

Infrastructure Setup for Software Testing

The 'Test Bed Setup', an initiative of Software Development Unit (SDU) at NIC HQ, New Delhi, comprise of six high end servers in Linux and Windows environment. These servers can be used as staging environment temporarily, for deployment of applications under testing apart from generating virtual users for Performance Testing.

Edited by
Vivek Verma

Software Testing is a process that provides information about the quality of a product or service under test. It is tested to determine how it performs in terms of responsiveness and stability under an expected workload. It also serves to investigate, validate or verify other quality attributes of a system, such as scalability, reliability and resource usage. Some form of testing can be performed at every stage of software development life cycle. Since the cost of defect resolution is inversely proportional to the stage at which it is captured hence the primary aim of software testing is to identify them as early as possible to minimize the expense and time.

Depending upon the methodology, approach and the stage of project execution, various ways of classifying the testing process are -

1. White box - tester has access to the code and algorithm implemented.
2. Black box - tester examines the software without any knowledge of the internal implementation.
 - a. Performance Testing - where the system is checked for the responsiveness and stability under a particular workload
 - b. Functional testing - where the expected functionality is verified against the software specification.
3. Unit or component testing - when the individual components of software is under test purview,
4. Integration testing - when the test is being done to check only the interface between different units
5. System testing - where the completely integrated system is tested for the expected functionality with optimum performance yardsticks.

The scope of this article is to cover

Functional Testing and Performance Testing, usage of various tools and the Test Bed Setup, created at Shastri Park, Data Centre at New Delhi.

FUNCTIONAL TESTING

The Functional Testing is done to ensure that the system requirements and/or specifications have been implemented in the application. This can either be requirement based testing (also called reliability testing) to match the functionality vis-à-vis the requirements mentioned in the specification testing (as in User Acceptance Testing) or regression based testing wherein the tester ensures that new changes to the application have not impaired existing functionality, thereby minimizing the ripple effects.

Tools like Rational Functional Tester is a software test automation tool used to perform automated regression testing. Testers create scripts by using a test recorder which captures a user's actions against their application under test. The test script is produced as either a Java or Visual Basic.net application and is also represented as a series of screen shots that form a visual storyboard. Testers can edit the script using standard commands and syntax of these languages or by acting against the screen shots in the storyboard. Test scripts can then be executed by Rational Functional Tester to validate application functionality.

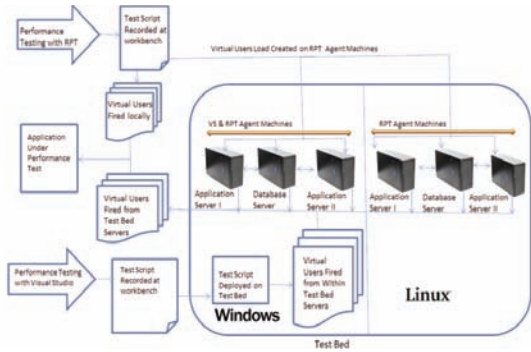
During the recording phase, the user must introduce verification points. Verification points capture an expected system state, such as a specific value in a field, or a given property of an object, such as enabled or disabled. During playback, any discrepancies between the baseline captured during recording and the actual result achieved during playback are noted in the Rational Functional Tester log. The log can then be reviewed to determine if an actual

MS RACHNA SRIVASTAVA
Senior Technical Director
rachna_sri@nic.in

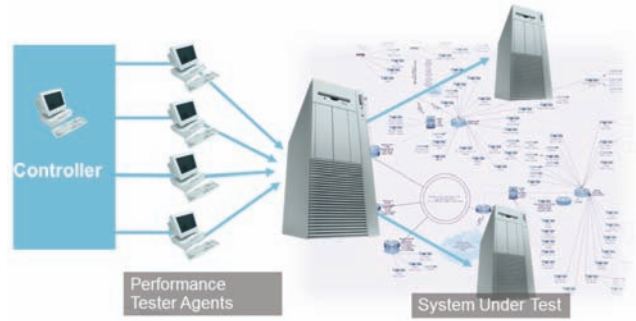


MR. ALOK PATHAK
Senior System Analyst
NIC, Shahstri Park
alok.pahak@nic.in





Performance Testing using the Test Bed setup



Performance testing illustration

software bug was discovered.

COMPONENTS IN FUNCTIONAL TESTING

Storyboard Testing enables testers to edit test scripts by acting against screen shots of the application. This is akin to modifying the test script code.

The Object Map is automatically created by the test recorder when tests are created and contains a list of properties used to identify objects during playback.

Script Assure technology enables Tester to ignore discrepancies between object definitions captured during recording and playback to ensure that test script execution runs uninterrupted.

Data Driven Testing enables tester to add additional test data cases to the test data pool without having to modify any test code. Object Proxy Mechanism allows users to program in Java or .NET to add functional testing support for Java and .NET custom controls.

PERFORMANCE TESTING

Executed before the deployment of the application in production environment, it is a vital sub-component of the software development life cycle as it gives clear idea about sustainability of the application when exposed to concurrent user loads expected in the real time environment.

Tools like IBM Rational Performance Tester, Visual Studio Ultimate 2010 etc are used for examining system behaviour while generating actual load. The functionality to be tested is recorded as test scripts which can be scheduled to be played back by virtual users thus generat-

ing the expected user load. This also examines the run time quality aspects of the application in terms of response time, CPU and memory usage.

TEST CONTROLLER OR TEST WORKBENCH

This can be a local machine where the test script is created, by recording the functionality to be tested using the wizard or guided steps available in the tool. The test script can also be created from scratch in programming languages supported by the testing tool or by using some third party tools like Fiddler.

The Test Controller or Test Workbench is also used to configure the load distribution on local or remote machines. For greater user load, the test is configured to execute on remote machines running the Agent software. The user load can be applied all at once or with appropriate staging intervals with suitable 'Think Time' between test iterations to simulate the actual real time test load scenario.

Agents

The load test agent machine is a remote machine with high end configuration that runs 'Agent' Constituent of the Testing suite. From the controller or workbench machine, the Virtual User load is distributed to multiple agent machines and thus the test is virtually fired from remote locations. The number of virtual users can also be increased or decreased dynamically while the test is running. Once the test is complete, the Agent machines send their results to the test controller that

collects this data to create a comprehensive performance report.

iTest Bed Setup

The 'Test Bed Setup', an initiative of Software Development Unit (SDU) at NIC HQ, New Delhi, comprise of six high end servers in Linux and Windows environment. These servers can be used as staging environment temporarily, for deployment of applications under testing apart from generating virtual users for Performance Testing.

For User load up to 1200, the Performance Tests can be carried out in Rational Test Environment

- Test script can be created on the local machine with Rational performance tester
- Performance test can be configured to be executed on RPT agents either locally or from the Test Bed Setup.

For User load of greater than 1200, the Performance Test can be carried out on Visual Studio Ultimate Testing suite, installed on the Test Bed Setup.

- Test script can be created on the local machine with Visual Studio Ultimate 2010.
- Test Script is then deployed on the Visual Studio Controller installed on the Test Bed Setup.

For further information

MS RACHNA SRIVASTAVA
Sr. Technical Director
7th Floor, NIC Head Quarters,
CGO Complex, Lodhi Road
rachna_sri@nic.in