Load Testing for Web Applications

Pu yunming
School of computer engineering, Jimei University
xiamen, china, 361021
yunmingpu@163.com

Xu mingna
School of computer engineering, Jimei University
xiamen, china, 361021
na1202@126.com

***Abstract*: The performance testing criteria was analyzed, including response time, concurrency users, throughout and performance counter. Performance testing is necessary for the system reliability. Load testing can be used for software troubleshooting and optimizing. With the LoadRunner and TestDirector testing tools, a load testing scheme based on an online examination system was designed.**

Keywords: load testing; web applications; automatic testing; system performance

I. INTRODUCTION

Software testing is an important stage in software life cycle, and it is an assurance of software quality. Software testing exists in each stage of software life cycle, and verifies the expected results are achieved or not, correct the bugs as soon as possible. In software development processing, bugs are always existed no matter what technology is adopted. Testing is applied to find bugs, and used to calculate software bugs density [1] [2]. In a typical software project, the percentage of software testing workload is about 40%.

As the web applications become popular, it is an urgent and necessary issue to how to test them. For the WEB applications, the kernel is performance testing, and finds the faults of the software products. In this aspect, load testing is the important evaluation measure. Load testing is due to verify the system performance requirements, and whether to satisfy the load increase. Load testing is used to debug the software exceptions, and optimize the system performance.

II. LOAD TESTING FOR WEB APPLICATIONS

As the network technology development and increase of web applications, users pay more and more attentions to the system performance. Since the web applications mixed lots of technology, such as HTML, Java, JavaScript, Database, network and EJB, result in the testing for web applications becomes more complex and difficulty, so the load testing for web applications is suggested.

The web applications normally have lots of user at the same time, and the employee is always limited within one project team or one company, so the requests of lots of users simultaneously is difficult to test, for the effective testing, the automatic testing system is necessary.

Automatic testing is defined as the management or implementation of each testing activity, which controls the testing execution by software, compares the real output and expected output, constructs the testing condition and testing report. Automatic testing replace the manual testing procedure by testing software, there are many normally web load testing software, such as LoadRunner, webLoad, Etest and OpenSTA, the basically processing is coding and recording the testing script, run the testing script, analyze testing data and fulfill testing report [3].

Load testing is an activity to test system performance in one level of load. The load level can be the amount of web user at one time, or amount of the online processing data, for example, how many user can be online at the same time, and what happened if extra lots of users added.

A. Performance testing criteria

- Response time

Response time is a time defined by start client request and end response from server. Response time is the key software performance, for a web application, page response time is defined by network time (N1+N2+N3+N4) and application time (A1+A2+A3), see Fig.1

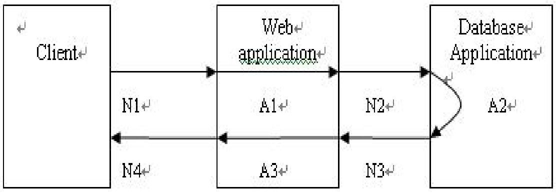


Figure.1 web application pages response time

- Concurrency user

In practical testing, testing engineer pays more attentions to business concurrency users, that is how many concurrency users in the business model is available. In formula (2), the C is the mean concurrency users, n is the amount of login session, L is the mean length of login session, and T is the inspected time. A login session is a time interval defined by a start time and end time.

$$C = n * L / T \qquad (1)$$

$$Cp = C + 3 * \sqrt{C} \qquad (2)$$

In formula (2), if the login session is fit to Poisson distribution, Cp is peak concurrency users. For example, in an OA system, there are about 3000 users, average users who visited this system are 400 every day, for a typical application, the mean time starts from users logs on to the system and ends when users logs out is about 4 hours, users only works 8 hours one day.

So, according to formula (1) and (2), the C and Cp are:

$$C= 400 * 4 / 8 = 200$$

978-0-7695-3887-7/09/$26.00 ©2009 IEEE

$$Cp = 200 + 3 * \sqrt{C} = 242$$

- Throughout

Throughout is the amount of users requests processed within one second. Throughout is the directly load performance, it is normally defined by hits per second or pages per second, there are two aspects role. One is used to design performance testing scenery, verify performance testing scenery achieved test object or not, the other one is to analyze performance bottleneck, the limit of throughout is the mainly aspect of performance bottleneck. Throughout is related to concurrency users when no bottleneck happened, defined as:

$$F = Nvu * R / T \qquad (3)$$

In formula (3), F is throughout, Nvu is the amount of virtue users, and R is the hits sent by virtual user (VU), T is the performance testing time.

- Performance counter

Performance counter is used to describe the performance of web server or OS, such as Processor Time, Memory Available Megabyte, Physical disk Time, Successful or Failed Hits, Hits Per Second, Attempted Connections, User 0 Connections, Number of deadlocks, Buffer Cache hit [2].

B. *The evaluation and choice of testing tools*

The followings are to be considered in the evaluation and choice of software testing tools:

- Functions, evaluate the functional conformity of different tools, define the priority and weight of each functions, compare each testing tools using the method of weighting scores,
- Product support capability, including the support capability of the product from original development company or dealer, the average replacement cycle of the product, whether the support is convenient, the pattern and timeliness of the support,
- Company reputations, this information mainly acquired from customer.

TABLE I. LOAD TESTING SOFTWARE

Software	Protocol	Monitoring resource
LoadRunner V8.0	*Windows Sockets, SOAP, Web, RMI, IMAP, POP3, SMTP, WAP etc.*	*Web connection and throughout capacity, HTTP response time, Virtual user Hits, Client Request per second, CSPS, RP, RET, RWT, RQ and % Processor Time etc.*
Rational TestManager	*HTTP, Socket, TUXEDO, IIOP, Flow Control etc.*	
QALoad V 5.1	*HTTP, SSL, IIOP, FTP, SMTP, POP3, WAP, PeopleSoft, Windows Sockets, RMI/EJB etc.*	
WAS	*Only for Web protocol.*	

Basically, the load testing can not be done manually, must use testing software. Table 1 is the usually tools in software company, listed the protocol and monitoring resources in load testing [2] [4].

C. *Load Runner for web application*

LoadRunner of MI is a standardize load testing tools, validate and find problem within system through virtual user and concurrency, shorten the testing time, optimize system performance, and deploy the web applications. There are commonly components in LoadRunner, such as Virtual User Generator (VuGen), Controller and Analysis. The procedure of load testing for web applications includes load testing plan, testing script development, execution scenery creation, testing, scenery monitoring and testing result analyses. The VuGen component is for script development, creating virtual user activity, node definition, transaction definition. Controller component creates testing scenery, run VU list and execute testing script list. The Analysis component analyzed the testing result [5].

III. LOAD TESTING FOR ONLINE EXAMINATION SYSTEM

A. *Experimental scheme*

LoadRunner is applied for load testing of one online examination. The web server and database server is physical independent, system must satisfy lots of users login on the same time, load testing is used to find the performance bottleneck, verify the system whether meet the need of performance.

- The tested system is an online examination system, the maximum users are 2000, the client pc installed windows XP professional, 512M ram and IE6.0, examination system is developed with MyEclips and SQL Server 2000, Web Server installed Tomcat 5.0.
- The tested system is a web system, so the network bandwidth, concurrency and server response time is the important performance to be monitored. The peak time of the web server is when the large users log into the server at the same time. The main monitoring objects include the end user response time, reliability, system capacity and System bottleneck.

B. *Web application monitoring*

Create Vuser to measure the user response time, create a typically script and execute, record the login script, set up virtual users and simulate the multiple user login at the same time, according the online examination system, set up the maximum user 2000.

- Transaction monitoring
- Network monitoring
- Server resources monitoring
- Web resources monitoring

C. *Experimental environment setup*

When begin a new testing project, first setup project materials, create user group or role, grant priority to user group, define testing requirement, create load testing, crate testing set and plan, then start application testing. Testing elements for applications are described in requirement document. The definition is conductive to testing plan and management. Requirement and testing are related with bugs, so testing procedure can be traced, helped to make decision in testing.

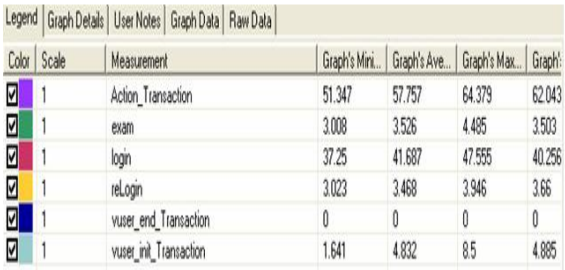
In testing procedure, setup testing date, time and condition. Condition is another testing result based on execution procedure. TD (test director) can setup testing sequence or postpone current execution until the complement of another defined testing. LoadRunner and TD can store, index and collect testing result effectively.

IV. Experimental Analysis

Study the mean transaction response time and performance summary from the analysis of experimental scheme. Determine the maximum long transaction response time, analyze when the maximum value occurred, in the some points or all execution process, then solve the problem and get the optimal performance.

A. *Transaction performance analysis*

Transaction performance summary describe the shortest response time, mean response time and longest response time of each transaction during action execution, decide the transaction which has longer response time. In Fig.2, the action is described for each transaction every minutes, the mean transaction response time is recorded. In experimental result, the mean response time of action transaction is about 57.575 seconds, and the mean response time of user login is about 41.687 seconds. The login transaction and action transaction have the longer response time, and the response time increased obviously when the user increased.



Color	Scale	Measurement	Graph's Mini...	Graph's Ave...	Graph's Max...	Graph'
☑	1	Action_Transaction	51.347	57.757	64.379	62.043
☑	1	exam	3.008	3.526	4.485	3.503
☑	1	login	37.25	41.687	47.555	40.256
☑	1	reLogin	3.023	3.468	3.946	3.66
☑	1	vuser_end_Transaction	0	0	0	0
☑	1	vuser_init_Transaction	1.641	4.832	8.5	4.885

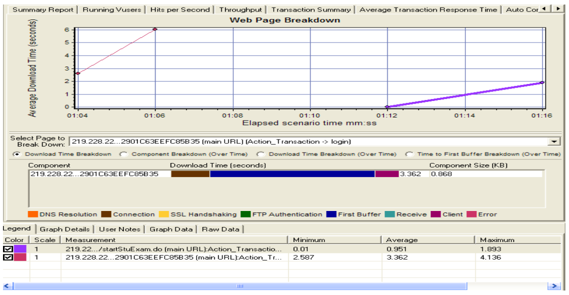
Figure 2. Mean transaction response time

For the analysis of each transaction, and know the reason why the login transaction and examination page need longer response time, so the transaction need to be separated and analyze the performance of each page component.

B. *Ttransaction separation*

Analyze the mean time from the client request to server response, determined the distribution of each component response time, analyze the holding time of server and network transportation [6] [7]. The following is the procedure from client request to server response step by step.

- When client send a request to web server, the request is sent to DNS server firstly, and analyze the client IP address. The analysis time is called DNS resolution, and this metric time can determine whether the DNS configuration is set up correctly. The DNS resolution is small if the DNS server runs correctly.
- When the IP address is analyzed, the request is sent to web server, the initial link between the client and server is connected, this is called connection. The metric time is used to identify whether the network and server response to the client request, the value of connection is small normally.
- After the connection is setup, when the first data packet from web server is sent back to client through network, the transportation time is called first buffer, this metric time is related to the delay of server and network response time.
- The time starts from the client received first byte and ends received last byte successfully, this time is called Receive time, received speed RCVT=size/time, RCVT can be used to identify network quality.


Summary Report | Running Vusers | Hits per Second | Throughput | Transaction Summary | Average Transaction Response Time | Auto Cor
Web Page Breakdown
Average Download Time (seconds)
0
1
2
3
4
5
6
01:04
01:06
01:08
01:10
01:12
01:14
01:16
Elapsed scenario time mm:ss
DNS Resolution
Connection
SSL Handshaking
FTP Authentication
First Buffer
Receive
Client
Error
Legend | Graph Details | User Notes | Graph Data | Raw Data

Select Page to Break Down: 219.228.22...2901C63EEFC85B35 (main URL) (Action_Transaction -> login)

● Download Time Breakdown ○ Component Breakdown (Over Time) ○ Download Time Breakdown (Over Time) ○ Time to First Buffer Breakdown (Over Time)

Component	Download Time (seconds)	Component Size (KB)
219.228.22...2901C63EEFC85B35	3.362	0.868

Color	Scale	Measurement	Minimum	Average	Maximum
☑	1	219.22.../startStuExam.do (main URL):Action_Transactio...	0.01	0.951	1.893
☑	1	219.228.22...2901C63EEFC85B35 (main URL):Action_Tr...	2.587	3.362	4.136

Figure 3. Transaction Separation

In Fig.3, the execution time of each component is displayed, and the First Buffer of component is longer, but can't determine it comes from server or network, so need to separate the detailed figure of server and network. In Fig.4, the server time is much more than the network time, so the problem is from server.

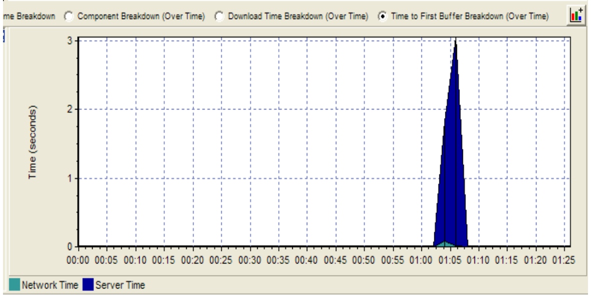


Figure 4. Server and Network Separation

C. *Load testing procedure*

By the load testing based on the online examination, the technological process of the load testing for the web application can be concluded. Simulating large scale users for the load testing simultaneously, determined the main factors in web system, shorten the testing time and releasing time extremely.

- When testing a new project, firstly setup the project database in TD, create new user group and grant the user priority, user group includes Developer, Project Manager, QATester, TD Admin and Viewer.
- After setup the project and user group, define testing requirements, testing plan and starts load testing. Analyzed and finished the testing report.
- In testing, setup testing time and condition, delay or advance the currently testing procedure according to the testing requirement changes.

V. CONCLUSION

Application must implement the performance testing, find the potential bugs, analyze and locate the software defects. Load testing is the kernel of performance testing, the performance criteria and testing tools selection is the key too. TD can control the whole testing procedure, create the testing infrastructure and improve the testing management effectively. Within the performance testing of the online examination system, analyzes the system bottleneck, concludes the basically testing procedure of load testing for the trusted web application. In the other hand, the testing data analysis and system bottleneck detected need accumulation of experience and technology. Based on the examination system, fulfill the testing object, plan, execute and analyze the load testing for the web applications, improve the system performance. In this project, the other problems are appeared, such as the bottleneck located intensively, performance analysis in detailed and testing management optimized, these problems should be studied in the future researching work.

ACKNOWLEDGEMENT

The author would like to thank for the funding and support from the science project of fujian province, china (grant No. JA07133), xiamen science bureau (grant No.3502Z20083025) support.

REFERENCES

[1] Paul C.Jorgensen. Software Testing: A Craftsman Approach [M].CRC Press，2002.

[2] Duan Nian, Software performance testing and practical [M]. Tsinghua university Press, 2006.

[3] LoadRunner 8.0 user Manual, Mercury Interactive,2005.

[4] Connie U.Smith, Lioyed G.Williams. Performance solutions, a practical guide to creating responsive scalable software [M]. China Machine Press,2003.

[5] John D. Musa. Software Reliability Engineering[M]. McGraw-Hill Education, 1999.

[6] C Kallepalli, J Tian. Measuring and modeling usage and reliability for statistical web testing [J], IEEE Trans on Software Engineering, 2001,27(11):1023-1036.

[7] Deng Xiaopeng, etc. Progress in Testing for web Applications [J], Journal of Computer Research and Development, 2007, 44(8):1273-1283