Empir Software Eng (2009) 14:371–396
DOI 10.1007/s10664-008-9084-6

Another viewpoint on "evaluating web software reliability based on workload and failure data extracted from server logs"

Toan Huynh · James Miller

Published online: 1 August 2008

Editor: Laurie Williams

Abstract An approach of determining a website's reliability is evaluated in this paper. This technique extracts workload measures and error codes from the server's data logs. This information is then used to calculate the reliability for a particular website. This study follows on from a previous study, and hence, can be regarded as a "partial replication" (technically, as both studies are case studies not formal experiments, this description is inaccurate. Unfortunately, no corresponding definition exists for case studies, and hence the term is used to convey a general sense of purpose) of the original study. Although the method proposed by the original study is feasible, the effectiveness of just using a specific error type and a specific workload to estimate the reliability of websites is questionable. In this study, different error types and their usefulness for reliability analysis are examined and discussed. After a thorough investigation, we believe that reliability analysis for websites must be based on more specific error definitions as they can provide a superior reliability estimate for today's highly dynamic websites.

Keywords Partial replication · Empirical evaluation · Case study · www reliability · Defect classification

1 Introduction

Reliability is becoming increasingly important to web systems due to the popularity of web applications. The need for highly reliable systems will only grow as companies continue to move their operations online. In order to increase reliability, a method to measure current

T. Huynh · J. Miller (✉)
Electrical and Computer Engineering Research Facility, Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2 V4, Canada
e-mail: jm@ece.ualberta.ca

T. Huynh
e-mail: huynh@ece.ualberta.ca

systems' reliability is required. However, existing methods to measure reliability (Lyu 1995; Musa et al. 1987; Trivedi 2001) cannot be applied directly to web systems due to their specific nature (Alagar and Ormandjieva 2002; Offutt 2002). Thus, these existing methods will need to be modified to include new workload characteristics to estimate the reliability of web systems (Tian et al. 2004). More specifically, they defined two special characteristics:

- *Massiveness and diversity*: Web systems can interact with many different external systems. For example, one application may interact with Internet Explorer 6.5 and MySQL 3.23; another application may interact with Internet Explorer 5.5, Mozilla FireFox 1.5, SQLite 3.4.2 and Google Maps API 2.1. Not only that, every user with an Internet connection is considered to be a potential user of the web system. The workload characteristics selected need to reflect this diverse software configuration and massive and ill-defined user population.
- *Document and information focus*: Traditional workload concentrates on the computational focus whereas web systems principally have a document and information focus. Newer web systems have increased computation; however, search and retrieval remains the dominant usage for web users. The workload types for computational focus are fundamentally different than the workload types for document and information focus.

To measure web workloads to ensure accurate reliability estimation, generic workloads suitable for traditional computation-intensive cannot be used. Hence, Tian et al. (2004) defined four different web workload characteristics for reliability calculations:

- *The number of hits*: This workload is popular because each hit corresponds to a specific request to a web server, and each entry in the access log is a hit which allows for easy extraction of the data. However, this workload is misleading if it shows high variability with the individual hits (Tian et al. 2004).
- *The number of bytes* transferred may be used as a workload of finer granularity than the hit count; the number of bytes of transferred for each hit is recorded in the server logs and can be extracted with relative ease.
- *The number of users*. This alternative workload can be used by organizations that support various web systems and want to examine reliability at the user level. To count the number of users per day, the total number of unique IP addresses for that day is counted, and each unique IP address is assumed to correspond to a unique user. In other words, all hits originating from the same IP address (which may be associated with one computer or multiple computers sharing the same IP address) are considered to be requests from a single user. A disadvantage of the user workload is its coarse granularity. This problem can be remedied by counting the number of user sessions.
- *The number of sessions* can be calculated from the IP address and the access time. If the time between each hit from one IP is within a time period, then all of these hits are considered to be one session. The session workload is better than the user workload because each session is typically associated with a change in user activity or a change in user. The same user may have several different usage patterns for each session; this can be revealed by the session workload characteristic.

Given the issues related to these workload estimates, this study will also examine simply using "days" as a workload characteristic. A "day" is defined as a 24 h period within a log file. Clearly this alternative has a substantially coarser granularity than the alternatives discussed above. While the most obvious temptation is to utilize a fine-grain workload metric; since issues exists in their estimation, the question of are they actually a superior choice of normalizing term needs to be considered.

Although web traffic characteristics have been explored in detailed—such as the characterization of the workloads (Alagar and Ormandjieva 2002), traffic trends and patterns (Crovella and Bestavros 1997), response times (Cremonesi and Serazzi 2002), etc.—only a few studies have investigated web error behavior and the measurement of web reliability. Although several hypothetical approaches exist; they lack empirical validations (Alagar and Ormandjieva 2002; Wang and Tang 2003). One practical approach to measuring the reliability of web systems is to use the information contained in server logs (Huynh and Miller 2005; Kallepalli and Tian 2001; Tian et al. 2004), such as system usage and failure codes. This information can be extracted and used to evaluate the system's reliability and identify "areas" for reliability improvement.

In this paper, the approach of measuring reliability from server logs, as presented by Tian et al. (2004), will be evaluated and analyzed to determine the viability and effectiveness of this approach. Results from the original study and from our new study will be used in the analysis. Two websites were examined in the original study; and two additional websites will be investigated in this new study. Initially, these two websites are analyzed using the same methodology as proposed in the original study (Tian et al. 2004). That is, the server logs from these two websites were parsed for all errors that occurred while the websites were serving content to their visitors. A reliability estimate is then calculated from the extracted errors. This paper extends the original study (Tian et al. 2004) by:

- Applying the technique to two new websites. One of which is a commercial website; in fact the site can be considered as being mission critical to the commercial organization. The logs investigated for this commercial website cover a 15 month period, which is an extensive time period. It is believed that this log represents the longest period of capture, and the only truly "mission critical" log reported within the research literature.
- Examining the error codes more rigorously; this will allow web administrators to focus on high value error codes.
- Re-examining the workload models to provide alternative methods for web administrators to analyze and interpret reliability information.

The remaining sections of this paper are organized as follows: Section 2 describes the research methodology. Section 3 provides a brief overview of the characteristics of the websites used in the previous and the current study. Section 4 examines the workloads, the limitations of the workloads proposed, and the results from the two websites. Finally, Section 5 presents our conclusions.

2 Research Methodology

Tian et al. (2004) demonstrated by performing an experiment on two websites that the operational reliability of websites could be estimated from server logs. They identified three failure sources:

- Host, network, or browser failures that prevent the delivery of requested information to web users. These errors can be analyzed and assured by existing techniques (Lyu 1995; Musa et al. 1987; Trivedi 2001) because they are similar to failures in regular computer systems, network or software (Tian et al. 2004).

- Source content failures that prevent the acquisition of the requested information by web users because of problems such as missing or inaccessible files, trouble with starting JavaScript, etc. These failures have unique characteristics to web systems (Crovella and Bestavros 1997; Montgomery and Faloutsos 2001; Offutt 2002); hence, special workload characteristics need to be defined before their reliability can be estimated.
- User errors, such as improper usage, mistyped URLs, etc. These errors also include any external factors that are beyond the control of web service or content providers.

They noted that host, network, browser failures and user errors can either be addressed by existing approaches or are outside of the responsibility and control of the content provider. However, source content failures represent a significant part of the problem and the content providers can address these issues. Hence, Tian et al. (2004) focused on web source content failures contained in error and access log files in their study. These files are created by all commercial HTTP Daemons.

The Nelson model (Nelson 1978), a widely used input domain reliability model, was used by Tian et al. (2004) to calculate reliability after the necessary information was extracted from the server logs. The formula for the Nelson model is:

$$R = \frac{n-f}{n} = 1 - \frac{f}{n} = 1 - r \tag{1}$$

where f is the total number of failures, n is the number of workload units and r is the failure rate. The mean time between failures (MTBF) was then calculated as:

$$\mathrm{MTBF} = \frac{1}{f}\sum_i t_i \tag{2}$$

where t_i is the usage time for each workload unit i. If the usage time is not available, the number of workload units is then used as an approximation of the time period. Thus, the MTBF can be calculated as:

$$\mathrm{MTBF} = \frac{n}{f} \tag{3}$$

2.1 Removal of Automated Requests

The log files contain requests from robots and other automated systems that should be removed as they are not actual requests from web users. Automated systems are classified as systems that repeatedly request a resource from the website after a set period of time. For example, upon investigation of Site A's server log, requests from two monitoring services are identified. The first service requests a resource from Site A every 30 min while the second service requests a resource from Site A every 66 min. The resources these services request are unique and not publicly available; hence removing them simply involves identifying these resources in the log files. Robots that automatically request the "robots.txt" resource are also removed from both Site A and ECE log files.

Although, it is infeasible to remove all automated requests from the server logs, web administrators need to remove all identifiable requests. Several techniques to identify them can be used by web administrators to remove automated requests. Most well known robots have a signature line that is included with every request as part of the USER AGENT field of the log file. For example, "Googlebot-Image/1.0" can be used to identify a robot from Google that is indexing the website's images. For web monitoring services, web administrators can simply dedicate a special resource that only these services can access. This resource can then be easily identified within the log files.

2.2 Analysis of Error Code Information

Error response codes can be extracted from either access or error logs. Due to the lack of error log files for the K Desktop Environment (KDE) website and Site A, only the access log files were used to extract the error information (Tian et al. 2004). Error response codes are embedded in the access logs, and these codes can be mapped to the error entries in the error log, for example, a "file not found" error in the error log usually corresponds to a 404 error code in the access log. Hence as stated in Tian et al. (2004), using just the access logs is a reasonable method to gather error information unless detailed information about the errors is required. Figure 1 provides a sample entry that can be found within the access logs.

This figure shows that on November 3, 2005, a remote user with an IP address of 129.194.12.3 used the POST protocol to access a file called search.php. The server responded with a 200 code and returned 50482 bytes of data. The previous URL that the user visited is http://www.sitea.com/database/form.php. The user used Microsoft Internet Explore version 6.0 to access the webpage.

The Nelson model and MTBF calculation require that the server logs capture the entire workload for the period under investigation. To ensure that the logs are complete, the parser used was customized to report suspicious gaps, which can be defined as long periods of inactivity between two recorded hits. These gaps were manually examined and discussed with the web administrators to ensure that the gaps are naturally occurring and not due to external factors such as the hard drive being full.

The error response codes in Tables 3, 4 and 5 are the standard HTTP error response codes as defined by the Request For Comment 2616 (http://www.w3.org/Protocols/rfc2616/rfc2616.html) as part of the HTTP protocol. The following is a list of the codes encountered, their descriptions, and what the implications are when they are used for reliability analysis:

- 400 (Bad request)—the request could not be understood by the server due to its malformed syntax. This code should not be used for reliability analysis because the code is caused by a client that is not following the HTTP standard. Since this is a client-side issue, it does not make sense to estimate a website's reliability based on this code.
- 401 (Unauthorized)—the server does not accept the client's authorization credentials (or they were not supplied). This error occurs when a user requests a resource that the user does not have permission to retrieve. If the referrer for this resource is external to the website then this error can be ignored because the web administrators cannot control these external referrers. However, if the referrer is internal to the website and it is not the expected behavior of the server, then this error needs to be included in the reliability analysis. This situation of an error response code encompassing error types which are source content failure and external sources (human and system errors) occurs repeatedly; hence, the situation needs to be resolved to provide accurate reliability information. This issue is resolved later in the paper.

```
129.194.12.3 - - [03/Nov/2005:15:44:34 -0500] "POST /data/search.php HTTP/1.0" 200
50482 "http://www.sitea.com/data/form.php " "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1; SV1; .NET CLR 1.1.4322)"
```

Fig. 1 A sample entry in an access log

- 403 (Forbidden)—the server is refusing to fulfill the client's request. The cause for this error is similar to the 401 error code. Depending on the configuration of the HTTP daemon, this error may be returned instead of the 401 error code. Hence, it has the same issue as the 401 error response code, and will be discussed later.
- 404 (Not found)—the server cannot find anything matching the Request-URI. This error is currently the dominating error code and represented the focus of result of Tian et al.'s paper (2004). However, again, this error response code covers a multitude of different error types some of which are source content failure but others lie outside the system or what seem to be source content failures are actually not source content failures. For example, an attacker utilizing a scanner can (Spitzner 2001) spoof the referrer field of the log file when scanning for a system's vulnerability; the spoofed referrer field appears to be an internal link when it is actually from an external source. Links to old versions of the website can also create 404 error codes that appear to be internal bad links because the old version of the website is hosted on the same server as the current website. However, these internal bad links should be discarded because the user is using an incorrect version of the website. With the availability of powerful link checkers (NetMechanic HTML Toolbox[1], W3C Link Checker[2]), it is highly likely that actual source content failures are on the decline.
- 405 (Method not allowed)—the method specified in the Request-Line is not allowed for the resource identified by the Request-URI. The client performs a request that is not allowed by the server. For example, the client tries to perform a PUT request, but the server is configured to not accept PUT requests; hence, a 405 error code is generated. Since this error code only occurs due to a configuration issue, it should be discarded.
- 406 (Not acceptable)—this error is returned if the web server detects that the client cannot accept the data it wants to return. This error code should be discarded because the server's content does not support the client used to access it.
- 407 (Proxy authentication required)—if the client does not authenticate itself with the proxy then this error is returned. This error code can be discarded because the client did not authenticate with the server before attempting to access restricted content.
- 408 (Request timeout)—the client did not produce a request within the time that the server was prepared to wait. This is a network failure rather than a source content failure, and hence, it should be discarded.
- 409 (Conflict)—the client is attempting to perform a request that conflicts with the server's established rule. For example, the client is attempting to upload a file that is older than the file currently available on the server, this results in a version control conflict. This error can be discarded because it is a browser failure, not a server failure.
- 410 (Gone)—the server cannot find the requested resource and no alternative location can be found. This error code is related to the 404 response code, and hence it should follow the same rules as the 404 response code.
- 411 (Length required)—the server is denying the data the client is uploading because the client is not specifying the size of the data. Because this error is a browser failure and not a server failure, it can be discarded.
- 412 (Precondition failed)—the resource requested failed to match the established preconditions. This error should be included because the server failed to satisfy the preconditions; this implies that this error response code is a server failure.

[1] http://www.netmechanic.com/products/maintain.shtml

[2] http://validator.w3.org/checklink

- 413 (Request entity too large)—the server is rejecting the data being uploaded from the client because the data size is too large. The size limit can be adjusted within the server configuration. Since this error code only occurs due to a configuration issue, it should be discarded.
- 414 (Request-URI Too Long)—the server returns this error code in the following situations:

– The client (usually a browser) has converted values from a POST request to a GET request. The POST request can handle larger values than the GET request; thus, the error occurs when an extremely large POST request is converted to a GET request.

– The client is attempting to exploit some type of vulnerability in the server. Usually, these exploits involve a large amount of malicious code being injected into the Request-URI. Some of these vulnerabilities include: buffer overflows (Cowan et al. 1998; Evans and Larochelle 2002; Wagner et al. 2000), SQL injections (Boyd and Keromytis 2004; Grossman 2004; Huang et al. 2003), cross-site scripting (CGISecurity 2002; Cook 2003), etc.

Generally, the first situation is rare, and hence it is usually safe to assume that a majority of 414 errors will correspond to attacks on the server or other users who are accessing the vulnerable website. Thus, by identifying these 414 errors, system administrators can identify attacks on their server system and take appropriate actions against the attackers. Although the 414 error code is useful to system administrators, it is not a source content failure and, hence, will be excluded from reliability analysis.

- 415 (Unsupported media type)—the server is refusing the request because the resource is in a different format from the requested format. For example, the browser requests a resource and specifies it as a text document; however, the server recognizes the requested resource as a binary file and not a text document. A 415 response code would be generated in this scenario. Since this error code is a browser failure and not a source content failure, it should be discarded.
- 416 (Requested range not satisfiable)—the client is requesting a file size's range that is invalid. This error occurs when the client, usually a download manager such as Getright (http://www.getright.com) or Wget (http://www.gnu.org/software/wget/wget.html), erred in its resume point calculation. Hence, this error code should not be used in reliability analysis.
- 500 (Internal error)—the server encountered an unexpected condition which prevented it from fulfilling the request. Bugs within various dynamic scripts running on the server cause this error code. Therefore, it must be included in any reliability calculation.
- 501 (Not implemented)—the server does not support the request type that the client is sending. For example, the browser tries to retrieve the header information of an ASP enabled web page, so it sends a HEAD request to the server. However, the server does not understand this request for ASP enabled web pages, so it returns 501 error response code. This error code should be included in reliability analysis.
- 502 (Bad gateway)—this error has two definitions depending on the HTTP daemon used. For Apache, this error occurs when the server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request. Because this error response code only occurs when the Apache HTTP Daemon is acting in a different mode rather than actively serving web pages, this error should be discarded for servers using the Apache HTTP daemon. For IIS, Microsoft

IIS' support page (http://support.microsoft.com/default.aspx?scid=kb;en-us;318380) describes this error as "You receive this error message when you try to run a CGI script that does not return a valid set of HTTP headers." In other words, this error code can be triggered by an error in the web application's output code. Thus, this error should be included in reliability analysis if the web software is running on the IIS platform.

- 503 (Service unavailable)—the server is overloaded and cannot serve further requests. For example, due to a popular marketing campaign for a website, many users decide to visit this site. The unexpected load caused by this sudden increase in traffic causes a major strain in the server's resources, which then leads to extremely slow response time or a server crash. For example, Toys R Us' website received a surge in traffic after it released its Big Book catalog. This surge in traffic overloaded the system's resources, which lead to an extremely slow response time. Numerous potential purchasers were turned away because of this slow response time (Masterson 1999).
- This failure response code is a host failure that can lead to extended availability issue if not resolved properly. Tian et al. (2004) stated that availability problems are generally perceived by web users as less serious than web software problems. They argued that users are more likely to be successful accessing required information after temporary unavailability whereas software problems would persist unless the underlying causes are identified and fixed. We believe this argument is questionable because web users are much more impatient and less forgiving than traditional users, as discussed by many studies (Galletta et al. 2004; Grant 2000; Masterson 1999; Nah 2002; Rose et al. 2001; Williams 2001). They typically move on to the next site if they encounter issues with the current site that they are browsing. From their perspective, if they cannot access the information they want then it is an error. Hence, although the 503 error response code corresponds to a host failure and not a source content failure, it must be included in reliability analysis.
- 504 (Gateway timeout)—this error only occurs when the server is acting as a gateway or proxy server, hence it should be discarded.
- 505 (HTTP version not supported)—the server does not support the HTTP protocol version used by the client. This error can be discarded because the client is not using the proper HTTP protocol version.

It should be noted that web systems can be configured to catch error codes and respond with a 200 OK code instead. While this strategy hides technical information from users, it does not allow the error codes to be logged properly if configured incorrectly. Hence, web administrators should ensure that error codes are still logged if this strategy is to be used.

3 Overview of the Websites

Tian et al. (2004) applied the proposed approach to two websites. The first website analyzed was www.seas.smu.edu, the official web site for the School of Engineering and Applied Science at Southern Methodist University (SMU/SEAS). The log files contained data covering 26 consecutive days in 1999. The second website analyzed was www.kde.org (KDE). This is the official website for the KDE project. The overall traffic and user population for this website is significantly larger than the SMU/SEAS website. The logs contained 31 days of traffic data. During these 31 days, over 13 million hits were recorded.

Both of these websites used the popular Apache HTTP Daemon (http://httpd.apache.org) to serve their web pages.

3.1 Overview of the Websites in This Study

This paper re-analyzes the approach presented in the original study (Tian et al. 2004). It initially applies this approach to two new websites, and based on these results postulates an alternative approach. The first website is www.ece.ualberta.ca, the website for the Department of Electrical and Computer Engineering at the University of Alberta. This site—similar to SMU/SEAS and KDE—although important to the organization, it is non-commercial and not mission critical. This website is a dynamic website that utilizes the ColdFusion (http://www.macromedia.com/software/coldfusion) scripting language, and the Apache HTTP Daemon (http://httpd.apache.org). To investigate the stability of the data, the log files were chosen to cover approximately 30 consecutive days in January 2005 (ECE1) and 30 consecutive days in March 2006 (ECE2). For the month of January, the ECE website received approximately 500,000 hits, 53,100 "unique" visitors and transferred a total amount of 4.8 Gbytes of data. During March 2006, the ECE website handled 470,000 hits, 61,000 "unique" visitors and transferred a total amount of 6.2 Gbytes of data.

The second website is the website for a publishing company that specializes in online databases (Site A). This website differs from the previous websites in that it is very critical to Company A's operation and hence it needs to be extremely reliable. The website utilizes the PHP (http://www.php.net) scripting language, MySQL (http://www.mysql.com) for the backend database and is hosted on an Apache HTTP Daemon. In order to observe potential trends and patterns for this mission critical website, the log files chosen cover 15 months of operation from January 2005 to March 2006. This website's traffic is lower than the ECE website. However, it represents a typical business website. That is, the site is a dynamic website with a mixed amount of static and dynamic pages—these are pages generated dynamically depending on the customers' inputs; its users are customers who are either looking to purchase a product or to register for a training course. For the 15 months covered, Site A received approximately 1.9 million hits and 92,000 "unique" visitors. The site transferred 34 Gbytes of data. Table 1 displays the technologies used by, and reliability requirements for, the two websites under investigation. Unfortunately, the ECE site administrator only has an approximate reliability target for their installation. These two websites were selected for this investigation because they utilize similar web development technologies while having different reliability requirements. The two websites use a scripting language in addition to an HTTP daemon; with one of the sites (A) also using a DBMS for data management. Although the technologies used are similar, their reliability objectives are quite different. ECE—due to its non-mission critical nature—is expected to experience between one to ten failures per month. Site A requires high reliability because the loss of customers and sales will occur if the site's failure occurs. In other words, Site A is expected to experience no more than one failure per month. Note: the two sites are not related in any way, nor have any personnel in common.

Table 1 Sites examined

Site	Technologies	Reliability requirement
ECE	CodeFusion, Apache	A few failures per month
Site A	PHP, Apache, MySQL	No more than 1 failure per month

Table 2 provides a summary of the properties of the logs used in previous studies and this study. Websites with an asterisk are commercial websites.

This table shows that the longest period that previous studies have collected data is over a 7 month period, compared to 15 months in this study. Furthermore, studies that use logs from commercial websites cover extremely short periods (1 to 2 weeks). This study investigates the log file from a commercial website for a much longer period (15 months). Hence, it is believed that this study presents the first long-term analysis of a (mission-critical) commercial website.

4 Results and Discussions

This section presents the results for the four websites, and discusses various issues encountered during this experiment and explains the similarity and differences between the original study and this study.

4.1 Results from the Original Study

Tian et al. (2004) discovered many issues associated with the extraction of workload data for reliability estimation. However, the log files provide information that allows available data for the hit count, byte count and user count to be extracted with ease. The session count can be derived based on a timeout value which can provide more granularity than the user count.

They found that the four proposed workload characteristics allow reliability assessments from different perspectives. Hence, systems administrators can choose the best workload characteristic depending on the situation. For example, administrators concerned with data traffic measurement can examine the byte count whereas the hit count can provide more

Table 2 Comparison of data sets

Reference		Log duration	Requests	Bytes transferred (GB)
Goseva-Popstojanova et al. (2006a)	NASA-Pvt1	20 week	23,000	0.5
	NASA-Pvt2	20 week	92,000	0.2
	NASA-Pvt3	20 week	489,000	2.2
	NASA-Pub1	20 week	93,000	9
	NASA-Pub2	20 week	732,000	6.7
	NASA-Pub3	20 week	108,000	4.6
	CSEE	6 week	5.8 million	80.9
	WVU	3 week	37.9 million	97
	ClarkNet*	2 week	3.3 million	27.6
	NASA-KSC	2 month	3.5 million	62.5
	Saskatchewan	7 month	2.4 million	12.3
Goseva-Popstojanova et al. (2006b)	WVU	1 week	15.8 million	34.5
	ClarkNet*	1 week	1.7 million	13.8
	CSEE	1 week	397,000	10.1
	NASA-Pub2	1 week	39,000	0.3
Tian et al. (2004)	SMU/SEAS	26 day	763,000	7.8
	KDE	31 day	14 million	110
This study	Site A*	15 month	1.9 million	34
	ECE1	1 month	500,000	4.8
	ECE2	1 month	470,000	6.2

useful information regarding web users. The next section will present results found in this study and whether they confirm findings from Tian et al. (2004) study.

4.2 Results from This Study

Tables 3, 4 and 5 provide a summary of the error response codes for all four websites. These tables contain the actual number of error counts and their corresponding percentages; these tables follow the analysis performed by Tian et al. (2004). That is, the access logs are parsed, and the errors are grouped together according to the error code without explicit considering of their cause. The original study provided only limited information for the KDE website; hence all the cells containing "n/a" are missing information that cannot be derived. Furthermore, the total percentage of errors recorded does not equal to 100 percent for this website. While Goseva-Popstojanova et al. (2006a, b) also performed analysis on the error codes, the results are combined into groups such as 4xx (all 400 level error codes) and 5xx (all 500 level error codes). Hence, results from Goseva-Popstojanova et al. (2006a, b) cannot be included in these tables.

These tables show that the 404 error type dominates, as noted by Tian et al. (2004). They discovered that, for SMU/SEAS, 99.9% of the errors encountered were of types 403 and 404; with 404 errors accounting for 93.1% of the recorded errors. For KDE, 98.9% of the recorded errors were of type 404. According to the survey results from 1994 to 1998 by the Graphics, Visualization, and Usability Center of Georgia Institute of Technology (http://www.gvu.gatech.edu/user_surveys/), 404 errors are the most common errors that users encounter while browsing the web. Ma and Tian (2003) found that a majority of these 404 errors are caused by internal bad links while only a small percentage are caused by external

Table 3 Recorded errors

Sites	Error code			
	400	401	403	404
SMU/SEAS	2 (0.02%)	14 (0.046%)	2,085 (6.78%)	28,659 (93.17%)
KDE	n/a	n/a	n/a	785,211 (98.90%)
ECE1	202 (0.15%)	6 (0.00%)	44 (0.03%)	136,143 (99.81%)
ECE2	52 (0.05%)	4 (0.00%)	211 (0.19%)	112,751 (99.74%)
Site A (Jan05)	1 (0.06%)	3 (0.17%)	188 (10.90%))	1,500 (86.96%)
Site A (Feb05)	0	10 (0.53%)	162 (8.50%)	1,722 (90.44%)
Site A (Mar05)	1 (0.05%)	28 (1.29%)	194 (8.90%)	1,938 (88.94%)
Site A (Apr05)	2 (0.09%)	17 (0.72%)	190 (8.07%)	2,121 (90.06%)
Site A (May05)	4 (0.20%)	27 (1.33%)	130 (6.39%)	1,849 (90.86%)
Site A (Jun05)	1 (0.05%)	36 (1.65%)	213 (9.78%)	1,920 (88.11%)
Site A (Jul05)	0	36 (1.53%)	146 (6.19%)	2,158 (91.44%)
Site A (Aug05)	0	28 (1.04%)	194 (7.20%)	2,448 (90.87%)
Site A (Sep05)	0	13 (0.59%)	167 (7.54%)	2,018 (91.15%)
Site A (Oct05)	0	12 (0.46%)	159 (6.03%)	2,434 (92.30%)
Site A (Nov05)	0	19 (0.68%)	214 (7.69%)	2,525 (90.76%)
Site A (Dec05)	1 (0.04%)	13 (0.54%)	156 (6.43%)	2,223 (91.56%)
Site A (Jan06)	0	19 (0.58%)	231 (7.04%)	2,758 (84.11%)
Site A (Feb06)	0	19 (6.66%)	164 (5.66%)	2,602 (89.82%)
Site A (Mar06)	0	22 (0.61%)	259 (7.12%)	3,321 (91.31%)
Site A (Total)	10 (0.03%)	302 (0.81%)	2767 (7.40%)	33,537 (89.69%)

Table 4 Recorded errors (cont)

Sites	Error code				
	405	408	414	415	416
SMU/SEAS	0	0	0	0	0
KDE	n/a	6,225 (0.78%)	n/a	n/a	n/a
ECE1	0	0	0	0	6 (0.00%)
ECE2	2 (0.00%)	1 (0.00%)	0	0	14 (0.01%)
Site A (Jan05)	1 (0.06%)	0	0	30 (1.74%)	2 (0.12%)
Site A (Feb05)	0	0	0	10 (0.53%)	0
Site A (Mar05)	0	0	0	17 (0.78%)	1 (0.05%)
Site A (Apr05)	0	0	0	25 (1.06%)	0
Site A (May05)	2 (0.10%)	0	0	17 (0.84%)	0
Site A (Jun05)	0	0	0	9 (0.41%)	0
Site A (Jul05)	0	0	0	20 (0.85%)	0
Site A (Aug05)	0	0	0	24 (0.89%)	0
Site A (Sep05)	0	0	0	16 (0.72%)	0
Site A (Oct05)	0	0	0	32 (1.21%)	0
Site A (Nov05)	0	0	0	24 (0.86%)	0
Site A (Dec05)	98 (4.04%)	0	0	26 (1.07%)	0
Site A (Jan06)	254 (7.75%)	0	0	17 (0.52%)	0
Site A (Feb06)	83 (2.87%)	0	0	29 (1.00%)	0
Site A (Mar06)	5 (0.14%)	0	0	30 (0.83%)	0
Site A (Total)	443 (1.19%)	0	0	326 (0.87%)	0

Table 5 Recorded errors (cont)

Sites	Error code			
	500	501	502	503
SMU/SEAS	0	0	0	0
KDE	n/a	n/a	n/a	n/a
ECE1	7 (0.01%)	0	0	0
ECE2	10 (0.01%)	0	0	0
Site A (Jan05)	0	0	0	0
Site A (Feb05)	0	0	0	0
Site A (Mar05)	0	0	0	0
Site A (Apr05)	0	0	0	0
Site A (May05)	0	0	0	6 (0.30%)
Site A (Jun05)	0	0	0	0
Site A (Jul05)	0	0	0	0
Site A (Aug05)	0	0	0	0
Site A (Sep05)	0	0	0	0
Site A (Oct05)	0	0	0	0
Site A (Nov05)	0	0	0	0
Site A (Dec05)	0	0	0	0
Site A (Jan06)	0	0	0	0
Site A (Feb06)	0	0	0	0
Site A (Mar06)	0	0	0	0
Site A (Total)	0	0	0	6 (0.02%)

factors such as the user mistyping the URL, robots from search engines, external links (links from other websites), old bookmarks, etc.. Tian et al. (2004) discovered that only 8.7% of the 404 errors encountered were caused by external factors for SMU/SEAS. Despite this conclusion, they did not provide convincing evidence that the majority of the recorded errors are in fact from source content failures. Furthermore, these tables shows that, although the 404 error type dominates, other error response codes also exist; and while the 404 error type may dominate numerically; no analysis exists as to the "value" (of the information) encoded within the various error types for web site administrators. Therefore, we will examine all of the error codes encountered to determine which errors are truly source content failures (have value) and which are attributed to other uncontrollable factors (no value). For example, we will show that the 404 response errors have no value for Site A because all of the 404 recorded errors are caused by factors outside of the site administrator's control; whereas the 503 error response code is high in value because the site administrator is expected to respond and correct the 503 errors immediately due to the potential loss in sales and customers that this error code can cause.

One common argument is that if information is available, external failures can also be resolved. This argument is not valid for several reasons. A site administrator can only be reactive to external failures rather than being proactive. That is, until an external failure occurs, a site administrator will not have enough information to resolve that failure. Furthermore, depending on circumstances, the failure may not be resolvable. For example, an external website has a link to a web page on the web system under examination. However, due to recent changes, that web page is no longer valid. The site administrator will not be aware of this issue until a user follows the link from the external website. Once the failure occurs, the site administrator can attempt to resolve it by attempting to contact the external website's Webmaster to get the link updated. However, this process requires cooperation from the external website's Webmaster. Furthermore, the process becomes tedious when there are thousands of websites linking to this invalid web page. The site administrator can also attempt to redirect the user to the correct page. However, this requires the site administrator to have a complete mapping of all invalid pages to valid pages which is clearly infeasible. Because of these potential issues, the site administrator cannot resolve external failures adequately.

Based on the information above, the error response codes can be associated to one or more failure sources. Table 6 displays this association for the error codes discussed. Error codes that do not have an association with a source content failure or host failure will not be investigated because they are beyond content providers', or website administrators', control.

Table 6 shows seven error codes, 401, 403, 404, 500, 501, 502 (IIS), and 503 that have either source content failure (SCF) or host failure as a potential failure source; hence, these seven error codes will be examined in detailed in order to determine their exact failure sources. Further, the 401, 403 and 404 error codes have both source content failure and external failures as failure modes or sources. After intensively investigating the log files for the two web sites under study (Site A and ECE), we discovered that, for these web sites, the source content failures can be classified into two types:

- SCF1—these are errors on the website that should be identified and corrected by the site administrators or content providers. These errors can be identified by close examination of the referrer field:

If the referrer field of an error contains the website's URL, then the error belongs to the SCF1 category.

Table 6 Failure sources for the error codes

Error code	Host	Source content	Network or browser	User and external
400				√
401		√		√
403		√		√
404		√		√
405				√
408			√	
415			√	
416			√	√
500		√		
501		√		
502		√ (IIS)	√ (Apache)	
503	√			

- SCF2—these are usually links from external websites pointing to an old version of the website under investigation. This old version still exists on the HTTP Daemon for archival purposes and has no connections to the current website. Hence, it is not maintained and can contain many bad links. When a user visits this old version—through search engines, old bookmarks, old emails, etc.—and clicks on one of these bad links, the log data will record that the error is caused by an internal source. Since, these errors are under the direct control of system administrators, we classify them as source content rather than external failures. However, an argument can be made that they are of lower value than SCF1 type errors. For example, for the ECE site, these errors are considered by the site administrator as a "non-issue"; and a case can be made for either including them or excluding them from reliability calculations. Errors belonging to the SCF2 type can be identified using the following method:

For each error, the referrer URL should be noted and visited. If the URL leads to an old version of the website, then the error is of SCF2 type.

External failure sources—which account for the majority of the failures—can also be classified into two categories:

- ES1—which are old links from external websites, search engines, old bookmarks, etc. These external links can be detected based on the referrer information—each entry in the log files contains a referrer field which provides the web page that links to the content the user is requesting:

All 401, 403 and 404 errors having URLs—not from the same domain as the website—or the character "-" in the referrer field are of the ES1 type.

- ES2—which are scanners being executed by attackers looking for known vulnerabilities contained in various web applications. These scanners can send spoofed information to the web server. The web server will generate internal 401 or 403 errors if the web administrators have set up security permissions for these applications, or internal 404 errors if the website does not use these web applications. ES2 errors can be identified by close examination of the errors:

If the requested resources belong to web applications not installed for the website, then the errors are of ES2 type.

Errors 401, 403 and 404 belonging to the ES1 and ES2 types should be detected and discarded during the data analysis phase. Tables 7 and 8 display the percentages of the different failure categories for the 401, 403 and 404 error codes, respectively. Due to unavailable information, the errors from the original study cannot be classified into the types discussed. These tables show that ECE (1 and 2) and Site A have extremely low (less than 0.5%) or no 401, 403, and 404 error codes as source content failures. All 500, 501, and 502 error codes were discovered to be source content failures, which is expected because of the associations shown in Table 6.

Finally, Tables 9 and 10 display the error codes generated from source content and host failures that will be used for reliability analysis in this study. This table contains the 500, 501, 502, and 503 error codes in addition to a subset of the error response codes from Tables 7 and 8. The 401 error code is not included in this table because they do not contain any source content failures as shown in Table 7. Tables 9 and 10 effectively demonstrate the low number of "errors" of interest, or value, experienced by live web sites (ECE and Site A). These numbers have significant implications of reliability analysis and models for these types of systems.

This section discussed various different error codes and how they may or may not contribute to reliability analysis. Care has to be taken when dealing with these error codes as they do contain limitations that may affect the accuracy of a reliability estimate. The next section will discuss the workloads and any limitations they may have and how those limitations can further impact reliability analysis.

Table 7 Possible error codes for reliability analysis

Sites	Error code							
	401				403			
	SCF1	SCF2	ES1	ES2	SCF1	SCF2	ES1	ES2
ECE1	0	0	6 (100%)	0	0	0	38 (86.36%)	6 (13.64%)
ECE2	0	0	4 (100%)	0	0	1 (0.47%)	164 (77.73%)	46 (21.80%)
Site A (Jan05)	0	0	3 (100%)	0	0	0	186 (98.94%)	2 (1.06%)
Site A (Feb05)	0	0	4 (40.00%)	6 (60.00%)	0	0	158 (97.53%)	4 (2.47%)
Site A (Mar05)	0	0	28 (100%)	0	0	0	193 (99.48%)	1 (0.52%)
Site A (Apr05)	0	0	17 (100%)	0	0	0	189 (99.47%)	1 (0.53%)
Site A (May05)	0	0	27 (100%)	0	0	0	130 (100%)	0
Site A (Jun05)	0	0	36 (100%)	0	0	0	213 (100%)	0
Site A (Jul05)	0	0	33 (91.67%)	3 (8.33%)	0	0	146 (100%)	0
Site A (Aug05)	0	0	25 (89.29%)	3 (10.71%)	0	0	193 (99.48%)	1 (0.52%)
Site A (Sep05)	0	0	13 (100%)	0	0	0	167 (100%)	0
Site A (Oct05)	0	0	12 (100%)	0	0	0	159 (100%)	0
Site A (Nov05)	0	0	19 (100%)	0	0	0	214 (100%)	0
Site A (Dec05)	0	0	13 (100%)	0	0	0	153 (98.08%)	3 (1.92%)
Site A (Jan06)	0	0	19 (100%)	0	0	0	230 (99.57%)	1 (0.43%)
Site A (Feb06)	0	0	19 (100%)	0	0	0	163 (99.39%)	1 (0.61%)
Site A (Mar06)	0	0	22 (100%)	0	0	0	239 (92.28%)	20 (7.72%)
Site A (Total)	0	0	290 (96.03%)	12 (3.97%)	0	0	2733 (98.77%)	34 (1.23%)

Table 8 Possible error codes for reliability analysis (cont)

Sites	404 error code			
	SCF1	SCF2	ES1	ES2
ECE1	0	16 (0.01%)	135,950 (99.86%)	177 (0.13)
ECE2	0	10 (0.01%)	112,643 (99.90%)	98 (0.09%)
Site A (Jan05)	0	0	1,479 (98.60%)	21 (1.40%)
Site A (Feb05)	0	0	1,683 (97.74%)	39 (2.26%)
Site A (Mar05)	0	0	1,881 (97.06%)	39 (2.94%)
Site A (Apr05)	0	0	2,075 (97.83%)	46 (2.17%)
Site A (May05)	0	0	1,814 (98.11%)	35 (1.89%)
Site A (Jun05)	0	0	1,877 (97.76%)	43 (2.24%)
Site A (Jul05)	0	0	2,087 (96.71%)	71 (3.29%)
Site A (Aug05)	0	0	2,377 (97.10%)	71 (2.90%)
Site A (Sep05)	0	0	1,986 (98.41%)	32 (1.59%)
Site A (Oct05)	0	0	2,391 (98.23%)	43 (1.77%)
Site A (Nov05)	0	0	2,477 (98.10%)	48 (1.90%)
Site A (Dec05)	0	0	2,139 (96.22%)	84 (3.78%)
Site A (Jan06)	0	0	2,686 (97.39%)	72 (2.61%)
Site A (Feb06)	0	0	2,344 (90.08%)	258 (9.92)
Site A (Mar06)	0	0	2,983 (89.82%)	338 (10.18%)
Site A (Total)	0	0	32,279 (96.25%)	1,258 (3.75%)

Table 9 Error codes to be used for reliability analysis

Sites	Error codes		
	403	404	500
ECE1	0	16 (69.565%)	7 (30.435%)
ECE2	1 (4.762%)	10 (47.619%)	10 (47.619%)
Site A (Jan05)	0	0	0
Site A (Feb05)	0	0	0
Site A (Mar05)	0	0	0
Site A (Apr05)	0	0	0
Site A (May05)	0	0	0
Site A (Jun05)	0	0	0
Site A (Jul05)	0	0	0
Site A (Aug05)	0	0	0
Site A (Sep05)	0	0	0
Site A (Oct05)	0	0	0
Site A (Nov05)	0	0	0
Site A (Dec05)	0	0	0
Site A (Jan06)	0	0	0
Site A (Feb06)	0	0	0
Site A (Mar06)	0	0	0
Site A (Total)	0	0	0

Table 10 Error codes to be used for reliability analysis (cont)

Sites	Error codes		
	501	502	503
ECE1	0	0	0
ECE2	0	0	0
Site A (Jan05)	0	0	0
Site A (Feb05)	0	0	0
Site A (Mar05)	0	0	0
Site A (Apr05)	0	0	0
Site A (May05)	0	0	6 (100%)
Site A (Jun05)	0	0	0
Site A (Jul05)	0	0	0
Site A (Aug05)	0	0	0
Site A (Sep05)	0	0	0
Site A (Oct05)	0	0	0
Site A (Nov05)	0	0	0
Site A (Dec05)	0	0	0
Site A (Jan06)	0	0	0
Site A (Feb06)	0	0	0
Site A (Mar06)	0	0	0
Site A (Total)	0	0	6 (100%)

4.3 Workload Analysis and Discussions

Table 11 contains the workloads for the four workloads explored by Tian et al. (2004). Session count uses the standard 2 h of inactivity to mark an end of a session (Montgomery and Faloutsos 2001), while “session count 2” uses 30 min of inactivity period which was also used in many previous studies (Catledge and Pitkow 1995; Cooley et al. 1999; Fu et al. 1999; Goseva-Popstojanova et al. 2004; Goseva-Popstojanova et al. 2006a, b, Menasce et al. 2000a, b). This 30 min figure is based on a mean value of 25.5 min (rounded up) determined by Catledge and Pitkow (1995). This figure is also believed to be commonly used in many commercial web applications (Huang et al. 2004). For example, Google Inc. uses the 30 min timeout value for their Analytics web application[3].

Table 11 shows that when the timeout period is decreased, the session count increases. This behaviour is expected because a shorter timeout period means that some longer sessions will be split into multiple shorter sessions. Because the number of errors remains constant, the increased session count means the reliability estimation will increase. This effect can be seen in Tables 13 and 14. Hence, choosing the correct timeout period for the session count is important if an accurate estimation of reliability is to be obtained. This table shows that during the months of January to March 2006, there seems to be a steady increase in traffic for Site A; this “increase in traffic” is expected because there was a marketing campaign launched during this period to attract more users. However, the three available data points are not sufficient to numerically prove this conjecture.

[3] http://www.google.com/support/googleanalytics/bin/answer.py?hl = en&answer = 55463 last accessed May 18, 2008

Table 11 Workloads

Sites	Workload					
	Hit count	Byte count (Mb)	User count	Session count	Session count 2	Days
ECE1	369617	4531	53208	60922	72502	30
ECE2	347413	5874	59727	71141	82761	30
Site A (Jan05)	120699	2191	5015	5336	6036	30
Site A (Feb05)	108219	1953	4982	5353	6017	28
Site A (Mar05)	135282	2474	6175	6633	7572	31
Site A (Apr05)	117785	2229	5800	6144	6961	30
Site A (May05)	113304	2110	5539	5926	6707	31
Site A (Jun05)	120784	2309	5902	6220	6940	30
Site A (Jul05)	105950	2060	5664	5980	6715	31
Site A (Aug05)	112997	2068	5935	6321	7094	31
Site A (Sep05)	111592	1980	5680	6055	6905	30
Site A (Oct05)	117256	2167	6258	6749	7666	31
Site A (Nov05)	122300	2178	6321	6784	7574	30
Site A (Dec05)	107702	2042	5948	6303	7296	31
Site A (Jan06)	148865	2726	7325	7792	8724	30
Site A (Feb06)	134334	2653	6830	7255	8094	28
Site A (Mar06)	161266	3147	8233	8771	10405	31
Site A (Total)	1838335	34287	91607	97622	110415	453

In order to determine if any correlation between the workload characteristics exists, Principal Component Analysis (Jolliffee 1986) was performed. Table 12 shows the results for Site A (Total) and Fig. 2 shows the Scree plot. The plot shows that only one component has an Eigen value over 1 and all other components after Component 1 appear to level off. This suggests that only one component is of importance. Results for the other websites (ECE1 and ECE2) are a similar, but are omitted for brevity. These results show that all of the workload characteristics are highly correlated which suggests that any workload characteristic can be used for reliability estimation. However, website administrators should select the workload characteristic most suitable for their requirements.

Tian et al. (2004) discussed the potential issues in using the byte count as a workload because a variety of entries, including error entries, in the access log that do not contain information on the number of bytes transferred. Upon further investigation, they discovered that the missing entries are associated with binary files already stored in the user cache. The byte count also treats large file size resources as more important than smaller sized resources. For example, let's assume that resources A and B exist on a web server, and resource A is much larger in size than resource B. A user, who requires both resources A

Table 12 Correlation matrix

	Hit count	Byte count	User count	Session count	Session count 2
Hit count	1	0.95	0.91	0.91	0.91
Byte count	0.95	1	0.92	0.92	0.91
User count	0.91	0.92	1	0.998	0.98
Session count	0.91	0.92	0.998	1	0.99
Session count 2	0.91	0.91	0.98	0.99	1

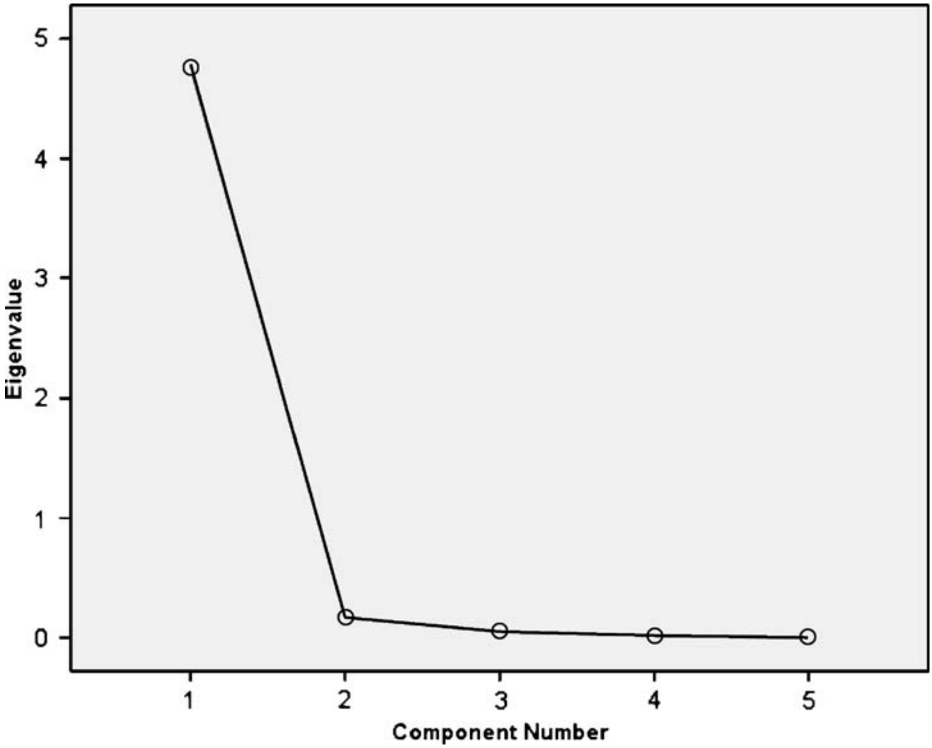

Fig. 2 Scree plot

and B, attempts retrieve these two resources. Resource A failing will have a greater effect on the reliability estimation of the system, which is inappropriate because the reliability of the server is the same regardless of the size of the resource. Figure 3 shows the file size (in Kbytes) histogram for Site A which illustrates this issue. The figure shows that the size of the resources on the furthest right is equivalent to the combined size of many resources on the left side.

Other issues also exist with using the user count and session count as workloads (Alagar and Ormandjieva 2002; Arlitt and Jin 1999; Rosentein 2000). In fact, since web workload characterization was extensively examined by Arlitt and Williamson (1997), many studies have been performed to further examine the individual workloads (Arlitt and Jin 1999; Cherkasova and Phaal 1998; Menasce et al. 1999, 2000). Tian et al. (2004) suggested that each unique IP address can be counted as one user. However, with the current explosion in the number of Internet users, the total amount of IP addresses available is shrinking rapidly. Thus, many methods now exist that allow one public IP address to be used for a group of machines; some of these methods include proxy servers, and personal routers. Since the original study suggests counting one unique IP as a user, there is a strong possibility that this “user” is actually a group of users. As personal routers and proxy servers become more dominant this issue is also becoming more prominent. The session count also suffers this same problem because “one session” may actually be several sessions from several different users who are sharing the same public IP. Thus, a methodology needs to be developed to distinguish different users before accurate reliability analysis can be performed. Websites

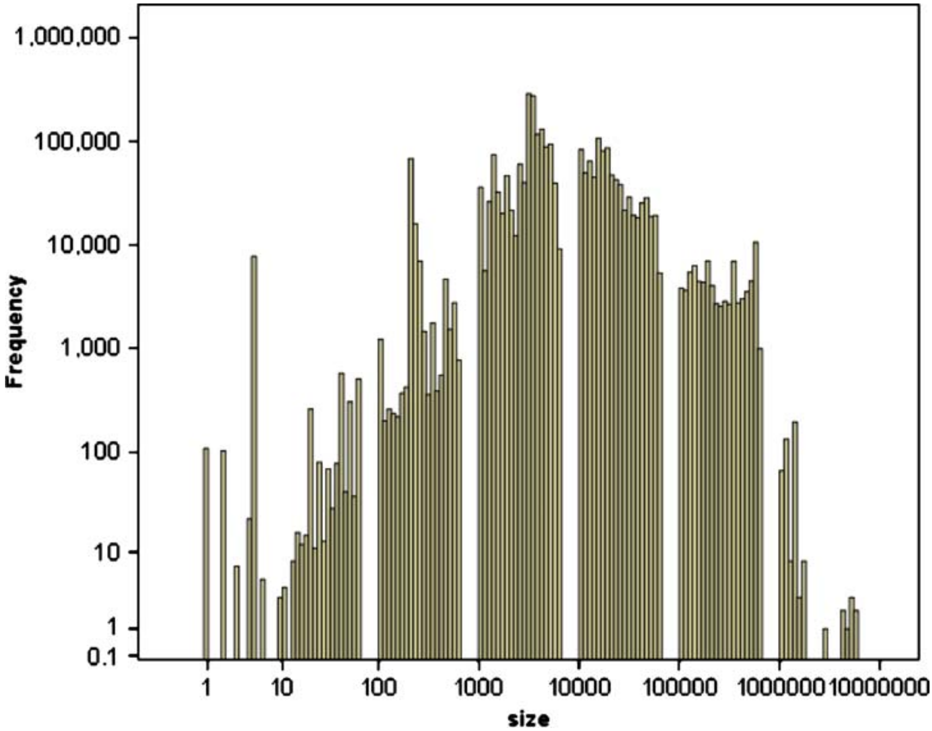

Fig. 3 File size histogram for site A

can use cookies to track user and sessions more effectively by placing a unique identifier and time related information inside the cookie. However, limitations still exist, such as two users sharing the same machine to access the website. The effectiveness of using cookies as a method to track user and session workloads will be explored in our future work.

Results from this section confirm issues with the extraction of workload data from the server logs as discussed in the original study (Tian et al. 2004). Issues not discussed in previous studies (Tian et al. 2004, Goseva-Popstojanova et al. 2006a, b) such as file size bias and proxy servers, are also presented to ensure that web administrators using this approach for reliability estimation are aware of these limitations.

4.4 Reliability Analysis and Discussions

The failures and workloads can be applied to the Nelson model to evaluate the overall operational reliability. Using Eq. 1, R, based on the hits workload, was calculated for the websites under examination; the results can be seen in Table 13. Not surprisingly, Site A, which has the highest reliability requirement, has a high reliability rate during the 15 month period (99.997% of the hits are successful). The sudden drop in reliability during May 2005 was examined; upon closer investigation and discussion with the administrator, we discovered that a configuration setting was not set up correctly; hence the website experienced several simultaneous server failures.

The hit reliability figures are consistent with previous studies (Tian et al. 2004; Goseva-Popstojanova et al. 2004, 2006a, b) in that they are very high. However, other workloads

Table 13 Reliability analysis

Sites	*R*
ECE1	0.999935
ECE2	0.999944
Site A (Except May05)	1
Site A (May05)	0.999960
Site A (Total)	0.999997

can be used to obtain different resolution for the reliability figure. As discussed by Tian et al. (2004) reliability based on other workloads (users, sessions, and bytes) can be calculated using:

$$R = 1 - \frac{f_w}{n_w} \tag{4}$$

where f_w is the number of workloads with at least one failure recorded. For example, f_{users} is the number of users who encountered at least one failure. n_w is the total number of workload units. Goseva-Popstojanova et al. (2004, 2006a), using the Nelson model, discovered that reliability based on the session workload is lower than reliability based on the hit count. However, there is no straightforward relationship between hit reliability and session reliability (Goseva-Popstojanova et al. 2006a); hence, web administrators should not use these two metrics interchangeably. Table 14 displays reliability using the other workloads. This table shows all workload units provide extremely high reliability number due to the low error count associated with the websites under investigation. However, the “days” workload characteristic contains rates that are lower, especially for ECE (closer investigation revealed that the ECE website experienced a high failure rate per day which results in the low reliability figure). Hence, the advantage of the four workloads—being able to provide better granularity than the daily error rate—is lost. In addition, significant issues still exist in accurately estimating the four proposed workloads. Hence, any future work on “live” (as opposed to test) websites should simply utilize days as their basis unless there are specific requirements that force web administrators to use other workload characteristics.

The mean workload between failures (MWBF) can also be calculated using the model discussed in Section 2. This model may provide better estimation due to the fact that it does not have the same limitations that the Nelson model has. Furthermore, it allows web administrators to analyze failure based on time. The original study calculated the MWBF by substituting the number of workloads units for time, effectively using Eq. 3 for analysis; hence, this study also uses this formula to calculate the MWBF for the websites under investigation. The resulting MWBFs for the two websites can be seen in Table 15. Sites (or months) with “n/f” experience no failures during the time period measured. The MWBF

Table 14 Reliability analysis using the other workloads

Sites	R_{bytes}	R_{users}	R_{sessions}	$R_{\text{sessions2}}$	R_{days}
ECE1	1	0.999574	0.999626	0.999682	0.281250
ECE2	1	0.999685	0.999729	0.999771	0.310345
Site A (Except May05)	1	1	1	1	1
Site A (May05)	1	0.999187	0.999229	0.999311	0.806451
Site A (Total)	1	0.999945	0.999947	0.999953	0.986413

Table 15 MWBF

Sites	Hits	Bytes	Users	Sessions	Sessions2	Days
ECE1	44,671	5.30×10^{08}	6,942	7,841	9,364	4.57
ECE2	34,320	4.74×10^{08}	5,931	6,965	8,225	2.90
Site A (Except May05)	n/f	n/f	n/f	n/f	n/f	n/f
Site A (May05)	24,878	4.46×10^{08}	1,230	1,297	1,452	5.17
Site A (Total)	365,138	6.54×10^{09}	18,048	19,034	21,309	75.83

data in Table 15 states that an error will be encountered for each of the workload (bytes, hits, users and sessions) values specified. This table shows that ECE1 has, on average, a failure for every 44,671 hits; Site A would experience one failure after every 365,138 hits. Looking at the "days" column shows that Site A does meet its reliability requirement of having no more than one failure per month (except in May), whereas ECE experiences at least one failure every week which is also expected.

The MWBF calculated using the second MTBF formula can only provide a rough estimate of the actual MTBF. Although using the workload units as a substitute for time is a reasonable method in situations where the time is not available, for this analysis, the time can be calculated from the daily failure. That is, MTBF=24(daily failure rate)

ECE is an academic website; hence it is not surprising to see its MTBF to be at 109.7 h (4.57 days) and 69.6 h (2.90 days) as opposed to Site A which has a MTBF rate of 1,820 h (75.83 days) for the entire 15 months. Again, the low MTBF (relatively) rate for Site A during May 2005 can be attributed to the web application upgrade issue.

This section shows that reliability can be estimated from server logs and expressed in different metrics. Different reliability metrics have been examined to provide system administrators with the flexibility of selecting the correct metric based upon the requirements. For example, the requirements of Site A and ECE were expressed in terms of failures per month. Hence, system administrators for these websites can choose the MTBF to express their estimated reliability.

4.5 Limitations of Log Files

Although log files can provide failure information, reliability can only be estimated from them. The actual reliability cannot be computed solely from web servers' log files due to several issues. The workload information cannot be accurately computed as mentioned in Section 4.1. However, with the help of web technology such as cookies, developers are beginning to be able to track the user session count and user count more accurately. Techniques on identifying the correct timeout value for the session workload are also being discussed by various researchers (He and Goker 2000; Huntington et al. 2008). As these technologies and new techniques are being utilized, more accurate workload data will be gathered which will increase the accuracy of reliability estimation.

Furthermore, errors that are not recorded in the log files may lead to an inflated reliability figure. For example, a website's link may point to an incorrect webpage rather than a missing one. This type of error requires human intervention as the error is only defined by a deviation from the specification rather than an exception. That is, the error codes in the server logs can only identify resource availability issues such as missing resources, moved resources, etc., and not whether the resources contain incorrect content. In this scenario, an error would not be recorded in the log files and the error would only be known when the customer reports the issue. Reliability estimation based on log files alone would not include this error. Because the link is available, automated web crawlers would not be able to detect this error. In fact, this scenario requires manual user intervention to detect the error; hence the error would have to be added manually to the data to increase the accuracy of the proposed reliability estimation method.

5 Conclusions

This paper investigates the validity of evaluating web site reliability based on information extracted from existing web server logs. The investigation is a partial follow up to a previously conducted study (Tian et al. 2004). Two additional websites were examined using the methodology proposed in the original study. The log data for ECE contain 2 months of data that are 1 year apart. The log data for the second website (Site A) cover a continuous 15 months of operation. These two websites belong to two organizations that have different reliability requirements for their websites. During this study several findings were discovered:

- Error codes such as 401, 403, and 404 error codes can be divided into different types. Based on the classification of the error types, we discovered that most errors are no longer source content failures, but are caused by external factors that cannot be controlled by website administrators and content providers. These external factors can be divided into two distinct categories.
- There are issues that exist with the workload information extraction process. The original study explained the difficulties with extracting the byte count workload. However, unique challenges also exist with the extraction of the user and session and hit count workloads. For example each IP may be shared by many users, thus counting each unique IP address as a user will lead to the situation where the counted number of users is actually less than the number of actual users.
- The number of high “value” errors is very low as seen in Table 9 which displays the numbers of errors encounter “per month”. Hence, the other workloads examined cannot provide better granularity than the daily error rate.
- The Nelson model, used for calculating reliability, is not applicable to some workloads without modifications. The MTBF for a website can be estimated because the total service time can be calculated from the total number of sessions. However, the MTBF will vary depending on the error codes used in the analysis. Thus, the correct error codes need to be selected before reliability evaluation is performed.
- Some of the error codes in response to requests are very similar to requests containing malicious payloads. For example, the 414 error is returned when the URI is too long. A benign client can generate a long URI due to some bug in its code; however the URI can also be too long when an attacker is trying to embed a large piece of JavaScript code to take advantage of a cross-site scripting vulnerability.

Our future works consist of detailed examination of the user and session workloads. In particular, we plan to investigate the intra/inter-session characteristics as defined by Goseva-Popstojanova (2006a) in order to examine the behaviors of new users (or sessions) versus repeat users (or sessions) and how these behaviors may affect the reliability of the web server. Furthermore, the effectiveness of using cookies as a method to track user and session workloads will be explored.

References

Alagar VS, Ormandjieva O (2002) “Reliability Assessment of Web Applications”. 26th Annual International Computer Software and Applications Conference, 405–412

Arlitt MF, Jin T (1999) Workload characterization of the 1998 world cup web site. HP Labs, Paolo Alto, Technical Report HPL-1999–35 (R.1)

Arlitt MF, Williamson CL (1997) Internet Web Servers: Workload Characterization and Performance Implications. IEEE/ACM Trans Netw 5(5):631–645 doi:10.1109/90.649565

Boyd S, Keromytis A (2004) Preventing SQL injection attacks. 2nd Applied Cryptography and Network Security (ACNS) Conference, Yellow Mountain, China, June 8–11, pp 292–304

Catledge L, Pitkow J (1995) Characterizing browsing behaviors on the World Wide Web. Comput Netw ISDN Syst 27(6):1065–1073 doi:10.1016/0169-7552(95)00043-7

CGISecurity.com (2002) The Cross Site Scripting FAQ. Accessed at May 15, 2008. http://www.cgisecurity.com/articles/xss-faq.shtml

Cherkasova L, Phaal P (1998) Session based admission control: a mechanism for improving the performance of an overloaded web server. HP Labs, Paolo Alto Technical Report, HPL-08-119

Cook. S (2003) A web developer’s guide to cross-site scripting. Accessed May 15, 2008. http://www.giac.org/practical/GSEC/Steve_Cook_GSEC.pdf

Cooley R, Mobasher B, Srivastava J (1999) Data preparation for mining World Wide Web browsing patterns. Knowl Inf Syst 1(1):5–32

Cowan C, Pu C, Maier D, Hinton H, Bakke P, Beattie S et al (1998) StackGuard: automatic adaptive detection and prevention of buffer-overflow attacks. 7th USENIX Security Conference, San Antonio, TX, USA, pp 63–78

Cremonesi P, Serazzi G (2002) “End-to-end performance of web services”, performance evaluation of complex systems: techniques and tools, performance 2002 tutorial lectures. Lect Notes Comput Sci 2459:158–178 doi:10.1007/3-540-45798-4_8

Crovella ME, Bestavros A (1997) Self-similarity in world wide web traffic: evidence and possible causes. IEEE/ACM Trans Netw 5(6):631–645

Evans D, Larochelle D (2002) Improving security using extensible lightweight static analysis. IEEE Softw 42–51, Jan/Feb: doi:10.1109/52.976940

Fu Y, Sandhu K, Shih M (1999) Clustering of web users based on access patterns. International Workshop on Web Usage Analysis and User Profiling (WEBKDD'99), San Diego, CA, USA

Galletta DF, Henry R, McCoy S, Polak P (2004) Web site delays: How tolerant are users. J AIS 5(1):1–28

Goseva-Popstojanova K, Mazimdar S, Singh A (2004) Empirical study of session-based workload and reliability for web servers. 15th IEEE International Symposium on Software Reliability, Saint-Malo, France, 403–414

Goseva-Popstojanova K, Singh A, Mazimdar S, Li F (2006a) Empirical characterization of session-based workload and reliability for web servers. Empir Softw Eng J 11(1):71–117 doi:10.1007/s10664-006-5966-7

Goseva-Popstojanova K, Li F, Wang X, Sangle A (2006b) A contribution towards solving the web workload puzzle. 2006 Intl Conf Dependable Syst Networks (DSN’06) pp. 505–516

Grant J (2000). Ten undeniable truths for web design. Accessed at May 15, 2008. http://www.htc.net/~joegrant/grantconsulting/articles/undeniable_truths_20000803.htm

Grossman J (2004) Thwarting SQL web hacks. VAR Business 20:41–42

He D, Goker A (2000) “Detecting session boundaries from Web user logs”, 22nd Annual Colloquium on Information Retrieval Research, British Computer Society, pp. 57–66.

Huang YW, Huang SK, Lin TP, Tsai CH (2003) Web application security assessment by fault injection and behavior monitoring. 12th International Conference on World Wide Web, Budapest, Hungary, pp. 148–159

Huang X, Peng F, An A, Schuumans D (2004) Dynamic web log session identification with statistical language models. J Am Soc Inf Sci Technol 55(14):1290–1303 doi:10.1002/asi.20084

Huntington P, Nicholas D, Jamali HR (2008) Website usage metrics: A re-assessment of session data. Inf Process Manage 44(1):358–372 doi:10.1016/j.ipm.2007.03.003

Huynh T, Miller J (2005) Further investigations into evaluating website reliability. 4th International Symposium on Empirical Software Engineering, Noosa Heads, Australia, pp 162–171

Jolliffee IT (1986) Principal component analysis. Springer, New York

Kallepalli C, Tian J (2001) Measuring and modeling usage and reliability for statistical web testing. IEEE Trans Softw Eng 27(11):1023–1036 doi:10.1109/32.965342

Lyu MR (1995) Handbook of software reliability. McGraw-Hill, Columbus

Ma L, Tian J (2003) Analyzing errors and referral pairs to characterize common problems and improve web reliability. 3rd International Conference on Web Engineering, Oviedo, Spain, pp. 314–323

Masterson M (1999) E-com tech tough enough? CNN Money. Accessed at May 15, 2008. http://money.cnn.com/1999/11/19/technology/etail_tech/

Menasce D, Almeida V, Fonseca R, Mendes M (1999) A methodology for workload characterization of e-commerce sites. ACM Conference on Electronic Commerce, Denver, CO, USA, pp. 119–128

Menasce D, Almeida V, Foneca R, Mendes M (2000a) Business-oriented resource management policies for e-commerce servers. Perform Eval 32(2–3):223–239 doi:10.1016/S0166-5316(00)00034-1

Menasce D, Almeida V, Ried R (2000b) In Search of Invariants for E-Business Workloads. 2nd ACM Conference on Electronic Commerce, Minneapolis, MI, USA, pp. 56–65.

Montgomery AL, Faloutsos C (2001) Identifying web browsing trends and patterns. IEEE Comput 34(7):94–95

Musa JD, Iannino A, Okumoto K (1987) Software reliability: measurement, prediction, application. McGraw-Hill, Columbus

Nah FH (2002) A study of web users' waiting time. In: Sugumaran, V (eds) Intelligent support systems technology: knowledge management. IRM, Hershey, pp 145–152

Nelson E (1978) Estimating Software Reliability from Test Data. Microelectron Reliab 17(1):67–73 doi:10.1016/0026-2714(78)91139-3

Offutt J (2002) Quality Attributes of Web Applications. IEEE Software. Spec Issue Softw Eng Internet Softw 19(2):25–32

Pitkow JE (1999) Summary of WWW characterizations. World Wide Web 2(1–2):3–13 doi:10.1023/A:1019284202914

Rose GM, Lees J, Meuter M (2001) A refined view of download time impacts on e-consumer attitudes and patronage intentions toward e-retailers. Int J Media Manage 3(2):105–111

Rosenstein M (2000) What is Actually Taking Place in Web Sites: E-Commerce Lessons from Web Server Logs. 2nd ACM Conference on Electronic Commerce (EC'00), Minneapolis, MN, USA, pp. 38–43.

Spitzner L (2001) Know your enemy: revealing the security tools, tactics, and motives of the Blackhat Community, chapter 6. Addison–Wesley, Boston

Tian J, Rudraraju S, Li Z (2004) Evaluating web software reliability based on workload and failure data extracted from server logs. IEEE Trans Softw Eng 30(11):754–769 doi:10.1109/TSE.2004.87

Trivedi KS (2001) Probability and statistics with reliability, queuing, and computer science applications, 2nd edn. Wiley, New York

Wagner D, Foster JS, Brewer EA, Aiken A (2000) A first step towards automated detection of buffer overrun vulnerabilities. Network and Distributed System Security Symposium, San Diego, pp 3–17

Wang W, Tang M (2003) User-oriented reliability modeling for a web system. 14th International Symposium on Software Reliability Engineering, Denver, CO, USA, pp 293–304

Williams J (2001) "Avoiding the CNN Moment", IT Pro, March-April, 68–72.

Toan Huynh received a B.Sc. in Computer Engineering from the University of Alberta, Canada. He is currently a PhD candidate at the same institution. His research interests include: web systems, e-commerce, software testing, vulnerabilities and defect management, and software approaches to the production of secure systems.

James Miller received the B.Sc. and Ph.D. degrees in Computer Science from the University of Strathclyde, Scotland. Subsequently, he worked at the United Kingdom's National Electronic Research Initiative on Pattern Recognition as a Principal Scientist, before returning to the University of Strathclyde to accept a lectureship, and subsequently a senior lectureship in Computer Science. Initially during this period his research interests were in Computer Vision; since 1993, his research interests have been in Software and Systems Engineering. In 2000, he joined the Department of Electrical and Computer Engineering at the University of Alberta as a full professor and in 2003 became an adjunct professor at the Department of Electrical and Computer Engineering at the University of Calgary. He has published over one hundred refereed journal and conference papers on Software and Systems Engineering (see www.steam.ualberta.ca for details on recent directions); and currently serves on the program committee for the IEEE International Symposium on Empirical Software Engineering and Measurement; and sits on the editorial board of the Journal of Empirical Software Engineering.