# Research and realization on the performance testing tool of Web application

Huarui Wu[1,2,3], Huaji Zhu[1,2,3(✉)]

[1]*Beijing Research Center of Intelligent Equipment for Agriculture, 100097 China;*

[2]*National Research Center of Intelligent Equipment for Agriculture, 100097 China;*

[3]*Key Laboratory of Beijing Research Center of Intelligent Equipment for Agriculture Ministry of Agriculture*

whr1998@126.com

zhuhj@nercita.org.cn

**Abstract:** The performance guidelines such as the response time and the connection unsuccessful rate are the key factors to determine the quality of a Web application. To calculate these performance guidelines quickly, show result to users with visual way and reduce the cost of testing tools, a performance testing tool ASTD aiming at web application is developed. ASTD provides many functions such as testing plan management, testing plan execution and testing report producing. Based on simulating users' behavior, an E-government system developed for Fangshan is tested by the ASTD. By analyzing the test results, the bottlenecks of the system were found and the performance of this system was optimized.

**Keywords:** Web application· performance testing· testing plan· testing report

## 1 Introduction

With the rapid development of Internet and e-commerce, the web application system based on B/S structure appears in large numbers. In these Web applications, only simple functions, such as browse, query, form input are completed in clients(Alberto and Elaine, 2004.). Most of the computing work is done by server-side applications, which makes the server application a heavy load. If the server application can respond to concurrent access by a large number of users, this requires the web application system to have very high performance (Subraya and Subrahmanya, 2000). But in fact many web application systems have not been rigorously tested. When excessive user access to web application in peak time, will cause the Web server unable to timely respond to user requests, and even make the system collapse and paralysis, bring great economic losses.

To avoid this, a test tool for web application is needed in order to simulate user access to web application (Ye and Feng, 2002; Tan et. Al. 2006.). When the Web application in the design and development stage, we use the test tool to maximize the analog access pressure monitoring system and real user concurrency, performance index, system performance and timely detection of hidden defects, quantitative

analysis of the capacity of the system in order to take the corresponding improved measures and countermeasures (Li and Wang 2005).

In order to overcome the limitations such as expensive and high learning curve of existing commercial performance testing software, such as LoadRunner of mercury and QALoad of Compuware company. A set of web application performance test tool ASTD is designed and implemented. In this work, the structure and the specific implementation of each module of the tool is introduced. With a e-government system of a town in Fangshan as the test object, simulating the operation of a large number of users with ASTD in order to test the number of concurrent users which the system can withstand and find out the bottleneck problems existing in the system. These test results can provide a reference for further optimization of the system. The results show that ASTD can performance index calculation software system quickly and effectively.

## 2 Web application performance testing

Web application performance testing is a class of tests that describe and evaluate Web applications. The Web application performance test mainly tests whether the software meets the performance specifications specified in the requirements specification and satisfies some performance related constraints and restrictions.

Web application performance testing includes two aspects: load test and stress test. Load tests are designed to determine the performance of the load system at various levels. The objective is to test the corresponding output items of the system components when the load increases gradually, such as response time, connection failure rate, CPU load, memory usage, etc., to determine system performance. The stress test is a test to determine the bottleneck of the Web application system or the ultimate performance point that can be received, so as to obtain the maximum service level test provided by the system.

## 3 Design and realization of the performance testing tool ASTD

### 3.1 Module structure design

Based on the analysis and research of Web application performance test, the Web application performance testing tool ASTD is designed and implemented the structure is showed in Figure 1. The tool is composed of three parts, the test plan management module, the test plan execution module and the test report generation module. The specific instructions for each module are as follows:

(1) The test plan management module

The test plan management module describes the policies and methods used to plan and organize test activities, determine the test object, request parameters, load parameters, expected results, and so on. The test plan will ensure that the testing process is carried out in an all-round way. According to the test plan, the test objectives are descripted

with the specific target. Around the test target, the test object is implemented with the specific test cases and the test scripts.

(2) The test plan execution module

The test plan execution module is the core function module of the tool. In a specific test scenario and the specific time, the test plan execution module implements the load test and stress test by simulating a number of real users to request the test system. After running, the test results are passed to the test report generation module.

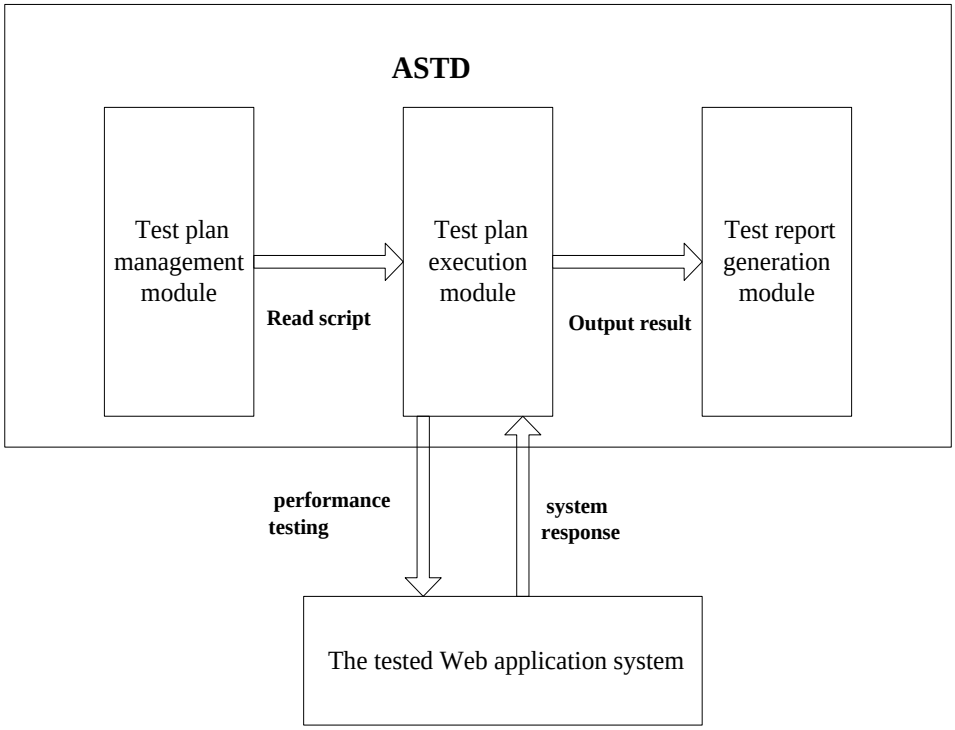



Figure.1. Structure of the ASTD

(3) The test report generation module

After the test result is obtained, the test report generation module compares the test result with the expected result, gives the measurement result analysis and makes the statistical chart, and finally generates the test report. The test report shows the performance and behavior of the tested system in the form of diagrams to analyze the location performance bottleneck problem.

### 3.2 Implementation of the module function

(1) Implementation of the Test plan module

1) Systems analysis

The system's performance metrics can be turned into specific goals for performance testing. In the above step, the testers required to have a comprehensive and systematic grasp of the structure and implementation of the tested web system. Through the analysis of the software and hardware architecture of the tested system, and combining

with the user performance index, a specific and targeted performance test implementation scheme is worked out. In a common web application system, browsing and querying services are the most frequently used services by users, so browsing and querying should be tested as typical services.

2) Request parameter configuration

After determining the typical business, you need to configure the parameters of the request. Including: measured server IP address, port number, relative to the service path, the method of parameter submission (POST or GET), the parameters needed to pass. The test execution phase sends requests to the specified page based on the request parameter configuration information.

3) Load parameter configuration

The load parameters required configuration by user include: the number of users who send requests 'UN', the time of send request 'RT', the number of repetitions of the request 'LN'. In the process of the test, the test tool opens UN threads and cycle LN times to request access to the system under test with the RT time.

4) Expected data entry

The expected data is the user's expectation of the performance index of the tested web system. The parameters of the performance index includes Maximum response time, Mean response time, Connection failure rate, throughput and so on. After the test is completed, the data comparator of the test report generation module compares the expected data and the test result data.

5) Generating scripts

The ASTD stores test requests, request parameters, load parameters, and expected data in the XML file of the same name as the test plan. The test script is formed. Some script instances are as follows:

```
<plan name="testXnc">
  <HttpConfig>
    <protocol>http</protocol>
    <ipaddress>localhost</ipaddress>
    <port>8080</port>
    <path>/xnc/index.jsp</path>
    <method>GET</method>
    <parameters/>
  </HttpConfig>
  <ThreadConfig>
    <userNum>100</userNum>
    <requestTime>10</requestTime>
    <loopNum>1</loopNum>
  </ThreadConfig>
  <Expect>
    <maxPerResponseTime>5</ maxPerResponseTime >
    <maxFailPercent>5%</ maxFailPercent >
    < throughput >200</ throughput >
  </Expect>
```

</plan>

The script above records basic information about performance testing using ASTD. For instance, the name of the test plan is ‘testXnc’, in 10 seconds, 100 pages of the user's URL address http://localhost:8080/xnc/index.jsp were accessed by GET. The maximum average response time expected by the user is 5 seconds, the maximum connection failure rate is 5%, and the system throughput is 200 requests per second.

(2)Test execution module implementation

According to the test script generated by the test plan module, the simulation of multi-user request access to the specified page, in order to obtain relevant performance index information. Using Java language multi-threaded programming and network programming features, open multiple threads, each thread alone send Http request on the Web page, access to response information. The thread class URLThread that accesses the Web application is obtained by inheriting the Thread class of the java.lang package and overriding the run method of the Thread class. Part of the code is as follows:

```
public class URLThread extends Thread {
// Covers the same name method of the Thread class and starts the test thread
public void run() {
// Generating the object of the request class
    SendRequest sr=new SendRequest();
// Pass the access address URL
    sr.setUrl(this.url);
// Gets the response status code of the request
  int state=sr.send();
……// The response time is recorded and the connection status is judged based on the response status code
}
}
```

In the connection class SendRequest, the URL and HttpURLConnection classes of the java.net package are used to implement the connection and test of the Web application.. Part of the code is as follows:

```
public class SendRequest {
  // Send the request connection information to the test URL and return the status code.
 public int send() {
  try {
            URL url = new URL(this.getUrl());//Building a URL object
            URLConnection urlc = url.openConnection(); // Gets the corresponding URLConnection object
            // Converts a URLConnection object to a HttpURLConnection object
            HttpURLConnection http = (HttpURLConnection) urlc;
            ……// Set request methods and properties
            return http.getResponseCode();//Return response status code
          }catch (Exception e) {
```

```
e.printStackTrace();
return 0; // If an exception occurs, return 0
            }
}
}
```

When the test execution is completed, the response information is written to the XML file for use by the test report generation module.

### 3.3 Test report generation module implementation

1) Data comparator

The read test executes the resulting XML file, gets response information, calculates performance metrics, and compares the user's expected performance metrics with the test script, resulting in deviations.

2)Statistical mapping

In order to describe the system average response time, intuitive link failure rate increases with the number of concurrent users increased by ASTD, the open source project JFreeChart draw line graph and bar chart, the horizontal axis is the number of users, the vertical axis is the average response time or link failure rate.

3) Generate report

According to the result of test performance test of Web application system, including the maximum response time, average response time, throughput, failure rate, to evaluate the performance of the system, combined with the statistical graph to generate the final test report.

## 4 Application and analysis

### 4.1 Application example

ASTD is used to test the performance of e-government system in Chengguan Street of Beijing, Fangshan District. The e-government system is developed with J2EE and was put into operation in 2006.

In the same network environment, 100, 200, 300, 400, 500, 600 virtual users is respectively throw in the e-government system in equal time. The ASTD can record the server delay response, and the failure rate of connection in each test. The maximum response time is the amount of time HTTP requests are requested for the longest response time recorded during each test. The average response time is the average of all request response times recorded during each test. The connection failure rate is the ratio of the number of connections rejected and the total number of connections requested by the Web server.

| Number of users | Maximum corresponding time | Mean corresponding time | Connection failure rate |
|---|---|---|---|
| 100 | 2271 Milliseconds | 332Milliseconds | 0% |
| 200 | 5911 Milliseconds | 965Milliseconds | 0% |
| 300 | 9165Milliseconds | 2254Milliseconds | 0% |
| 400 | 15942Milliseconds | 3917Milliseconds | 2% |
| 500 | 24783Milliseconds | 6983Milliseconds | 8% |
| 600 | 39986Milliseconds | 13957Milliseconds | 35% |

Figure.2. The response of the system under different online users

As showed in Figure 2, when 100 virtual users are online, the maximum response time is 2.271 seconds and the average response time is 0.332 seconds. This shows that the vast majority of users click behavior in a second will be able to respond. The users feel pretty good. Generally, the user experience is satisfying when the page response is controlled within 5 seconds. When 600 users are online at the same time, some requests have to consume more than more than 30 seconds to get feedback, and most of the operation response time exceeds 10 seconds, the user feel worse.

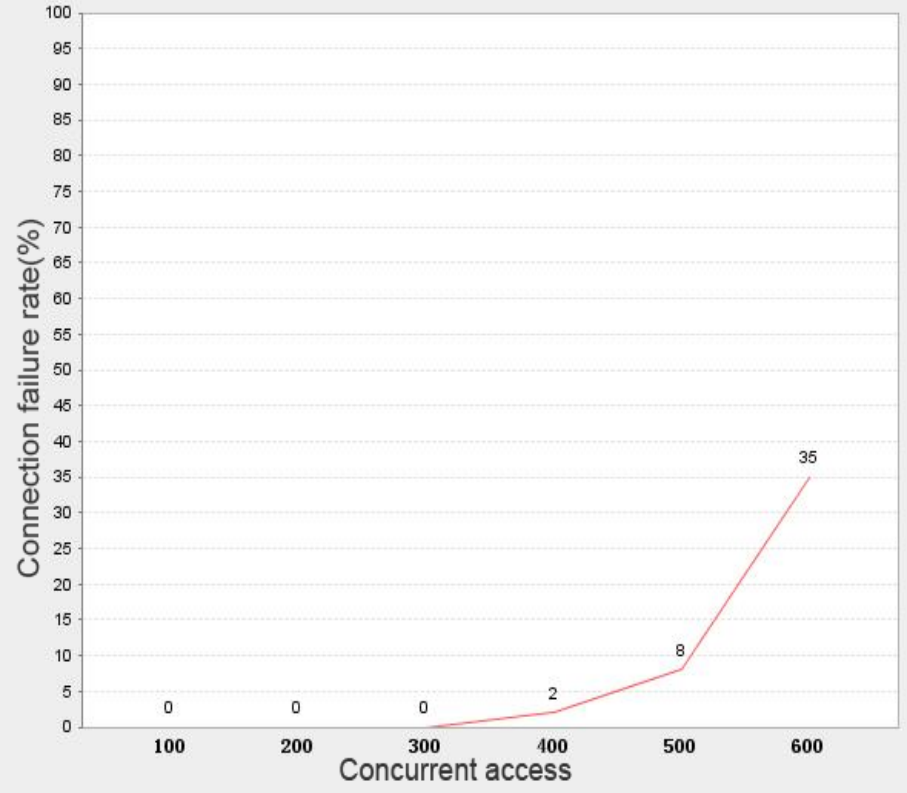



Figure.3. The connection failure rate varies with concurrent access numbers

As you can see from Figure 3, when the number of concurrent accesses is below 300, the Web server can correctly respond to all client requests. When the number of people online reaches 500, the connection failure rate is 8%, that is to say, 40 users' requests can’t be answered and the performance of the server has a downward trend. Once the number of users reached 600, the server could not respond to requests made by users to become very serious, only 65% of requests can be answered. That is to say, when the server is under the load of 600 online users, the system can’t run basically, and it needs to update the server hardware or optimize the system.

### 4.2 Result analysis

There are many factors that can affect system performance, such as network environment, server hardware performance, database, and Web application server. Any link may affect the performance of the whole system. Because the test environment is completed in LAN, it is possible to eliminate the bottleneck of network bandwidth. When you find that the number of online users is higher during the test, the functionality related to the database will increase rapidly. Therefore, the database is the bottleneck of the e-government system, and the next step can be considered to improve database design, or optimize the database connection pool.

## 5 Conclusions

In this paper, a set of Web performance testing tools ASTD is designed and implemented by using the characteristics of Java multi thread and network programming. The testing tool is adopted to simulate the real load, WEB application of the response time, the user request connection performance failure rate, which are comprehensive, accurate and reliable performance evaluation, help to find the bottleneck of the system, to ensure the quality of software products.

**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. Alberto A., Elaine J. 2004. The Role of Modeling in the Performance Testing of E-commerce Applications. *IEEE Transactions on Software Engineering*, 30(12):1072-1083.

2. Subraya B. M., Subrahmanya S. V. 2000. Object Driven Performance Testing of Web Applications. *Quality Software. Proceedings of the 1st Asia-Pacific Conference*, 17-26.

3. Ye X.M., Feng X.L. 2002. The Flow of Stress Test in Software. *Acta Scientiarum Naturalium University NeiMongol*, 33(1):107-108.

4. Tan H., Guan X., Ma L. 2006. Principle of performance test and realization of its automatic tools. *Computer Engineering and Design*, 27(19): 3660-3662.

5. He J.Y., Yan H.H., Liu C., Jin M.Z. 2007. Effective Web Performance Testing Method and Its Application. Computer application research, 1:275-277.

6. Li N., Wang C. 2005. Java based design of web application automation testing. *Computer Engineering and Design*, 26(12): 275-277.