2005

Empirical study of error behavior in Web servers

Ajay Deep Singh
West Virginia University

Empirical Study of Error Behavior in Web Servers

Ajay Deep Singh

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Committee Members
Dr. Katerina Goseva – Popstojanova, Ph.D., (Committee Chair)
Dr. Bojan Cukic, Ph.D.
Dr. Matthew C. Valenti, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2005

ABSTRACT

Empirical Study of Error Behavior in Web Servers

Ajay Deep Singh

The World Wide Web has been a huge success, bringing the Internet to widespread popularity. For Web based systems to deal effectively with increasing number of Web clients, it is very important to understand the basic fundamentals of Web workload and error characteristics. In this thesis we focus on detailed empirical analysis of Web server error characteristics and reliability based on the data extracted from eleven different web servers. First, we address the data collection process and describe the methods for extraction of workload and error data from Web logs. Then, we analyze the Web error characteristics which include unique errors, frequency of occurrence of unique errors and top files causing errors. Furthermore, we analyze the relationship between errors among Web workload and estimate request-based and session-based reliability. The discussion presented in this thesis shows the sessions-based reliability is better indicator of user perception of Web quality than request-based reliability. Finally, we analyze and develop heuristic search criteria to identify sessions which indicate unusual server behavior, such as extremely long sessions and sessions with large number of server errors. The results of our study provide valuable measures for tuning and maintaining of Web servers.

ACKNOWLEDGEMENT

I am eternally grateful to my advisor Dr. Katerina Goseva – Popstojanova for her constant support and encouragement. I am also grateful to her for introducing me to interesting research area in software development.

I am also grateful to my other committee members, Dr. Bojan Cukic and Dr. Mathew C. Valenti for their support. I would like to thank NASA IV & V Facility, Fairmont, West Virginia which provided financial support for my graduate studies through NASA Office of Safety and Mission Assurance (OSMA) software assurance research proposal. Finally I would like to thank my parents, my brother and sister in law and friends for their constant help and support.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Background

The World Wide Web (WWW) has become the most popular part of Internet. It is essentially a huge client-server system with millions of clients and servers distributed worldwide. An exponential growth in clients and servers has been noticed in the past couple of years [8]. The growing availability of the Internet has led to significant increase in the use of the World Wide Web [17]. Due to this enormous growth of Web and according to Web users demand of 24/7 Web availability to satisfactory levels, certain factors like performance, scalability, availability, and security are necessary to address. In order to address each of these factors, it's very important to understand the basic trend of network traffic flow, general user behavior on Web, network failures, network congestions, request load on server, etc.

A lot of empirical research has been done to understand the patterns of Web traffic and Web server's behavior towards this traffic. But due to the exponential growth of Web users and rapid change in Web technologies, more studies and analysis are still required to be done in this area.

The information about all the Web traffic (requests to and responses from server) is stored in Web server logs. Every Web server available today maintains Web logs as well as provides the feature to choose log format from several available log formats. These logs contain a lot of information about each request made to the server.

There are different types of logs, containing different types of information. These include access logs, error logs, and referrer logs.

- Access logs contain information about all the requests & responses coming in to the server.
- Error logs contain information about the errors encountered by the Web server (requests not successfully fulfilled). These errors can be either client side errors or server side errors.

- Referrer logs are similar to access logs but with additional information of the referrer from where that request was generated.

Proper study and analysis of the Web logs can show the right picture of the Web server's reliability and the ways to improve the performance in various dimensions like making more profit (E-Commerce sites), addressing the Web site failures, finding out Web user's needs, etc. There are many profit and non-profit organizations that are working in the field of log analysis.

In this thesis we study the characterization of the error behavior, both on request and session level where sessions are termed as sequence of requests coming from the same user within a given time threshold. This thesis also includes the characterization of request-based and session-based reliability.

1.2. Motivation and Research Objective

Realizing the increasing Web-based system's dependency in almost all the fields (banking, schools, science, Web marketing etc), forces us to do more detailed and rigorous studies to avoid any kind of server failures and improve reliability. It is estimated that the economic loss because of unavailability due to failures or poor performance is in the range of billions of dollars per year in United States alone [17]. In addition, Web technology is now used even in real-time critical application, which forces us to address performance (response time) issues. An example is the Web Interface for Telescience (WITS) developed at Jet Propulsion Laboratory which enables scientists and engineers to collaborate in daily mission operations from multiple geographically distributed locations via the Internet [6].

There are many tools available in the market that analyzes Web logs [26]. Most of these tools aim towards the commercial need *i.e.* identify Web trends improve the profit and number of clients, instead of server performance analysis, reliability and cost effective Web quality improvement.

The analysis done in this thesis is based on real data extracted from Web logs of 11 different servers. The aim of this thesis is:

- Characterizations of the errors behavior at request level.
- Characterization of request-based reliability.
- Characterization of session-based reliability.
- Analysis of so called weird sessions that contain large number of errors.

Chapter 2: Related Work and Our Contributions

2.1. Background and Related Work

All the communication in the Internet takes place in request-response fashion *i.e* a client always makes request to the server and in response to that request the server responds [34][24]. There has been a lot of research done in the past, focusing on Web traffic characteristics.

In [4], analysis is done on six different access logs and emphasis was placed on characterization of document type, document size, document referring behavior, and geographic distribution of requests. Distribution of the file size in web server requests was discussed in [9]. The WWW transfers from the actual Web logs are consistent with self-similarity notion, characterized by bursts and heavy tail distributions, were shown in [12]. Similar findings were reported in the recent study [11] of the end-end response time required to download Web pages from a set of well-known Web sites. Tool for measurement of Web server activity was developed in [2], to help identify bottlenecks.

The concept of sessions was introduced for the first time in [10] as a unit of Web workload. Session is described as a sequence of requests coming from the same user during single visit to the Web Site. Session boundaries are delimited by a period of inactivity by a user. Some Web sites enforce a threshold and close inactive sessions to save resources allocated to these sessions. In [3], authors studied how the number of sessions is affected by changing the threshold (period of inactivity). They also focused on other session characteristics like distribution of number of requests per session, session length, and inter-session arrival times. Authors in [18] studied the request, function, and session characteristics of two actual e-commerce sites.

Although lot of research is done on characteristics of Web workload [5], there are very few papers published that focus on analysis on error behavior and characterization of errors. The information about unsuccessful requests is reported by the server in access logs in the form of response codes (status codes). Analysis has been done in [3] on server response codes from the access logs of 1998 World Cup Web Site and reported

distribution of number of successful requests partial content responses, not modified responses, and responses with errors. In [16], authors talk about the information extracted from the error logs of Web Server of School of Engineering and Applied Science at the Southern Methodist University. They also reported number of different types of errors per day, computed request-based reliability as the number of successful request over the total number of requests.

Apart from academic research, there are many commercial log analysis tools available in the software markets that are used in the industrial applications. Most tools available [26] provide limited analysis and generate predefined fixed reports such as Kbytes transferred, number of hits, unique visitors, user's geographical location, information about the browser and operating system, and so on. They neither consider sessions at all nor provide limited information about sessions. There are very few tools which analyze the error logs or errors occurring at the server. Moreover most of tools available such as WebSideStory HBX Analytics [29], focus on commercial aspect for example the track of request coming from referrals so that the company can decide the area they need to advertise. Commercially available tools are more targeted at marketing and business than at information technology departments. However it is worth mentioning that there are few tools that provide some kind of session and error analysis. For example, Webtrax [34] provides limited information about sessions, but no information on errors. Sawmill [32] provides information on sessions, as well as error analysis based on error logs. NetTracker [31] and FastStats [28] also provide information on sessions and some information about errors based on error and access logs.

2.2. Our Contributions

In this thesis we empirically characterize the error behavior, request-based and session-based reliability based on data extracted from eleven real web servers. This thesis includes part of our work presented in [14] and [15]. In this thesis we presented the analysis done on logs of eleven different servers, having significant quantity of data. Our contributions include:

- **Development of prototype tool:**

  Although there are many log analysis tools available, either they do not consider sessions or provide limited predefined session reports. Moreover most of the tools available do not analyze error logs, and do not perform analysis considering server reliability issues. To overcome these limitations we have developed a prototype tool that extracts detailed workload and error information from Web access and error logs, having flexibility and addressing Web reliability issues.

- **Characterization of errors encountered by server:**

  We empirically analyze web access and error logs for this purpose. This research work includes detailed analysis of Web error characteristics. Analysis includes type of errors, severity of errors, unique errors, frequency of error occurrence and top three and ten files most frequent with errors.

- **Characterization of request and session-based reliability:**

  For this purpose, we analyze number of errors and total number of requests for both private and public servers and compared them. We also empirically analyze session based reliability, arguing that session-based reliability is better indicator of server's quality than request-based reliability. Unlike some of the earlier papers focused on Web reliability that presented models that were not supported by real data [23] [1], in this paper we present empirical analysis of the request-based and session-based reliability based on actual logs from eleven Web servers.

- **Weird session analysis:**

  Analysis is also done on suspicious sessions i.e. those having unusual behavior compared to other sessions. Filtering of weird sessions is done using different parameters like number of requests, number of errors, and duration of sessions. Further discussion is done on how such analysis can help identify attacks, and unusual activity at Web server.

Chapter 3: Data Extraction

In this chapter we describe types of logs used for analysis, information contained in logs and the steps involved in data extraction from the logs. Web server logs contain highly relevant information about the Web workload and errors encountered by the Web server. Therefore, data extraction should be done carefully especially when there are huge log files. We collected and analyzed the log files from eleven real operational Web servers. This type of empirical study is called observational [7], [25] since, unlike controlled experiments, there are no treatments or controlled variables, that is, the subject under study is not perturbed.

3.1. Information in Logs

The Web servers maintain different types of logs to keep track of all requests coming in to the server and server's response to those requests. These logs are necessary to keep track of activity and performance of the server and also for the checking the errors encountered by the server. The logs which are widely used are as follows:

3.1.1. Access logs

Access logs have an entry for each request coming in to the server [27]. The format of access log is highly configurable. There are a few types of access log formats available, for example custom log format, and combined log format. Example of such entry from access log (using combined log format) of a Web Site using Apache Web Server is shown in the Figure 3.1.

```
1.1.1.2 -- [023/Dec/2003:00:15:27 -0500] "GET /stats-usage/www/index.shtml HTTP/1.1" 200 14351
```

Figure 3.1: Access Log Sample Entry

Information contained in the access log entry is explained below:

a) Client IP Address:

   If the *HostnameLookups* feature is on, then server tries to get the Hostname before IP address. It usually affects the server's performance so it is not generally recommended.

b) Client Identity:

   It gives the information about the identity of the client. This information is highly unreliable so it is seldom used [27]. The "hyphen" in output indicates that this information is not available.

c) Authenticated Client Userid

   It represents the userid of the client for the HTTP authentication.

d) Date and Time of Request

   It tells the exact time at which server finishes processing the request.

e) Method of Request

   It represents what type of method is used by the client to put the request. Most of the times it's either GET or POST.

f) URI of File Requested

   This piece of log entry indicates the requested server resource by the client. Note that it is not the complete path but the URI.

g) Protocol used

   Next item in the log entry is Protocol used by the client for example "HTTP/1.0".

h) Status code

   Status code or response code is one of the most useful pieces of information contained in the log entry. It is generally a 3 digit value and there is predefined meaning of each value. Status code gives idea about server's response for the request. We will study response codes in detail in the following section.

i) Bytes Transferred

   The last piece of information in the log entry is the number of bytes transferred to the client to fulfill the request.

Status Codes:

Status code is a very valuable information contained in the access log entry. This piece of information is also sent to the client along with the response. It is a 3 digit number representing the status of the server's response to the request made by the client. Before getting into details of each status code here is the brief overview of the types of status codes:

- 2xx - OK, i.e., request was successful
- 3xx - The request was redirected
- 4xx - Client side error
- 5xx - Server side error

Explanation of each status code:

The possible status codes with brief explanation [33] are given below categorized according to their range:

2xx - Successful Client Requests

- 200 OK
- 201 Created
- 202 Accepted
- 203 Non-Authorative Information
- 204 No Content
- 205 Reset Content
- 206 Partial Content

3xx - Client Request Redirected

- 300 Multiple Choices
- 301 Moved Permanently
- 302 Moved Temporarily
- 303 See Other
- 304 Not Modified
- 305 Use Proxy

4xx - Client Request Errors

- 400 Bad Request
- 401 Authorization Required
- 402 Payment Required (not used yet)
- 403 Forbidden (Permission Denied)
- 404 Not Found (File does not exist)
- 405 Method Not Allowed
- 406 Not Acceptable (encoding)
- 407 Proxy Authentication Required
- 408 Request Timed Out
- 409 Conflicting Request
- 410 Gone
- 411 Content Length Required
- 412 Precondition Failed
- 413 Request Entity Too Long
- 414 Request URI Too Long
- 415 Unsupported Media Type

5xx - Server Errors

- 500 Internal Server Error
- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Timeout
- 505 HTTP Version Not Supported

3.1.2. Error logs

Error logs [27] are used by the server to record the information about any kind of errors that server encounters while processing the request. For each entry of error reported in access log (with 4xx or 5xx status code) there is corresponding entry in the error log. The format of the error log is relatively free-form and descriptive. Error logs provided by Apache server cannot be customized i.e. information cannot be added or removed. Example of entry in error log is shown in Figure 3.2.

```
[0Sun Oct 26 06:40:00 2003] [0error] [0client 66.196.90.18] File does not exist:
/projects/www/htdocs/~grove
```

Figure 3.2: Error Log Sample Entry

The information contained in each entry of error log is as follows:

a) Date and Time

   It gives information about the date and time of occurrence of error encountered by the server.

b) Level of severity

   This piece of log entry informs the severity level of the error and the detailed discussion is provided in the next section.

c) Client IP Address

   As the Access Log, error log also contain client IP address.

d) Error message

   Error logs also give some kind of error message that is generally a text message containing information about the reason of error occurrence.

e) Exact URL of the error:

   The exact path of the file (causing error) requested is also reported in the error log entry.

Error Severity Level:

The error logs provide very important information about the severity of errors, which help to prioritize errors while fixing them. Table 3.1 gives the brief information about the severity levels of possible errors in logs of Apache Web server [27].

Severity	**Description**
Emerg	Emergencies - system is unusable
alert	Action must be taken immediately
crit	Critical Conditions
error	Error conditions
Warn	Warning conditions
Notice	Normal but significant condition
Info	Informational
Debug	Debug-level messages

Table 3.1: Severity Level of errors in Error Logs

In *Apache Web Server*, the severity level of error logs can be configured using LogLevel directive and the default level set by server is "warn". The server only logs the errors which are equal or more severe than the severity level set at the time of server configuration. It should be noted that server also logs cgi errors which are merely used for debugging purposes. For cgi errors there is no information of IP address and severity level reported in error logs. Moreover, there is no corresponding entry in access log. Furthermore, for '*notice*' level errors there is no information of IP address & file request in error log and there is no corresponding entry in access log. For a particular type of status code (4xx or 5xx) in access log, the corresponding message in the error log might not be same.

3.1.3. Referrer logs

Referrer logs [27] are just an extension of access logs and they contain same information as access logs, with some additional information at the end of each entry in the log. This additional information is referrer information which basically tells from where the request is generated originally. For example it is possible that the link to your Web site is given in someone else's Web site and a user accesses your Website using that link. In this case referrer information in your server's referrer log will be the other Web sites URL.

The referrer information is sometimes useful when you want to know from which Websites the user is entering your Web site. It is mainly used for commercial purposes as if you know that most requests are coming from particular advertisement of your Web site link and not from other advertisements then you might want to change or improve the other advertisements and thus get more clients.

From the example shown in figure 3.3, it is clear that the all the information in referrer log is the same as access log entries except the last referrer information. The server just appends the access log entry with Referrer header of the incoming request.

```
61.18.186.130 - - [026/Dec/2004:06:56:24 -0500] "GET /~trapp/wvumatlab.htm HTTP/1.1" 200 9664

http://www.google.com.hk/search?hl=zh-TW&q=solve+equation+by+matlab&meta= -> /~trapp/wvumatlab.htm
```

Figure 3.3: Referrer Log Sample Entry

3.2. Logs used for analysis

The logs used in this thesis were obtained from eleven Web servers : three public and three private Web servers at NASA independent verification and validation (NASA IV & V), the Web server at Lane Department of Computer Science and Electrical Engineering (CSEE) at West Virginia University, the campus wide Web server at West Virginia University (WVU), Web server of commercial Internet provider ClarkNet, the Web server at NASA Kennedy Space Center (NASA-KSC), and the campus wide Web Server at the University of Saskatchevwan. The data sets obtained from NASA IV & V, CSEE and WVU consists of access logs and error logs, while the rest of datasets consists of access logs only. The three data sets, for which only access logs were available, were downloaded from the Internet traffic archive [30].

The datasets used in this thesis for the analysis are from different domains: seven of these are from research institutions, three are from educational institutions and one from commercial Web site. Moreover, three of the servers are private and the other are public.

3.3. Data extraction from logs

Web logs are in ASCII format. Direct analysis on raw logs directly is generally not very flexible and efficient. Therefore data extraction includes parsing each log entry into its smallest units of information and recording them into relational databases. The data extraction and analysis process is shown in Figure 3.4. After generating relational database from raw logs, sessions are created using database scripts and then the data is processed to obtain valuable results.

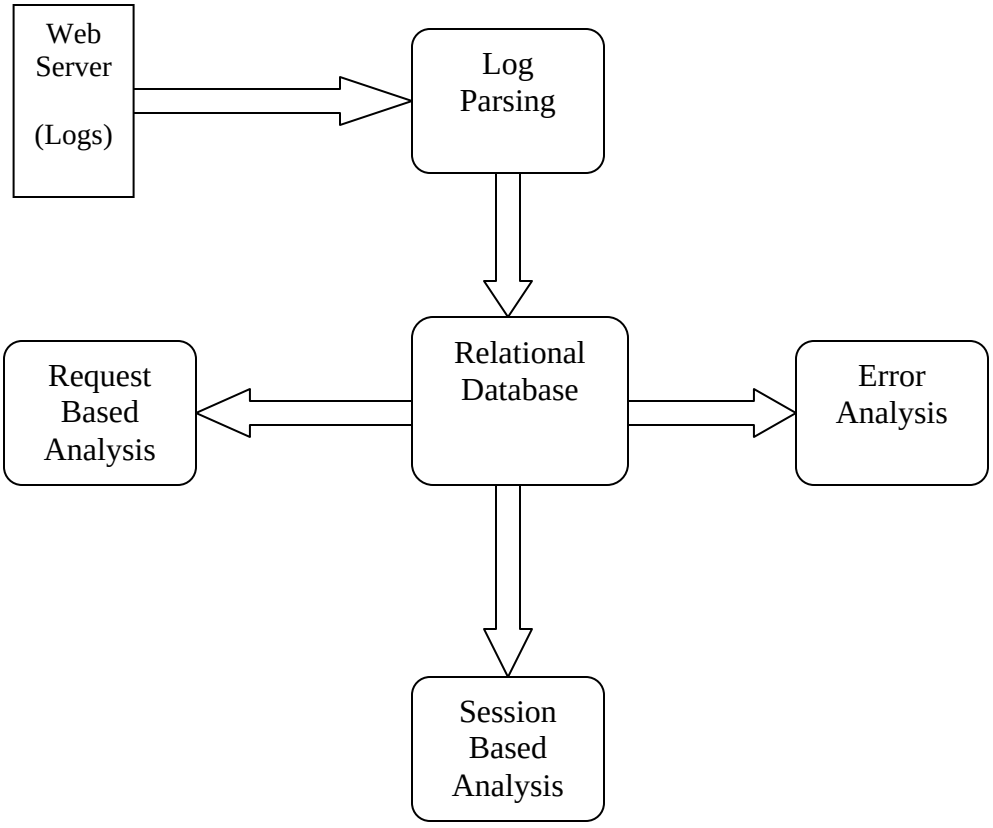


Figure 3.4: Data Extraction and Analysis Process

3.3.1. Request Data Extraction

Access logs contain the information of all the requests received by the server. Data extracted from access logs at the eleven servers is summarized in Table 3.2 which includes the log duration along with the information about the workload. Since the span of duration of all logs varies from three weeks to seven months, in order to compare the results of all the servers we normalized the workload by calculating average requests per day, sessions initiated per day, bytes transferred per day etc. It is clear from the table that WVU Web server has the maximum traffic and NASA-Pvt2 has the lowest traffic. Also number of visitors at WVU is the highest as the number of sessions per day of WVU is highest.

Data Set	Log Duration	Start Date	Requests	Average Requests Per Day	Sessions	Average Sessions Per Day	MB Transferred	Average MB per Day
NASA-Pvt1	20 weeks	Apr 6 2004	22,623	159	921	6	474	3.33
NASA-Pvt2	20 weeks	Apr 6 2004	92,112	649	4,544	32	162	1.14
NASA-Pvt3	20 weeks	Apr 6 2004	489,004	3,444	23,907	168	2,192	15.43
NASA-Pub1	20 weeks	Apr 6 2004	92,541	652	18,443	130	8,988	63.30
NASA-Pub2	20 weeks	Apr 6 2004	731,504	5,151	57,889	408	6,665	46.93
NASA-Pub3	20 weeks	Apr 6 2004	108,200	762	15,850	112	4,572	32.20
CSEE	6 weeks	Mar 3 2003	5,815,202	135,237	252,753	5,873	80,913	1,881
WVU	3 weeks	Jan 1, 2004	37,870,087	1,803,337	487,637	23,220	96,953	4,616
ClarkNet	2 weeks	Aug 28, 1995	3,328,632	237,759	283,961	20,282	27,646	1,974
NASA-KSC	2 months	July 1, 1995	3,461,612	59,682	306,523	5,284	62,488	1,974
Saskatchwan	7 months	June 1, 1995	2,408,623	11,255	463,684	2,166	12,344	57

Table 3.2: Summary of the server workload

3.2. Session Extraction

Session is very important concept for analyzing Web workload characteristics. It is defined as a sequence of requests from the same user (same IP address) during a single visit to the web site. The web session starts when a user requests service for first time and ends when there is no request from that IP for a set threshold time. For example, making any monetary transaction through an online banking web site, the user establishes sessions with the bank web server. All the requests during that transaction until the user logs out or sits idle for specific session threshold time belong to that session. There are two main points to discuss about extracting sessions from the logs:

1) *IP address used for user identification*

   Most of the research papers consider each IP address as a distinct user, which is clearly not true in all cases [3] [20]. There is a possibility of existence of proxy server between user computer and Web server due to which the proxy IP address is reported in the logs, rather than the address of the original generator of the request. It is possible that the machine used is for public access which means different users create different sessions at the server from same IP address. This directly affects total number of users generating sessions. Despite the inaccuracy, we believe that using the IP address for user identification provides good approximation.

2) *Time threshold to delimit sessions*

   Threshold to delimit sessions is defined as the time of inactivity between two sessions from the same IP address. We examined the number of sessions by varying this threshold parameter from time duration of 1 minute to 40 minutes. Figure 3.5 depicts the variation in total number of sessions by varying the threshold. As the threshold increases from one minute, the number of sessions decreases rapidly. Furthermore when threshold goes beyond 30 minutes there is a little decrease in the number of sessions even with substantial increase in threshold. The result of this analysis confirms the fact of standard 30 minute [18]

threshold value therefore all the session-based analysis in this thesis is done using 30 minutes threshold value.

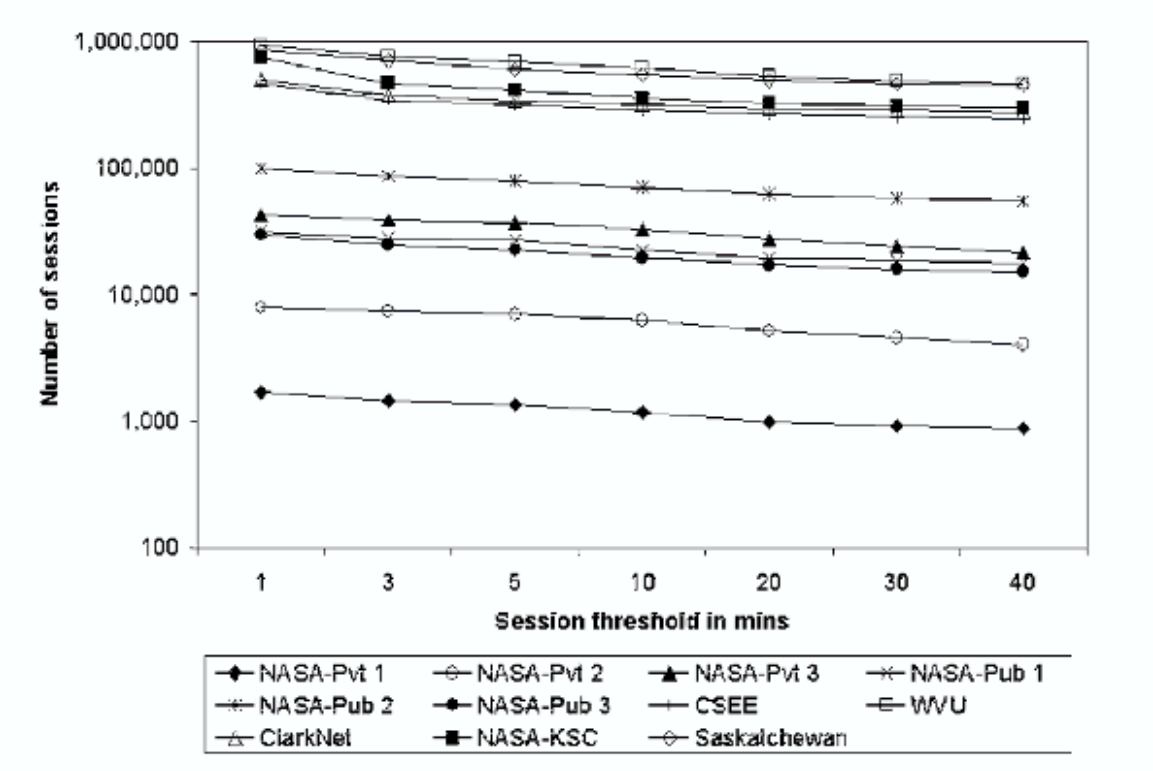


Figure 3.5 Effect of the session threshold on the number of sessions

Chapter 4: Analysis and Results

The analysis of data to extract the valuable information about server's quality is categorized as request-based error analysis and session-based error analysis. Also sessions showing unusual behavior are analyzed in case the web administrator needs to track down or get details about weird server activity.

4.1. Error Analysis

In this section we present more detailed analysis of the errors based on the data extracted from web access and error logs. Not all errors that are encountered by the server are different. Most of them are the same errors but have different time of occurrences. Most of the errors reoccur again and again thus raising the number of errors in logs. For example if there is a link for another page in the web site but actually that page does not exist then 'File does not exist' error will be observed in error logs. This error will occur as many times as the link is clicked, causing the error log to grow. Errors must be prioritized before fixing them and for this purpose we analyze unique errors, frequency of errors and top frequent files causing errors.

4.1.1. Severity

Percentages of errors with different level of severity are presented in table 4.1. It is clear from the table that most of errors from error logs fall in '*error*' severity category. Very few percentages of errors have alert, crit, warn or notice severity levels. Errors should be prioritized according to their severity level before fixing them.

Severity Level	NASA-Pvt 1	NASA-Pvt 2	NASA-Pvt 3	NASA-Pub 1	NASA-Pub 2	NASA-Pub 3	CSEE	WVU
emerg	0	0	0	0	0	0	0	0
alert	0	0	0	0	0	0	0.005	0
crit	0	0	0	0	0	0	1.196	0
error	100	100	100	100	98.761	100	97.382	100
warn	0	0	0	0	0.003	0	0.837	0
notice	0	0	0	0	1	0	0.579	0
info	0	0	0	0	0	0	0	0
debug	0	0	0	0	0	0	0	0

Table 4.1: Error Severity Distribution in Error logs

4.1.2. Unique Errors

An error with same error message and same file requested is defined as a unique error. Note that in error logs, it is quite possible for the same file to cause errors with different error messages. For example, the error messages *unable to include "/top_footer.html" in parsed file* and *unable to include "/bot_footer.html" in parsed file* both associated with file *AB-help.html* are considered as different unique errors.

For this analysis, we excluded the CGI errors which also occur in error logs. The CGI errors are just debugging messages which appear in the logs when CGI scripts do not run successfully. It is noticed that one of the CGI scripts in CSEE server generated half a million debugging error messages in the error log, causing it to grow enormously. Fixing this cgi script improved the quality of server, as well as saved the resources wasted for logging.

Figure 4.1 shows the percentage of unique errors over total number of errors in six NASA IV&V servers, CSEE server and WVU server. As we can see the web server NASA-PVT 3 has the lowest (2.04%) percentage of unique errors which means that most of the errors encountered by this server are the same and fixing this small percentage of

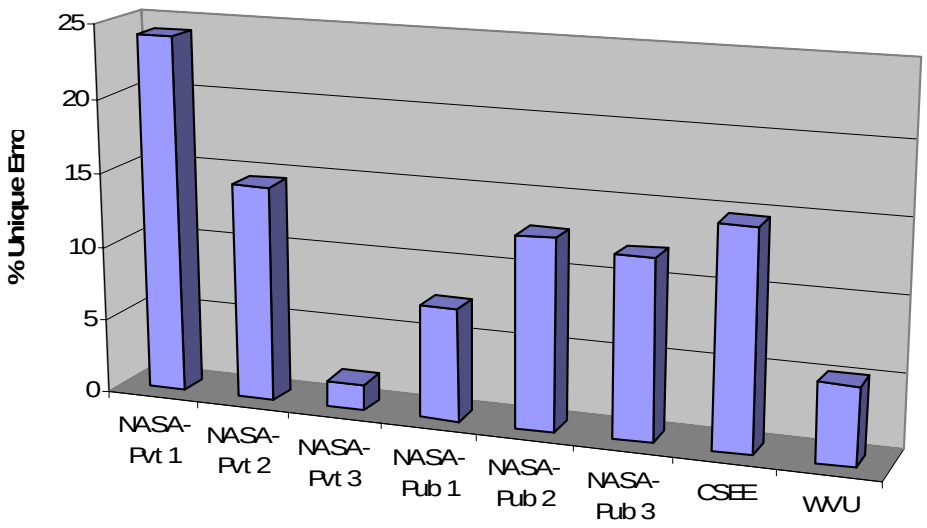


Figure 4.1: Percentage of unique errors

unique errors leads to cost effective improvement. NASA-PVT 1 has highest percentage of unique errors which shows that lot of distinct type of errors are present.

Figure 4.2 represents the number of unique errors, total errors and total requests. We can see that the total requests follow the same trend as total errors which is thoroughly analyzed in the next section. From the figure it is clear that most servers with higher number of requests have higher number of total errors and unique errors. The only exception are the NASA-Pvt2 and NASA-Pub1. In the case of NASA-Pvt2 and NASA-Pub1, both servers have the almost the same Web workload (92,112 and 92,541 requests), but NASA-Pub1 has almost 17 times more errors compared to NASA-Pvt2, which leads to lower request-based reliability (as explained in the next section). Despite the significantly higher number of total errors, NASA-Pub1 has almost half the percentage of unique errors than NASA-Pvt2. This shows that there are few errors in NASA-Pub1 which occur again and again, hence total number of errors will be high but not the fixing time/effort. These observations confirm widely accepted fact that software error behavior depends not only on the existence of faults, but also on the usage patterns [13].

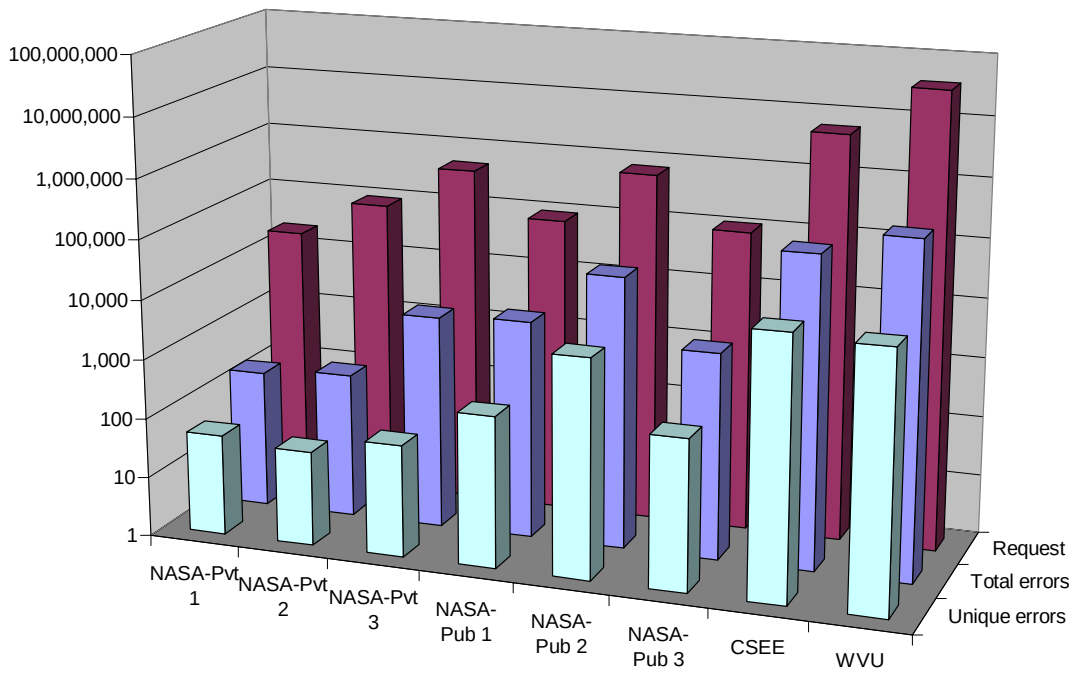


Figure 4.2: Number of unique errors, total errors and request

4.1.3. Frequency of Occurrence of Unique Errors

Building on the concept of unique errors, here we analyze the frequency of occurrence of these errors. Before fixing the errors we should prioritize them i.e. knowing which ones to fix first. The error prioritization constitutes towards more cost effective improvement of Web server's quality.

In Figure 4.3, we present the data from analysis done on frequency of occurrence of unique errors. It is clear that most of the errors have frequency of occurrence less than 1200 approximately. For example, there are 1,062 errors of NASA-Pub 2 which occur only once in 20 weeks, similarly 15,356 of CSEE unique errors occur

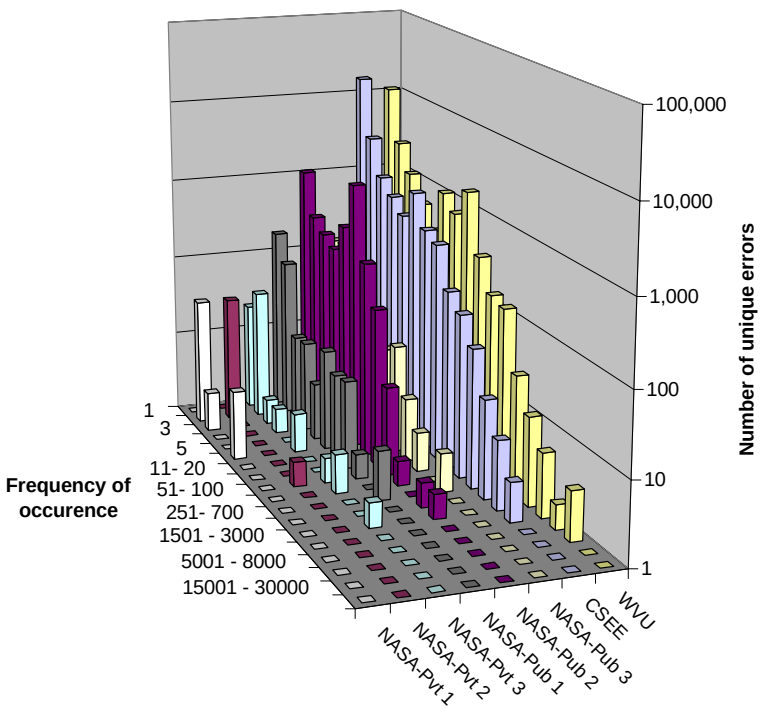


Figure 4.3: Frequency of occurrence of unique errors

only once in 6 weeks and 10,714 of WVU unique errors occurred only once in 3 weeks. Apart from this, there are errors which have extremely high frequencies and are important should be considered first at the time of fixing errors which will lead to cost effective improvement of Web quality. The highest occurrence of unique errors in NASA-Pvt 1, NASA-Pvt 2, NASA-Pvt 3, NASA-Pub 1, NASA-Pub 2, NASA-Pub 3, CSEE, and WVU servers are 50, 91, 1512, 990, 1666, 542, 7752 and 47415 number of times respectively. Thus, for example fixing a single error in WVU can basically remove 47,415 occurrences in the error logs.

4.1.4 Unique Files causing Errors

As we have discussed in section 4.1.3 ‘Unique Errors’, there is a possibility of a single file causing more than one type of error. In this section we have introduced the concept of unique files causing errors. It is noticed that in all the error data sets, the total

number of unique files causing errors is slightly smaller than the total number of unique errors, confirming the fact that some files have more than the one error message associated with them.

As an illustration, in Figure 4.4 we present the percentage of total errors that occur due to the top three most frequent files involved in generating errors at the Web server. From the figure it is clear that a significant percentage of the total number of errors (10.03% - 84.52 %) is due to only three files for each Web server. This analysis shows that fixing errors in these files can greatly improve the reliability of the server. For example, fixing the errors

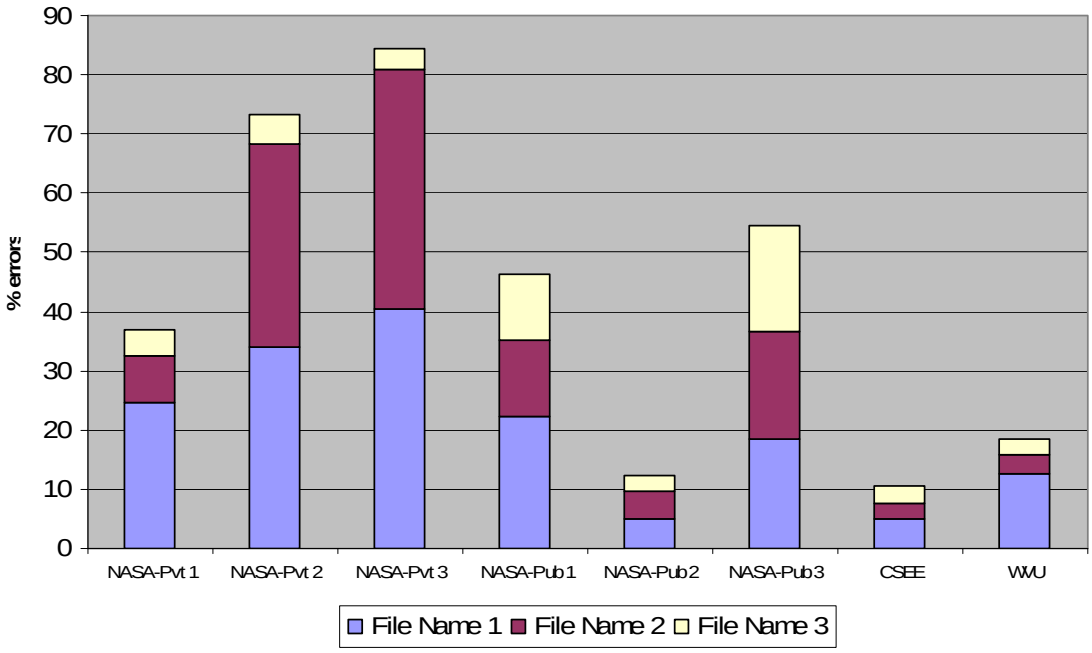


Figure 4.4: Error percentage due top 3 most frequent files

related to only three files with the highest frequency of occurrence in NASA IV&V Web servers eliminates significantly higher percentage of errors than fixing 36 – 1643 unique files with errors that occur 1 -3 times. For CSEE Web server fixing three most frequent unique files gives better results than fixing 13,390 unique files with errors that occur only once. Even more impressive for WVU (as it has the highest traffic in comparison to

other servers used for analysis), fixing three most frequent files eliminates three times more than fixing 15,334 unique files with errors that occur 1-5 times.

It must be noted that in the process of prioritization of errors for fixing purpose, in addition to frequency of occurrence of unique files with errors, one must consider the severity level of errors.

The same analysis is also performed on the top ten frequent files causing errors, which is shown in Figure 4.5. It is clear from the figure that by increasing the number of most frequent files number from three to ten, there is not significant or drastic change in percentage of errors which interprets that with fixing the top ten files would not lead to as significant improvement as by fixing the top three files.

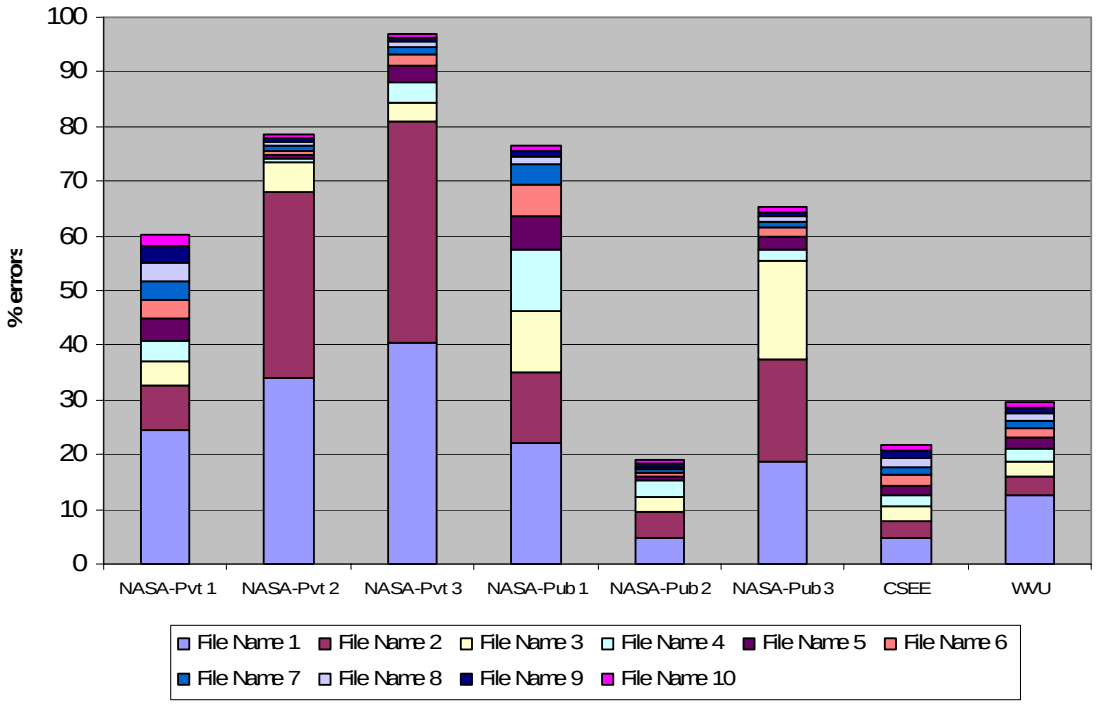


Figure 4.5: Error percentage due top 10 most frequent files

From the analysis it is concluded that *10 – 85 % of total number of errors are due to only the three files with errors that occur most frequently in each data set. It follows that fixing only three files in each web server results in significant increase of the Web reliability*.

4.1.5 Types of errors

The previous section shows the analysis done on the errors only but in order to measure the quality of web server it is necessary to know error characteristics with respect to requests. As mentioned in section 3.1.1, each log entry in access logs contains a response code which gives information about the server's response to that request.

Only few of the response codes occur in most of the requests (more than 99% of requests). These response codes are 200, 206, 301, 302, 304, 4xx (client side errors) and 5xx (server side errors). The percentage values of the data obtained by parsing the requests with respect of their response codes are presented in table 4.2.

Status Code	200	206	301	302	304	4xx	5xx	Rrequest
NASA Pvt1	76.1054	1.5487	0	0	20.3968	1.6411	0.0000	0.9836
NASA Pvt1	55.2221	0.0000	0	0	44.3788	0.2895	0.0000	0.9971
NASA Pvt1	52.9922	0.1282	0.0035	0	46.0442	0.8315	0.0000	0.9996
NASA Pub1	77.3020	4.1331	0.5232	0	12.8826	4.9875	0.0097	0.9500
NASA Pub1	75.4434	2.0400	0.1887	0.0814	17.0935	4.8358	0.0297	0.9513
NASA Pub1	71.2410	8.3939	0.6110	0.0000	16.9023	2.7032	0.0083	0.9729
CSEE	26.5694	1.2581	1.2719	22.3007	45.7964	2.8031	0.0004	0.9720
WVU	54.8073	0.1964	0.2604	0.1149	43.6235	0.9833	0.0167	0.9900
Clarknet	88.7764	0	0	0.8737	8.0736	2.2037	0.0616	0.9773
NASA KSC	89.5687	0	0	2.1109	7.7066	0.6107	0.0031	0.9939
Saskat- chewan	91.0692	0	0	1.6904	6.2955	0.9216	0.0233	0.9906

Table 4.2: Breakdown of status codes and request-based reliability

Table 4.2 reveals that the majority of requests resulted in responses without errors (response codes 2xx and 3xx). Four of the web sites (NASA-Pvt 2, NASA-Pvt 3, CSEE and WVU) have significantly higher percentage (43.62 – 46.04%) of requests which result in 304 response codes (Not Modified). In case of CSEE server, 304 (Not modified) response codes are even more than 200 (Successful) response codes. For NASA-Pvt1, NASA-Pub2, and NASA-Pub3 percentage of requests with 304 response codes is in range 12.90 to 20.39%. But for old data sets (ClarkNet, NASA-KSC, and Saskatchewan) this percentage with 304 response codes is less than 9%, which clearly shows, improved caching capability of Web which is especially effective for certain usage patterns that include revisiting the same content and/or Web sites that contain pages with large amount of static content.

4.1.6. Request-Based Reliability

There are very few requests resulting in 4xx and 5xx response codes in comparison to total number of errors in all data sets. Moreover, 4xx errors (client side errors) are comparatively one to four times more frequent than 5xx errors. Also most of these 4xx errors are 404 (*File Not Found*) errors. This implies that the server is unable to find the requested resource. The 404 errors that occur due to broken/bad links are counted as web errors.

Due to errors encountered by the server the reliability of the server goes down. The reliability of the server, R_{reques} can be measured using Nelson's model [19] as follows:

$$R_{request} = 1 - \frac{f_r}{n_r} = \frac{n_r - f_r}{n_r}$$

where f_r is number of requests which results in erroneous codes (4xx and 5xx), and n_r is total number of requests.

The results obtained by estimating the request-based reliability are shown in Table 4.1 and their graphical representation is shown in Figure 4.6.

The analysis done to estimate request-based reliability shows that:

The request based reliability is in the range of 0.9500 -0.9971.

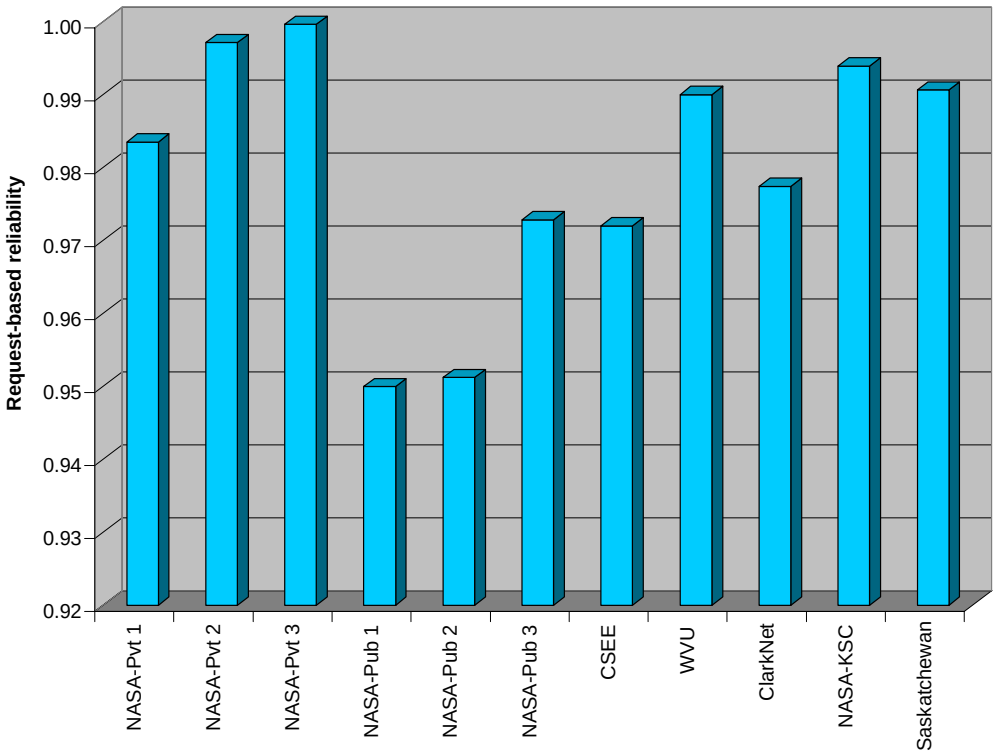


Figure 4.6: Request Based Reliability

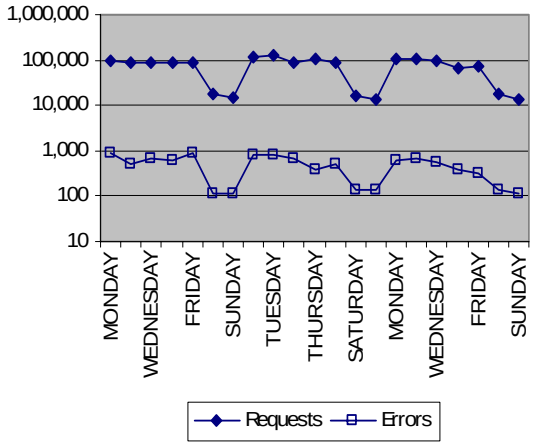
It should be noted that estimates of request-based reliability are conservative due the fact that some of the errors, such as unauthorized access (with response code 401), may not be errors but the behavior we expect from the server.

Another example of the same form is that not all 404 errors (which are basically file not found error) are errors, as might be possible that instead of clicking any link to get access of some resource at server, user actually types the whole URL itself and by mistake misspells the name of the file. For such requests, the server is going to search for the misspelled resource and eventually result in *file not found* error. The response code 403 (Permission denied) can also be considered as one of the examples which can fall in this category, making the reliability estimate more conservative. As 403 occurs when user tries get access to some password protected resource with wrong credentials, then it is as expected by the server to behave this way, but such response of the server counted as web error.

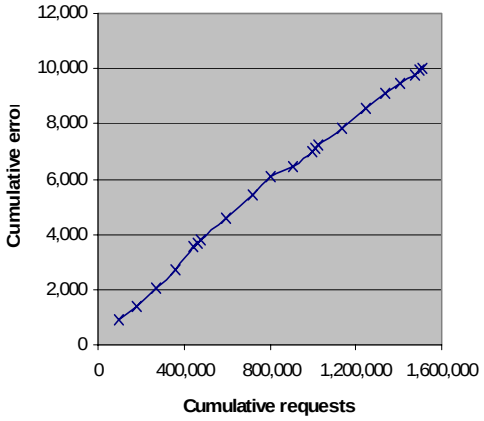
To make the reliability estimates more accurate, analysis of different types of errors is very important and part of a future work.

4.1.7. Relationship between errors and workload

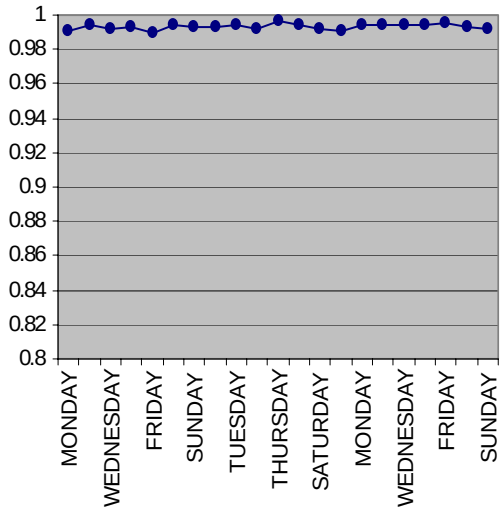
This section presents the study on the relationship between the workload (number of requests) and error behavior, as well as the variability of request-based reliability over



(a) Request and Error per Day



(b) Cumulative Error vs Request



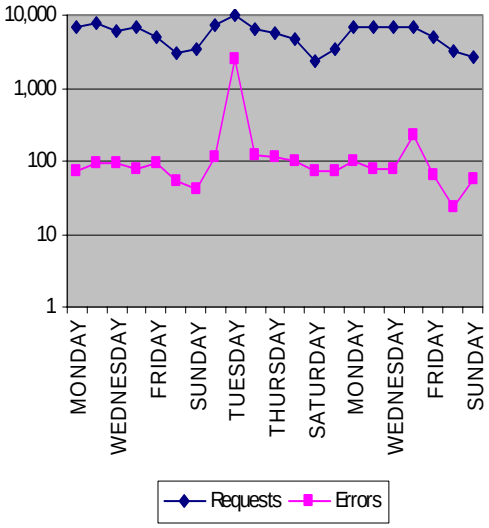
(c) Request-based reliability per day

Figure 4.7: Request vs Error Analysis for NASA-Pub3 data set

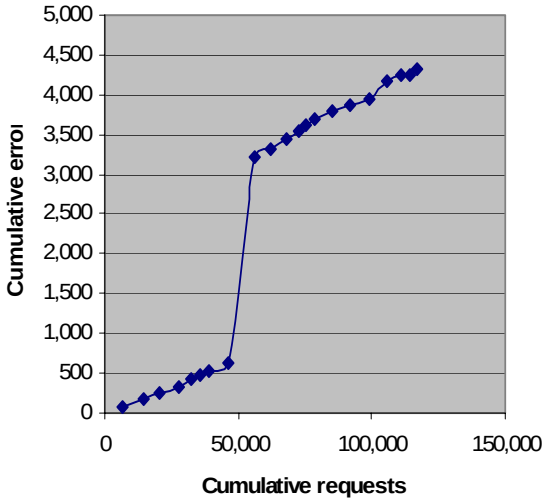
time. For these estimates, eight servers (six NASA servers, CSEE server and WVU server) are used since the error logs were not available for the remaining three servers. This analysis is done on 3 weeks of data.

Figure 4.7 shows the variation of the total number of errors and requests (web workload) for each day of NASA-Pub3 web server. In Figure 4.7 (a), it is perceptible that the total number of errors follows the same trend as total number of requests, and this is very obvious behavior as if the number of request varies then accordingly the number of errors also change. The valleys in the graph show a decrease of web workload during weekends which is expected since there are less number of requests/users during weekends.

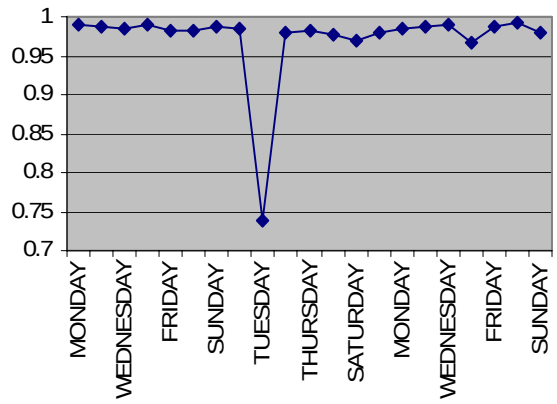
Figure 4.7(b) shows a graph of the cumulative requests verses the cumulative errors and it shows the linear behavior which confirms that errors and requests accumulate with same rate. Request-based reliability per day presented in figure 4.7 (c) shows that reliability per day remains almost constant, further confirming that the number of errors follows the exact pattern as of number of requests.



(a) Request and Error per Day



(b) Cumulative Error vs Request

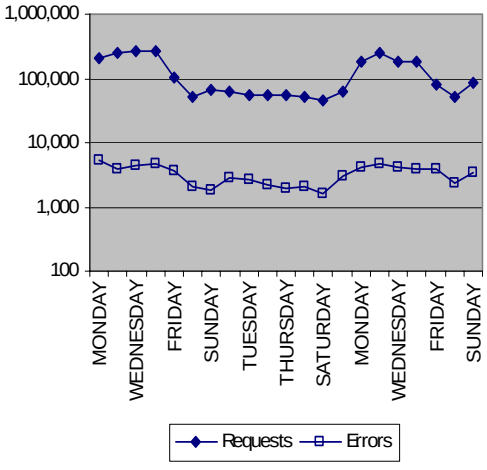


(c) Request-based reliability per day

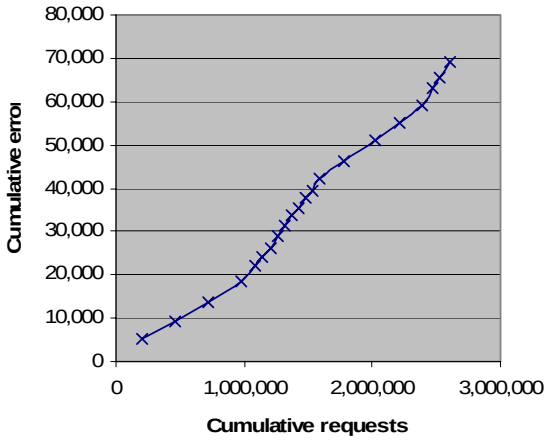
Figure 4.8: Request vs Error Analysis for NASA-Pub2 web server data set

Figure 4.8 shows request vs error behavior of another server (NASA-Pub2). It is seen that errors follow the same path as requests but for a particular day (Tuesday), the number of errors increase significantly. The behavior was due to some scripts that were run intentionally that day, which resulted in many errors. Cumulative request vs error graph confirms the same effect; as well as the significant downfall in reliability for that day, see (Figure 4.8 c).

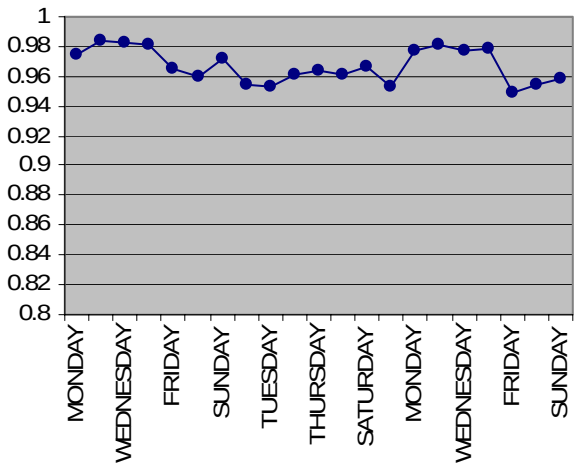
Figure 4.9 and 4.10 represents the same analysis for CSEE and WVU servers. Both these servers exhibit more web traffic and therefore their graphs show a clear picture about errors following same pattern as requests.



(a) Request and Error per Day



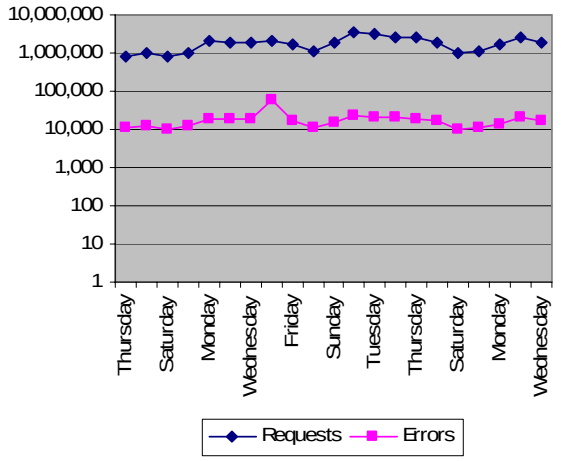
(b): Cumulative Error Vs Request



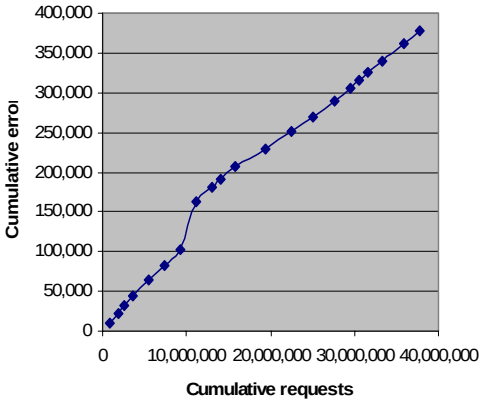
(c) Request-based reliability per day

Figure 4.9: Request vs Error Analysis for CSEE web server data set

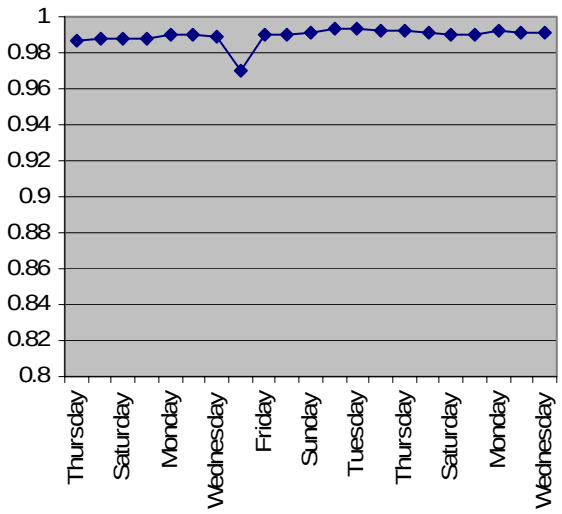
The valley in figure 4.10 (c), which is basically reliability per day of WVU server, shows the more errors encountered that particular day and this can be confirmed by looking at other two plots of WVU server i.e. figure 4.10 (a) and figure 4.10 (b).



(a) Request and Error per Day



(b) Cumulative Error vs Request



(c) Request-based reliability per day

Figure 4.10: Request vs Error Analysis for WVU web server data set

From this analysis it is concluded that both error intensity and workload intensity have a periodic component, with smaller values during weekends. It is also observed that the number of errors encountered per day is closely related to the workload intensity, that is, increased usage is accompanied by increased number of errors encountered. Furthermore, reliability for each day is examined which falls into a tight range between 0.9899 and 0.9664. Similar kind of study of relationship between workload intensity and errors was shown in [16].

This kind of analysis can be very valuable for web server administrator to find any unusual behavior of the server, which can be tracked down by looking into logs for that time period. Such plots prove to be really handy to monitor the errors and the workload of the web servers. More detailed discussion of unusual server activity in terms of sessions is presented in section 4.2.

4.2. Session-Based Error Analysis

In this section we present the error analysis done on the data extracted from the access logs in the form of sessions. Session-based reliability is also discussed in this section. For this analysis, distribution of errors within session is studied.

4.2.1. Error distribution within sessions

Figure 4.11 represents histogram of errors per session. It is obvious from the figure that most of the sessions do not show any error which means that requests result in error free responses in most of the sessions. Approximately 77 – 98% of sessions have requests with no erroneous status codes (4xx and 5xx).

Furthermore, as the number of errors increases, percentage of sessions decreases rapidly showing errors in most of erroneous sessions are very less (0-4 errors).

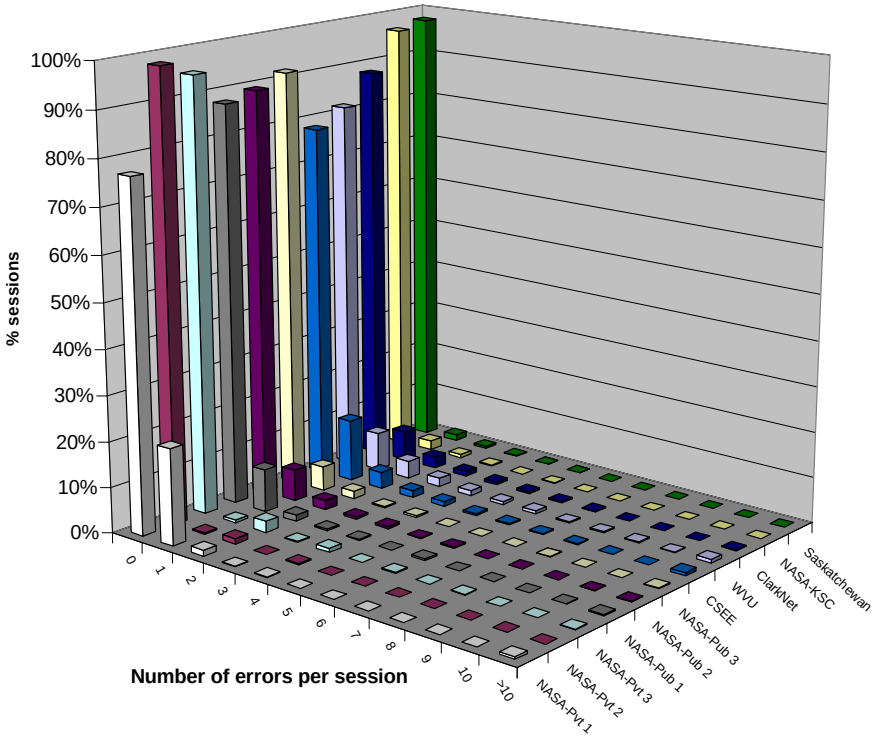


Figure 4.11: Histogram of number of errors per session

4.2.2. Session-based reliability

Session-based reliability can be interpreted as the probability that a user of a web server will not experience error in any of requests that constitute user's session. Even having one request with erroneous status code sessions are considered failure otherwise successful with absolutely no errors.

We believe that the session based reliability estimates are very important for measuring the ability of Web servers to process the entire sequence of requests without any error and they are the better indicators of the user's perception of the quality of the Web servers. We can estimate session-based reliability, $R_{session}$, as:

$$R_{session} = 1 - \frac{f_s}{n_s} = \frac{n_s - f_s}{n_s}$$

where f_s is the number of sessions having at least one request with erroneous request and n_s as the total number of sessions.

	NASA Pvt1	NASA Pvt1	NASA Pvt1	NASA Pub1	NASA Pub1	NASA Pub1	CSEE	WVU	Clarknet	NASA KSC	Saskat-chewan
$R_{session}$	0.7676	0.9806	0.9505	0.8770	0.8928	0.9182	0.7814	0.8166	0.8806	0.9650	0.9782

Table 4.3: Session-based reliability

The outcome of session-based reliability analysis is shown in table 4.3. An important observation is that session-based reliability is always lower than request based reliability for all web servers used for analysis. This is due to the fact that the sessions even with single erroneous response is considered as a failed session. In particular, sites exhibiting large number of sessions with very few errors will show smaller session-based reliability than request-based reliability. This means that many users will experience at least one error during a session.

It should be noted that the session-based reliability can be higher than request-based reliability. This can happen in the case when there are very few sessions having

significant number of errors and most of the sessions do not have any request with erroneous code. Thus, there might be relatively high number of erroneous responses, if they are distributed only within few sessions; session-based reliability will be high reflecting that only a small percentage of users will experience errors.

Since session-based reliability depends on the distribution of erroneous responses within sessions, there is no straightforward relationship between request-based and session-based reliability. Let us consider a simple hypothetical example. Consider there are two web servers, both having the same number of total requests, requests resulting in erroneous status codes, and same number of sessions. Then the server which has a uniform error distribution of erroneous responses over the sessions will exhibit smaller session-based reliability than request-based reliability. On the other side, the server that has skewed error distribution (very few sessions with significant number of erroneous responses) will exhibit higher session-based reliability than its request-based reliability.

4.3. Weird Session Analysis

In this section we develop heuristic search criteria for identifying sessions with unusual behavior. There are situations when the server's reliability goes significantly down or other unusual behavior of server, when it is required to track the details of cause of such cases. In section 4.1.6, it is seen that request based reliability of the server on certain days goes significantly down which put the web-maintenance group in doubt. In the previous section, details of such unusual server's behavior were shown in terms of request-based analysis. Here the similar analysis is presented in form of session-based analysis, which gives more insight to such server behaviors. The sessions behaving unusual are termed as weird sessions.

Weird session is interpreted as session which has unusual behavior in terms of significant percentage of erroneous responses in its duration. The heuristic criteria we used to extract weird sessions is the sessions having more than 50 requests and more than 50% of total requests resulting in erroneous status codes. We set this criterion for weird sessions considering the fact that there are few cases in which normal user surf the web site with significant number of request and experience lot of errors.

In some of web servers considered in this thesis, weird sessions were noticed. By looking into details of such sessions, it was observed that such behavior is not due to human users rather to some kind of scripts that run and result in many erroneous responses. The reasons for presence of such weird sessions can be:

- **Web robots:** A web robot is a program that traverses the web's hypertext structure by retrieving a document and recursively retrieving all documents that are referenced. These programs are sometimes called "spiders", "crawlers" or "worms". There are some advantages of web robots [21] which include their usage in search engines, maintenance of web sites etc. Also there are certain disadvantages like they consume web resources and bandwidth, overload servers, increases Internet traffic, etc.
- **Security breaches:** Existence of such sessions in the logs can also be result of some kind of unusual activity by users or by running some scripts to breach the security and get access of resources which are not authorized to those users.

- **Testing scripts:** Another possible reason for such sessions can be only the scripts run by the server administrator or authorized person to test the server or with some other motive which results in lots of errors.

This concept of weird session is introduced to help Web administrator to look into the suspicious sessions which may be due to web hacking, or other security related issues.

Results of this analysis are not presented considering the fact that their might be some kind of activity which web server administrator/authorized person wants to keep it undisclosed.

Chapter 5: Conclusions

In this thesis we have presented a detailed empirical analysis of request-based and session-based Web error characteristics on real data extracted from logs of eleven different Web servers. The results obtained from such analysis prove that a solid understanding of Web error behavior is fundamental to improve Web quality attributes such as reliability, performance, and security.

First, we analyzed the Web error characteristics in terms of unique errors. The analysis of unique errors proved that most of the errors encountered by the server reoccur at different times. The presented analysis of severity and frequency of occurrence of errors is extremely useful in deciding on the priority for fixing errors. The analysis of unique files with errors proved that fixing the errors associated with only a few files is the most cost effective way to improve the Web server quality, leading to a significant reduction of total number of errors.

Then, we analyzed the request-based reliability and the trend of total errors per day compared to the total requests per day. This analysis shows that the number of errors follows the same trend as requests in general, unless there is some unusual server's activity. The abrupt change in request-based reliability per day confirmed dissimilar patterns of errors to requests. This analysis is useful in finding any weird behavior of server for particular day. Then, we have introduced and empirically evaluated the session-based Web reliability and argued that it is better indicator of the user's perception of the Web quality than the request-based reliability.

The last contribution of this thesis is development of heuristic search criteria for finding sessions which indicate unusual server behavior, such as extremely long sessions, and sessions with unexpectedly large number of errors. This kind of search can be helpful to administrators for tracking attacks or other security issues

The future work with respect to this research should address the challenging problems of identifying unusual sessions and further differentiating between robot sessions, server attacks, and testing scripts. Another important aspect can focus on detailed analysis of different types of errors and more rigorous analysis of the

relationship between request-based and session-based reliability. Characterizing actual errors versus human errors e.g. is it a user typing mistake or a broken link? , is also one of the areas to be explored. Detailed statistical approach to model the error distribution to a particular probability distribution function can be done. The final goal of this research should concentrate on automating the process of error characterization and its real time implementation.

References

[1] V. S. Alagar and O. Ormandjieva, “Reliability assessment of web applications”, *Proc. 26th Annual International Computer Software and Applications Conference* (COMPSAC’02), 2002.

[2] J. Almeida, V. Almeida, and D. Yates, “Measuring the behavior of a World-Wide Web server”, *in Proc. 7th Conf. High Performance Networking (HPN),* White Plains, NY, Apr. 1997, pp. 57 72.

[3] M. Arlitt and T. Jin, “Workload characterization of the 1998 World Cup Web site”, *Hewlett-Packard Technical Report, HPL-1999-35(R.1),* Sep. 1999.

[4] M. Arlitt and C. Williamson, “Internet Web Servers: Workload characterization and performance implications”, *IEEE/ACM Transactions on Networking,* Vol.5, No.5, October 1997, pp. 631-645.

[5] M. Arlitt and C. Williamson, “Web server workload characterization: The search for invariants,” *in Proc. 1996 ACM SIGMETRICS Conf.*, Philadelphia, PA, May 1996, pp. 126–137.

[6] P. G. Backes, K. S. Tso, J. S. Norris, G. K. Tharp, J. T. Slostad, R. G. Bonitz, and K. S. Ali, "Internet-based operations for the Mars Polar Lander Mission," *in Proceedings of the IEEE International Conference on Robotics and Automation* Vol. 2, (San Francisco, CA), pp. 2025-2032, Apr. 2000.

[7] V. R. Basili., “The Role of Experimentation in Software Engineering: Past, Current, and Future”, *in 18th IEEE International Conference on Software Engineering (ICSE-18),* pages 442–449, May 1996.

[8] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Nielsen, and A. Secret,"The World-Wide Web", *Commun. ACM*, vol. 37, no. 8, pp. 76–82, Aug. 1993.

[9] H. Braun and K. Claffy, "Web traffic characterization: An assessment of the impact of caching documents from NCSA's Web server", *in Proc. 2nd World Wide Web Conf.'94: Mosaic and Web*, Chicago, IL, Oct. 1994.

[10] L. Cherkasova and P. Phaal, "Session Based Admission Control: A mechanism for improving the performance of an overloaded Web servers", *HP Labs Technical Reports*, HPL-98-119, 1998.

[11] P. Cremonesi and G. Serazzi, "End-to-End Performance of Web Services", *Performance 2002, M.C.Clzarossa, S.Tucci (Eds.), LNCS 2459*, Springer-Verlag, 2002, pp. 158-178.

[12] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", *IEEE/ACM Transactions on Networking*, Vol.5, No.6, Dec.1997, pp. 835-846.

[13] K. Goseva-Pospstojanova, and S. Kamavaran, (2003), "Assessing Uncertainty in Reliability of Component-Based Software Systems", *in 14th International Symposium on Software Reliability Engineering (ISSRE 2003)*, pages 307-320.

[14] K. Goseva-Popstojanova, S. Mazimdar, and A. Singh, "Empirical Study of Session-based Workload and Reliability for Web Servers", *in proceedings of the 15th IEEE International Symposium on Software Reliability Engineering (ISSRE 2004)*, Saint-Malo, France, November 2004, pp. 403-414.

[15] K. Goseva-Popstojanova, A. Singh, S. Mazimdar and Fengbin Li, "Empirical Characterization of Session-based Workload and Reliability for Web Servers" *Empirical Software Engineering Journal, Springer US, (This paper is an*

expanded version of the paper presented at 15th IEEE International Symposium on Software Reliability the which was selected among papers with strong archival value.), accepted for publication. Vol.11, No.1, Jan. 2006.

[16] C. Kallepalli and J. Tian, "Measuring and Modeling Usage and Reliability for StatisticalWeb Testing", *IEEE Transaction on Software Engineering*, Vol.27, No.11, Nov. 2001, pp. 1023-1036.

[17] D. A. Menasce, V. A. F. Almeida, R. Foneca, and M. A. Mendes, "Business–oriented Resource Management Policies for E-commerce Servers", *in performance evaluation*, Vol.42, No.2-3, 2000, pp. 223-239.

[18] D. Menasce, V. Almeida, and R. Ried, "In Search of Invariants for E-BusinessWorkloads", *in Proc. 2nd ACM Conference on Electronic Commerce (EC'00)*, Minneapolis, MI, Oct. 2000, pp. 56-65.

[19] E. Nelson, "Estimating Software Reliability from Test Data" *in Microelectronicsand Reliability*, Vol.17, No.1, 1978, pp. 67-73.

[20] M. Rosenstein, "What is Actually Taking Place in Web Sites: Ecommerce Lessons fromWeb Server Logs", *in Proc. 2nd ACM Conferenceon Electronic Commerce (EC'00)*, Minneapolis, MI, Oct. 2000, pp. 38-43.

[21] P. N. Tan, V. Kumar, Discovery ofWeb Robot Sessions Based on their Navigational Patterns, *Data Mining and Knowledge Discovery*, Vol 6, No. 1, Jan. 2002, pp. 9-35.

[22] A. Tanenbaum, Computer Networks, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[23] W. Wang and M. Tang, “User–Oriented Reliability Modeling for a Web System”, *Proc. 14th International Symposium on Software Reliability Engineering (ISSRE 2003)*, Denver, CO, Nov. 2003, pp. 293-304.

[24] N. Yeager and R. McGrath, “Web Server Technology: The Advanced Guide for World Wide Web Information Providers”, San Francisco, CA: Morgan Kaufmann, 1996.

[25] M. V. Zelkowitz and D. R.Wallace., “Experimental models for validating technology”, *IEEE Computer*, pages 23–31, May 1998.

[26] A Listing of Access Log Analyzers, http://www.uu.se/Software/Analyzers/Access-Analyzers.html

[27] Apache Logging Documentation: http://httpd.apache.org/docs/1.3/logs.html

[28] FastStats, http://www.mach5.com.

[29] HBX Analytics, http://www.websidestory.com/products/web-analytics/hbx-analytics.

[30] Internet Traffic Archive, http://ita.ee.lbl.gov/html/traces.html

[31] NetTracker, http://www.sane.com/demo/NetTracker/web/index.html.

[32] Sawmill, http://sawmill.net.

[33] Status Code Definitions part of Hypertext Transfer Protocol -- HTTP/1.1 RFC 2616 Fielding, et al.: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

[34] Webtrax, http://www.multicians.org/thvv/webtrax-help.html.

Appendix

Appendix A: *Tool documentation*

Approach to develop a prototype graphical user interface includes developing JAVA routines, Database design and Database side queries, on windows based system.

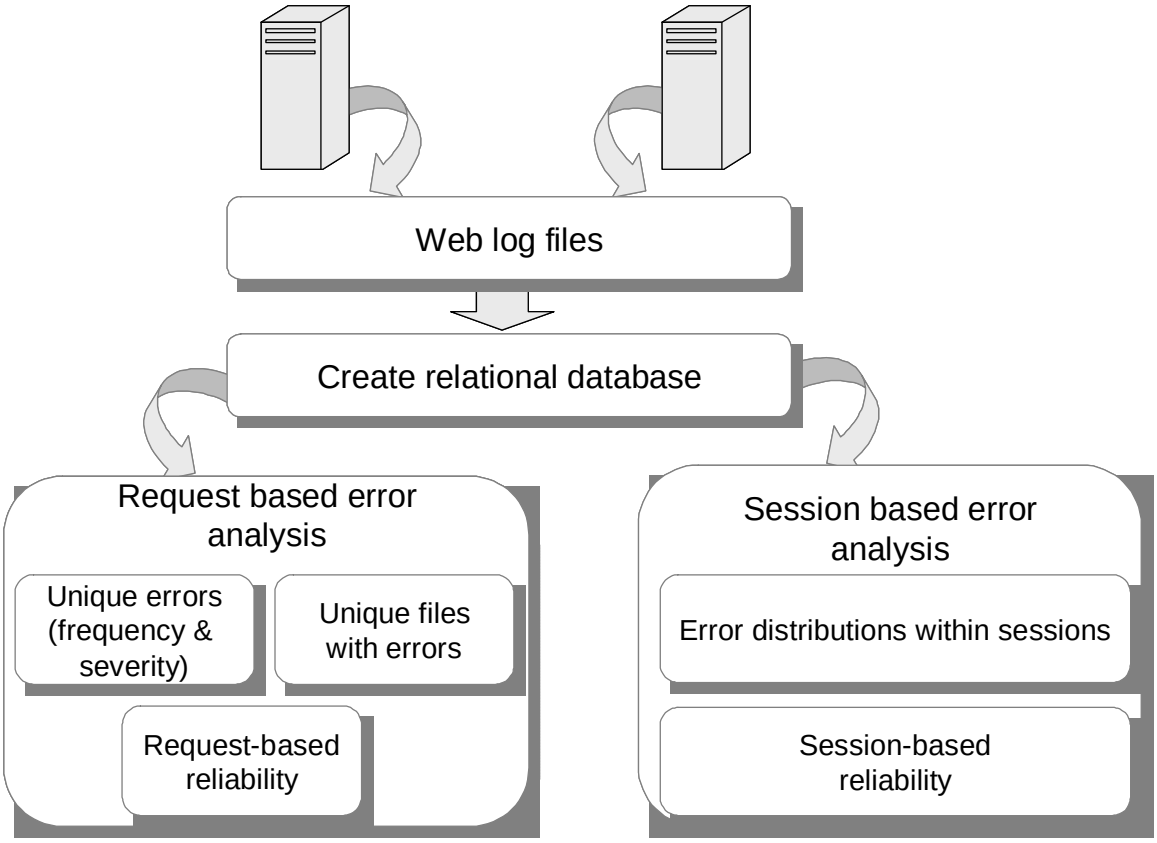


Figure A.1: Process Design

Figure 3.4 and Figure A.1 explains the general dataflow and process design respectively. The raw logs files (ASCII format) are passed to the java program, which parses them and inserts the relevant information into the oracle database. Different queries are then executed on the database to obtain valuable results. Few other java programs are used to obtain the results from the database and export the values in excel

sheets. The results include unique errors, unique files, frequencies, severity of errors, and unique error messages.

We used PL/SQL stored procedures to create sessions from the data collected in the database. Other stored procedures are used to obtain results which include workload intensity per day and number of errors per day, request-based reliability and session based reliability. All the results from stored procedures and java programs are combined to plot graphs in excel to make results viewable and easily understandable. We also used stored procedures to identify weird sessions i.e. sessions showing unusual server behavior.