hyu Chan Park, Mining Weighted Frequent Patterns from Path Traversals on Weighted Graph, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.4, April 2007

[4] Maximilian Viermetz, Carsten Stolz, Relevance and Impact of Tabbed Browsing Behavior on Web Usage Mining, Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence.

[5] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, SIGKDD Explorations, Jan 2000, Volume 1, Issue 2 – page 12

[6] R. Iváncy and I. Vajk. Frequent Pattern Mining in Web Log Data. Acta Polytechnica Hungaria, Journal of Applied Sciences at Budapest Tech Hungary, Special Issue on Computational Intelligence. Vol.4, No.1, pp.77-99, 2006.

[7] Kurt D. Fenstermacher and Mark Ginsburg,"Mining Client- Side Activity for Personalization", International Workshop on Advanced issues of E-Commerce and Web- Based Information Systems,IEEE,2002

[8] Robert.Cooley,Bamshed Mobasher, and Jaideep Srinivastava, "Web mining:Information and Pattern Discovery on the World Wide Web,",In International conference on Tools with Artificial Intelligence, pages 558-567, Newport Beach, IEEE,1997.