

## ABSTRACT

The goal of this paper is to document the importance of performance testing and the consequences that can be paid off if performance testing is neglected. Where and how Performance testing term often used interchangeably with 'stress' and 'load' testing?

Is it true that performance testing refers to the testing of how well an application complies with performance requirements? All answers to these questions have been discussed in this paper. The paper also discusses about the different ways available to conduct the performance testing process in IT industry, the process of Performance Test and how automated performance testing is better than manual performance testing?

## INTRODUCTION

Performance testing, by its name itself, is to analyze the quality of a product whether it's some hardware or software in terms of its speed, efficiency and effectiveness. The product might contain a comprehensive set of features in the way to attract the users. At the end, it's all about the bottom line stating how fast the product features work and the response time matters when a user interacts with the product. When the product is slow even though it has all the features what exactly user wants from the product, eventually it will lead to failure just because it's not triggering the output as fast as the user expects. Needless to say in this fast-moving technology world, people want their daily happenings instant and easy to move the life going on and on from modular kitchens & toiletries to finding their life partner matches.

"The standard philosophy of '**Test to Destruction**' will probably give you an idea of roughly how many users your site can handle at once, but it won't always tell you why the site fails to function properly. And without knowing why, you're not likely to be able to do much about it".

**--Extreme Tech**

Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device. Qualitative attributes such as reliability, scalability and interoperability may also be evaluated. Performance testing is often done in conjunction with stress testing.

The goal of performance testing is not to find bugs, but to eliminate bottlenecks and establish a baseline for future regression testing. To conduct performance testing is to engage in a carefully controlled process of measurement and analysis. Ideally, the software under test is already stable enough so that this process can proceed smoothly.

The process can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency or reliability.

Determining the most suitable performance testing technique can prove to be quite challenging especially with limited testing tools and available resources. However, successfully identifying and implementing an appropriate performance testing technique can be equally rewarding leading to an application developed which meets the stringent requirement specifications and exceeds customer needs and expectations.

**"The peace of mind that it will work on go live-day alone justifies the cost of performance testing".**

Effective performance testing can quickly identify the nature or location of a software-related performance problem.

### Importance of Performance Testing

With the ever growing need for quick access to data, and with the large volumes of data at our disposal, the need for high performance systems and applications has become increasingly important.

The performance testing is imperative as the internet and IT infrastructure becoming crucial to business and User, employees, business partners, customer etc all rely on portals, applications, and data to do their jobs. At this the cost of failure can be devastating.

Performance testing in the enterprise is intermittent that is often prompted by upgrades as it determines the usability/effectiveness of a system under load, helps to detects bottleneck before a new system or upgrade is deployed and tune it for better performance.

Valuable time can be wasted due to poorly designed hardware layout or badly developed application. Slow data transfer rate may be inherent in hardware but can also result from software-related problems, such as:

- Too many applications running at the same time.
- A corrupted file in a Web browser.
- A security exploit.
- Heavy-handed antivirus software.

Performance testing of an application under development aims to thoroughly address a multitude of performance affecting factors to result in an enhanced user experience when using an application. By overlooking the performance testing can lead to pay “High costs until Disaster strikes”.

Various techniques, depending on the application type, project duration, available performance testing tools and resources i.e., budget and time constraints are taken into consideration when devising the most suitable performance testing strategy.

The performance is evaluated for a product’s particular feature when the functionality is stuffed with certain amount of load and stress that means load and stress testing goes in conjunction with performance testing. The performance of a product feature accounts to its enduring capabilities under an expected amount of load, which is gradually increased, thereby allowing the product to experience stress when the load applied is beyond the limit.

The output response time is measured for every application of load to determine what amount of load creates hindrance to the normal operation? How far the application is robust at times of overload or extreme load conditions?

**Limitation of Manual Performance Testing**

The manual testing may be our best tool but the limitation that it cannot handle is the rate of change which leads to frequency of testing requirements which involves extensive testing.



Figure 1

**Advantage that Automated Performance Testing gives**

The automated software testing tools play a crucial role in the development and testing cycle of any product. If we want our software to be released on time, employing automated tools makes good sense. It’s a wise decision to carry out the performance test by using an automated test tool to simulate large number of users as it replaces real users with virtual users and generate a consistent, measurable, and repeatable load, managed from a single point of control. Moreover, it helps in efficiently isolates performance bottlenecks.

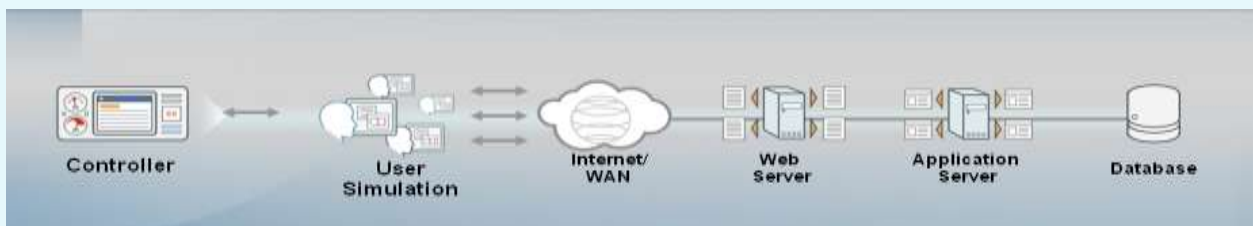


Figure 2

Once a performance test is initiated, it can run for several minutes to several days, depending on the test goal.

During the test time, the test tool monitors and collects performance data from all of the components within the system under test, such as the Web server, application server, or database server. All of the monitor data along with the performance test data collected at the generating client end to determine the overall performance as well as the potential system bottlenecks. In a typical performance test cycle, the performance bottlenecks are located, fixed, and iteratively retested to ensure that they are fixed as designed.

While doing automated performance testing, we have to keep in mind few questions like;

- How does a client/server environment affect testing?
  - What are the expected loads on the server (e.g., number of hits per unit time?), and what kind of performance is required under such loads (such as web server response time, database query response times)?
  - What kinds of tools will be needed for performance testing (such as web load testing tools, other tools already in house that can be adapted, web robot downloading tools, etc.)?
  - What kind of performance is expected on the client side (e.g., how fast should page appear, how fast should animations, applets, etc. load and run)?

Client/server applications can be quite complex due to the multiple dependencies among clients, data communications, hardware, and servers. Thus testing requirements can be extensive. At this stage performance testing may be useful in determining client/server application limitations and capabilities.

The one critical configuration requirement for the load-generating systems is that they have adequate network bandwidth throughput capability to access the system under test in a realistic manner without bandwidth constraints.

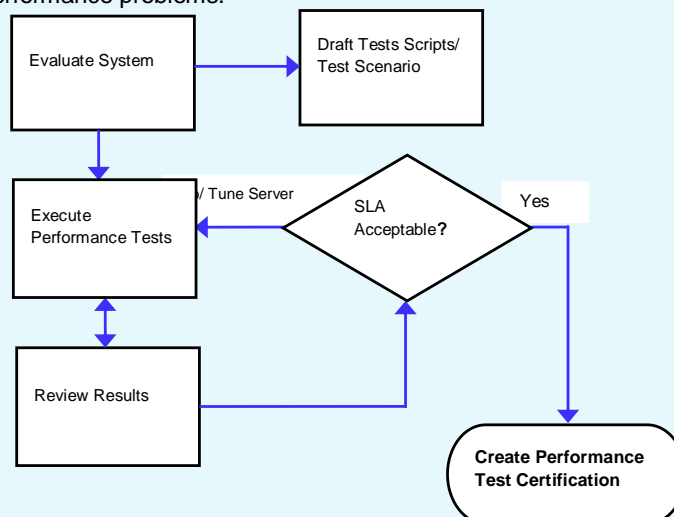
If bandwidth constraints become a problem, adding additional load generators to the pool of load generators will typically fix this problem.

If the test is global or local, the Internet will be an important factor in the configuration. For an isolated configuration, the Internet is not a factor.

**Process of Performance Testing**

Performance Test process is magnificent to Evaluate the compliance of a system or component with specified performance requirements. As Performance testing is intended to determine the responsiveness, throughput, reliability, and/or scalability of a system under a given workload, it is commonly conducted to accomplish the following:

- Evaluate against performance criteria.
- Compare performance characteristics of multiple systems or system configurations.
- Find the source of performance problems.



**Evaluate System** is the most important process because it involves:

• Predicting actual user experience	• Agree on goals of testing
• Accessing any system limitations	• Identify and define stakeholder expectations
• Determining budget, schedule constraints and staffing plan	• Agree to promotion strategy
• Outline resources available	• Engage test lab (if needed)
• Verify basic functionality	• Generate use cases
• Capture user activity logging information	• Analyze user activity profile and Model user activity
• Choose the tool(s)	• Assign resources needed for scripting, testing
• Create test environment	

After initiating the test, the load generator systems to begin accessing the system under test using the designed usage patterns. Depending on whether the test is a global, local, or isolated configuration, the load generators may be located worldwide or completely contained within a test lab.

**Drafts Tests Scripts/Scenario** is the second step which involves writing test scripts/test scenarios. The script/scenario will contain transitions which are the most intensive activities performed on the application. It includes:

1. Design scripts
2. Create scripts
3. Validate scripts
4. Build script library
5. Identify re-usable script components

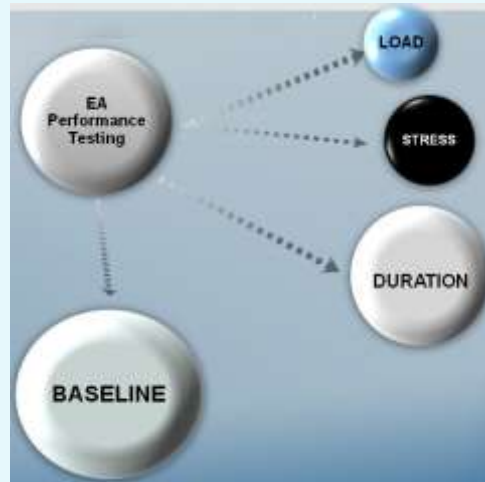
**Execute Performance Tests** will leads to review results. We will keep on reviewing results till SLA would be in an acceptable condition. Performance requirements (PR) are necessary for system design and development. If there are no written performance requirements, it just means that they exists in heads of stakeholders, but nobody bothered to write them down and made sure that everybody agrees with them. Then PR will be input for performance testing (where they will be validated) as well as capacity is planning and production monitoring (SLA - Service Level Agreement).

These are criteria that the team wants to meet before the application release date, with some of the criteria being negotiable, thus performance goals are somewhat flexible in comparison to performance requirements. Another commonly used term which equates to performance requirements is performance thresholds which indicate maximum values for identified metrics.

1. Execute tests	2. Collect data
3. Analyse test results	4. Run possible iterations
5. Trouble shoot bottle necks	6. Tune system
7. Rest	8. Long non-performance failure

Review Result will draft and then promote results with the following points:

1. Outline context
2. Draft results
3. Provide feedback to stakeholders
4. Deliver action items
- 5.
6. Promote results



Load tests is the subcategory which checks the end to end performance test under anticipated product load and concerned with testing the application under workloads and load volumes estimated during actual operations.

Stress testing is aimed towards determining performance characteristics when the application is subject to conditions beyond those anticipated during actual operations. Stressful conditions may include limited memory, insufficient disk space, or server failure. Stress tests are designed to determine when and how an application will fail, and what indicators will give warning of an approaching failure.

Duration tests are test with a constant load over a period between 8 and 24 hours to determine if an application's performance degrades over an extended period of time.

Baselines are the subcategory of performance test and are used to determine whether performance is improving or declining and to find variations across different builds. Baselines can be created at the application, component or system level. It is important to note that baselines are metrics and can include a broad set of key performance indicators i.e., response times, processor capacity, memory usage, network usage etc.

### Conduct the “Right Test” to get the “Right Results”

Performance Testing is not a one off event, so we have to be sure that at which stage performance which type of performance test is crucial decision.

**Smoke Test** is used to check an application can perform under normal load an initial run of performance test takes place.

**Capacity Test** is designed to determine your server's ultimate failure point. This is done to plan for future growth, such as an increased user base or increased volume of data. Capacity testing helps you to identify a scaling strategy in order to determine whether you should scale up or scale out.

**Endurance testing** is a subset of load testing, focused on determining or validating the performance characteristics of the product under test when subjected to workload models and load volumes anticipated during production operations over an extended period of time. Metrics such as Mean Time between Failure (MTBF) and Mean Time to Failure (MTTF) are valuable outcomes of endurance testing.

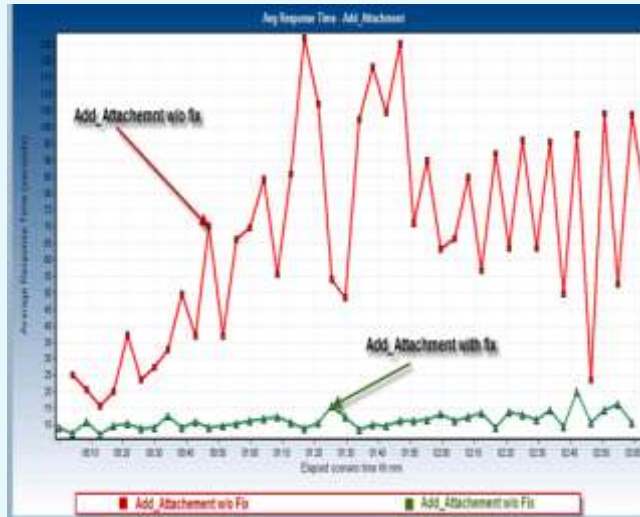
**Spike testing** is a subset of stress testing, focused on determining or validating the performance characteristics of the product under test when subjected to workload models and load volumes that repeatedly increase beyond anticipated production operations for short periods of time.

**Volume Testing** determines the problems that occur during long-term user activity.

**Component Testing** determines the performance and behavior of a specific component.

**Benchmark Testing** measures the performance of a system or component relative to a standard.

**Transaction Cost Analysis** determines the system resources consumed by a single transaction.



### Response Time Degradation Curve

When the load placed on the system is raised beyond normal usage patterns, in order to test the system's response at unusually high or peak loads, it is known as stress testing. The load is usually so great that error conditions are the expected result, although no clear boundary exists when an activity ceases to be a load test and becomes a stress test.

### ENDING NOTES

Though it may seem counterintuitive at first to slow your deployment for performance test planning and execution, the payoff in time, money, and quality will be big and will come soon. A fully integrated performance testing program is the preventative medicine that keeps your system from becoming an inaccessible and costly resource.

Companies rely on systems to conduct business efficiently and effectively and Performance testing ensures that our users are getting reliable and timely access to the resources they need because Performance testing mitigates the risk of lost time and money due to poor performance.

### REFERENCES

1. <http://projects.webappsec.org/w/page/13246989/Web-Application-Security-Statistics>
2. <http://www.aldeid.com/index.php/WackoPicko/Directory-Traversal>
3. <http://programming4.us/security/614.aspx>