Working Experiences of Planning, Design and Implementation for Software Load Testing

Chen Yiju

***Abstract*—Software Testing is one parts in Software Engineering whatever your development model. For Software quality assurance, we may need Functional Test, Non-Functional Test or both. Load Testing is one kind of Performance Test which belongs to Non-Functional Test. And Load Testing is the most frequently-executed testing type in Performance Test because it's relevant to us. For example, the ticket-booking website may fail or crash as soon as a Lady Gaga concert begins to sell. In this case, the sudden huge number of users (equivalent to heavy loads) may slow down the system, causing some transactions to become failed and even crash the system if it doesn't have Load Testing before the website gets online. This can cause huge loss whatever in business revenues or reputations. As you can see, it's so important to understand your system's max loading before it goes to production environment, because system providers still have chance to do performance tuning to improve the system quality. Besides, I notice that there are almost no articles on the Internet which provide the practical steps or complete process for Load Testing. In view of this, I've summarized the 4 steps for proceeding Load Testing which can be used both by testers or system providers.**

***Index Terms*—Soak test, spike test, configuration testing, isolation testing, transaction response time, successful transaction rate, fail-open, fail-closed.**

I. Introduction

Generally Speaking, Software Performance Testing covers Load Testing,Stress Testing,Soak Testing, Spike Testing, Configuration testing and Isolation Testing [1]. Before introduce today's topic as Load Testing, I'd like to make a simple clarification for Stress Testing which is often being confusing even by some testers. The purpose of Stress Testing is used to confirm a system has Robustness even under extremely high loads which may be tens or hundreds of times than Load Testing. As you can see in Fig. 1, the load applied on the system has to be higher than the acceptable max loading of the system.

In general, in the execution of Stress Testing, abnormal behaviors of the system can be anticipated. But when this abnormality leads to system crash, the system has to respond correctly via exception handlings and adopts fail-closed rather than fail-open rules is the key point.

The purpose of Load Testing is to check if the system's response can meet users' requirements or expectations under the specific workloads, then can get the maximum loads for this system.

Manuscript received December 19, 2016; revised March 30, 2017.

Chen Yiju is with the Institute for Information Industry, Taiwan (e-mail: wichchen@yahoo.com.tw).

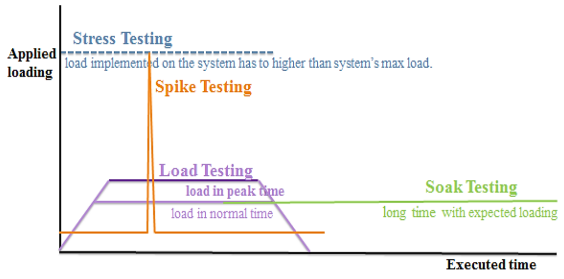


Fig. 1. Stress testing V.S load testing.

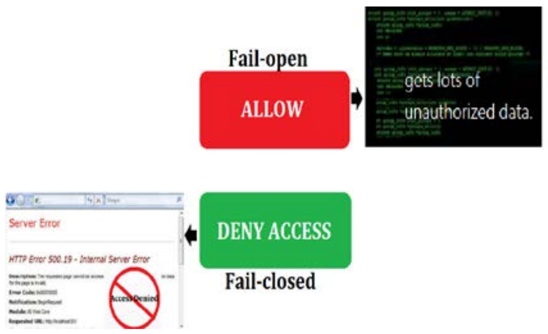

Fail-open
ALLOW
gets lots of unauthorized data.
Server Error
HTTP Error 500.19 - Internal Server Error
Access Denied
DENY ACCESS
Fail-closed

Fig. 2. Fail-open vs. failed-closed rule for website.

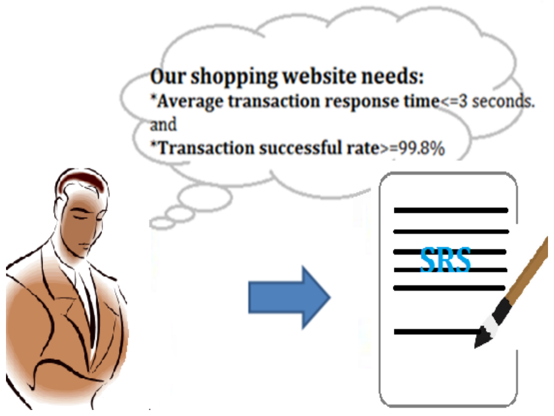

Our shopping website needs:
*Average transaction response time<=3 seconds.
and
*Transaction successful rate>=99.8%
SRS

Fig. 3. Users' expectations.

The loads mentioned here are coming from requirements or expectations for the system getting online in the future both in daily operations or for a particular event to meet a large number of surged users in a period of time. The requirements or expectations are stated in several formal documents like SRS (Software Requirements Specification) or Test Plan. The system's responses include several indicators such as throughput, average transaction response time and average successful transaction rate, etc. During the implementation of the test, in addition to use automated Load Testing tools to generate loads on the targeted system (maybe a website of a mobile application), the related backend servers like Database or AP servers should also be monitored. This is because if the final testing result is unsatisfied, we can use this information to help determine system's bottlenecks are coming from hardware, software or both.

II. The Key Contents

Based on my working experiences in software testing field, I've developed a methodology for proceeding load testing as Fig. 4.

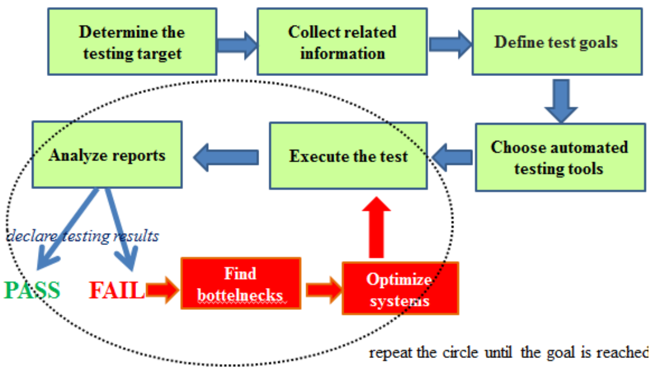

Determine the testing target
Collect related information
Define test goals
Choose automated testing tools
Execute the test
Analyze reports
declare testing results
PASS
FAIL
Find bottlenecks
Optimize systems
repeat the circle until the goal is reached

Fig. 4. Methodology for proceeding load testing.

And here are 4 practical steps for executing a load test:

1) Determine the testing target.
2) Collect related information for the system and define test objectives.
3) Choose automated testing and monitoring tools to perform Load Testing.
4) Analyze reports and repeat the circle as Fig. 5 until the expectation is reached.

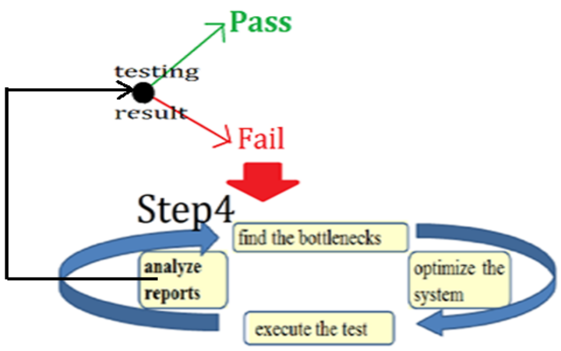
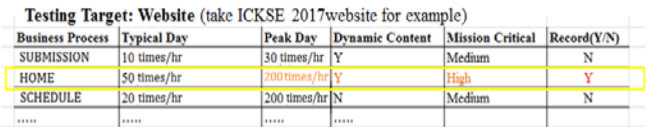


Fig. 5. Circulation process in step 4.

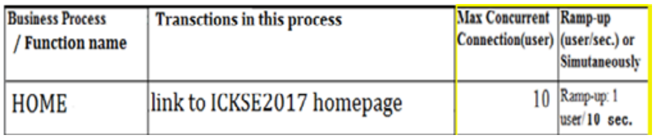
Step 1: First, the test manager would understand the testing requirements and testing target for the test this time in high level view, such as: the testing target is a website, or a mobile application? What kind of system architectures and its program languages? to allow the testing team to evaluate feasibilities and required resources for this testing project initially.

Step 2: The test manager will start to collect information covering core business processes, the frequency in both normal and peak time for each process which can refer to historical data or expectations, content type of each process (dynamic or static web pages),the importance to stakeholders ,estimated maximum number of online users in peak time when system getting on line, the way for users to enter the system(simultaneous or ramp-up),entering rate, etc. Use this information to discuss with your clients and stakeholders before arranging the test. The more complete information to collect (as Fig. 6 and Fig. 7), the easier testing experts can pick out business processes, define transactions and arrange the subsequent testing scenarios as yellow parts.


Testing Target: Website (take ICKSE 2017website for example)

Business Process	Typical Day	Peak Day	Dynamic Content	Mission Critical	Record(Y/N)
SUBMISSION	10 times/hr	30 times/hr	Y	Medium	N
HOME	50 times/hr	200times/hr	Y	High	Y
SCHEDULE	20 times/hr	200 times/hr	N	Medium	N
.....	.....	.....	.....		

Fig. 6. Information to be collected-1.



Business Process / Function name	Transctions in this process	Max Concurrent Connection(user)	Ramp-up (user/sec.) or Simutaneously
HOME	link to ICKSE2017 homepage	10	Ramp-up: 1 user/ 10 sec.

Fig. 7. Information to be collected-2.

Transaction is a complete process which starts from an online user sending a request to the web server, and then the server returns a response to the user as Fig. 8.

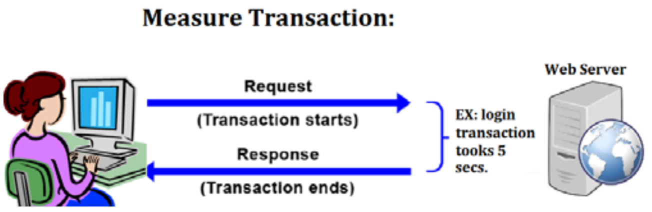
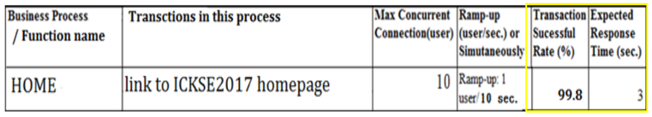


Fig. 8. Transaction definition.

A request is possibly from clicking a hyperlink or pressing a "Send" button on the dynamic page which would interact with the web server. Transactions are defined by testing experts and clients together. There are 2 transaction indicators that would often be used in Load Testing: "Average Transaction Response Time" and "Average Successful Transaction Rate". Transaction Response Time means the time it takes to complete the specific Transaction. And Successful Transaction Rate is the percentage of transactions that did not produce an error divided by all transactions for the defined transaction in the test execution. The test objectives of Load Testing should be quantifiable indicators such as the above mentioned indicators covering "Average Transaction Response Time" and "Average Successful Transaction Rate". Expected values of these indicators are defined according to users' requirements and expectations as Fig. 9 with considering the industry requirements (or your competitors' criteria).



Business Process / Function name	Transctions in this process	Max Concurrent Connection(user)	Ramp-up (user/sec.) or Simutaneously	Transaction Sucessful Rate (%)	Expected Response Time (sec.)
HOME	link to ICKSE2017 homepage	10	Ramp-up: 1 user/ 10 sec.	99.8	3

Fig. 9. Quantifiable indicators in load testing.

The test manager will place the above collected information, testing objectives and Pass-Fail Criteria in a Test Plan before the actual test is carried out. After the client has confirmed the Test Plan, the Step3 is going to be started soon. Besides, information for system architecture in a production environment including hardware, software information and their configurations should also be also collected and put into the Test Plan together. This information is not only used to build a testing environment but also an important reference to consider possible causes for affecting the testing result (like the load balancer or hardware storage capacity would affect the system's performance).

Step3: Automated testing tools play very important roles in Performance Test because it's hard and inefficient to generate heavy loading by asking lots of real men to execute the same action at the same time. Test experts and a test manager will choose suitable automated testing tools according to program language,protocols, software attributes with the triangle limit of Project Management. After accomplish this, testers will use these performance tools to record procedures selected in step2, this kind of tools will record a user's operating procedure for the testing target (a website or a mobile applications). Not every testing tool can support all program languages or system protocols. So the key point of this step is to determine the supportability and applicability for automated tools. Besides, remember to execute Load Testing in an Intranet rather than an Internet testing environment in the first few rounds. Because we need to try to avoid the networking issue in the first few rounds of tests to lessen variables. Then the testing team will continue to use these tools to set up testing scenarios for those scripts recorded in step1 based on the Pass-Fail Criteria in Test Plan. Many kinds of commercial and open-source tool can do this like Apache Jmeter [2], Soap UI [3], and HP Load Runner [4]. The testing tool will generate many virtual users by simulating your operations, the put loads on you system(your backend servers)for a period of time.

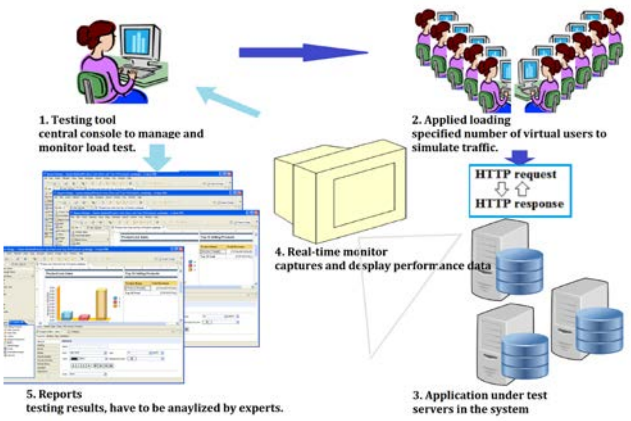

1. Testing tool
central console to manage and
monitor load test.
2. Applied loading
specified number of virtual users to
simulate traffic.
HTTP request
HTTP response
4. Real-time monitor
captures and desplay performance data
5. Reports
testing results, have to be anaylized by experts.
3. Application under test
servers in the system

Fig. 10. Operation diagram for performance testing tools.

As mentioned earlier, in testing execution, in addition to use a load-testing tool, testers would also enable monitoring functions which may be built-in in the load-testing tool or add other specialized tools to monitor hardware resources on related servers in the whole system. For example, to build a website may need a webs server, AP server and a database server. Monitoring for hardware's resource can help confirm the system bottleneck is coming from software or hardware. The popular monitoring indicators for hardware resources are CPU usage,Memory,Disk usage and so on. The reasonable ranges of these values during the testing period should meet the following requirements which refer to MSDN materials [5] in Reference page.

Network/Bandwidth:

- Network Interface usage should not higher than 70%.
- Network Interface\Output Queue Length should not higher than 2.

CPU usage:

- Processor\% Processor Time should not higher than 85%.
- Processor\% Interrupt Time should not higher than 15%.

Memory:

- Memory\% Committed Bytes in Use should not higher than 80%.
- Memory\% Available Mbytes should be higher than 5% of total physical RAM.
- Memory\Pages per Second should be lower than 1000.

Disk:

- PhysicalDisk\% Idle Time should be higher than 20%.
- PhysicalDisk\Avg.Disk Sec/Read should be lower than 10 millisecond.
- PhysicalDisk\Avg.Disk Sec/Write should be lower than 10 millisecond.

Step4: After the test completes, the testing manager and experts will interpret data and reports generated by all tools, analyze the final result and develop action plans. Is the test in this round passed or not? Does it meet the customer's expectations? If system's performance is not ideal, the bottleneck is coming from the hardware, software or both? Of course, the important thing is "how to improve"? Sometimes, in result-analysis stage, the crux can't be discovered only from few data generated by a single tool, so the other load-testing tools can be added in. This can help testing experts get more details then make accurate and objective conclusion for the test. After the program has been optimized or hardware resources are enlarged by system developers (note: only 1 adjustment can be made at a time, so we just can know if this measurement is working out), the test team will execute the same scenarios in the same testing environment again to confirm the system's performance has been improved or may need to adopt other action plan.

III. Conclusion

From the above contents, we can see all events in pre-test phase are very important including information collections and communications with your clients. Also, during the testing-execution phase, the testing team can add other testing tools depending on test results and available resources then, to provide more accurate results and recommendations. And after the action plan is executed (whatever for the software program or hardware), the test has to be executed again to verify the system's performance has been improved until meeting the requirements and expectations in SRS.

As I mentioned before, operations for automated performance testing tools is not the most important part in Load Testing, because now many automatic testing tools on the markets are very smart and user friendly. Automated tools are important but still needs complete thinking and planning as the four points in this paper. Both testers and system providers can use these points as the guide to proceed Load Testing. Through the 4 steps in this article, you can save lots of time, because you can have clear idea for what information us need to collect from customers, how to plan and design the correct test plan and test scenarios. Do the right things in the beginning of the test then you just can get the valuable testing results in the final phase. Otherwise, even if you use the most advanced automated testing tools, due to the initial poor planning and design, the wrong conclusion would be generated then you won't find the real bottleneck for your system.

IV. Index Term

A. Soak Test

Soak test is also called durability test, usually to determine whether the system can still run properly for a long period of time under the expected load capacity. During the testing execution, memory utilization would be monitored to detect if memory leak happens potentially. After the system has been running for a long period of time, it is also important to check the throughput or response time would degrade or can be as good as its first implementation or even better.

B. Spike Test

The spike test is performed by rapidly increasing the load to a very large amount then observing the behavior of the system. The goal is to determine whether the system's performance will be kept the same,affected or will crash under such a drastically varying load.

C. Configuration Testing

Configuration testing is not derived from the perspective of the load testing. Its point is to confirm the impacts for system's performance and behaviors after a change of system configuration and settings has been made. For example, with different load balancing mechanisms for a server, the performance and behaviors of the whole system may be different.

D. Isolation Testing

Isolation tests are tests that perform repeated testing on areas that may cause system problems. Such tests are usually isolated and executed independently to identify the specific component that makes system failure.

E. Transaction Response Time

Transaction Response Time means the time it takes to complete the specific Transaction. For on-line web systems under the usual operation loadings, generally, the transaction response time is set for no more than 3 seconds.

F. Successful Transaction Rate

Successful Transaction Rate is the percentage of transactions that did not produce an error divided by all transactions of this test execution. In software industry, for on-line non e-commerce websites, the average transaction success rate is usually set higher than 99%.

G. Fail-Closed

In general, a website or mobile application usually adopts fail-closed rule, meaning that when the system crashes, its access authority should be set as all deny rather than open to anyone.

H. Fail-Open

When exceptions are thrown, fail-open systems allow access as opposed to fail-closed systems that block access.

References

[1] Types of software performance testing From Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Software_performance_testing

[2] Apache JMeter official website. [Online]. Available: http://jmeter.apache.org/index.html

[3] SoapUI official website. [Online]. Available: https://www.soapui.org/

[4] LoadRunner official website. [Online]. Available: http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/

[5] Evaluation criteria when monitoring resource usage by Microsoft. [Online]. Available: https://msdn.microsoft.com/zh-tw/library/ms178072.aspx

Chen Yiju is a senior testing engineer from Institute for Information Industry, Taiwan, who has been working in this field for over 10 years and specializes in: software project management; software testing consulting service; testing tutoring such us how to clarify testing requirements, design of test scenarios, adopts suitable test techniques / tools for functional, performance, usability, and security test. Make the summary of testing results then provide recommendations. Problem tracking management.

She also gets several international IT certificates covering Project Management Professional (PMP), ISTQB Testing FL, EC-COUNCIL CERTIFIED INCIDENT HANDLER, CompTIA Security + and Cloud Computing Foundation Certificate. Those knowledge can also help in software testing field