## **Performance Testing**

**Performance Testing** is a software testing process used for testing the speed, response time, stability, reliability, scalability and resource usage of a software application under particular workload. The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application. It is a subset of performance engineering and also known as The focus of Performance Testing is checking a software program's

Big Data ∨

Modernize your Fishing

Secure your catch with a modern and user

**OPEN** 

100

Want to know your IQ?

Answer 20 questions to find out

115

130

1.9% 0.19

Work From Home Positions And

Become A Gaming Support Agent

vessel

bluectrl.io

IQ

0.1% 1.9%

www.test-iq.org

**Tutorials** 

**Smoke-Sanity Testing** 

**Basis Path Testing** 

**Performance Testing** 

**Regression Testing** 

Non Functional Testing

**Testing** 

**Automation** 

friendly system.

"Perf Testing".

• Speed - Determines whether the application responds quickly

- Scalability Determines maximum user load the software application can handle. Stability - Determines if the application is stable under varying loads In this tutorial, you will learn-

• Why do Performance Testing? • Types of Performance Testing

• Common Performance Problems **Performance Testing Process** 

• What is Performance Testing?

- Performance Testing Metrics: Parameters Monitored
- Performance Test Tools FAQ

• Example Performance Test Cases

- Why do Performance Testing?

**FEATURED VIDEOS** 



performance bottlenecks.

poor usability.

deviations.

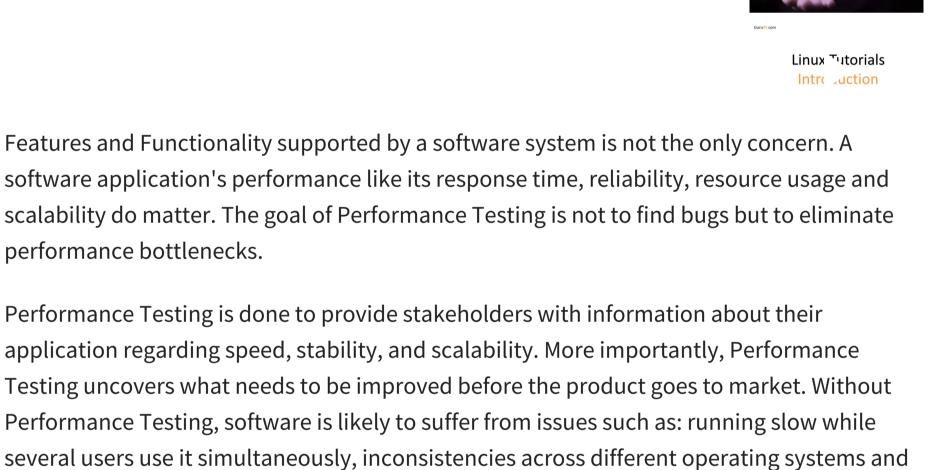
Web Service Outage.

an application.

over a long period of time.

Hence, performance testing is important.

**Types of Performance Testing** 



Linux owtorials IrRLAYINGn

Performance testing will determine whether their software meets speed, scalability and stability requirements under expected workloads. Applications sent to market with poor performance metrics due to nonexistent or poor performance testing are likely to gain a bad reputation and fail to meet expected sales goals. Also, mission-critical applications like space launch programs or life-saving medical equipment should be performance tested to ensure that they run for a long period without

such an organization would be \$896,000 weekly, translating into more than \$46 million per year. Only a 5-minute downtime of Google.com (19-Aug-13) is estimated to cost the search giant as much as \$545,000. It's estimated that companies lost sales worth \$1100 per second due to a recent Amazon

• Load testing - checks the application's ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live. • Stress testing - involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of

generated by users. • Volume testing - Under Volume Testing large no. of. Data is populated in a database and the overall software system's behavior is monitored. The objective is to check software application's performance under varying database volumes.

• Scalability testing - The objective of scalability testing is to determine the software

application's effectiveness in "scaling up" to support an increase in user load. It helps

• Spike testing - tests the software's reaction to sudden large spikes in the load

plan capacity addition to your software system. **Common Performance Problems** 

performance bottlenecks are

Operating System limitations

CPU utilization

Disk usage

Memory utilization

Network utilization

running application will lose potential users. Performance testing is done to make sure an app runs fast enough to keep a user's attention and interest. Take a look at the following list of common performance problems and notice how speed is a common factor in many of them:

Most performance problems revolve around speed, response time, load time and poor

scalability. Speed is often one of the most important attributes of an application. A slow

- range of users. Load Testing should be done to be certain the application can handle the anticipated number of users. • Bottlenecking - Bottlenecks are obstructions in a system which degrade overall system performance. Bottlenecking is when either coding errors or hardware issues cause a decrease of throughput under certain loads. Bottlenecking is often caused by
- **Performance Testing Process** The methodology adopted for performance testing can vary widely but the objective for performance tests remain the same. It can help demonstrate that your software system
- Configure **Identify Test** Plan & Implement erformance Test **Run Tests** Environment Design **Test Design** Criteria Environment Re-test 1. Identify your testing environment - Know your physical test environment, production environment and what testing tools are available. Understand details of the hardware, software and network configurations used during testing before you begin the testing process. It will help testers create more efficient tests. It will also help identify possible challenges that testers may encounter during the performance testing procedures.

2. **Identify the performance acceptance criteria** - This includes goals and constraints

project success criteria outside of these goals and constraints. Testers should be

specifications will not include a wide enough variety of performance benchmarks.

Sometimes there may be none at all. When possible finding a similar application to

3. Plan & design performance tests - Determine how usage is likely to vary amongst

end users and identify key scenarios to test for all possible use cases. It is necessary to

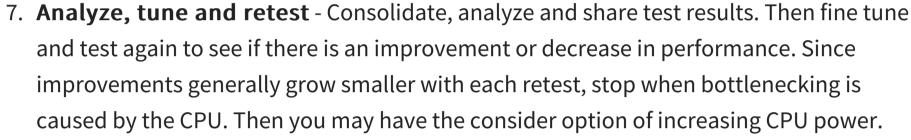
simulate a variety of end users, plan performance test data and outline what metrics will

empowered to set performance criteria and goals because often the project

for throughput, response times and resource allocation. It is also necessary to identify

be gathered. 4. Configuring the test environment - Prepare the testing environment before execution. Also, arrange tools and other resources.

compare to is a good way to set performance goals.



Performance Testing Metrics: Parameters Monitored

5. Implement test design - Create the performance tests according to your test design.

- disk during a sample interval. • Network output queue length - length of the output packet queue in packets. Anything more than two means a delay and bottlenecking needs to be stopped. • Network bytes total per second - rate which bytes are sent and received on the interface including framing characters. • Response time - time from when a user enters a request until the first character of the response is received. • Throughput - rate a computer or network receives requests per second. • Amount of connection pooling - the number of user requests that are met by pooled connections. The more requests met by connections in the pool, the better the performance will be.
- running and currently active. • Garbage collection - It has to do with returning unused memory back to the system. Garbage collection needs to be monitored for efficiency.

• Verify response time is not more than 4 secs when 1000 users access the website

• Verify response time of the Application Under Load is within an acceptable range when

- During the actual performance test execution, vague terms like acceptable range, heavy load, etc. are replaced by concrete numbers. Performance engineers set these numbers as per business requirements, and the technical landscape of the application.
- This tool is capable of simulating hundreds of thousands of users, putting applications under real-life loads to determine their behavior under expected loads. Loadrunner features a virtual user generator which simulates the actions of live human users. • Jmeter - one of the leading tools used for load testing of web and application servers.

• HP LoadRunner - is the most popular performance testing tools on the market today.

integrates into your existing Continuous Delivery pipeline. With NeoLoad, teams test 10x

faster than with traditional tools to meet the new level of requirements across the full

Agile software development lifecycle - from component to full system-wide load tests.

All provided FREE!!! What is the difference between Performance Testing & Performance

It is of significance to understand the difference between Performance Testing and

**Performance Testing** is a discipline concerned with *testing and reporting* the current

Performance engineering is the process by which software is tested and tuned with the

intent of realizing the required performance. This process aims to optimize the most

Performance Engineering. An understanding is shared below:

performance of a software application under various parameters.

success, the concept of coupling performance testing with performance tuning has caught on, and now we call it performance engineering. **Conclusion** 

What is Web Load **Testing vs TEST HARNESS?** Stress Tools & **Testing vs Examples** 

Report a Bug

**SOFTWARE TESTING** 

product. It ensures customer satisfaction & protects an investor's investment against product failure. Costs of performance testing are usually more than made up for with improved customer satisfaction, loyalty, and retention. Prev

**Engineering** 

YOU MIGHT LIKE: **SOFTWARE TESTING Testing** 

Read more »

Test the final System

conditions. **Performance Test Tools** testing tools. NeoLoad - is the performance testing platform designed for DevOps that seamlessly

**Example Performance Test Cases** 

There are a wide variety of performance testing tools available in the market. The tool you choose for testing will depend on many factors such as types of the protocol supported, license cost, hardware requirements, platform support etc. Below is a list of popularly used • LoadNinja – is revolutionizing the way we load test. This cloud-based load testing tool empowers teams to record & instantly playback comprehensive load tests, without complex dynamic correlation & run these load tests in real browsers at scale. Teams are able to increase test coverage. & cut load testing time by over 60%.

FAQ Which Applications should we Performance Test? Performance Testing is always done for client-server based systems only. This means, any

Historically, testing and tuning have been distinctly separate and often competing realms. In the last few years, however, several pockets of testers and developers have collaborated independently to create tuning teams. Because these teams have met with significant

**Application Testing: 8 Performance Testing:** Test Harness in Software **Difference Discussed Step Guide to Website** Testing Test Harness in

**SOFTWARE TESTING** 

Selenium

• Check the maximum number of users that the application can handle before it crashes. • Check database execution time when 500 records are read/written simultaneously.

GUYU Guru99 is totally new experience.

important application performance trait i.e. user experience.

What is Performance Software Testing is a Before we pen down more testing? Performance collection of stubs, drivers...

> **Python** What 45 Linux Linux Beginner Tutorial Linux Tutorials

SAP eBook **Execute online** Execute Java Online **Execute Javascript** 

Advertise with Us

**Career Suggestion** 

**SAP Career Suggestion Tool** 

Software Testing as a Career

Write For Us

**Contact Us** 

© Copyright - Guru99 2021

• Long Load time - Load time is normally the initial time it takes an application to start. This should generally be kept to a minimum. While some applications are impossible to make load in under a minute, Load time should be kept under a few seconds if possible. • Poor response time - Response time is the time it takes from when a user inputs data into the application until the application outputs a response to that input. Generally, this should be very quick. Again if a user has to wait too long, they lose interest. • **Poor scalability** - A software product suffers from poor scalability when it cannot handle the expected number of users or when it does not accommodate a wide enough one faulty section of code. The key to fixing a bottlenecking issue is to find the section of code that is causing the slowdown and try to fix it there. Bottlenecking is generally fixed by either fixing poor running processes or adding additional Hardware. Some common

meets certain pre-defined performance criteria. Or it can help compare the performance of two software systems. It can also help identify parts of your software system which degrade its performance. Below is a generic process on how to perform performance testing

6. Run the tests - Execute and monitor the tests.

The basic parameters monitored during performance testing include:

**Bandwidth** - shows the bits per second used by a network interface. • **Private bytes** - number of bytes a process has allocated that can't be shared amongst other processes. These are used to measure memory leaks and usage. • Committed memory - amount of virtual memory used. • Memory pages/second - number of pages written to or read from the disk in order to resolve hard page faults. Hard page faults are when code not from the current working

processor. This again occurs when a process requires code from outside its working set.

• CPU interrupts per second - is the avg. number of hardware interrupts a processor is

• **Disk queue length** - is the avg. no. of read and write requests queued for the selected

• Page faults/second - the overall rate in which fault pages are processed by the

• **Processor Usage** - an amount of time processor spends executing non-idle threads.

Memory use - amount of physical memory available to processes on a computer.

**Disk time** - amount of time disk is busy executing a read or write request.

set is called up from elsewhere and retrieved from a disk.

receiving and processing each second.

once.

tuned.

simultaneously.

the network connectivity is slow

• Maximum active sessions - the maximum number of sessions that can be active at • **Hit ratios** - This has to do with the number of **SQL** statements that are handled by cached data instead of expensive I/O operations. This is a good place to start for solving bottlenecking issues. • Hits per second - the no. of hits on a web server during each second of a load test. • Rollback segment - the amount of data that can rollback at any point in time. • Database locks - locking of tables and databases needs to be monitored and carefully • Top waits - are monitored to determine what wait times can be cut down when dealing with the how fast data is retrieved from memory • Thread counts - An applications health can be measured by the no. of threads that are

• Check CPU and memory usage of the application and the database server under peak load conditions • Verify response time of the application under low, normal, moderate and heavy load

application which is not a client-server based architecture, must not require Performance Testing. For example, Microsoft Calculator is neither client-server based nor it runs multiple users; hence it is not a candidate for Performance Testing.

details on the type of web testing is a type of testing Read more »

Informatica

**Interesting SAP** Java eBook Blog Quiz **Execute HTML Jmeter Execute Python** 

testing, lets quickly define for determining the speed Web Testing. What is... of... Read more » **SOFTWARE TESTING ENTERPRISE TESTING SOFTWARE TESTING** Levels of **Database** IoT **Testing in** (Data) **Testing** Data Integrit **Software Tutorial: Testing** What is, **Testing Tutorial** with Process, Tests are **Challenges & Tools Sample Test Cases** grouped together based on What is Database Testing? What is Internet of Things? where they are added in Database Testing is a type The Internet of Things SDLC or the by the level of of software testing that popularly known as IoT is of... checks the schema,... the network that... Read more » Read more » Read more » **Top Tutorials** f y in D **About** About Us

Privacy Policy | Affiliate Disclaimer | ToS

**Testing** 

According to Dunn & Bradstreet, 59% of Fortune 500 companies experience an estimated 1.6 hours of downtime every week. Considering the average Fortune 500 company with a minimum of 10,000 employees is paying \$56 per hour, the labor part of downtime costs for

• Endurance testing - is done to make sure the software can handle the expected load

In Software Engineering, Performance testing is necessary before marketing any software Next 🔰

Hacking

 $\otimes$