Research the performance testing and performance improvement strategy in web application

Kunhua Zhu Junhui Fu Yancui Li
School of Information Engineering
Henan Institute of Science and Technology
Xinxiang ,Henan Province 453003,China
zwkh100@163.com

Abstract— **With the web services used widely in all aspects of social life, the web application performance testing is gaining wide attention. In the paper, we firstly analyses and research the types, indicators and testing methods of the performance testing of the web, and then we put forward some testing process and methods to optimize the strategy.**

Keywords- web performance testing; test type; load; test method

I. INTRODUCTION

Today many companies, enterprises and Web sites set up Web-based applications, Web scale has been expanded, and more and more people pay their attention to how to ensure the accuracy and reliability of Web applications. Performance testing is a process of information collection and analysis of collected data used to predict how the load level will be run out of system resources. It studies Web's response to user's requests under different user load. so as to ensure the future security of system operation, reliability and efficiency in the implementation. The goal of performance testing by simulating the true load is to identify performance bottlenecks, optimize system performance so as to ensure the practical operation of the program can provide a good and reliable performance.

II. TESTING OF WEB PERFORMANCES

A.. Web performance testing types

We usually associated Web testing with black box testing. Web performance testing is an imitation of the end-users of the tested system, by recording and describing the real user's behavior, using an automated, controlled way to repeat the implementation of these user's behaviors.

Because it is self-executing, the system can simulate the high-traffic user's behavior. Test systems mainly through the test generator (computer) to simulate the user's events. Figure 1 schematic diagram for performance testing.

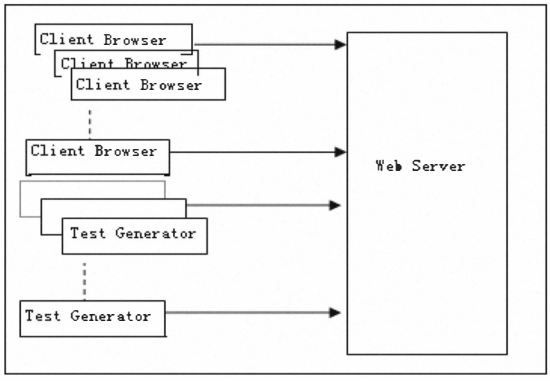

Figure 1 Performance Diagram

A test generator can typically simulate user's behavior running tens to hundreds of Web client software. Virtual users and Web servers communicate directly without having to use the Web-browser (such as IE or FireFox). In the performance tests, the running and testing number of the virtual users can be set in the generator. If more virtual users need to be simulated, it can be connected with multiple load generator, and centralized control, thereby generating a flow close to the limit. In addition, delay time can also be set between the acts of two tests.

Web Performance testing generally can be divided into three types: stress testing, load testing and strength test. Each test using the same script, testing tool, and environment, but different testing time intervals.

(1) Stress test

Pressure testing is done by gradually increasing the system load, testing the changes in system performance, and ultimately determine the system performance under any load conditions in a failure state, and thus to get the system can provide the greatest level of service tests. In order to discover the conditions which the application performance will become unacceptable, mainly by changing the application's input to the corresponding procedures imposed by an increasing load, until the discovery of the inflection point decline in application performance, to identify bottlenecks in a system and performance point that can not be received, so as to get the maximum service level test that system can provide.

Stress tests examine the current hardware and software environment the system can withstand the maximum load and to help identify system bottlenecks.

978-1-4244-6370-1/$26.00 © 2010 IEEE

(2) Loading test

Loading test is done by gradually increasing the load, testing the changes in system performance, and ultimately determine the maximum load test the system can withstand, in meeting the performance indicators. Through the load testing, to determine system performance in a variety of work load, so as to test the changes of the various system performance indicators when the load increased gradually. Load testing is usually to describe a specific type of stress testing - to increase the number of users to stress test the application.

Load test begins actually from the relatively small and gradually increasing the number of simulated users until the application response time overtime, that is the load test. Load testing and stress testing may be in conjunction.

(3) Strength test

Strength test is a longer interval load test or stress test. Unlike other tests is that the weight-bearing or tension testing interval of only tens of seconds to maintain the strength test should be delayed a few hours or even days. Strength testing often finds some inexplicable errors. For example, memory leaks, that is, memories, rollback segments exist in the database transaction were not submitted, or have a cumulative impact on system resources, errors and so on.

B. The general indicators of performance testing

Performance testing through automated testing tools simulates a variety of normal and abnormal peak load conditions of the system of performance indicators for testing. Performance model provides measurable standards of performance, while the standard is constituted by a series of performance indicators. Typical performance metrics are Response Time, System Throughput, System Resource Utilization, the Number of Concurrent Users, HTTP Transactions / sec and the Number of Sessions / sec, Network Traffic Statistics, Resource Request Queue Length and other indicators to measure Web performance. The following discussion focused on the first three indicators.

(1)Response time

Response time is also known as the waiting time from the user point of view. It is from the client sends a request to receive the server's response to the delay experienced, usually measured in time units. In general, it is as the low-level user load increases slowly increasing, but once the system of one or a few resources have been exhausted, waiting time will be rapidly increased. Figure 2 shows the response time with the relationship between the user loads.

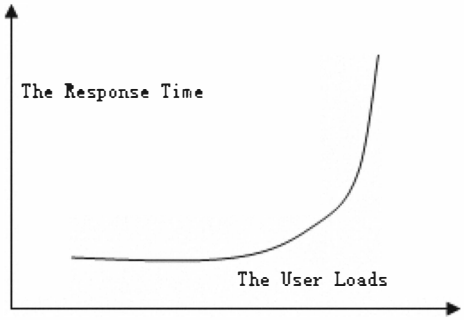

Figure 2 the Response Time with the Relationship between the User Loads

In figure2, a sudden increase in response time is often caused by one or a variety of the system resources achieve the maximum utilization. For example , to configure a Web server to use a fixed thread handle concurrent user requests, and when the number of concurrent requests exceeds the number of threads the server can afford, any incoming request will be placed in the request queue and wait for processing. Waiting in the queue at any time will naturally be added to the waiting time to go.

The waiting time period is divided into many small fragments and these fragments are divided into two main types: network latency and application latency. Network latency refers to data from one server to another server, it takes time. Application wait time is the data processed within a server consumes time. Figure 3 shows a full course of a typical web request processing the different waiting times.

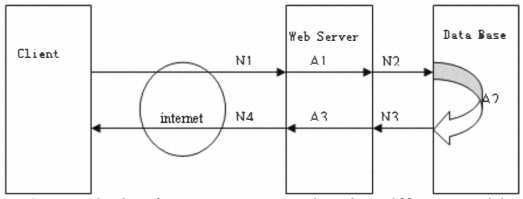

Figure 3 A Typical web Request Processing the Different Waiting Times

Shown in Figure 3: All waiting time (response time) = (N1 + N2 + N3 + N4) + (A1 + A2 + A3) where Nx on behalf of network latency,Ax on behalf of application wait time. Response time is usually the major decision by the N1 and N4, the waiting time represents the way customers access to Internet. To reduce the waiting time (N1 and N4), a common solution is to put Web server or Web application content as much as possible be placed close to the customer's location, which can be set up by the nearest server, or in some of the major Internet hosting provider's site done to achieve the mirror sites to reduce the client to the server network route between the forwarding time.

Network latency N2 and N3 are often dependent on the performance of the server switching equipment. When the back-end database, traffic increases, you can consider upgrading the exchange settings and network adapters to improve performance. To reduce the waiting time applications (A1, A2 and A3) is more difficult because of the complexity of the server application software will allow analysis of Performance Data and Performance Tuning has become very complicated. For example, multiple software

components on the server interaction for a particular service request, waiting time may be caused by any one of these components in the production. Thus as follows to solve this problem:

1) Applications should be designed to minimize round trip as far as possible, thereby reducing transmission time and resources to the request wait time, the ideal situation is to use a round trip.

2) Can be optimized to improve performance of many server components, database adjustment is one of the most noteworthy of the links, stored procedures and indexes can be optimized.

3) To find and remove competition bottleneck caused by public resources competition between threads and components.

4) In order to enhance the capacity of the system, you can upgrade the server hardware (by percentage increase). If not enough system resources to become a bottleneck, the use of multi-server as a cluster can ease the load on a single server so that it can improve the performance of the system, thereby reducing application latency.

(2) Throughput

Throughput refers to the number of user requests that system handled within a particular unit of time .Commonly used unit is the number of requests / second, or the number of pages / sec. From a market point of view, throughput can be the number of visitors per day or daily page visits to measure. As one of the most useful performance indicators, web applications, throughput is often in the design, development and release at different stages throughout the cycle measurement and analysis. For example, in capacity planning stage,, the throughput is the key parameters to determine the hardware and system requirements of the web site.In addition, the throughput in identifying performance bottlenecks and improves application and system performance, also play an important role. Whether web platform is to use a single server or multiple servers, throughput statistics indicate that the system for different user load levels be reflected by similar characteristics. Figure 4 shows the throughput with the user load between the typical characteristic curves.

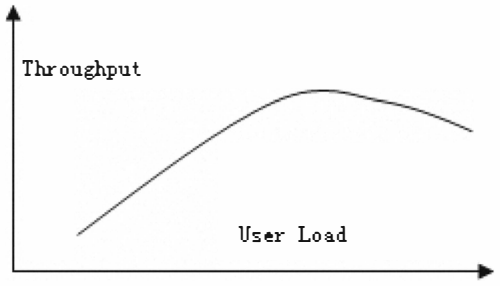

Figure 4 the Throughput with the User Load between the Typical Characteristic Curves

In the initial stage, the system throughput and user load proportionally increase in cases, However,due to system resource constraints, throughput can not be infinitely increased. It will gradually reach a peak, and then the whole system performance will decrease as the load increases. Maximum throughput is the figure the peak point is the system in a given unit of time be able to handle the maximum concurrent number of user requests. In many ways, the throughput is associated with the response time, and that they are different ways to consider the same issue. In general, site has a longer response time would have smaller throughput. At the same time, if throughput measurement without regard to waiting time is a misconception because the waiting time is often at the peak prior to the user throughput, loaded will suddenly increase. This means that the availability from the application point of view, the peak throughput will appear on the unacceptable response time.

(3) Resource utilization

Utilization refers to the systematic usage of different resources, such as the server's CPU, memory, network bandwidth and so on. It is often used to account for the largest amount of resources, measured as a percentage. Figure 5 shows the resource utilization with the relationship between the characteristics of the user load.

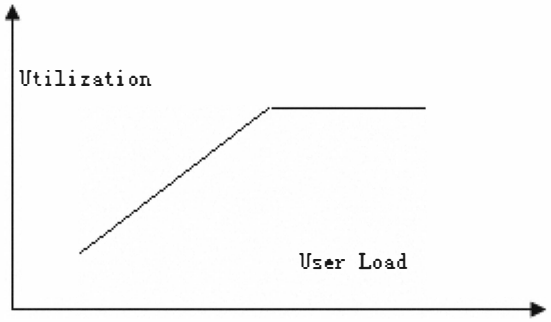

Figure 5 the Utilization with the Relationship between the Characteristics of the User Load

Figure 5 shows the utilization rate is usually directly proportional to the load cases with the user, but when reached a certain amount of number, as the continued growth in subscribers ,utilization will remain a constant value, here already reached the maximum availability of resources. When resources are maintained at a constant value of 100%, then the resource has become the system bottleneck. To enhance the capacity of such resources can increase the system throughput and shorten waiting times.

C. Web application performance test methods

For different application types and focus, there are mainly three kinds of Web Performance Testing Methods at present: The virtual user method, WUS method and object-driven approach.

(1) The method of virtual users

By simulating the behavior of real users of the tested program (application under test, AUT) applied load, to measure the performance of AUT index value, such as the transaction response time, server throughput and so on.It is based on real user's "business process" (the user for the completion of a commercial business and the

implementation of a series of operations) as the basic unit of the load, with a "virtual user" (simulated user behavior test script) to simulate real user. The load demand, such as the number of concurrent virtual users, implementation frequency of business processes obtained through the manual collection and analysis of systems information. Some load testing tools to support the methods ,you can use less hardware resources to simulate thousands of virtual users access AUT simultaneously, and simulate requests from different IP addresses, different browsers and different types of network connection, it also can monitor real-time system performance indicators to help testers analyze test results.

By simulating real users access to the system can help the system analyst find bottlenecks, optimize the system hardware and software configurations in the design phase of Web application systems. This method has sophisticated tools to support, more intuitive, suitable for e-commerce applications in test, but to determine the load of information relys on manual collection, accuracy is not high.

(2) WUS method

The concept of based on the "website usage signature (WUS)" designed test scenarios, emphasizing the establishment of a true load. The proposal of WUS is to measure the proximity between the test load and the actual load, which can fully characterize a series of load parameters and a collection of measure, including the number of pages viewed per hour / clicks, average visit duration, each visits average number of pages viewed / clicks, and page request distribution and so on. These parameter values can be obtained from the log file. Frequently accessed path is as constituent units of the load. This method is that only when the test of WUS and practical application is consistent, the test is valid. This method has the advantage of testing the operation of the load from the actual site data, so they can reflect and represent the actual load. Defect is too dependent on the log file, applicable to test newly developed procedures.

(3) Object-driven method

The basic idea is the act of the AUT can be broken down into testable objects. Objects can be links, command buttons, list boxes, messages, images, and downloadable files, audio and so on. Object-defined granularity depends on the application complexity. A Web page can use objects to recursion, performance, thus becomes the process of testing to test each object or a collection of some objects, these objects act as the load of the constituent units. The method by AUT decomposed into objects, so that test structure, a high degree of reusability is good, the results clearly, the page component type for rich, complex business applications; but too much emphasis on the performance of local components is difficult to reflect user's actual experience to the performance .

III. TEST PERFORMANCE OPTIMIZATION STRATEGIES

A. Local server tests

Local server uses apache combination and PHP, and now the server simulating user, respectively 50,100,150,200,250,300,350 analog user response time statistics. Statistical results shown in Figure 6:

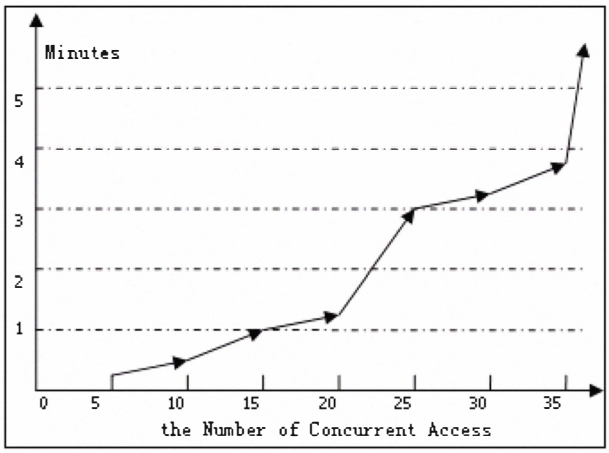

Figure 6 the Local Server Test

As can be seen from figure 6, the number of users from 200 to 250 when there is a mutation, and more than 350 test users when the maximum response time of more than 10,then the user will feel that the system becomes very slow. So the local server would be able to withstand more than 300 individuals.

B. Remote server tests

On the remote server (such as www.baidu.com) test, the test results are unstable, errors are varying, but the overall results and the figure is almost, but www.baidu.com can withstand about 200, Upper 200, the response time becomes very long.

C. Optimization

(1) Analysis

Web server response is too long for many reasons, from the top local testing and remote testing can be seen in the local test, the local server's hardware determines whether the stable and robust Web services. In the remote test, the network plays a decisive role in the situation; it's also an important way to improve its Web server's services if it has sufficient network bandwidth, when the local server hardware is ensured in a good situation.

In addition to optimizing the hardware, the server-side software on the optimization space is very large, for the existence of the database server to optimize database operations can greatly improve the query efficiency. Optimization of the database can expand the database of basic table (for table horizontal partition and vertical partition) and create the index and so on..

(2) Optimization strategy

1) Using polling mechanism. In order to reduce the impact of interrupts on system performance, in the load under normal circumstances a "second half of the treatment" method is very effective, while in high load condition, using this method will still result in Livelock phenomenon and then polling mechanism can be used. Although this method would cause waste of resources and response speed decreased when the load under normal conditions, but the

data in the network to reach the server frequently is more effective than interrupt-driven technology.

2) Reduce context switching. Reduce context switching. This approach regardless of the circumstances under which the server performance improvements are effective, then can introduce the core-level (kerne1-leve1) or hardware-level data flow approach to achieve this objective Core-level data flow is the data from the source transmitted through the system bus without making the data through the application process, the process, because the data in memory, requiring CPU operates data.

3) Hardware-level data stream from the source through the malpractices of private data bus, such as DMA, or while carried forward through the system bus without the need to make the data through the application process, this process does not require CPU operates data. So that the data transfer process does not require the involvement of the user thread, reducing the number of times data is copied and the context switching overhead.

4) To reduce the frequency of interruption (mainly for high-load situation). Here there are two main methods: batch suspension and the temporary closure of disruption. Batch interrupts can effectively inhibited the phenomenon of live lock when it is overloaded, but the performance of the server doesn't have any fundamental improvement; when the system appears to receive live-lock signs, you can use the temporary closure of disruption methods to ease the burden on the system, when the system cache is available again, you can then open the interruption, but this method is not big enough buffer to receive the case will cause packet loss.

IV. CONCLUSIONS

The diversity and complexity of the Web technology is determined difficulty of its tests. In order to ensure the quality of Web applications, gives it better reliability, security, and adaptability, the text from the types of performance testing, test targets, test methods, and several other aspects to analyses and research the Web server performance tests, and put forward several optimization strategies, which can more accurately identify problems and optimize system performance for users planning the configuration of the entire operating environment to provide a basis.

REFERENCES

[1]Internet Testing: Keeping Bugs Off the Web. http://www.stlabs.com/bugs_off.htm

[2] J. Bach. Testing Internet Software. http://www.stlabs.com/testnet/docs/inet.htm

[3] Tongren Compatibility and Security Testing of Web-Based Applications. TTN Online，1999，3

[4]Hypertext Transfer Protocol--HTTP/1.1 〔EB/OL〕.http://www.w3.org/Protocols/

[5]Rick Stout WORLD WIDE WEB Reference Handbook〔M〕.Bei-jing:OceanPress,1996

[6] YuLi,ShiBing-xin, LinQi-wen.A Performance analysis model for WWW server system〔J〕.Journal of Huazhong University of Science and Technology.199927(4):28～30

[7]angHui,ChenXin-meng.PerformanceanalysisoftheinternetnetworkingcomputingbasedonWWWJ〕.ComputerEngineer-ingandApplications.200137(14):72～73