Data Pre-processing on Web Server Logs for Generalized Association Rules Mining Algorithm

Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd, Hafizul Fahri Hanafi, Mohamad Farhan Mohamad Mohsin

Abstract**—** Web log file analysis began as a way for IT administrators to ensure adequate bandwidth and server capacity on their organizations website. Log file data can offer valuable insight into web site usage. It reflects actual usage in natural working condition, compared to the artificial setting of a usability lab. It represents the activity of many users, over potentially long period of time, compared to a limited number of users for an hour or two each. This paper describes the pre-processing techniques on IIS Web Server Logs ranging from the raw log file until before mining process can be performed. Since the pre-processing is tedious process, it depending on the algorithm and purposes of the applications.

Keywords**—**web log, www, data mining, web server.

I. Introduction

The brisk grow of the Information and Communication Technology has made an information dissemination become critical. Especially the web plays an important role and medium of information dissemination. There is a need for data log to track any transaction of the communications. Log file data can offer valuable information insight into web site usage. It characterizes the activity of many users over a potentially long period of time, compared to a limited number of users for an hour or two each. As more organizations view the Web as an integral part of their operations and external communications, interest in the measurement and evaluation of Web site usage is increasing [1] Server logs can be used to glean a certain amount of quantitative usage information. Compiled and interpreted properly, log information provides a baseline of statistics that indicate usage levels and support and/or growth comparisons among parts of a site or over time. Such analysis also provides some technical information regarding server load, unusual activity, or unsuccessful requests, as well as assisting in marketing and site development and management activities [3,4,5]. This paper presents the pre-process the web server logs in order to ensure its reliability to use in data mining algorithm.

II. Web server logs

A. Background

In dynamic systems such as the Internet, it is a common practice to periodically record samples of activity [7]. Those samples are then used to characterize the activity in the system and to evaluate new mechanisms to be used in this system. This is certainly true of HTTP traffic.

On the World Wide Web (WWW), logs of HTTP traffic are recorded continuously as a function of most origin web servers as well as intermediate proxies. The primary function of these logs is to chronicle the operation of these systems. However, as mock-up of HTTP activity, logs generated by these systems are also used for characterization, evaluation and usage reporting. Occasionally, researchers will capture HTTP traffic via other means, such as from augmented client browser.

Web Server logs are plain text (ASCII) files, that is independent from the server platform. There are some distinctions between server software, but traditionally there are four types of server logs:

1. Transfer Log
2. Agent Log
3. Error Log
4. Referer Log

The first two types of log files are standard. The referrer and agent logs may or may not be “turned on” at the server or may be added to the transfer log file to create an “extended” log file format. Each HTTP protocol transaction, weather completed or not, is recorded in the logs and some transactions are recorded in more than one log.

Mohd Helmy Abd Wahab is with the Department of Computer Engineering, Universiti Tun Hussein Onn Malaysia, P.O. Box 101, Batu Pahat, Johor, Malaysia. (Phone: +607-4537646; Fax: +607-4536060; e-mail: helmy@uthm.edu.my).

Mohd Norzali Haji Mohd., is with the Department of Computer Engineering, Universiti Tun Hussein Onn Malaysia, P.O. Box 101, Batu Pahat, Johor, Malaysia. (e-mail: helmy@uthm.edu.my).

Hafizul Fahri Hanafi is with the Faculty of Information and Communication Technology, Universiti Perguruan Sultan Idris, Tg. Malim, Perak, Malaysia (e-mail: hafizul@ftmk.upsi.edu.my).

Mohamad Farhan Mohamad Mohsin is with the College of Art and Science, Universiti Utara Malaysia, Sintok, Kedah Malaysia (e-mail: farhan@uum.edu.my).

B. Access Log

Fig. 1 is an example of a single line in a common transfer log collected from Portal Pendidikan Utusan. This typically displays as one log line of ASCII text, separated by tabs and spaces (useful for importing it into a spreadsheet program).

```
2003-11-23   16:00:13   210.186.180.199   -
CSLNTSVR20    202.190.126.85    80    GET
/tutor/images/icons/fold.gif - 304 140 470
0          HTTP/1.1          www.tutor.com.my
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows
+98;+Win+9x+4.90)
ASPSESSIONIDCSTSBQDC=NBKBCPIBBJHCMMFIKMLNN
KFD;+browser=done;+ASPSESSIONIDAQRRCQCC=LB
DGBPIBDFCOKHMLHEHNKFBN
http://www.tutor.com.my/
```

Figure 1: Single entry of log file from Portal Pendidikan Utusan

Portal Pendidikan Utusan normally known as Tutor.com and its server log consists of 19 attributes. The attributes are:-

a) Date

The date from Greenwich Mean Time (GMT x 100) is recorded for each hit. The date format is YYYY-MM-DD. The example from Fig. 1 above shows that the transaction was recorded at 2003-11-23.

b) Time

Time of transactions. The time format is HH:MM:SS. The example from Fig. 1 above shows that the transaction time was recorded at 16:00:13.

c) Client IP Address

Client IP is the number of computer who access or request the site.

d) User Authentication

Some web sites are set up with a security feature that requires a user to enter username and password. Once a user logs on to a Website, that user's "username" is logged in the fourth field of the log file.

e) Server Name

Name of the server. In Fig. 1 the name of the server is **CSLNTSVR20.**

f) Server IP Address

Server IP is a static IP provided by Internet Service Provider. This IP will be a reference for access the information from the server.

g) Server Port

Server Port is a port used for data transmission. Usually, the port used is port 80.

h) Server Method (HTTP Request)

The word request refers to an image, movie, sound, pdf, txt, HTML file and more. The above example in Fig. 1 indicates that folder.gif was the item accessed. It is also important to note that the full path name from the document root. The GET in front of the path name specifies the way in which the server sends the requested information. Currently, there are three formats that Web servers send information [8] in GET, POST, and Head. Most HTML files are served via GET Method while most CGI functionality is served via POST.

i) URI-Stem

URI-Stem is path from the host. It represents the structure of the websites. For examples:-

```
      /tutor/images/icons/fold.gif
```

j) Server URI-Query

URI-Query usually appears after sign "?". This represents the type of user request and the value usually appears in the Address Bar. For example:-

?**q=tawaran+biasiswa&hl=en&lr=&ie=UTF-8&oe=UTF-8&start=20&sa=N**

k) Status

This is the status code returned by the server; by definition this will be the three digit number [2]. There are four classes of codes:

i. Success (200 Series)
ii. Redirect (300 Series)
iii. Failure (400 Series)
iv. Server Error (500 Series)

A status code of 200 means the transaction was successful. Common 300-series codes are 302, for redirect from http://www.mydomain.com to http://www.mydomain.com, and 304 for a conditional GET. This occurs when server checks if the version of the file or graphics already in cache is still the current version and directs the browser to use the cached version. The most common failure codes are 401 (failed authentication), 403 (Forbidden request to a restrict subdirectory, and the dreaded 404 (file not found) messages. In the above transmission a status is 200 means that there was a successful transmission.

a) Bytes Sent

The amount of data revisited by the server, not together the header line.

b) Bytes Received

Amount of data sent by client to the server.

c) Time Stamp

This attribute is used to determine how long a visitor spent on a given page.

d) Protocol Version

HTTP protocol being used (e.g. HTTP/1.1).

e) Host

This is either the IP address or the corresponding host name (www.tutor.com.my) of the remote user requesting the page.

f) User Agent

The user agent reported by the remote user's browser. Typically, this is the string describing the type and version of browser software being used.

g) Cookies

Cookies can be used to track individual users thus make the sessionizer task easier. However, the use of cookies also raises the concern of privacy thus it requires the cooperation of the users.

h) Referer

The refering page, if any, as reported by the remote user's browser.

It is possible to analyze the following variables in the access log:

- Domain name or Internet Protocol (IP number).
- Date and Time
- Item accessed

It is possible to generate the following data from these variables:

- The percentage of users accessing the site from a specific domain type (e.g., .com, .edu, .net, .mil, .gov). This can be analyzed further by ***hits*** versus ***accesses.***

- The number of hits the server is getting from various IP groups. Such data can inform server administrator as to the primary client of their servers.

- The number of unique IP addresses accessing the site. While not a measure of unique users, this can provide server administrators with some indication of the number of users by stripping IP addresses from the log data. This data is an important indicator of the breadth of penetration of the servers.

- The quantity of accesses/hits the server receives during specific hours and days of the week. These statistics can be useful to server administrators who need to know the optimal time/day to perform server maintenance and/ or upgrades.

- The path – known as "threading" – a user takes through a site. Knowing this allows a server administrator to determine the average length of a user's session, specific location duration (e.g. average time on a page), average download times, and how the user navigated through the site (e.g. entrance and exit points).

The data from Access Logs provides a broad view of a Web server's and users (as indicate by IP address). Such analysis enables server administrators and decision makers to characterize their server's audience and usage patterns.

C. Agent Log

The Agent Log provides data on a user's browser, browser version, and operating system. This is the significant information, as the type of browser and operating system determines what a user is able to access on a site (e.g. Java, forms). Below is the sample agent log entry (see Fig. 2)

```
Mozilla/3.0 (Win 95; 1)
```

Figure 2: Agent Log entry

Analysis of Agent Log enables server administrators to determine:

- ***Browser***

The type of browser used to access a website. There are several different Web browsers on the market today (e.g. Netscape, Microsoft Internet Explorer, Lynx, Mosaic), Each of which has different viewing capabilities.

- ***Browser Version***

Each browser has its own capabilities. In this study, an Internet Explorer version 6.0 is used so that all components of a website can be viewed.

- ***Operating System.***

The type of computer and operating system used to determine the Graphical User Interface (GUI) of a website depending on the computer platform (e.g. Windows, Win 95, Macintosh).

The Agent Log information is essential for the design and development of Websites. Without such information, server administrator could design sites that require viewing capabilities that a vast majority of the site's users do not posses. This could lead to wasted effort by the server administrators. Worst still, this can lead to improperly displayed web content, thus effectively rendering the site useless to the user.

D. Error Log

The average Web user will receive an "Error 404 File Not Found" message several times a day. When a user encounters this message, an entry is made in the Error Log. Below is a sample Error Log entry (see Fig 3).

```
2003-11-23   16:07:09   210.186.79.43   -
CSLNTSVR20    202.190.126.85    80    GET
/tutor/images/tagline.gif - 304 141 404 0
HTTP/1.1                    tutor.com.my
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows
+NT+5.1)SPSESSIONIDCSTSBQDC=NLKBCPIBNAHCLL
MNEJLCIHLC;+browser=done
http://tutor.com.my/
```

Figure 3: Error log file entry

The Error Log contains the following data for analysis:

- ***Error 404***.

The Error Log tells a server administrator the time, domain name of the user, and page on which a user received the error. These error messages are critical to Web server administration activities, as they inform server administrators of problematic and erroneous links on their servers.

- ***Stopped transmission***.

This updates a server administrator of a user-interrupted transfer. For example, a user clicking on the "stop" button would generate a "stopped transmission" error message. The Error Log tells a server administrator the time, domain name, and page that a user was on when the transmission was stopped (as in the above sample Error Log entry). This information is useful as it can indicate patterns with large files such as image, movie, and other files that users consistently stop downloading.

The analysis of Error Log data can provide afford vital server information such as missing files, erroneous links, and aborted downloads. This information can enable server administrators to modify and correct server content, thus decreasing the number of errors users encounter while navigating a site.

E. Referer Log

The Referer Log indicates what other sites on the Web link to a particular server. Each link made to a site generates a Referer Log entry which is which is depicted in Fig. 4.

```
http://search.yahoo.com/search?p=utu
san&ei=UTF-8&fr=fp-tab-web-
t&cop=mss&tab=
```

Figure 4: Referer Log Entry

In this particular example, the referer was AltaVista, indicating that the user entered the Web site after performing a search using the AltaVista search facility.

The Referer Log entry provides the following data:

- ***Referral***.

If a user is on a site (e.g., ericir.syr.edu), and clicks on a link to another site (e.g., www.sun.com), then www.sun.com will receive an entry in their Referer Log. The log will show that the user came to the sun site (www.sun.com) via ericir.syr.edu (the referral). Such referral data is critical to alleviating missing link (Error 404) data. For an example, when the URL of a page within www.sun.com changes, the server administrator of www.sun.com could notify all referrals (e.g., ericir.syr.edu) of the change. This can alleviate future "Error 404 - File Not Found" messages.

Through the analysis of the four log files, Web service providers can begin the process of assessing and evaluating their networked information services. Current Web usage statistics generally center on the analysis of the Access Log, thus limiting the ability of Web-mounted service extensiveness measures. There are, however, means to analyze the Agent, Error, and Referer log files. Such techniques can provide important additional insight into the use of Web-based services by users.

III. Log File format

Currently, there are three formats available to record log files:-

- W3C Extended Log file Format
- Microsoft IIS Log File
- NCSA Common Log file Format

The W3C Extended log file format, Microsoft IIS log file format, and NCSA log file format are all ASCII text formats. The W3C Extended and NCSA formats record logging data in four-digit year format. The Microsoft IIS format uses a two-digit year format for years 1999 and earlier and a four-digit format thereafter. The Microsoft IIS log format is provided for backward compatibility with earlier IIS versions.

3.1.1 W3C Log File Format

W3C Extended format is a customizable ASCII format with a variety of different fields. Fields can be included when important, while limiting log size by omitting unwanted fields. Fields are separated by spaces. Time is recorded as UTC (Greenwich Mean Time). The example in Fig. 5 shows lines from a file using the following fields: Time, Client IP Address, Method, URI Stem, Protocol Status, and Protocol Version.

```
  #Software:      Microsoft      Internet
Information Services 5.1
  #Version: 1.0
  #Date: 1998-05-02 17:42:15
  #Fields: time c-ip cs-method cs-uri-stem
sc-status cs-version
  7:42:15 172.16.255.255 GET /default.htm
200 HTTP/1.0
```

Figure 5: W3C Log File Format

The preceding entry designates that on May 2, 1998 at 5:42 P.M., UTC, a user with HTTP version 1.0 and the IP address of 172.16.255.255 issued an HTTP GET command for the /Default.htm file. The request was returned without error. The #Date: field indicates when the first log entry was made, which is when the log was created. The #Version: field indicates that the W3C logging format used.

Any of the fields can be selected but some fields may not have information available for some requests. For fields that are selected, but for which there is no information, a hyphen (—) appears in the field as a placeholder.

3.1.2 IIS Log File Format

Microsoft IIS format is a fixed (non-customizable) ASCII format. It records more items of information than the NCSA Common format. The Microsoft IIS format includes basic items such as the user's IP address, user name, request date and time, Service status code, and number of bytes received. In addition, it includes detailed items such as the elapsed time, the number of bytes sent, the action (for example, a download carried out by a GET command) and the target file. The items are separated by commas, making the format easier to read than the other ASCII formats, which use spaces for separators. The time is recorded as local time.

When you open a Microsoft IIS format file in a text editor, the entries are similar to the following example in Fig. 6.

```
192.168.114.201, —, 03/20/98, 7:55:20,
W3SVC2, SALES1, 192.168.114.201, 4502,
163, 3223, 200, 0, GET, /DeptLogo.gif, —,
172.16.255.255, anonymous, 03/20/98,
23:58:11, MSFTPSVC, SALES1,
192.168.114.201, 60, 275, 0, 0, 0, PASS,
/intro.htm, —,
```

Figure 6: IIS Log File Format

In the log file, all fields are terminated with a comma (,). A hyphen (—) acts as a placeholder if there is no valid value for a certain field.

3.1.3 NCSA Log File Format

NCSA Common format is a fixed (non-customizable) ASCII format, available for Web sites but not for FTP sites. It records basic information about user requests, such as remote host name, user name, date, time, request type, HTTP status code, and the number of bytes sent by the server. Items are separated by spaces; time is recorded as local time.

When you open an NCSA Common format file in a text editor, the entries are similar to the following example:

```
172.21.13.45        —          REDMOND\fred
[08/Apr/1997:17:39:04      -0800]      "GET
/scripts/iisadmin/ism.dll?http/serv
HTTP/1.0" 200 3401
```

Figure 7: NCSA Log File Format

IV. METHODOLOGY

A. Raw Log File

Use The log files are text files that can range in size from 1KB to 100MB, depending on the traffic at a given a website [10]. In determining the amount of traffic a site receives during a specified period of time, it is important to understand what exactly; the log files are counting and tracking. In particular, there is a critical distinction between a hit and access, wherein: -

- A ***hit*** is any file from web site that a user downloads. A Hit can be text document, image, movie, or a sound file. If a user downloads a web pages that has 6 images on it, then that user “hit” the web site seven times (6 images + 1 text page).

- An ***access***, or sometimes called a page hit, is an entire page download by a user regardless of the number of images, sounds, or movies. If a user downloads a web page that has 6 images on it, then that user just accessed one page of the web site.

In this study, raw log file were collected from Portal Pendidikan Utusan Malaysia or known as Tutor.com. This portal focuses on education and provides more information related to education purposes such as Tutorials, Question Banks, Teaching Guidelines, and etc. For the analysis purposes, data dated on 24 November 2003 that consists of 82 683 records was retrieved from the server and needed to be preprocessed.

The raw log files consists of 19 attributes such as *Date, Time, Client IP, AuthUser, ServerName, ServerIP, ServerPort, Request Method, URI-Stem, URI-Query, Protocol Status, Time Taken, Bytes Sent, Bytes Received, Protocol Version, Host, User Agent, Cookies, Referer*. One of the main problems encountered when dealing with the log files is the amount of data needs to be preprocessed (Drott, 1998). A sample of a single entry log file is displayed in Fig. 8.

```
2003-11-23 16:00:13 210.186.180.199 - CSLNTSVR20
202.190.126.85 80 GET /tutor/include/style03.css - 304 141
469 16 HTTP/1.1 www.tutor.com.my
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+98;+Win+9
x+4.90)
ASPSESSIONIDCSTSBQDC=NBKBCPIBBJHCMMFIKML
NNKFD;+browser=done;+ASPSESSIONIDAQRRCQCC=L
BDGBPIBDFCOKHMLHEHNKFBN
http://www.tutor.com.my/
```

Figure 8: Single entry of raw log file

B. Data Preprocessing

From the technical point of view, Web usage mining is the application of data mining techniques to usage logs of large data repositories. The purpose of it is to produce result that can be used to improve and optimize the content of a site [9]. In this phase, the starting point and critical point for successful log mining is data extraction. The next task after data extractions are data cleaning and data filtering. Since the origin web logs data sources are blended with irrelevant information, data pre processing acts as an important steps to filter and organize only appropriate information before presenting to any web mining algorithm [6].

An entry of Web server log contains the time stamp of a traversal from a source to a target page, the IP address of the originating host, the type of request (GET and POST) and other data. Many entries that are considered uninteresting for mining were removed from the data files. The filtering is an application dependent. While in most cases accesses to embedded content such as image and scripts are filtered out. However, before applying data mining algorithm, data preprocessing must be performed to convert the raw data into data abstraction necessary for the further processing (see Table 1). We only select the record for 1 day which consists of 1377738 records.

Table 1: Preprocessed Log File

T	ClientIP	Datetime	Met hod	ServerIP	Por t	URI Stem
0	202.185.122. 151	11/23/200 3 4:00:01 PM	GET	202.190.126.8 5	80	/index.asp
1	202.185.122. 151	11/23/200 3 4:00:08 PM	GET	202.190.126.8 5	80	/index.asp
2	210.186.180. 199	11/23/200 3 4:00:10 PM	GET	202.190.126.8 5	80	/index.asp
3	210.186.180. 199	11/23/200 3 4:00:13 PM	GET	202.190.126.8 5	80	/tutor/inclu de/style03. css
4	210.186.180. 199	11/23/200 3 4:00:13 PM	GET	202.190.126.8 5	80	/tutor/inclu de/detectB rowser_coo kie.js

Table 1 exhibits the sample of preprocessed log file. All attributes after preprocessing cannot be shown in the table above due to the space restriction. After preprocessing completed, the pattern mining was performed to mine the access pattern.

C. Tool used

Active Server Pages (ASP) is one of the popular scripting languages used for developing web-based application. This study focuses on this language in order to develop the application that can manipulate the server logs. To access the server logs from windows 2000, the *.dll file named *logscrpt.dll* is used to load the class object MSWC.IISLog. The MSWC.IISLog class contains several *methods* and *properties* that can be used either to retrieve log entries or write log entries. (See Fig. 9)

Methods

AtEndOfLog	To indicates that all records have been read from the log files
CloseLogFiles	Closes all open log files
OpenLogFile	Opens a log file for reading or writing.
ReadFilter	Filters records from the log file by date and time.
ReadLogRecord	Reads the next available log record from the current log file.
WriteLogRecord	Writes a log record to the current log file.

Properties

BytesReceived	Indicates the number of bytes received.
BytesSent	Indicates the number of bytes sent.
ClientIP	Indicates the client's host name.
Cookie	Indicates the client's cookie.
CustomFields	Indicates an array of custom headers.
DateTime	Indicates the date and time, in GMT.
Method	Indicates the operation type.
ProtocolStatus	Indicates the protocol status.
ProtocolVersion	Indicates the version string.
Referer	Indicates the referrer page.
ServerIP	Indicates the server's IP address.
ServerName	Indicates the server name.
ServerPort	Indicates the port number.
ServiceName	Indicates the service name.
TimeTaken	Indicates the total processing time.
URIQuery	Indicates any parameters passed with the request.
URIStem	Indicates the target URL.
UserAgent	Indicates the user agent string.
UserName	Indicates the user's name.
Win32Status	Indicates the Win32 status code.

Figure 9: Method and Properties of MCSW.IISLog Class

In order to perform pattern mining and generalized association rules, a tool was written using Active Server Pages (ASP) to perform preprocessing techniques. The algorithm for preprocessing is shown in Fig.10.

```
Const ForReading = 1
Const ForWriting = 2

Sub ReadLog( Physical-Path, ModeFile-1, TypeOfLogFile, ModeFile
            2,StrTypeOfLogFormat)

        RecordCounter = 0
        Set LogReader =
    Server.CreateObject("IISLog")
        LogReader.OpenLogFile
LogFilePath, ModeFile-1,
        TypeOfLogFile,
ModeFile-2, StrTypeOfLogFormat

LogReader.ReadLogRecord
While NOT LogReader.EndOfLogRecord
        Retrieve Log Attributes
        ……
        ……
```

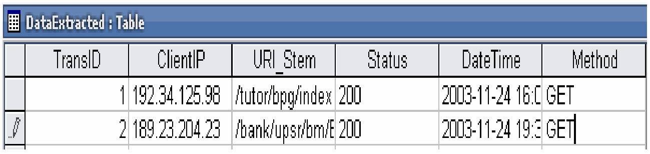
```
19                  RecordCounter = RecordCounter + 1
20                  LogReader.ReadLogRecord
21     Loop
23     LogReader.CloseLogFile
24     End Sub
```

Figure 10: Algorithm for Reading Server Logs

Fig. 10 depicts the algorithm used for reading the server logs. Since the server logs consists of three common formats as described above, the algorithm for reading the file is also different depending on the format of the log file. In Fig. 10, line 1 and 2 shows the definition of the mode of the file. Since the log files is read for preprocessing purposes, a mode 1 is utilized. In line 8, an object to read the file is instantiated and line 9 causes the file to be opened does open the file. In line 12, the first record of the log file is read then the ***RecordCounter*** will count the record that represents the transaction.

5.1 Transfer Server Logs to database

After reading the log files, several attributes are ignored because they were considered not important for the analysis. The read logs records will be stored in a database. Fig. 11 shows the database to store the data.



DataExtracted : Table

TransID	ClientIP	URI_Stem	Status	DateTime	Method
1	192.34.125.98	/tutor/bpg/index	200	2003-11-24 16:0	GET
2	189.23.204.23	/bank/upsr/bm/E	200	2003-11-24 19:3	GET

Figure 11: Table after data is transferred to database

Fig. 11 shows the server log data after transferring to database and note that not all attributes are shown in the Fig. 11 due to the space restrictions. Several attributes are ignored and the interesting fields are included in the database. The algorithm that implements this function is written as:

```
Declare Variables

  Set DB = Server.CreateObject("ADODB.Connection")
  Set RS = Server.CreateObject("ADODB.Recordset")

  ConnStr = {MsAccess Driver}
  DB.Open ConnStr
  RS.Open TableName, ActiveConnection,


      Add Data
      … …
      ……
      RS.Update
  Set Rs = Nothing
  DB.Close
```

Figure 11: Algorithm transfer to database

Fig. 5.4 illustrates the algorithm to perform data transferring from flat file (original log file) to database. Mining task can be performed to produce the useful patterns, while data already in database.

V. CONCLUSION AND FUTURE WORK

After conclude the pre-processing. The cleaned data is stored in databases to be used for fitting in the Generalized Association Rules for rules generations. Conversely, the raw data before analyze is about 1377738 records.

Nevertheless, the progression of preprocessing data are prepared discretely due to the system is not currently incorporated. The data preprocessing are prepared separately suitable to the massive amount of data for each log files.

Extraction is a process of removing out uninteresting data or attributes. The web server logs contains 18 attributes, however removing process has taken out 17 attributes considered uninteresting and only 1 attribute known as "URL" are left in the databases.

Data filtering perform by removing unwanted patterns from each record in the database. Since the pre-processing techniques performed is to mine the interesting patterns, the data end with *.jpg, *.gif, *.bmp be removed. The final data after all process completed is about 38,890 records. The final data will be fed into Generalized Association Rules for rule generation and calculating the interesting rules by producing the support and confidence value.

In future work, more semantic information will be introduced into mining system so queries of similar meanings can be clustered and generalized. In addition, more log files of longer periods of time (such as months) are required to fabricate more reliable and more useful Rules Mining Algorithm, which will improve further the performance of the Web Servers.

REFERENCES

[1] Haigh, S. and Megarity, J. Measuring Web Site Usage: Log File Analysis. Network Notes #57, 1998.

[2] Pramudiono, I. Parallel Platform for Large Scale Web Usage Mining. PhD Thesis, 2004

[3] Tsuyoshi, M and Saito, K. Extracting User's Interest for Web Log Data. Proceeding of IEEE/ACM/WIC International Conference on Web Intteligence (WI'06), 2006.

[4] Ciesielski, V. and Anand, L. Data mining of web access logs from an academic web site. Design and application of hybrid intelligent systems, 2003. Pp 1034 - 1043

[5] Jeffrey, X. Y., Yuming, O, Zhang, C, Zhang, S. Identifying Interesting Customers through Web Log Classification. IEEE Intelligent Systems #20, 2005. pp 55-59.

[6] Natheer, K. and Chan, C.C. Active User-Based and Ontology-Based Web Log Data Preprocessing for Web Usage Mining. Proceedings of theIEEE/WIC/ACM International Conference on Web Intelligence (WI '06). 2006.

[7] Davidson, B. D. (2001). Web Traffic Logs: An Imperfect Resource for Evaluation. Ninth Annual Conference of The Internet Society.

[8] Rubin, Jeffrey (2003), *Integrating Content Management Systems with Legacy Applications.* Collegiate Sports Information Director's Association of America (CoSIDA), Cleveland, Ohio, July 2003

[9] Drott, M. C. (1998). Using Web Server Logs to Improve Site Design. *Association for Computing Machinery (ACM) Proceeding of the Sixteenth Annual International Conference on Computer Documentation. pp. 43 – 50.*

[10] Novak and Hoffman. (1996). *New Metrics for New Media: Toward the Development of Web Measurement Standards.*
http://www2000.ogsm.vanderbilt.edu/novak/web.standards/webstand.html
[Date Accessed: 28 February 2008].