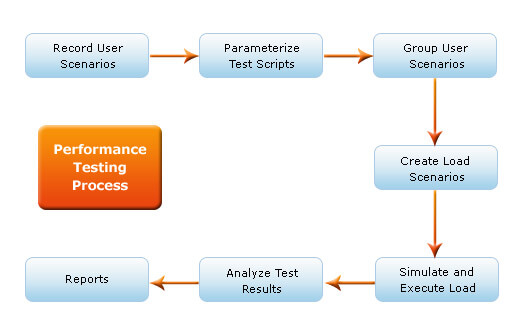
Snehal Thomake

Batch=7670

Date= 28/10/2022

Performance Testing Steps



Also known as the test bed, a [testing environment](https://stackify.com/azure-devtest-labs-test-env-dev-machines/) is where software, hardware, and networks are set up to execute performance tests. To use [a testing environment for performance testing](https://msdn.microsoft.com/en-us/library/bb924376.aspx), developers can use these seven steps:

### 1. Identify the testing environment.

Identifying the hardware, software, network configurations and tools available allows the testing team to design the test and identify performance testing challenges early on. [Performance testing environment options](http://www.kualitatem.com/blog/performance-test-environment-setup) include:

* Subset of production system with fewer servers of lower specification
* Subset of production system with fewer servers of the same specification
* Replica of productions system
* Actual production system

### 2. Identify performance metrics.

In addition to identifying metrics such as response time, throughput and constraints, identify what are the success criteria for performance testing.

### 3. Plan and design performance tests.

Identify performance test scenarios that take into account user variability, test data, and target metrics. This will create one or two models.

### 4. Configure the test environment.

Prepare the elements of the test environment and instruments needed to monitor resources.

### 5. Implement your test design.

Develop the tests.

### 6. Execute tests.

In addition to running the performance tests, monitor and capture the data generated.

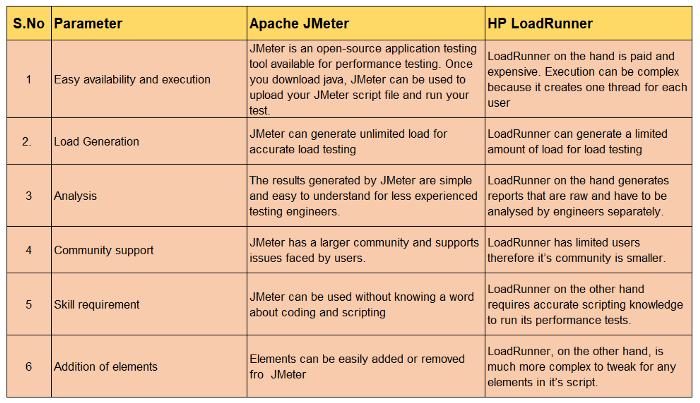
### 7. Analyze, report, retest.

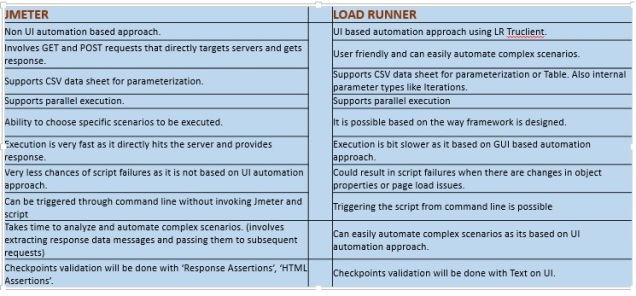
Analyze the data and share the findings. Run the performance tests again using the same parameters and different parameters.

Given below are the seven steps that are performed during performance testing:

1. First a proper requirement study is done, followed by analyzing test goals and objectives. Also, the testing scope along with a test initiation checklist is determined. Identifying the logical and physical production architecture for performance testing, identifying the software, hardware, and network configurations required to initiate the performance testing form an integral part of this step. Both the test and the production environments, while identifying the testing environment, are compared.
2. The desired performance characteristics of the application like response time, throughput, and resource utilization must be identified.
3. Some of the other key factors that need to be identified include key usage scenarios, determining appropriate variability across users, identifying and generating test data, and specifying the metrics to be collected. Finally, these items provide the foundation for workloads and workload profiles.
4. Before the execution of the test, the conceptual strategy, available tools, and the designed tests must be prepared. Also, the test environment must be configured. This includes preparing the test environment, tools and resources, necessary to implement each strategy as features and components are made available for the test.
5. A performance test needs to be created according to the test planning and design.
6. Executing the test comes next where the tests are run and monitored. Validating the tests, test data, and collecting the results are the next steps.
7. Then the results are consolidated and data is shared. The data must be analysed both individually and as a cross-functional team. The remaining tests must be reprioritized and re-executed. When all the metric values are within limits and accepted and none of them have violated the thresholds, the test in that scenario on that configuration is over. It is time to collect the desired information.







## Performance Testing Tools

### **1. WebLOAD**

Web application load and performance testing tool for enterprises. WebLOAD is the tool of choice for organizations with high user traffic and sophisticated testing needs. It enables you to load and stress test any internet application by simulating load from the cloud and on-premises machines. WebLOAD’s benefits are its versatility and ease of use, allowing you to quickly define the tests you require using capabilities such as DOM-based recording/playback, automatic correlation, and the JavaScript scripting language. The tool gives a detailed study of your web application’s performance, identifying flaws and bottlenecks that may be impeding you from meeting your load and response requirements.WebLOAD supports hundreds of technologies, ranging from web protocols to corporate

applications, and offers built-in connection with Jenkins, Selenium, and many other tools to enable DevOps continuous load testing.WebLOAD testing tool supports HTTP, HTTPS protocols and Enterprise applications, Network Technology as well as  Server Technologies.

Key Features:

* Correlation: Automatically correlates dynamic data such as session IDs, allowing scripts to run on different virtual clients.
* HTTPS, HTTP, and XML are all supported protocols.
* Integration: Works with technologies like Selenium, Jenkins, and others.
* Customer Service Representatives- You can contact them by phone, fax, or through a contact form.

### **2. LoadNinja**

It enables you to construct advanced load tests without using scripts and cuts testing time in half. It also replaces load emulators with real browsers and provides actionable, browser-based analytics at ninja speed. LoadNinja enables teams to expand test coverage without sacrificing quality by eliminating the time-consuming tasks of dynamic correlation, script translation, and script scrubbing. It supports HTTP, HTTPS, SAP GUI Web, WebSocket, Java-based protocol, Google Web Toolkit, Oracle forms. Engineers, testers, and product teams may use LoadNinja to focus on designing apps that scale rather than load testing scripts. Client-side interactions can be easily captured, debugged in real-time, and performance issues can be identified quickly. LoadNinja enables teams to expand test coverage without losing quality by automating dynamic correlation, script translation, and script scrubbing.

**Key Features:**

* Automated tests utilizing bespoke CI/CD plugins or a REST API.
* Customer Support: You can get answers from the LoadNinja user community or by reading their extensive documentation and FAQs.

### 3. LoadRunner

Micro Focus LoadRunner is a software testing tool. It is used to test programs, as well as to measure system behavior and performance under load. It can mimic thousands of users utilizing application software at the same time. Load Runner is a performance testing tool that is used to test the load on an application. In the load basically, there are three application processes; the Controller, Load Generators, and Load

Analyzers.LoadRunner supports all the protocols. It reduces hardware and software expenses by precisely forecasting system capacity. LoadRunner rapidly and accurately identifies the fundamental cause of application performance problems. It has effective tool utilization tracking, browser-based access to worldwide test resources, and optimized load generator farm utilization.

Key Features:

* It will support XML, which means we’ll be able to see and manipulate XML data in the test scripts with ease.
* It supports a wide range of applications, which cuts down on the time it takes to comprehend and interpret reports.
* We may generate extensive performance test reports with this tool.
* The expense of distributed load testing will be reduced.
* It will serve as an operational tool for tracking deployments.

### **4. Apache JMeter**

JMeter is an open-source performance and load testing tool that can be used to analyze and measure the performance of a wide range of services. This tool is mostly used in online and web service applications. It’s a Java platform program that’s an open-source load testing tool. It’s primarily used for performance testing, although it can also be combined with a test strategy. In addition to the load test plan, you can create a functional test plan. This tool can be installed on a server or network to evaluate its performance and see how it performs under various conditions. It was originally created to test web applications, but its capabilities have subsequently grown. It’s perfect for testing the functioning of Servlets, Perl Scripts, and Java objects. You’ll need JVM 1.4 or higher to operate it. Requirements for the system: It is compatible with both Unix and Windows operating systems.It supports HTTP,HTTPS,XML,SOAP,Java-based protocols and FTP protocols.

Key Features:

* HTTPS, HTTP, SAOP, XML, FTP, and more protocols are supported.
* PHP, NodeJS, ASP.NET, database, MOM, LDAP, Java objects, TCP, and other technologies are supported.
* Correlation: this allows you to extract data from popular
* response formats like HTML, XML, and JSON with ease.
* **Extensibility:** It can be used with pluggable samplers, Groovy-scriptable samples, pluggable timers, data visualization, analysis plugins, and more.
* Integrates with open source libraries like Jenkins, Gradle, and Maven, among others.
* **Customer Service:** You can interact with a vast community of developers and contributors. They’ve also kept tutorials up to date so you can get a better understanding of the product.

### **5. NeoLoad**

NeoLoad is the most automated performance testing platform for enterprises that need to test applications and APIs on a regular basis. NeoLoad offers testers and developers automated test design and management, the most realistic user behavior simulation, quick root cause analysis, and built-in integrations with the whole SDLC toolchain. From functional testing tools to analytics and metrics from APM tools, NeoLoad allows you to reuse and share test files and results. To meet all testing needs, NeoLoad supports a wide range of mobile, online, and packaged apps, such as SAP. Schedule, manage and disseminate test resources and findings across the organization on a regular basis to ensure application performance. System Requirements: This tool is compatible with Microsoft Windows, Linux, and Solaris operating systems.

Key Features:

* It works with HTML, Angular, HTTP/2, WebSocket, and other web frameworks and protocols, as well as packaged apps from Salesforce, SAP, Oracle, and IBM.
* Dynamic parameters are linked to automatic handling, and app-specific parameters are detected using established criteria such as.Net, Siebel, and JSF.
* SOAP/REST support, Selenium, Tricentis Tosca, Dynatrace,

Azure, Jenkins, Git, and other DevOps tools; SOAP/REST support, Selenium, Tricentis Tosca, Dynatrace, Azure, Jenkins, Git, and other DevOps tools.

### **6. LoadUI Pro**

LoadUI allows you to build and edit test cases as they run. The focus on usability through its visual interface and easy design, combined with the flexibility that comes with the ability to make changes during the test, is what makes load UI so strong. LoadUI Pro enables you to rapidly construct sophisticated load tests without using scripts, distribute them on the cloud using load agents, and track the performance of your servers as the demand on them increases. You can get thorough reports and automate your load tests rapidly.

**Key Features:**

* **Compatibility:**Mac OS, Windows, and Linux are all supported.
* **Test Reuse:** You can save time by reusing functional tests that already exist in your pipelines.
* **Integration:** Works with SoapUI, a functional testing tool.

### 7. LoadView

Using LoadView’s enterprise-level platform, enable your IT team to stress-test your websites, APIs, and web applications with thousands of simultaneous connections in real browsers.

LoadView’s cloud network is managed by AWS and Azure, allowing you to create multiple tests on even the most complicated projects. Utilizing load injectors from 30 worldwide locations spanning the US, South America, Canada, APAC, and Europe, you may define users, duration, and behavior using various scenarios and realistically imitate people. To analyze traffic spikes, scalability, and infrastructure restrictions, the tool includes three load curves: Load Step curve, Dynamic Adjustable curve, and Goal-based curve.

**Key Features:**

* LoadView has dedicated IPs that you may allow and control, so you can run tests behind a firewall.
* Video recording: Use video recording to capture the rendering of a website or app for better inspection and assessment.
* Reference servers, thorough waterfall charts, dynamic variables, and load injector controls are among the other features.

### Storm Forge

### StormForge enables you to execute load testing for the speed and scalability of your apps at a minimal cost, right within your CI/CD workflow. It enables you to boost application uptime, throughput, latency, and application faults while also allowing you to scale to additional users. The application provides all of these capabilities while using fewer resources, requiring no manual processes, promoting environmental sustainability, and assisting you in lowering your monthly cloud expenditures. To ensure that the test reflects real-world traffic patterns, you can capture real-world traffic. It operates on an open-workload model, accurately models real-world scenarios, and solves error detection issues.

Key Features:

* Use Performance Testing as Code in your CI/CD process to make it more repeatable.
* Cloud-native: On Kubernetes, it works nicely.
* Java, Nginx, Go, and Python is among the supported programming languages.
* Integration: It works seamlessly with your ecosystem, including cloud providers (AWS, DigitalOcean, GCP, IBM, Azure), monitoring tools (Prometheus, Dynatrace, Datadog, New Relic, and Circonus), and DevOps tools (Prometheus, Dynatrace, Datadog, New Relic, and Circonus) (Jenkins, Puppet, Chef, and Rancher Labs).

### 9. LoadComplete

It’s yet another tool for performance (load) testing. It’s used to build and perform automated tests for web services and servers. It works with all browsers and web services. When we have a large load, it will examine the performance of our web server. Throughout the test runs, we can use this program to monitor numerous server metrics such as CPU utilization.

Key Features:

* It will allow us to produce a big load for stress testing by providing load modeling for performance testing.
* We may record and playback our actions in the web browser using this.
* It works on a variety of platforms, including Windows and UNIX.
* During load testing, template-based criteria will be used to evaluate the server message body, ensuring that the server is running properly.
* It can test Flash, Flex, Silverlight, and Ajax apps, among others.
* It will generate load test reports, which will contain user interface customization.

### 10. Gatling

Gatling is an open-source performance and load testing tool for online services, mostly apps, that was launched in 2012 and built-in Scala. It allows you to minimize crashes by anticipating crashes and poor response times, as well as to detect issues early on to save time to market, improve user experience, and grow your business. Gatling’s code-link scripts make it simple to manage and automate test scenarios. It’s designed for continuous load testing and may easily be integrated into your development cycle. A web recorder is also included.

Key Features:

* Readable DSL: Their domain-specific language (DSL) makes it simple for anyone to read test scenarios.
* Analysis: Simulate thousands of user requests per second with ease and get very accurate stats in a colorful and dynamic report.