Estimating the Error Rate in an Apache Web Server System

Xiao XIAO and Tadashi DOHI
Department of Information Engineering, Graduate School of Engineering
Hiroshima University, 1--4--1 Kagamiyama, Higashi-Hiroshima, 739--8527 Japan
E-mail:{xiaoxiao/dohi}@rel.hiroshima-u.ac.jp

Abstract

In this paper we focus on the relationship between the error rate which is one of the representative reliability measures in Apache web servers and the system parameters which reflect on the web server's system performance, and develop a probability model to describe it. More specifically, we implement a simple client server system and carry out an experiment to measure both the error rate and the system parameters. As the result on quantitative evaluation of the proposed logit model, it is shown that our model could fit the empirical error rate with varying the number of threads.

Keywords: *web server reliability, Apache web server, system parameters, error rate, measurement-based approach, logit model.*

1. Introduction

In the informational society where internet is widely spreaded, a very important issue is to keep the performance of web service systems. In this point of view, quantitative evaluation of the workload becomes necessary in operational web server performance assessment. Barford and Crovella [1] developed a tool that automatically measures six metrics that characterize the web server's workload such as the size of server file, the request size and so on. Hu, Nanda and Yang [2] focused on the Apache server that can operate on a single processor or 4-CPU-SMP (symmetric multi-processor) system, and executed the experimental performance evaluation of the Apache web server. van der Mei, Hariharan and Reeser [3] proposed an end-to-end performance evaluation model of the web server system which is based on a tandem type of queueing model, and investigated effectiveness of the model through simulation experiments. Nahum, Barzilai and Kandlur [4] also evaluated the performance of www server on UNIX platform, and developed a method that can efficiently reduce the server throughput. Cao *et al.* [5] considered analytically a queueing model that evaluates the performance of the web server, and derived the performance evaluation measures such as the average response time and the blocking probability, *etc*.

In this paper, similar to Hu, Nanda and Yang [2], we consider a quantitative evaluation of the Apache web server which is the most commonly used web server software. Due in good part to the broad utilization of Apache web server, it is considered that our challenge in this paper may be capable of wide application. Moreover, we choose Apache web server, but not IIS or AN HTTPD, also for the following advantages; (i) it is a freeware, (ii) adequate responses for questions by the users are easily available from the Apache community when any technical problem occurred, (iii) the reliability as a server software is quite high, (iv) it possesses a variety of technical functions, (v) it corresponds to many kinds of operational circumstances. In this way, the Apache web server can demonstrate an enough performance even as it is, but a higher performance can be attained by ascertaining the bottleneck and by evaluating the performance through *load test*. Sugiki, Kono and Iwasaki [6] showed that the server throughput can be improved by the self-adjustment of the keep-alive time, and implemented it on a library of the Apache web server. Though a lot of tools have been developed up to now as load test tools, the Apache JMeter [7] is often used in the actual performance assessment of server systems.

In general, there are three performance evaluation measures in the Apache JMeter; mean response time, error rate and system throughput. In this paper, we focus on the error rate to assess the reliability of the Apache web server system. Unlike the general performance evaluation problem, there are very a few works related to the web server reliability. Shereshevsky *et al.* [8] constructed a simple client server system composed of two machines, and found the multifractal nature between software aging phenomenon [9] and the memory resource parameters. Also they examined the problem of forecasting the time to the system failure. Gokhale, Vandal and Lu [10] applied a stochastic model which is called *stochastic reward net* to carry out the throughput/reliability analysis based on the architecture of the Apache web server. However in the previous works, the system parameters which influence both the system performance and the error rate were not clarified. Hence the quantitative evaluation model to describe the error-occurrence mechanism of the web server system should be explored.

In this paper, we take a measurement-based approach under the load test circumstance and propose a simple logit model which can describe the relationship between the error rate and the system parameters. The logit model is the well-known nonlinear regression model that is frequently used in the field of econometrics and the medical statistics. We construct an experiment system to measure the system parameters under the assumption that multiple clients send HTTP (Hypertext Transfer Protocol) requests to a web server randomly. We define the error rate as a ratio of the number of failed requests recognized as time-out errors when there is no response from the server for a constant response time period. Based on the logit model and its related standard techniques for statistical inference, we estimate the model parameters and perform the goodness-of-fit test on it. The rest part of this paper is organized as follows. In Section 2 we briefly explain the experimental setup in our study and describe the data sets collected in the experiments. Section 3 is spent for introduction of the logit model and its related parameter estimation with the least squares principle. The goodness-of-fit test result is presented in Section 4. Finally Section 5 contains some concluding remarks and the possible directions to the future research.

2. Experiments

2.1. Experimental Setup

In our experimental setup, the client-server system consists of an Apache server (version 2.2.4) on a Windows platform and a client connected via a local LAN. The system components and system structure are illustrated in Figure 1. In this experimental setup with only one client, we put the homepage, index.jsp, in the home directory of the web server and use the Apache JMeter [7] to generate HTTP requests that access the

	Client	Server
OS	Windows XP	Windows 2000
Processor	2.59GHz	850MHz
RAM	2.00GB	128MB
Software	Apache JMeter (version 2.1.01082007)	Apache Server (version 2.2.4)

Figure 1. Experimental setup.

homepage randomly. In this way, we represent an environment where multiple clients randomly access the web server. Basically, there are three parameters we can use to control the workload of web server; (i) *number of threads* (users), (ii) *ramp-up period* (in seconds) and (iii) *loop count*. For instance, if these three parameters are set to be 20, 10 and 5, respectively, then it means that 20 clients send HTTP requests 5 times severally while all the 100 requests are started within 10 seconds. In our experiments, we fix the ramp-up period (in seconds) as 10 seconds and the loop count an infinity ("Forever" checkbox is selected) meanwhile the number of threads is set to be 30, 40 and 50. So, three kinds of workload environments are generated under this experiment.

2.2. Data Collection

We are interested in the error rate which is defined by

$$R_i = \frac{\text{the number of HTTP requests failed by time } i}{\text{the number of HTTP requests generated by time } i}, \qquad (1)$$

where the failed HTTP requests are those caused by the time-out error and $i = 1, 2, \ldots$ denote the discrete time index. The error rate is automatically recorded on the *Aggregate Report* of the Apache JMeter, so we only need to collect the data of system parameters that reflect on the web server's performance. Since we focus on the system parameters concerned with memory resource usage, we employ the *Performance*, which is one of the Windows 2000 system tools, to take log files of these system parameters. Table 1 presents 9 system parameters which describe the momentary statement of the web server system, where for the notational convenience, we denote AB ～ SCRB as

Table 1. System parameters.

AB (k=1)	size of the virtual memory available currently
CaB (k=2)	number of bytes currently in use by the system cache
CoB (k=3)	size of virtual memory that has been committed
PNA (k=4)	number of calls to allocate space in the system nonpaged pool
PNB (k=5)	number of bytes in the nonpaged pool
PPA (k=6)	number of calls to allocate space in the system paged pool
PPB (k=7)	number of bytes in the paged pool
PPRB (k=8)	size of paged pool resident in core memory
SCRB (k=9)	number of bytes of system code total bytes currently resident in core memory

Table 2. An example of the sample data.

Time	Error Rate	AB	CaB	...	SCRB
1	0.00	12455936	2084640		4612096
2	0.00	12029952	20680704		4612096
3	10.68	12148736	20701184		4632576
4	41.37	12406784	20717568	. . .	4648960
5	56.94	12455936	20738048		4669440
. . .	. . .	. . .	. . .		. . .
179	32.73	11493376	21512192		5443584
180	32.59	11407360	21549056		5464064

system parameters $k = 1 \sim 9$ hereafter.

In our experiments, we set the operation time (time limit for web server system working per experiment) as 30 minutes and take the record every 10 seconds. Then since we obtain a sample with 180 data from one experiment, this procedure is repeated for $n = 30$ times when the number of threads is set to 30, 40 or 50. An example of the sample is shown in Table 2. The main reason why the operation time is set to be 30 minutes is that the error rate tends to converge to a constant value after a while, so we stop the experiment every 30 minute which can be considered as a proper time limit.

3. Logit Model
3.1. Regression-based Model

From the above discussion we know that system parameters concerned with memory resource usage may reflect the total performance of the web server. On the other hand, the error rate can be considered as one of the important reliability measures of the web server. Then the question is what the dependence of the system parameters on the error rate is. To this end, we propose a simple probability model to describe the relationship between multiple system parameters and the error rate. It is worth mentioning that in our situation the usual linear regression model is inappropriate because the error rate takes positive values less than one. The logit model under consideration can represent the error rate as a function of multiple system parameters used as explanatory variables and can bridge between them.

Let R_i $(0 \le R_i \le 1)$ and $\boldsymbol{x}_i = (x_{0,i}, x_{1,i}, ..., x_{9,i})$ denote the error rate and the system parameter vector observed at time i (= 1, 2, …) with $x_{0,i} = 1$. Then the error rate can be expressed as a function of system parameters by

$$R_i = R(\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{\beta}\boldsymbol{x}_i^T)}{1+\exp(\boldsymbol{\beta}\boldsymbol{x}_i^T)}, \tag{2}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_9)$ is the regression coefficient and T means transposition. Next, applying the logit transformation to function R_i yields

$$Y_i = Y(\boldsymbol{x}_i) = \log\left(\frac{R(\boldsymbol{x}_i)}{1-R(\boldsymbol{x}_i)}\right). \tag{3}$$

Based on the above transform, we consider the following linear regression model:

$$Y_i = \boldsymbol{\beta}\boldsymbol{x}_i^T + \varepsilon, \tag{4}$$

where ε is the Gaussian error term. More formally, we make the following assumptions:
(Assumption 1) The mean of error term is zero, *i.e.*, $\mathrm{E}[\varepsilon] = 0$.
(Assumption 2) The variance of error term is constant, *i.e.*, $\mathrm{Var}[\varepsilon] = \sigma^2$.
(Assumption 3) The covariance of error term is zero, *i.e.*,

$$\mathrm{E}[\varepsilon_i \varepsilon_{i'}] = 0 \; (i \ne i', \; i = 1, ..., n, \; i' = 1, ..., n).$$

(Assumption 4) The probability density function of ε is given by

$$p(y) = (2\pi\sigma^2)^{-1/2} \exp[-y^2/2\sigma^2].$$

At each observation time point, i = 1, 2, …, 180, we repeat to observe Y_i and $\boldsymbol{x}_i$ for n = 30 times and estimate the regression coefficients $\boldsymbol{\beta}$ by means of the well-known mean squares error method. It is noted that this statistical estimation method can be validated by the well-known Gauss-Markov Theorem, so an estimate of regression coefficient $\boldsymbol{\beta}$ is given by the solution of

$$\min_{\boldsymbol{\beta}} \sum_{j=1}^{n} \sum_{i=1}^{180} \left(y_{i,j} - \sum_{k=0}^{9} \beta_k x_{k,i,j} \right)^2, \tag{5}$$

where $y_{i,j}$ denotes the logit transformed value of the error rate observed at time i in the j th experiment, while $x_{k,i,j}$ denotes the value of system parameter k observed at time i in the j th experiment.

In order to predict the long-term behavior of the error rate, it is needed to know the future values of system parameters. If the system parameters are known in advance it is easy to estimate regression coefficients with the least squares method. In the case where system parameters can not be observed, on the other hand, we need to predict both of regression coefficients and the system parameters. In our experiences through the experiments, it is found that the system parameters, CoB, PPB and PPRB, almost take constant values, while the other parameters tend to show non-constant trends with stationary trend. To estimate the future system parameter values we apply a simple linear regression model and use the estimated values to predict the long-term behavior of the error rate.

Let $x_{k,i}$ denote the value of system parameter k observed at time i = 1, 2, …, 180. Define the time vector I = (1, i) and assume that the system parameter is a function of time i. Then $x_{k,i}$ takes the following form:

$$x_{k,i} = \boldsymbol{\alpha}_k \bullet \boldsymbol{I}^T = \alpha_{k,0} + i \bullet \alpha_{k,1}, \tag{6}$$

where $\boldsymbol{\alpha}_k = (\alpha_{k,0}, \alpha_{k,1})$ is the regression coefficient of system parameter k. Then an estimate of system parameter is given by $\boldsymbol{\alpha}_k$ which is the solution of the following minimization problem:

$$\min_{\boldsymbol{\alpha}_k} \sum_{i=1}^{180} \left[x_{k,i} - (\alpha_{k,0} + i \bullet \alpha_{k,1}) \right]^2, \tag{7}$$

Finally, from estimates of $\boldsymbol{\alpha}_k$ it is possible to obtain estimates of $x_{k,i}$ and the error rate in an arbitrary time.

4. Real Data Analysis

In this section, we apply the logit model to the measurement data. To evaluate the prediction performance of the logit model, we use the standardized residual sum of squares (RSS) as a criterion to quantify the estimation/prediction ability. For the fixed number of threads, we execute 33 experiments in parallel, where 30 data sets are used for estimation (training) and 3 data sets are used to investigate the prediction performance. The regression coefficients, $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}_k$, are estimated 6 times while the number of experiment, n, is set to be n = 5, 10, 15, 20, 25 or 30. All the statistical analysis executed in this paper is employed with the well-known statistical analysis package, R [11].

Table 3 presents the goodness-of-fit result when the number of threads is fixed as 30, 40 or 50. The values in this table show the RSS between the observed error rate and its estimates via the least squares method, where n denotes experiment times in the estimation of regression coefficient $\boldsymbol{\beta}$. It is evident that as the number of samples increases, the estimation accuracy becomes much better. This is not a surprising result, because it is no wonder that the more available measurement data, the better the accuracy will be. However, it seems to be interesting to take notice on the results

Table 3. RSS with monitoring system parameters.

(1) no. threads = 30

RSS	n = 5	n =10	n =15	n = 20	n = 25	n = 30
SampleEva01	817.51	131.03	87.12	68.34	56.94	47.83
SampleEva02	1125.75	976.43	205.12	202.35	187.05	165.95
SampleEva03	2588.02	134.67	116.89	122.15	113.90	121.15

(2) no. threads = 40

RSS	n = 5	n =10	n =15	n = 20	n = 25	n = 30
SampleEva01	2340.30	144.59	87.86	85.10	79.26	78.93
SampleEva02	764.98	150.68	101.08	86.74	117.78	95.78
SampleEva03	1850.64	160.73	101.68	83.56	69.95	69.77

(3) no. threads = 50

RSS	n = 5	n =10	n =15	n = 20	n = 25	n = 30
SampleEva01	1069.71	182.56	161.20	160.84	141.05	149.22
SampleEva02	1552.62	322.55	227.00	189.76	254.09	205.14
SampleEva03	1990.38	182.71	177.87	172.81	156.76	154.76

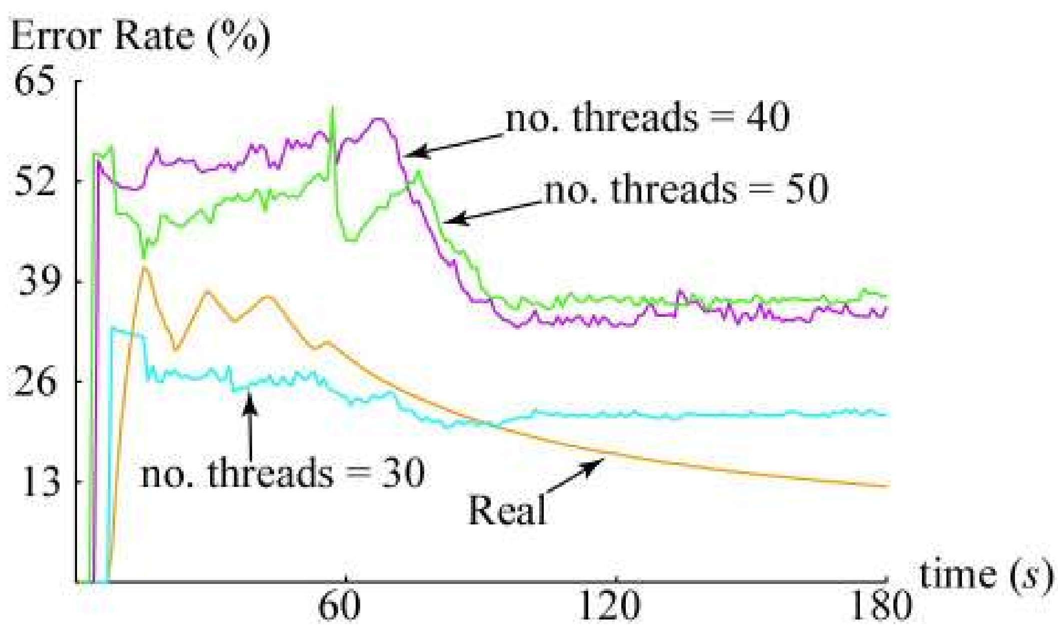

Figure 2. Comparison among different no. threads.

Table 4. RSS with predictions of system parameters.

(1) no. threads = 30

RSS	n = 5	n =10	n =15	n = 20	n = 25	n = 30
SampleEva01	464.18	236.39	133.81	117.20	92.96	98.53
SampleEva02	634.64	346.72	202.75	176.29	134.97	145.21
SampleEva03	382.46	196.86	122.98	113.23	100.74	103.08

(2) no. threads = 40

RSS	n = 5	n =10	n =15	n = 20	n = 25	n = 30
SampleEva01	407.96	330.16	333.28	284.82	270.96	259.55
SampleEva02	539.67	434.92	439.24	367.98	344.96	366.02
SampleEva03	332.09	270.58	273.00	237.88	229.39	252.64

(3) no. threads = 50

RSS	n = 5	n =10	n =15	n = 20	n = 25	n = 30
SampleEva01	286.90	322.69	283.32	247.02	239.25	252.74
SampleEva02	488.27	540.77	462.08	373.89	348.89	328.13
SampleEva03	287.03	318.50	294.24	277.60	276.34	314.75

among different threads. SampleEva01 gives the smallest RSS when the number of threads is set to be 30, while SampleEva02 and SampleEva03 show smaller RSS in (2) than that in (3) of Table 3. This suggests that our logit model is more suitable for lower number of threads, or to say, it is more capable in a lighter workload situation. This result behaves much more clear in the following prediction part (refer to Table 4). As the result of parameter estimation, it can be checked that the error rate behaves as a multi-modal function of time and is right-skewed. This feature of error rate is captured by the proposed model rather excellently, and Figure 2 gives an obvious idea about this result.

Our next concern is the long-term prediction of the error rate as well as the system parameters. The RSS between the predicted and observed error rate is presented in Table 4, where the regression coefficient $\boldsymbol{\beta}$ is estimated by the least squares method. It should be noted that the value of system parameter $\boldsymbol{x}_i$ is a predictive one based on the linear regression model, but not the actual observations. It is a natural conclusion that $\boldsymbol{\alpha}_k$ can be estimated better as the number of experiments monotonically increases.

5. Conclusion

In this paper, we have proposed a simple logit model to describe the relation between the error rate, which is the representative reliability measure in the Apache web server, and the system parameters which reflect on the web server's system performance. By combining a simple linear regression model with the logit model, we have estimated the long-term behavior of the error rate and the system parameters simultaneously. It will be necessary, of course, to improve the estimation and prediction accuracy of the logit model in future. Especially, it is worth considering a novel approach to represent the situation where the workload varies dynamically.

6. References

[1] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation", Proceedings of 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'98), ACM, 1998, pp. 151-160.

[2] Y. Hu, A. Nanda and Q. Yang, "Measurement, analysis and performance improvement of the Apache web server", Proceedings of 18th IEEE International Performance, Computing and Communications Conference (IPCCC'99), IEEE CS Press, 1999, pp. 261-267.

[3] R. D. van der Mei, R. Hariharan and P. K. Reeser, "Web server performance modeling, Telecommunication Systems", 2001, vol. 16, no. 3/4, pp. 361-378.

[4] E. Nahum, T. Barzilai and D. Kandlur, "Performance issues in www servers", IEEE/ACM Transactions on Networking, 2002, vol. 10, no. 1, pp. 2-11.

[5] J. Cao, M. Andersson, C. Nyberg and M. Kihl, "Web server performance modeling using an M/G/1/K^*PS queue", Proceedings of 10th International Conference on Telecommunications (ICT' 03), 2003, pp. 1501-1506, IEEE.

[6] Sugiki, K. Kono and H. Iwasaki, "Tuning mechanisms for two major parameters of Apache web servers, Software -- Practice and Experience", Oct. 2008, vol. 38, no. 12, pp. 1215 - 1240.

[7] JMeter - Apache JMeter, http://jakarta.apache.org/jmeter/

[8] M. Shereshevsky, B. Cukic, J. Crowel, V. Gandikota, Y. Liu, "Software aging and multifractality of memory resources", Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2003), IEEE CS Press, 2003, pp. 721-730.

[9] M. Grottke and K. S. Trivedi, "Fighting bugs: remove, retry, replicate, and rejuvenate", IEEE Computer, 2007, vol. 40, no. 2, pp. 107-109.

[10] S. S. Gokhale, P. J. Vandal and J. Lu, "Performance and reliability analysis of web server software architectures", IEEE CS Press, 2006, Proceedings of 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06), pp. 351-358.

[11] http://www.r-project.org/

Authors

Xiao Xiao received the B.Sc. (Engineering) and M.Sc. (Engineering) from Hiroshima University, Japan, in 2008 and 2010, respectively. She is now working as a Ph.D. candidate in Graduate School of Engineering, Hiroshima University. She is a Student Member of IEICE.

Tadashi Dohi received the B.Sc. (Engineering), M.Sc. (Engineering), and Ph.D. (Engineering) from Hiroshima University, Japan, in 1989, 1991, and 1995, respectively. In 1992, he joined the Department of Industrial and Systems Engineering, Hiroshima University, Japan, as an Assistant Professor. Now he is a Full Professor in the Department of Information Engineering, Faculty of Engineering, Hiroshima University, Japan, since 2002. In 1992, and 2000, he was a Visiting Research Scholar at University of British Columbia, Canada, and Duke University, USA, respectively, on leave of absence from Hiroshima University. His research areas include Software Reliability Engineering, Dependable Computing, and Performance Evaluation. Dr. Dohi has published over three hundred journal and peer-reviewed conference papers in the above research topics. He is a Regular Member of ORSJ, JSIAM, IEICE, REAJ, and IEEE. He also served as General Co-chair of AIWARM’04-08, WoSAR’08-10, MENS'10, APARM'10 and General Chair of ISSRE'11, Tokyo, Japan.