Enhanced URMG Method for Automatically Testing Web Applications in the Cloud

Amol K. Rajput[1], Ajay Kumar Gupta[2]

[1]Student, Samanvay Pratishthan's Institute of Knowledge College of Engineering, Savitribai Phule Pune University,
Pimpale Jagtap, Pune, India

[2]Assistant Professor, Samanvay Pratishthan's Institute of Knowledge College of Engineering, Savitribai Phule Pune University,
Pimpale Jagtap, Pune, India

Abstract: *In real world large number of web applications have been developed and deployed and these applications are run in cloud. Every day in life on cloud web applications are visited by several or billions of users concurrently. Testing of this application manually takes lots of time or it's impossible to test like real user as manually. Thus the testing and performance evaluations of these applications are important. User model based evaluations can be reduce the manual work and enable us to determine the performance of the application under real time environments. There are different kinds of models using mining web access logs such as Markov model and Customer behavior model graph. This project proposed to Enhance of User Representation Model Graph (EURMG). URMG which is built based on CBMG. It uses an algorithm to redefine URMG and optimizes the evaluations execution process. Based on this model, an automatic testing and evaluation system for web applications is designed, implemented, and deployed in our test cloud, which is able to execute all of the analysis and testing operations using only web access logs.*

Keywords: Cloud Computing; Cloud Based Testing; Software Testing; Cloud Services; Testing Difficulties.

1. Introduction

Cloud-based testing has the potential to offer a compelling combination of lower costs, pay-per-use and elimination of upfront capital expenditures. It can address the ramping demands for sophisticated test environments. The benefits, however, extend beyond cost. The non-cost factors include utility-like, on-demand flexibility, freedom from holding assets, enhanced collaboration, greater levels of efficiency and, most importantly, reduced time-to-market for key business applications. At the same time, cloud-based testing introduces a new set of challenges, such as data security and a lack of standards, especially in the public cloud model. Testing is considered an important but non-business-critical activity. Moving testing to the cloud is seen as a safe bet because it doesn't include sensitive corporate data and has minimal impact on the organization's business-as-usual activities.

Web applications are becoming increasingly popular in daily life, and because of the rapid development of cloud technologies, these applications can support concurrent usage by several millions or billions of users. Applications are increasingly becoming dynamic, complex, distributed and component-based, creating a multiplicity of new challenges for testing teams. For instance, mobile and Web applications must be tested for multiple operating systems and updates, multiple browser platforms and versions, different types of hardware and a large number of concurrent users to understand their performance in real-time. The conventional approach of manually creating in-house testing environments that fully mirror these complexities and multiplicities consumes huge capital and resources.

To satisfy the rapid growth of cloud technologies, a large number of web applications have been developed and deployed, and these applications are being run in clouds. Due to the scalability provided by clouds, a single web application may be concurrently visited by several millions or billions of users. Thus, the testing and performance evaluations of these applications are increasingly important. User model based evaluations can significantly reduce the manual work required, and can enable us to determine the performance of applications under real runtime environments [5]. Hence, it has become one of the most popular evaluation methods in both industry and academia. Significant efforts have focused on building different kinds of models using mining web access logs, such as Markov models and Customer Behavior Model Graph (CBMG) [6].

To solve the problem of user model based evaluation in Customer Behavior Model Graph (CBMG) we propose a new model called the Enhance - User Representation Model Graph (E-URMG) to support high number of user sessions and minimize the session error by prioritization the service request as Low (L), Medium (M) and High (H) based on the existing URMG support. The URMG is built based on CBMG, which helps to optimize the evaluation execution. Based on URMG, we designed and implemented an automatic testing and evaluation system for web applications, which can execute the access log analysis, GBMG and URMG construction, virtual user creation, evaluation, and data management without the need for any manual operation.

2. Related Work

To solve the problem of user model based evaluation in Customer Behavior Model Graph (CBMG) we propose a new model called the Enhanced User Representation Model Graph (E-URMG) enhancing the URMG. The URMG is built based on CBMG, which helps to optimize the evaluation execution. Based on URMG, we designed and

implemented an automatic testing and evaluation system for web applications, which can execute the access log analysis, CBMG and URMG construction, virtual user creation, evaluation, and data management without the need for any manual operation.

Mostly, Large application systems are characterized by workloads that consist of sessions, where users interact with the system through a sequence of requests. The idea of this work is to analyze the logs of multi-tier systems so that sessions (sequence of interactions with the system) can be derived and modeled. One of the techniques used to model a repeated sequence of user interactions with application systems is the Customer Behavior Model Graph (CBMG). CBMG is a state transition diagram used to describe the behavior of a group of users exhibiting similar behavioral patterns while interacting with the application system. The CBMG is also referred to as a User Interaction Model, which describes the nature of interaction between the user and the system. The model describes the frequency of interaction and the sequence of requests submitted to the system [10].

3. Research Methodology

To implement URMG propose three rules to identify the particular session used to replace the states of CBMG. The rules are integrity, relevance similarity, and topology similarity.

3.1 Integrity

Integrity refers to the state integrity. Because we will use the access records of one session to replace the state of CBMG, the session has to contain all of the states. In other words, all of the states of CBMG have at least one access record from the same session. However, it is sometimes difficult to determine this kind of session. Here denoting set of session $S_i(i = 0, 1, 2, \ldots, N)$ N is number of session, set of state represent $C_j(j = 0, 1, 2, \ldots, M)$ M is number of state. And access record for session Si represent $S_{ik}(k = 0, 1, 2, \ldots, K)$ K is number of access record for session i then we get Eq.(1) and (2) below

$$S_i = (S_{i0}, S_{i1}, \ldots \ldots, S_{ik}) \tag{1}$$

$$C_i = (S_{0j}, S_{1j}, \ldots \ldots, S_{kj}) \tag{2}$$

3.2 Relevance Similarity

The relevance similarity considers the difference between the state-transfer-probability matrix of CBMG MT_C and the state-transfer-probability matrices of sessions MT_S. It considers the difference between CBMG and sessions of the same cluster. In order to reduce the possible loss incurred by the state replacement should choose access records from the session that are closest to CBMG.

3.3 Topology Similarity

The topology similarity considers the topological relationship between states within a session. The topological relationships represent the dependencies that exist between states. For example, the access operation of some states has to be started after the operation of some other states, and before some other operations. The topology similarity analysis is different from the relevance similarity in that it uses a sequence to analyze matrices other than the transfer matrices.

4. Mathematical model and Algorithm

4.1 Mathematical Model

$S = \{U,M,R\}$
Where :
U : User Access Interface
M : Model Generation
R : Result Evaluation
$U = \{q_0, q_1, q_2\}$

$q_2 \rightarrow q_1 \rightarrow q_0$
Where:
q_0 : Website access by users.
q_1 : Transaction log generated from user access record
q_2 : Save generated Transaction log

$q_1 = \{c_0, c_1, \ldots, c_n\}$
c : State of the session
n : Number of state in session.

$M = \{d_0, d_1, d_2, d_3\}$
$d_1 \rightarrow q_2$
$d_3 \rightarrow d_2$
d_0 : Configure test website
d_1 : Use web log
d_2 : Construct sessions using CBMG
d_3 : Construct sessions using URMG

$R = \{p_0, p_1, p_2, p_3\}$
$p_0 \rightarrow d_3$
p_0 : Use URMG constructed list of sessions
p_1 : Virtual user generation
p_2 : Evaluation execution
p_3 : Result management

4.2 Algorithm

For developing below algorithm steps we used GSP(Generalized Sequential Pattern) Algorithm concepts.

Steps:
1) Assign user access log randomly for different session
2) Retrieve the first common sequence from all user access log
3) Make the combination of possible sequence of different states with other session.
4) Store that combination in matrix of session i MTS_i $=\{UC_p, CC_1, \ldots CC_n\}$, CC is represent combination of states, UC_p is represent set of states randomly selected user

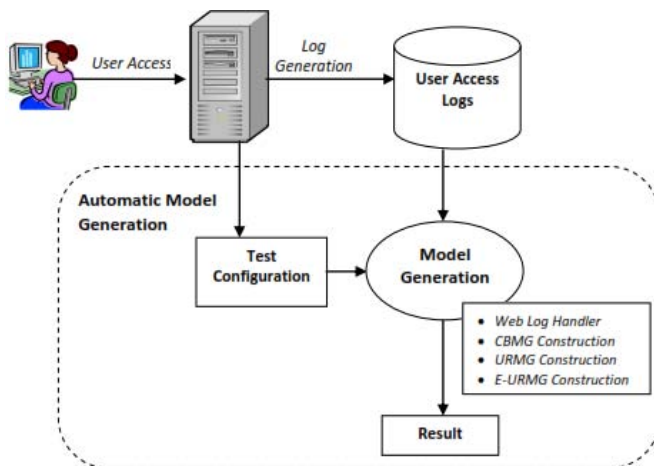# 5. System Design and Implementation

5.1 System Architecture

Figure 1: System Architecture

The system architecture is shown in Figure-5.1. The system consists of three modules as:
1) User Access Interface.
2) Automatic Model Generation
3) Result Evaluation

5.2 System Modules

5.2.1 User Access Interface
User access interface module (UAI) provides an interface to access the image hosting website in server. The user performs some actions like view, purchase, order and comments over the website to generate some transaction logs. The generated logs are stored in the database for result evaluation.

5.2.2 Automatic Model Generation
Automatic Model Generation (AMG) consists of two functions to perform the result generation as described below.

A. Test Configuration
To perform the evaluation we create a "*model.config*" configuration file which allows users to configure the operational parameters, which includes both necessary and unnecessary parameters. The necessary parameters include the URL of the application and the evaluation execution time. The unnecessary parameters include the evaluation mode and the form in which the evaluation results are presented. It will be used during the evaluation execution phase.

B. Model Generation
The model generation module is the processing module of the system, which builds models to create virtual users based on the web logs. It is made up of three sub-modules:

- *Web log handler*
  The web log handler module analyzes the logs and then generates a group of sessions. After all the operations are finished, it passes the sessions to the CBMG construction module.

- *CBMG Construction*
  The CBMG construction module creates the state-transfer-probability and user-thinking-time matrix for each session. It then uses a k-means algorithm to build CBMGs and delivers them to the URMG construction module.

- *URMG Construction*
  The URMG construction module creates URMGs using the CBMGs matrix obtained by the methodology described above, and then transmits them to the evaluation execution module.

- *E-URMG Construction*
  The E-URMG construction module creates enhanced-URMGs to support high number of user sessions and minimize the session error by prioritization the service request as Low (L), Medium (M) and High (H).

5.2.3 Result Evaluation
The result evaluation execution module manages the execution and result collection and processing. It also includes three sub-modules:

- *Virtual user generation*
  The virtual user generation module uses the URMGs to create virtual users, which are a group of scripts used to simulate real user visits. Then, it passes the scripts to the evaluation execution module.
- *Evaluation execution*
  The evaluation execution module uses the scripts and the configuration file to execute the evaluation. It will create several threads to run each single script. During the evaluation, it will collect the outputs and store them in a file where they can be found by the result management module.
- *Result management*
  The result management module automatically performs a statistical analysis on the evaluation results. It then generates a performance evaluation report with tables and charts, compiles all of the results, and sends them back to the users.
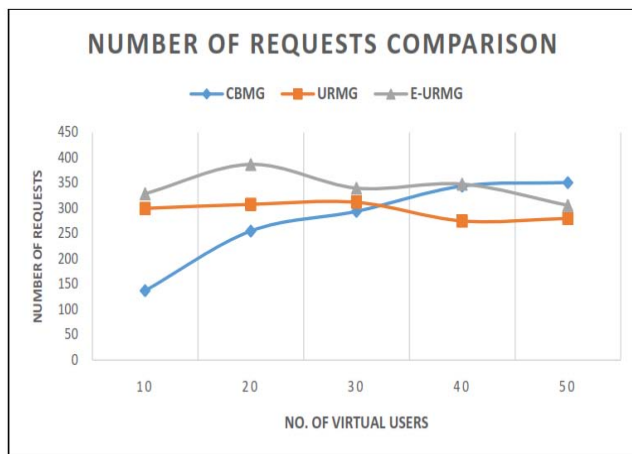
6. Result Evaluation

6.1 Number of User Request Comparison

The result in Figure-2 shown that E-URMG attains to handle more number of requests in compare to number of users in compared to CBMG and URMG. It shows this improvisation due to effectively identification of the users access behavior and prioritizing the access activities.

Table 1: Number of Request Comparison Table

| NUMBER OF REQUESTS			
Virtual Users	CBMG	URMG	E-URMG
10	137	300	329
20	255	308	387
30	294	312	340
40	344	275	348
50	351	280	306

Figure 2: No. of Request Comparison *Vs* No. of Virtual Users

6.2 Error Rate Comparison

The result in Figure-3 shows that E-URMG attains low error rate in compare to number of users in compared to CBMG and URMG. It shows this improvisation due to effectively identification of the users access behavior and controlling the user request based on the server load and prioritizing the access activities.

Table 2: Number of Error Rate Comparison Table

ERROR RATES			
Virtual Users	CBMG	URMG	E-URMG
10	24	20	18
20	23	27	15
30	25	21	19
40	22	18	15
50	25	16	16
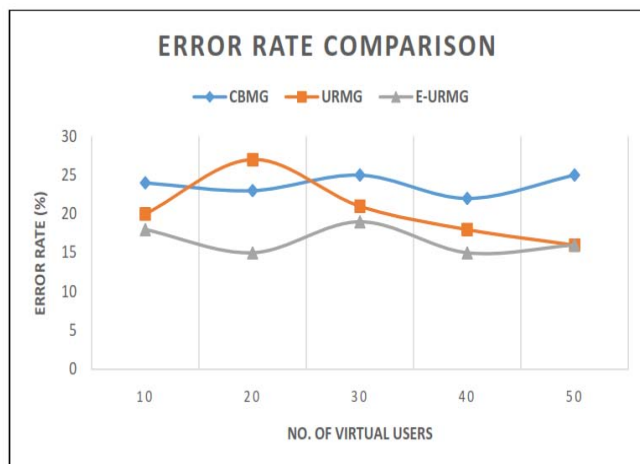
Figure 3: Error Rate Comparison *Vs* No. of Virtual Users

7. Future Work

Future work will concentrate on the optimization of E-URMG. Because the representational session sometimes does not exist, and there may be some complex operations in the session which will affect the URMG generation process, the selection of both the representation session and the access records should be optimized. Besides, the evaluation result evaluation module should be more intelligent, which can both analyze the evaluation results and identify the bottlenecks in the application.

8. Conclusion

We have presented an implementation approach for automatically testing web application testing enhancing URMG representation model as E-URMG, which extends the CBMGs using system logs. Specifically, we used the Apache Tomcat access to generate application access logs for users and to infer the repeatable user behavior, which is transformed into CBMGs. The CBMG output is utilized to build URMG representation model with more accurate access pattern.

E-URMG extends the URMG representation model into categorical access level based representational to control and to handle high number user request with low error rate. The experiment results show that our approach for enhancing URMGs is accurate and can be used to estimate the various system workloads.

References

[1] Xiaolin Xu, Hai Jin, Song Wu, Lixiang Tang, and Yihong Wang, "URMG: Enhanced CBMG-Based Method for Automatically Testing Web Applications in the Cloud", Tsinghua Science and Technology, February 2014, 19(1): 65-75

[2] M. T. Lu and W. L. Yeung, "A framework for effective commercial web application development", Internet Research: Electronic Networking Applications and Policy, vol. 8, pp. 166-173, 1998.

[3] Ricca and P. Tonella, "Analysis and testing of web applications", in Proceedings of the 23rd International Conference on Software Engineering, Los Alamitos, CA, USA, 2001, pp. 25-34.

[4] A. Day, "A model for monitoring web site effectiveness", Internet Research: Electronic Networking Applications and Policy, vol. 7, pp. 109-115, 1997.

[5] W. Lin and Y. Liu, "A novel website structure optimization model for more effective web navigation", in Proceedings of Workshop on Knowledge Discovery and Data Mining (WKDD 2008), Los Alamitos, CA, USA, 2008, pp. 36-41.

[6] K. Mark and L. Csaba, "Analyzing Customer Behavior Model Graph (CBMG) using Markov chains", in Proceedings of 11th International Conference on Intelligent Engineering Systems, Budapest, Hungary, 2007, pp. 71-76.

[7] M. Pinzger and G. Kotsis, "AWPS-simulation based automated web performance analysis and prediction", in Proceedings of Seventh International Conference on the Quantitative Evaluation of Systems (QEST 2010), Los Alamitos, CA, USA, 2010, pp. 191-192.

[8] J. Attenberg, S. Pandey, and T. Suel, "Modeling and predicting user behavior in sponsored search", in Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, Paris, France, 2009, pp. 1067-1075.

994

[9] Y. Bai and C. Cheng, "The performance estimation by queuing network models for a web-based medical information system", in Proceedings of 17th IEEE Symposium on Computer-Based Medical Systems, Bethesda, MD, USA, 2004, pp. 191-196.

[10] Ruffo, R. Schifanella, M. Sereno, and R. Politi,"WALTy: A user behavior tailored tool for evaluating web application performance", in Proceedings of Third IEEE International Symposium on Network Computing and Applications (NCA 2004), Los Alamitos, CA, USA, 2004, pp. 77-86.

[11] A.Vanitha Katherine and K. Alagarsamy, "Conventional Software Testing Vs. Cloud Testing", International Journal Of Scientific & Engineering Research, Volume 3, Issue 9, Spetember-2012.

[12] Eljona Proko, Ilia Ninka, Analysis and Strategy for the Performance Testing in Cloud Computing, Global Journal of Computer Science and Technology Cloud & Distributed, Volume 12 Issue 10 Ver1.0 July 2012.

[13] W. K. Chan, Lijun Mei, Zhenyu Zhang, "Modeling and Testing of Cloud Applications", In 2009 IEEE Asia-Pacific Service Computing Conference (APSCC 2009), Singapore; Dec 7-11, 2009.

[14] Lian Yu, Xiaohu Li, and Zhongjie Li. Testing tasks management in testing cloud environment. In Proc. of the 2011 IEEE 35th Annual Computer Software and Applications Conference, pages 76–85, Washington, DC, USA, 2011.

Author Profile

Mr. Amol K. Rajput received the B.E. degree in Computer Engineering from North Maharashtra university, Jalgaon in 2011 and pursuing M.E. degree in Computer Engineering from Samanvay Pratishthan's Institute of Knowledge College of Engineering, Savitribai Phule Pune University, Pimple Jagtap, Pune, Maharashtra, India in 2015.

Mr. Ajay Kumar Gupta is working as a Assistant Professor in Samanvay Pratishthan's Institute Of Knowledge COE, Savitribai Phule Pune University, Pimple Jagtap, Pune, Maharashtra, India.