Research on Web Application Load Testing Model

Zhu Huaji[1,2], Wu Huarui[1,2]*
1. Beijing Research Center of Intelligent Equipment for Agriculture, Beijing Academy of Agriculture and Forestry Sciences Beijing, China
2. Key Laboratory of Beijing Research Center of Intelligent Equipment for Agriculture Ministry of Agriculture Beijing, China
wuhr@nercita.org.cn, zhuhuaji@126.com
*Corresponding author

***Abstract*—As an important indicator of performance testing in Web applications - load capability is a key factor in determining the performance of Web applications, building load test model is an important prerequisite to obtain load capability of Web application. Based on the user group model and further analysis of the general actual user behavior, this paper increases the parameter personalization function in the user group model. The practical application shows that the improved model can simulate the real user behavior more realistically, and can make the data more statistically significant, so that the performance of web applications can be more accurately predicted.**

Keywords- Web Application; Load capacity; performance testing; Load test Model

I. INTRODUCTION

With the popularity of the Internet, more and more applications based on Web and the amount of data and access to users of Web applications are also increasing, and the system has to face the challenges of performance and reliability. Therefore, performance testing has become an important measure to guarantee the quality of web application system. Load testing, as an important part of performance testing, has a direct impact on the results and quality of performance testing. However, due to the complexity and unpredictable nature of the web application system, load testing is more difficult. If the load tests are not accurate, the test results will be unusable and can even mislead the enterprise to overestimate or underestimate the capabilities of the web application system. In the entire testing process of Web application system, the design of load test occupies a very important position. However, in many web application performance testing projects, because the load model can't match the real environment, a long duration test can exists or the real performance evaluation of the web system can be achieved. Therefore, based on the actual user behavior, a improved user group model is proposed order to better simulate the actual load in the actual situation.

II. THE CONCEPT OF WEB LOAD TESTING

Load testing is testing the response time of the client or server at different loads. Load tests can also be used to help testers calculate the maximum number of transactions that a server can handle within a given period of time. In addition, when the C/S system uses a workload balancing or distributed architecture, the load test should assess whether the load balancing or distributed methods are consistent with the design (Ming et.al. 2010). The essence of load test based on Web application is to evaluate the service quality of the system by simulating the behavior of the actual user. The service quality index of Web based application system has two main components: availability and response time (Song et. al. 2015).

A. Availability

Availability refers to the percentage of time that a web application can provide user access, and the standards for availability are different for different kinds of application systems. For real-time online trading systems, there is a more stringent standard than the general application system. However, availability is always a critical bottom line even in applications with low availability standards. In addition, the demand for availability may vary over different time periods. For example, in the trading peak, online trading system should be 100% available. Unfortunately, some web sites often reject the user's demand and lead to lower availability because they often face with a large number of users access to resources.

B. Response time

In a system environment based on web applications, response time refers to the end-to-end response time, that is to say, the time that the user issues a request to complete the requested HTML page from the web site. The HTML page refers to the base page and does not include other components such as pictures on the page. The end-to-end response time also varies with time and space, and users from different regions receive different response times at different times of the day.

C. Web load test principle

The essence of load testing is to accurately emulate the requests made by the original user through the browser by capturing or writing scripts that rely on the scripts in the tests. The script recorder transforms user requests into actual user session scripts, and then the load generator generates simulated loads on the web application based on scripts and test parameters. At the same time, the server's response is verified in the test client and the system performance

parameters with the real-time monitoring are recorded in the form of data (Chen et. al. 2014).

The function of simulating user behavior is accomplished by a load generator. That is, a request is sent continuously to the web application system through the browser. A new request will be sent after an answer is sent by the web application. A load generator can simulate thousands of concurrent users to test the quality of service of a Web application's system. In load testing, each simulation behavior is called a virtual user, and a virtual user is an instance script.

III. USER GROUP TESTING MODEL

The load test model defines how the test tool submits requests to the web application system, including the number of virtual users to send requests to the web application system, and the speed and frequency which each virtual user sends the request. It is the key to design the load test model by simulating the user's actual behavior as accurately as possible, and the user group model is a common test model. The user group model is modeled by the user's actual actions on the web application. In reality, the user's behavior in a web application can include a series of actions. For example, a web system of online bookstore, after user logins system, possible behaviors including: immediate return, exit or further browse the web view relevant information, registration, and then search books, buy books. From the client's point of view, a user performs all of these actions.

However, in the load test, each specific behavior of all users cannot be simulated in full conformity. When the load test is executed, the virtual test can simulate the same load level as the actual load and does not need to match the virtual tester with the actual user to simulate the specific behavior of each user.

In actual, in the load test, the user's serial behavior can be divided into a number of sub behaviors, and then the virtual tester executes each sub behavior in an appropriate way (Qiao et. al. 2015, Hui et. al. 2011). For example, in the above example, user behavior can be divided into 3 major sub behaviors, i.e. browse information, search books and buy books. The 3 sub actions are then executed in the proper order with 3 different virtual testers. From the server side, the load that the 3 virtual testers load to the server is the same as the actual load applied by the actual user to the server. After the user's behavior patterns are understood, the number or percentage of different users should be known because loads imposed on the system of each type of user's behavior is different. It is necessary to determine the number or percentage of various users in order to accurately simulate the actions of the actual user. The behavior of various combinations of different percentages should satisfy the expected page request distribution. For example, in the above example, there is a big difference between the load of two behaviors of browsing information and buying books. Browsing information may be just a few pure static pages (possibly cached by browsers) that will impose very small loads on the server. But the behavior of the buying books will communicate with the database, and even communicate with other subsystems. The load of the buying books on the server is much larger than the browsing information. Therefore, it is necessary to specify the number or percentage of different users according to the load situation of the Web application in order to accurately simulate the actual load imposed by the user on the server.

In a user group model, a horizontal line is used to represent the user's behavior. It only identifies the behavior of the user and does not include specific steps or pages to complete the behavior. The dotted line is used to indicate the behavior of the user other than the behavior indicated by the horizontal line. The arrow is used to indicate that users are exiting the Web application. Fig. 1 is an example of a user group model for web applications. In the model, multiple user behaviors are included, and the brackets in user's behaviors represent the percentage corresponding to each behavior. The user performs all the branches from left to right, and each branch corresponds to the corresponding percentage of the users

The user group model has some defects in the accuracy of the actual user behavior simulation. The user group model is idealized and ordered in the process of modeling the behavior of the actual user. The behavior of each real user starts with registration and ends with exit. This is not the case in actual.

IV. IMPROVED USER GROUP TESTING MODEL

In the above model, if a user wants to create or register a new user account, he can only fill in the relevant user information before accessing other pages of the web application. In reality, for an initial access to the web application, it is not always an immediate registration or creation of an account. Users often start browsing the page first. Only when they have to create an account to continue accessing the web application, they can register and fill in the relevant user information.

Thus, we should understand that the execution of user behavior should be distributed sequentially over a given time interval and be not distributed sequentially in the order which the individual user completes the task (Deng et. al. 2013). The performance test results of the web applications obtained by above way have statistical significance. The following figure is a further explanation of the concept.

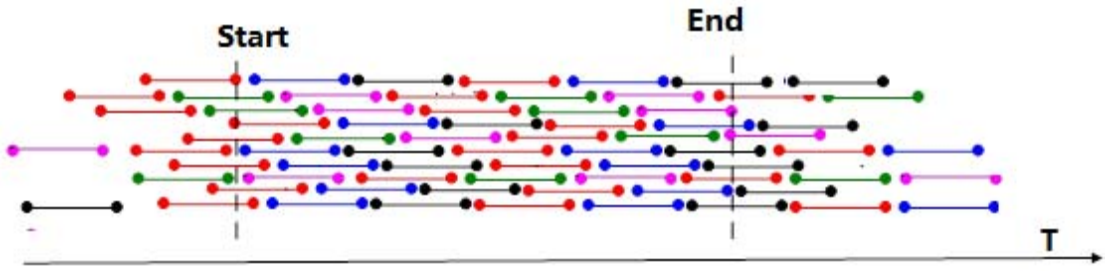

Fig.1. user group testing model

In the above figure, each line represents a behavior of the user and different colors represent different behaviors. In this work, we assume that the red line represents the behavior of the access home, and that the entire horizontal line in the diagram represents a user or a thread. In addition, we assume that each user spends the same time performing the same behavior. Assume that the test model spends one hour from start to end. We'll start to analyze with the web server perspective. In above figure, user 1 access the page behavior including , the first "red", then "blue", "black", "red", "blue" and "black". User 2 also starts with "red", followed by "green", "purple" and so on. We also note that within the 1 hour period, if we make any vertical line in the figure, the line will pass through 10 segments representing the user's behavior, which means that at any point in time, there are 10 concurrent users in the system. In the above figure, there are 17 "red" behaviors, 11 "purple" behaviors, 11 "blue" behaviors and 8 " green " behaviors. For the Web application server, there are only 10 concurrent user behaviors at each point, and the distribution of these behaviors is described in the figure above.

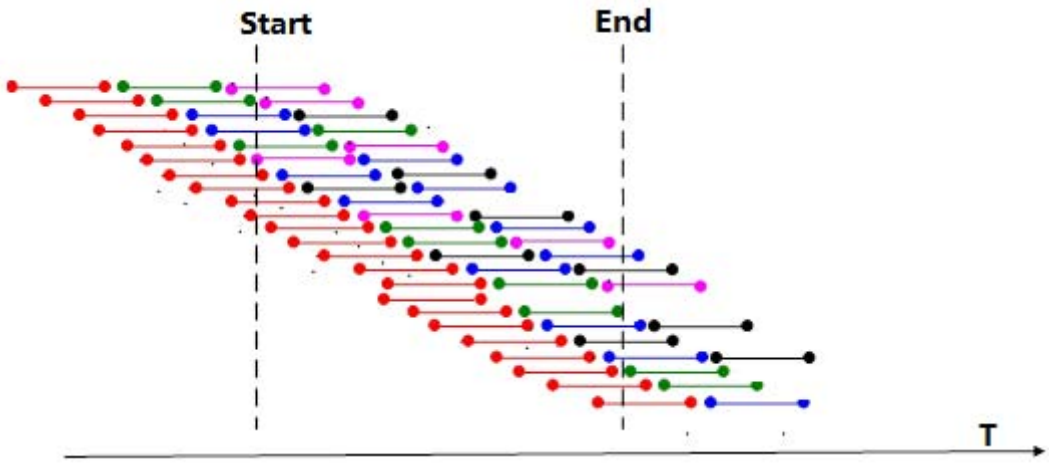

Fig.2. Improved user group testing model

As shown in Fig.2, this diagram describes the behavior of a single individual user. If we look at it from the point of view of the web server, the behavior of the user is shown in Fig. 1. In Fig. 2, the total number of users is 23. Each user has his behavior at the beginning of the model to the end of the entire time period, and each user's behavior starts at a different time.

In addition to start with the "red" represents the behavior, each user behavior and the order of no particular pattern, from the Web server's perspective, the 23 user representatives 10 concurrent users, so we can describe the test parameters, namely per hour to 23 users or 10 concurrent users. If we can overlap two graphs like the above, we may see that the distribution of each user behavior is almost consistent over a period of time. When the number of users increases to tens of thousands, the actual distribution of user behavior may be almost the same as that in the model. Even if there is a difference, there is no statistical significance.

For the model we discussed above, this can be simulated with the following way. First of all, when the occurrence of

the "red" to act, then "blue", "black" behavior and "green", "purple" behavior of the frequency is the same, so we can create 2 scripts, 1 scripts on behalf of "red", "blue", "black", on the other a foot of the "red", "green", "purple" behavior. Then you can design a test scenario to simulate the user behavior distribution within 1 hours, that is, by executing the first script 11 times and second scripts for 8 times. We look at the user group model of online bookstore example now, we assume that the above line represents the "purple" and "registered" this activity, in the actual distribution map, "and purple" always occurs in many different activities, and from the Web server perspective, it always occurs between the "red" and "green" behavior. In fact, for Web servers, there is always no way to distinguish the order in which these behaviors occur. So, when we simulate the user registration behavior, it can be used as a separate script occurs in the beginning of each test scenario, it is because these actions are completed before access to the home, then another script started, so we can design test scenarios easier to accurately simulate the model.

V. CONCLUSION

This work introduces a traditional load test model, and on this basis, according to the actual needs of the test model was further improved, to simulate real user behavior to make data more statistical significance obtained which can accurately predict the performance of web applications. However, in the realistic environment to carry out accurate load test, in addition to the establishment of a more consistent with the actual needs of the test model, must also consider other factors, such as hardware, browser type, user bandwidth and so on, these factors on load test accuracy has an important influence.

ACKNOWLEDGEMENTS

This work was supported by Research and innovation platform construction project of Beijing Academy of Agriculture and Forestry Sciences in 2017 (KYCXPT201702).

REFERENCES

[1] Z. Ming, J.F. Yin, W. Yang, H. Wang and Z.J. Xiao, "A Web Performance Testing Framework and Its Mixed Performance Modeling Process," Journal of Computer Research and Development. 2010, vol. 47(7), pp.1192-1200

[2] W. Song, C.L. Zhang and B.L. Wu, Research and Practice of Web System Performance Testing. Computer Applications and Software, 2015, vol.32(3), pp.4-6.

[3] M.J. Chen, H.Y. Liu, Z.Z. Wang, Y. Wang and S. Xu, "Performance Testing and Optimization of Web Exam System Based on LoadRunner," Chinese Agricultural Science Bulletin, 2014, vol.30(34), pp.250-256.

[4] Z.J. Qiao, J. Tian, "A Web Performance Test Model based on User Access to Features," Intelligent Computer and Applications, 2015, vol. 5(4), pp.5-8.

[5] B.W. Hui, M.R. Chen and D.P. Yang, "Research and application of performance test on web application system," Journal of Computer Applications, 2011, vol.31(7), pp.1769-1772.

[6] L.J. Deng, N.M. Fan, Y.K. Sun, "Research on performance test model of Web application," Computer Engineering and Applications, 2013, vol. 49(1), pp.75-77.