Real Time Application Analysis Tool for Customizable Web Log Mining

Dr. Dushyantsinh Rathod
Department of Computer Engineering
Associate Professor & HOD, D.A Degree Engineering & Technology

Abstract- *Analyzing Web data has become a must-have for businesses. Significant research has been done in studying clickstream data to understand the navigation behavior of users after visiting a Web site. Analyzing clickstream data is not easy for most companies because Web logs are stored in a form that is not suited for analysis. Before any meaningful analysis can be done, much effort is spent in transforming server logs to the right form so that they can be analyzed. This is one of the reasons why companies often use third-party services (such as Webtrends, Adobe, or Google Analytics) to analyze their Web log data. This paper demonstrates applying programming to prepare a data set from raw Web logs and to generate summary reports.*

Keywords- Component, formatting, style, styling, insert

I. INTRODUCTION

Advancement in technology and growing use of the internet has opened up different study areas for statisticians. Every time users visit websites; clicks are saved that can be used for extracting useful patterns [2]. Clickstream data could be considered as a very rich source of information, because they contain behavioral information of the web site visitor. However it is difficult to analyze since it is available as unstructured data [3] and many different formats depending on the web server. Many companies have their specific ways of collecting and analyzing data; for example, e-commerce companies can measure the sales and demand of their products and identify behavioral patterns of consumers. Even non-profits such as universities are using their web data to market their courses [4].At times' clickstream data may be very difficult and costly to manage for e-commerce companies who would be using data for their businesses. Dealing with these challenges has compelled companies to purchase web analytical tools [6]. These tools range from simple reporting applications to much advanced analytical software applications like Google Analytics. Web Analytics tool is among the popular sophisticated tools which help companies in analyzing and visualizing their web log data.

Most of the web analytics tools directly take web logs and give end users information in the form of charts, plots and reports. The end user lacks control over the raw data which if they had in an useful format(such as data set) can be used for various other types of analysis which are not available in the tool. For example, one can get excellent insights by using Google Analytics for your website. However, businesses cannot perform advanced analytical methods like sequence analysis or social network analysis because they do not have the data in the right form. Nowadays companies have started integrating customer-level behavior data from a website into their analytics environment. In such cases it is important that companies have control over their web log data and make it available in the right form for other enterprise applications to use it. The discussed in this paper provides a user with a dataset of weblogs processed in a form that can be used easily for any type of statistical analysis or modeling.

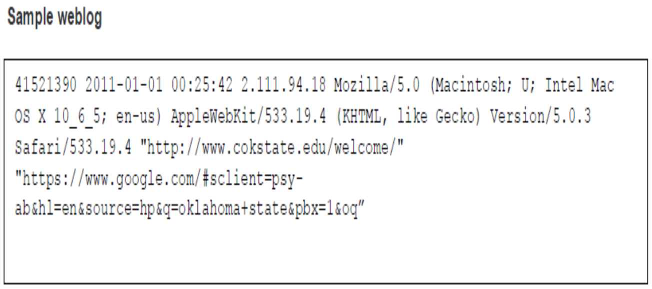
II. DATA COMPONENTS

2.1 SERVER WEBLOGS

Weblog can be defined as an electronic record of internet usage collected by web servers. Each web server has a separate configuration and settings which sometimes distinguishes weblog information from one server to another. The W3C maintains a standard format for web server log files, but other proprietary formats exist. Each record in the log usually contains IP address, html page name, date and time, referrer and additional information based on how it is setup. But these are the main elements that one will find in any setting. These logs can be stored as single file or can be separated as access logs, error logs, distinct logs etc. Site administrators usually have complete control over these files. We used a weblog collection with 6,633 entries collected over a week's time from a website. The name of the website is masked for confidentiality reasons. The information contained in the web log for each user includes following items.

Visitor Identification Number: This is a unique identification number for each user visit. In the case of this client company, the server was configured to create two separate variables that capture the unique identification number.
Date and time: Timestamp of the page visit.

IP Address: Every machine has a unique address. This field captures the IP address of the machine from where the page request is originating.
Page URL: URL of the current page the user is viewing.
Referral Page Information: Referral page captures the URL of the source page from where the request has originated
Browser and device information: Browser and device column provides information on type of browser and device used for accessing the web pages. Earlier we have just seen these requests coming from desktops or laptops. Now we find various mobile devices like smart phones and tablets that are used for accessing web pages.

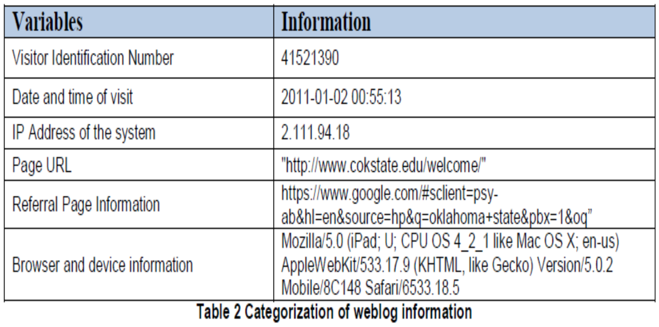


Sample weblog

```
41521390 2011-01-01 00:25:42 2.111.94.18 Mozilla/5.0 (Macintosh; U; Intel Mac
OS X 10_6_5; en-us) AppleWebKit/533.19.4 (KHTML, like Gecko) Version/5.0.3
Safari/533.19.4 "http://www.cokstate.edu/welcome/"
"https://www.google.com/#sclient=psy-
ab&hl=en&source=hp&q=oklahoma+state&pbx=1&oq"
```

Figure 1 Sample web log

Figure 1 shows a sample web log record. A Data Step program can used to prepare a Data set from this raw weblog. Table 1 shows the values in the Data set after identifying variables for the elements in the web log.



Variables	Information
Visitor Identification Number	41521390
Date and time of visit	2011-01-02 00:55:13
IP Address of the system	2.111.94.18
Page URL	"http://www.cokstate.edu/welcome/"
Referral Page Information	https://www.google.com/#sclient=psy-ab&hl=en&source=hp&q=oklahoma+state&pbx=1&oq"
Browser and device information	Mozilla/5.0 (iPad; U; CPU OS 4_2_1 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8C148 Safari/6533.18.5

Table 2 Categorization of weblog information

III. RELATED WORK

3.1 Implementation Tool

This tool is implemented in ASP.NET 2010. it is one of the popular Platform used for developing web-based application. This study focuses on this language in order to develop the application that can manipulate the server logs. The tool for preprocessing is shown in Fig.4. Using this tool we can upload three different log file format like W3C,IIS and NCSA log file. After uploading all three log file format user can select any important columns or attributes from gridview as per the user requirements.

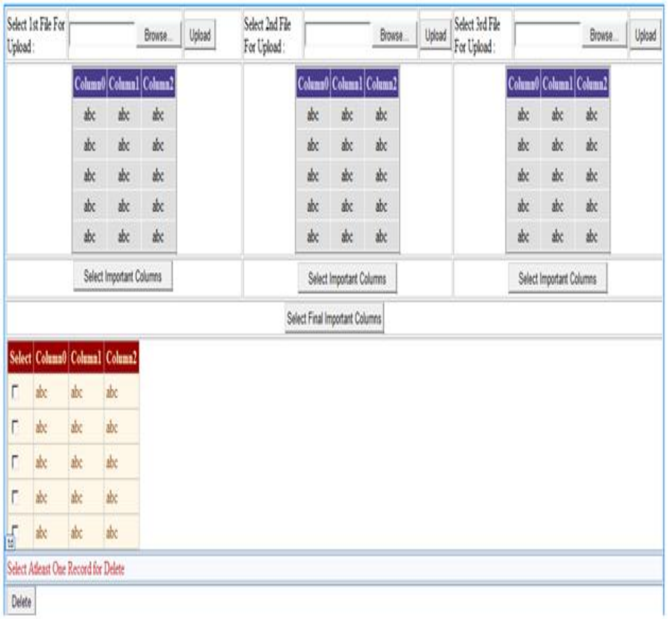


Figure 5 : Implementation tool

Figure 5 shows the implementation tool created in ASP.NET 2010.Using this tool user can upload their log files and this tool displays that log file in gridview and user grab important or useful atrributes from that gridview. And at the end we got combined log file .After that user can remove unnecessary data from combined file.

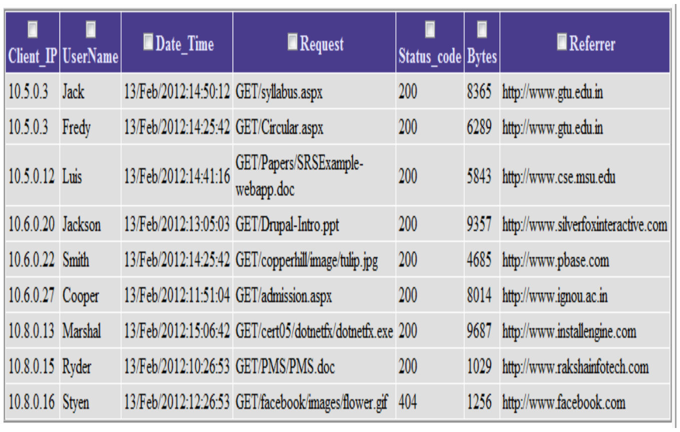
3.2 Used Algorithms

1) This algorithm read the data from different web log file from web server log

Input: Log File

Output: Data Source(CSVTable)

1. Create an instance of *StreamReader sr* to read from a file.
2. Give the file path in the *StreamReader* constructor.
3. Declare *String Line* variable to read the data line by line.
4. If *String Line* found "," then replace with " " (Space) and data row split with " "(space).
5. Take *While loop* to Read and display lines from the file until the end of the file is reached.
6. Records available in *L*.
7. Add records in *Data Source*.
8. Close the instance *Sr* of *StreamReader* class.



□ Client_IP	□ UserName	□Date_Time	□Request	□ Status_code	□ Bytes	□Referrer
10.5.0.3	Jack	13/Feb/2012:14:50:12	GET/syllabus.aspx	200	8365	http://www.gtu.edu.in
10.5.0.3	Fredy	13/Feb/2012:14:25:42	GET/Circular.aspx	200	6289	http://www.gtu.edu.in
10.5.0.12	Luis	13/Feb/2012:14:41:16	GET/Papers/SRSExample-webapp.doc	200	5843	http://www.cse.msu.edu
10.6.0.20	Jackson	13/Feb/2012:13:05:03	GET/Drupal-Intro.ppt	200	9357	http://www.silverfoxinteractive.com
10.6.0.22	Smith	13/Feb/2012:14:25:42	GET/copperhill/image/tulip.jpg	200	4685	http://www.pbase.com
10.6.0.27	Cooper	13/Feb/2012:11:51:04	GET/admission.aspx	200	8014	http://www.ignou.ac.in
10.8.0.13	Marshal	13/Feb/2012:15:06:42	GET/cert05/dotnetfx/dotnetfx.exe	200	9687	http://www.installengine.com
10.8.0.15	Ryder	13/Feb/2012:10:26:53	GET/PMS/PMS.doc	200	1029	http://www.rakshainfotech.com
10.8.0.16	Styen	13/Feb/2012:12:26:53	GET/facebook/images/flower.gif	404	1256	http://www.facebook.com

Figure 6 : NCSA log file in Gridview

Figure 6 shows one example of NCSA log file which can be read using algo 1 and it displays log file in gridview

2) This algorithm used to add Checkbox in the header of Gridview For Mining Data and Integration of Multiple Data Source.

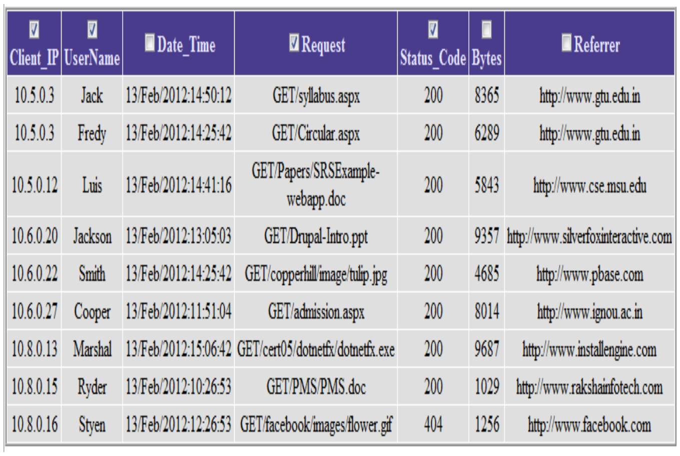
Input: Gridview's Data source(dtAll)

Output: New Data Source(dtFinal)

Steps:

1. Add *Checkbox* control to Gridview Headers Cell.
2. Bind *Data Source* to Gridview Control.
3. Take *for loop* to check *checkbox* in Gridview Header Cell.
4. Using *If* condition to check whether the checkbox is checked or not.
5. *If* true then take *for* loop to calculate the *Rows* for selected Columns.
6. Add *Selected Column's* and *Rows* in to Data Source.
7. *Copy* one Data Source data to another Data Source
8. *Merge* multiple Data Sources.
9. *Bind* Data Source to Gridview Control.

Output Ex 1: NCSA Customizable Log File

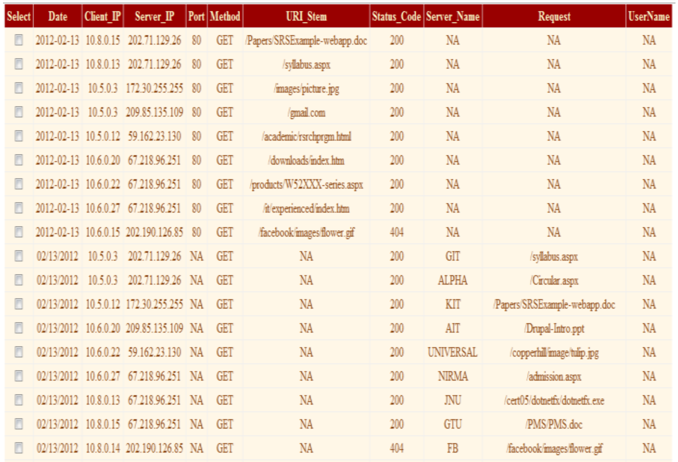


☑ Client_IP	☑ UserName	□Date_Time	☑Request	☑ Status_Code	□ Bytes	□Referrer
10.5.0.3	Jack	13/Feb/2012:14:50:12	GET/syllabus.aspx	200	8365	http://www.gtu.edu.in
10.5.0.3	Fredy	13/Feb/2012:14:25:42	GET/Circular.aspx	200	6289	http://www.gtu.edu.in
10.5.0.12	Luis	13/Feb/2012:14:41:16	GET/Papers/SRSExample-webapp.doc	200	5843	http://www.cse.msu.edu
10.6.0.20	Jackson	13/Feb/2012:13:05:03	GET/Drupal-Intro.ppt	200	9357	http://www.silverfoxinteractive.com
10.6.0.22	Smith	13/Feb/2012:14:25:42	GET/copperhill/image/tulip.jpg	200	4685	http://www.pbase.com
10.6.0.27	Cooper	13/Feb/2012:11:51:04	GET/admission.aspx	200	8014	http://www.ignou.ac.in
10.8.0.13	Marshal	13/Feb/2012:15:06:42	GET/cert05/dotnetfx/dotnetfx.exe	200	9687	http://www.installengine.com
10.8.0.15	Ryder	13/Feb/2012:10:26:53	GET/PMS/PMS.doc	200	1029	http://www.rakshainfotech.com
10.8.0.16	Styen	13/Feb/2012:12:26:53	GET/facebook/images/flower.gif	404	1256	http://www.facebook.com

Figure 7 : NCSA Customizable log file

Figure 7 shows customizable NCSA log file after uploading file into the tool

Output Ex 2: Combined Customizable Log File



Select	Date	Client_IP	Server_IP	Port	Method	URI_Stem	Status_Code	Server_Name	Request	UserName
□	2012-02-13	10.8.0.15	202.71.129.26	80	GET	/Papers/SRSExample-webapp.doc	200	NA	NA	NA
□	2012-02-13	10.8.0.13	202.71.129.26	80	GET	/syllabus.aspx	200	NA	NA	NA
□	2012-02-13	10.5.0.3	172.30.255.255	80	GET	/images/picture.jpg	200	NA	NA	NA
□	2012-02-13	10.5.0.3	209.85.135.109	80	GET	/gmail.com	200	NA	NA	NA
□	2012-02-13	10.5.0.12	59.162.23.130	80	GET	/academic/rsrchprgm.html	200	NA	NA	NA
□	2012-02-13	10.6.0.20	67.218.96.251	80	GET	/downloads/index.htm	200	NA	NA	NA
□	2012-02-13	10.6.0.22	67.218.96.251	80	GET	/products/W52XXX-series.aspx	200	NA	NA	NA
□	2012-02-13	10.6.0.27	67.218.96.251	80	GET	/it/experienced/index.htm	200	NA	NA	NA
□	2012-02-13	10.6.0.15	202.190.126.85	80	GET	/facebook/images/flower.gif	404	NA	NA	NA
□	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	GIT	/syllabus.aspx	NA
□	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	ALPHA	/Circular.aspx	NA
□	02/13/2012	10.5.0.12	172.30.255.255	NA	GET	NA	200	KIT	/Papers/SRSExample-webapp.doc	NA
□	02/13/2012	10.6.0.20	209.85.135.109	NA	GET	NA	200	AIT	/Drupal-Intro.ppt	NA
□	02/13/2012	10.6.0.22	59.162.23.130	NA	GET	NA	200	UNIVERSAL	/copperhill/image/tulip.jpg	NA
□	02/13/2012	10.6.0.27	67.218.96.251	NA	GET	NA	200	NIRMA	/admission.aspx	NA
□	02/13/2012	10.8.0.13	67.218.96.251	NA	GET	NA	200	JNU	/cert05/dotnetfx/dotnetfx.exe	NA
□	02/13/2012	10.8.0.15	67.218.96.251	NA	GET	NA	200	GTU	/PMS/PMS.doc	NA
□	02/13/2012	10.8.0.14	202.190.126.85	NA	GET	NA	404	FB	/facebook/images/flower.gif	NA

Figure 8: Combined customizable log file

In the above Fig 8 shows all three combined log file. In such columns shows NA ,which describes that the columns are not relevant or not belongs with such log file format.

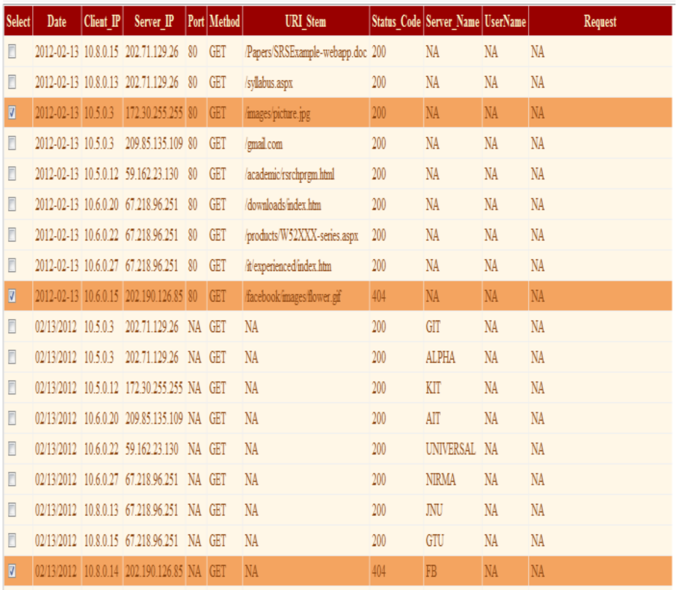
3) This algorithm used to removing irrelevant or unnecessary records

Input: Data Source(dtFinal)

Output: Final Data Source(dtFinalXml)

Steps:

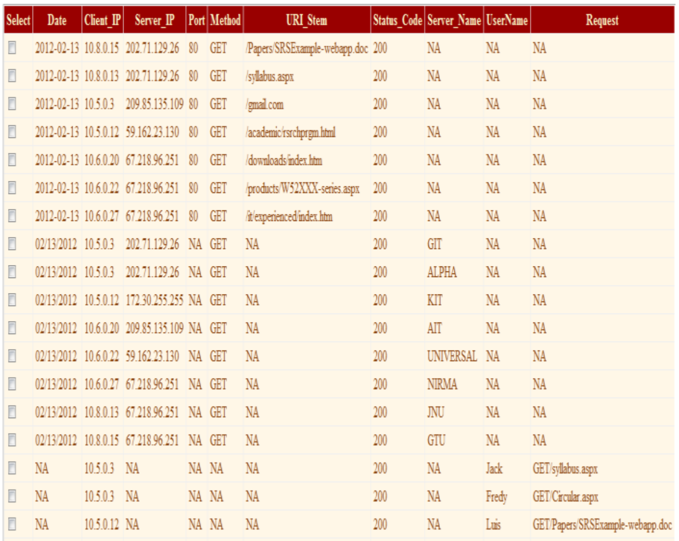
1. Read record in data source.
2. For each record in data source.
3. Read fields URL Field//In web server Log the requested object is the URL field
4. If requested URL field Contains/end with Substring = {*.gif,*.jpg,*.css,*?} then
5. Remove records
6. Else if Response code is
7. >299 or <200 then
8. Remove records
9. Else if Request method
10. not in {GET, POST}
11. Remove records
12. Else
13. Save records in output
14. End if
15. Next record.



Select	Date	Client_IP	Server_IP	Port	Method	URI_Stem	Status_Code	Server_Name	UserName	Request
□	2012-02-13	10.8.0.15	202.71.129.26	80	GET	/Papers/SRSExample-webapp.doc	200	NA	NA	NA
□	2012-02-13	10.8.0.13	202.71.129.26	80	GET	/syllabus.aspx	200	NA	NA	NA
☑	2012-02-13	10.5.0.3	172.30.255.255	80	GET	/images/picture.jpg	200	NA	NA	NA
□	2012-02-13	10.5.0.3	209.85.135.109	80	GET	/gmail.com	200	NA	NA	NA
□	2012-02-13	10.5.0.12	59.162.23.130	80	GET	/academic/rsrchprgm.html	200	NA	NA	NA
□	2012-02-13	10.6.0.20	67.218.96.251	80	GET	/downloads/index.htm	200	NA	NA	NA
□	2012-02-13	10.6.0.22	67.218.96.251	80	GET	/products/W52XXX-series.aspx	200	NA	NA	NA
□	2012-02-13	10.6.0.27	67.218.96.251	80	GET	/it/experienced/index.htm	200	NA	NA	NA
☑	2012-02-13	10.6.0.15	202.190.126.85	80	GET	/facebook/images/flower.gif	404	NA	NA	NA
□	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	GIT	NA	NA
□	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	ALPHA	NA	NA
□	02/13/2012	10.5.0.12	172.30.255.255	NA	GET	NA	200	KIT	NA	NA
□	02/13/2012	10.6.0.20	209.85.135.109	NA	GET	NA	200	AIT	NA	NA
□	02/13/2012	10.6.0.22	59.162.23.130	NA	GET	NA	200	UNIVERSAL	NA	NA
□	02/13/2012	10.6.0.27	67.218.96.251	NA	GET	NA	200	NIRMA	NA	NA
□	02/13/2012	10.8.0.13	67.218.96.251	NA	GET	NA	200	JNU	NA	NA
□	02/13/2012	10.8.0.15	67.218.96.251	NA	GET	NA	200	GTU	NA	NA
☑	02/13/2012	10.8.0.14	202.190.126.85	NA	GET	NA	404	FB	NA	NA

Figure 9 : Combined log file with like unnecessary data like .jpg, error page etc..

Fig 9 shows removing unwanted records from combined log file



Select	Date	Client_IP	Server_IP	Port	Method	URI_Stem	Status_Code	Server_Name	UserName	Request
□	2012-02-13	10.8.0.15	202.71.129.26	80	GET	/Papers/SRSExample-webapp.doc	200	NA	NA	NA
□	2012-02-13	10.8.0.13	202.71.129.26	80	GET	/syllabus.aspx	200	NA	NA	NA
□	2012-02-13	10.5.0.3	209.85.135.109	80	GET	/gmail.com	200	NA	NA	NA
□	2012-02-13	10.5.0.12	59.162.23.130	80	GET	/academic/rsrchprgm.html	200	NA	NA	NA
□	2012-02-13	10.6.0.20	67.218.96.251	80	GET	/downloads/index.htm	200	NA	NA	NA
□	2012-02-13	10.6.0.22	67.218.96.251	80	GET	/products/W52XXX-series.aspx	200	NA	NA	NA
□	2012-02-13	10.6.0.27	67.218.96.251	80	GET	/it/experienced/index.htm	200	NA	NA	NA
□	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	GIT	NA	NA
□	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	ALPHA	NA	NA
□	02/13/2012	10.5.0.12	172.30.255.255	NA	GET	NA	200	KIT	NA	NA
□	02/13/2012	10.6.0.20	209.85.135.109	NA	GET	NA	200	AIT	NA	NA
□	02/13/2012	10.6.0.22	59.162.23.130	NA	GET	NA	200	UNIVERSAL	NA	NA
□	02/13/2012	10.6.0.27	67.218.96.251	NA	GET	NA	200	NIRMA	NA	NA
□	02/13/2012	10.8.0.13	67.218.96.251	NA	GET	NA	200	JNU	NA	NA
□	02/13/2012	10.8.0.15	67.218.96.251	NA	GET	NA	200	GTU	NA	NA
□	NA	10.5.0.3	NA	NA	NA	NA	200	NA	Jack	GET/syllabus.aspx
□	NA	10.5.0.3	NA	NA	NA	NA	200	NA	Fredy	GET/Circular.aspx
□	NA	10.5.0.12	NA	NA	NA	NA	200	NA	Luis	GET/Papers/SRSExample-webapp.doc

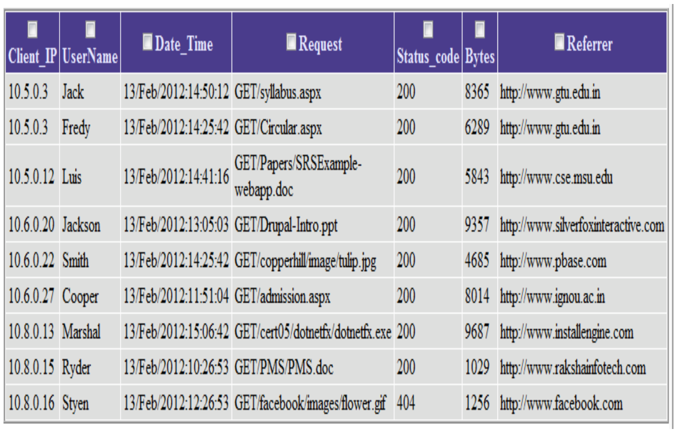
Figure 10 : Cleaned combined log file

Fig 10 shows Cleaned combined log file After removing unnecessary records.so we have got finally cleaned data

IV. PROPOSED SCHEME OF ANALYSIS TOOL

4.1 READING FROM WEBLOG

Weblogs can be extracted as .txt files from the server. If you are analyzing only access logs then all other types of logs like error logs should be filtered before you start making a data set. This task can be easily performed by your system administrator. The discussed in this paper only works with access logs. The first step in the is to read and convert the .txt files to a data set. The takes information from the log file and assigns appropriate data types and formats. It is required you understand the structure of the data in your web log so that you can modify the to suit your server environment. The full code is reported in appendix. Figure 2 displays a sample of web log entries after these were converted into a data set. As mentioned before, in the case of this client company, we had two variables representing the unique identification number. This may not be the case with other servers. Due to the differences in the way web logs are structured, you may have to tweak the slightly in this step to accommodate these differences.



□ Client_IP	□ UserName	□ Date_Time	□ Request	□ Status_code	□ Bytes	□ Referrer
10.5.0.3	Jack	13/Feb/2012:14:50:12	GET/syllabus.aspx	200	8365	http://www.gtu.edu.in
10.5.0.3	Fredy	13/Feb/2012:14:25:42	GET/Circular.aspx	200	6289	http://www.gtu.edu.in
10.5.0.12	Luis	13/Feb/2012:14:41:16	GET/Papers/SRSExample-webapp.doc	200	5843	http://www.cse.msu.edu
10.6.0.20	Jackson	13/Feb/2012:13:05:03	GET/Drupal-Intro.ppt	200	9357	http://www.silverfoxinteractive.com
10.6.0.22	Smith	13/Feb/2012:14:25:42	GET/copperhill/image/tulip.jpg	200	4685	http://www.pbase.com
10.6.0.27	Cooper	13/Feb/2012:11:51:04	GET/admission.aspx	200	8014	http://www.ignou.ac.in
10.8.0.13	Marshal	13/Feb/2012:15:06:42	GET/cert05/dotnetfx/dotnetfx.exe	200	9687	http://www.installengine.com
10.8.0.15	Ryder	13/Feb/2012:10:26:53	GET/PMS/PMS.doc	200	1029	http://www.rakshainfotech.com
10.8.0.16	Styen	13/Feb/2012:12:26:53	GET/facebook/images/flower.gif	404	1256	http://www.facebook.com

You can also see from figure 2 that the identifies appropriate formats for the variables. Web logs in this stage are still in a form that cannot be used for data mining or web analytics. Each record in the data set represents a single page visit per user with the latest visit at the bottom. Each entry captures the time of visit for a page. In order to calculate the time spent on a page you should know the time of visit for the next visited page and this goes on for all other pages until the visitor exits the web site. Therefore, you can never calculate the time spent on the last visited page in any session.

V. CREATING OUTPUT DATASET

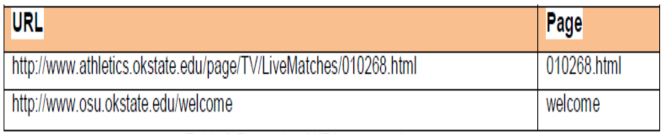
Once the raw data are available so we have to store all those data into database for permanently storage.Once the raw data set is available, we can use programming to transform the raw data set into a form that can be used for analysis. The structure of the output data set can be formulated based on the type of analysis an analyst wants to perform. New variables need to be created in order to extract insights from the data. This can include creating simple variables such as “Browser Type” and “Date” to complex variables like “Session Duration” and “Percent Page Duration”. The developed and reported in this paper creates these new variables with processed information but also retains the raw variables from the input data set. The new variables that are created by this are explained below:

Time Spent: This variable captures the time spent by the visitor on each page. The time spent on the page can be calculated only by knowing the start time of the next visited web page which is available only in the next following observation. We used SORT procedures to reverse the order of data along with RETAIN statements to calculate the time spent on a page.

Session: A session is defined as a series of page requests from the same uniquely identified client with a time of no more than 30 minutes. We track the time spent information to calculate the session for a visit.

Session Duration: Session duration captures the total time spent on all the pages visited in a session.

Page Name: Page Name is the actual page visited by the user. The identifies this page as the name with .htm or .html extension as found in the complete URL. If there is not .an html or .htm page, the last string in the URL is taken as the page name. Table shows two different examples for page names.



URL	Page
http://www.athletics.okstate.edu/page/TV/LiveMatches/010268.html	010268.html
http://www.osu.okstate.edu/welcome	welcome

Table 2 Example of URL page and page name

Exit Page: This is the last page visited by user. This value is identified based on the session.

Percent of Pages Visit: This variable captures the number of times a page was visited in a particular session in percentage.

VI. FILTERS – WEB ROBOTS

Multiple filters need to be applied to processed web log datasets prior to doing any kind of analysis on the data. One of the most important filters would be exclusion of web robots from the dataset. Web Robots are machine-generated search engines that provide necessary service to sites like Google and the other search engines by providing fast access to the internet resources[7]. Access to resources is possible by creating a worldwide index of available information. Identification and removal of robot becomes the vital part when activities like reporting the web site metrics is to be done. Variety of methods is used for removing robots; important ones of them is including user agent string exclusion. Usage of user agent string with the conjunction of IP addresses exclusion list could be one of the best ways to remove web robots. Sometimes, just using IP exclusion list may not solve the purpose as Internet Servers and IP addresses keep on changing. Code for some of the important exclusion list is mentioned in Appendix.

VI. REPORT GENERATION

The other important function of the is to generate relevant reports using the processed data. Reports help in answering various questions related to the website and visitor behavior like:

- Which is the most visited web page?
- Where are the visitors spending most of the page?
- Which is the most frequent exit page?
- What is the average time spent by a visitor on a particular page?

The processed data set can be used to answer these types of questions. The currently generates only basic reports. The can be modified to generate different types of reports according to the analyst's requirements.

The reports that the generates are:

- Top Ten visited pages
- Top Ten web pages where visitors spent most of the time
- Number of pages visited on a daily basis
- Top ten exit pages

VII. CONCLUSION

Clickstream data has a lot of valuable information about web site visitor's online behavior. However the server log data are not available in the right format for analysis. Asp.net programming can be used to prepare data in a form that can be reported and used for various modeling analysis. The programming in this paper can be easily used to prepare a dataset from server access logs and generate all user require reports. More sophisticated reports can be obtained by using any commercial web analytics applications that charge a lot of money (such as Adobe) or do not give researchers control over their data (such as Google Analytics).But, the data set that is created by this free gives more control to analysts in terms of applying wide range of advanced analytics techniques and defining customized variables. This can also be customized by uses to include more reporting capabilities. We hope many users can use this free and tweak it to create data sets from their own web logs and the apply sophisticated analytic techniques on those data set.

REFERENCES

[1] Randolph E. Bucklina and Catarina Sismeirob "Advances in Clickstream Data Analysis in Marketing", JOURNAL OF INTERACTIVE MARKETING

[2] Avi Goldfarb, "Analyzing Website Choice Using Clickstream Data", Joseph L. Rotman School of Management University of Toronto

[3] Alan L. Montgomery, Shibo Li, Kannan Srinivasan, and John C. Liechty, "Modeling Online Browsing and Path Analysis Using Clickstream Data"

[4] Peter I. Hofgesang and Wojtek Kowalczyk, " Analysing Clickstream Data: From Anomaly Detection to Visitor Profiling" Free University of Amsterdam, Department of Computer Science, Amsterdam, The Netherlands

[5] Wei Wang, “Parsing Web Logs with Base SAS®", Highmark Blue Cross Blue Shield, Pittsburgh, PA

[6] Kim Weller “Mainstreaming Web Data with SAS® Web Analytics 5.3", SAS Institute, Inc., Cary, NC

[7] Jenine Eason and Jerry Johannesen “CREATING MEANINGFUL DATA FROM WEB LOGS USING BASE SAS®”, Autotrader.com, Peachtree City, GA