Fortuitous Technologies, Inc
http://fortuitous.com
1-512-351-7783

The Importance of IT Systems Testing – Cost, Benefits, Tools, and Impact

Dominique A. Heger, Phil Carinhas, Fortuitous Technologies, Austin, TX

Introduction

One of the most neglected aspects of many IT operations is the stress testing, load testing, scalability testing, and overall benchmarking of server systems. Most IT organizations claim that testing server systems is regarded as additional work (overhead) that is not planned for and not billable. Further, the lack of tools and IT processes that facilitate systems testing results in many IT organizations abandoning the idea of incorporating systems testing into the IT business plan. The result of all this is that IT organizations are trapped in a vicious cycle of troubleshooting HW stability issues and fire fighting application and operating system performance problems. Hence, the ramifications are poor user acceptance of the IT infrastructure, a loss in productivity, and ultimately a dip in business revenue.

The reluctance of IT organizations to embrace systems testing as a strategic IT function results in missing a very important business point. A sound IT testing strategy allows a business to gain a competitive edge, to stand out from the competition, attract, win new, and keep customers and business, strengthen and improve existing business relationships, and ultimately safe both, time and money to operate the IT infrastructure. In general, an IT test strategy has to be in place as:

1. Testing the server systems and user requirements guarantees the ultimate user satisfaction. In the case a test plan is in place, the end-user community and IT organization are in sync. The appropriate systems requirements are being scrutinized and tested, and hence, the IT organization is able to deliver the systems and applications on time and within budget.

2. Testing the systems prior to deployment minimizes the IT hours necessary for handling support calls and troubleshooting exercises. Similar to the process of software development, the sooner an error is detected, the easier and more cost effective it is to resolve the problem. Even for smaller IT organizations, the same law of economics applies.

3. End-users and management alike appreciate the security and added value of a well-defined, well-executed, and well-documented test strategy. For any IT organization, what matters most is the quality of the product that they deliver to the user community. Having a sound test plan in place *certifies* the IT equipment, and clearly communicates to the end-users and upper management what is planned, build, and ultimately delivered.

IT Systems Testing

HW component testing involves verifying the systems building blocks, such as the CPU's, the memory subsystem, the IO channels, or the network interfaces to insure that they meet the user community's needs and requirements. Sometimes, it is very beneficial to verify (via testing) some of the manufacture's claims that affect user requirements, a process that safes both, time and money. This is especially true when the HW is new, and the claims are subject to interpretation. To illustrate, a 4Gbit/second Fibre Channel (FC) Host Bust Adapter (HBA) used to connect to a SAN has a bandwidth of 500MB/sec, but protocol overheads limit the actual throughput to approximately 400MB/sec. In any case, testing the HW allows insuring that the actual configured FC HBA in a system is even *capable* of achieving the 400MB/sec throughput target.

Unit and system testing revolves around testing the systems subsystems as they work in conjunction with each other. In other words, testing the CPU, memory, and IO subsystems together (by utilizing a workload generator) allows insuring that all the individual components are able to operate

efficiently and effectively as a unit (the HW and SW). Unlike a HW component test that focus on the individual building blocks, the unit and system testing process focuses on the entire environment to confirm that what was designed, setup, and implemented actually matches the user requirements and system specifications. Depending on the IT organization or the IT project, the result of the unit and system testing phase can be described as an actual *acceptance test*. If there are (performance, reliability, availability, or maintainability) requirements stipulated in a test document, this test phase verifies that all expectations are met. The recommendation made is to always formulate these requirements up-front, and to meticulously insure that the goals are met or exceeded.

If problems are identified during the systems testing phase (and they will), the implemented test process worked. Identifying systems issues during testing is considered a success. By effectively testing the systems, IT organizations prevent customer dissatisfaction due to faulty HW or SW components, broken interfaces, or under-performing applications. Further, identifying the issues in the test phase safes time, aggravation, and embarrassment of having to constantly repair the systems while they are in production. Overall, a sound test plan allows an IT organization to focus on its core business objectives and not on costly fire fighting and troubleshooting exercises. While identifying systems issues may seem like bad news, experience has shown the opposite, as management prefers to hear that the IT organization has uncovered and resolved some problems during the test phase, compared to having to report systems issues while the systems are in production. The most important factors of an efficient and effective test process can be summarized as:

- It is paramount to insure that an IT test plan is in place
- The goals and expectations have to be documented and understood by all the involved parties
- The test process (methodology and tools) is used to verify the test plan
- Utilize a tool that allows testing all the components of the environment (synergy effect)
- The progress made while testing the systems has to be well documented
- An organization should make testing an official part of any systems deployment project

Testing adds value and credibility to a company and discloses and underlines the professionality of its IT department. IT systems testing can be decomposed into the following *testing disciplines*:

- *Performance Testing* is the critical component of an entire testing process. It determines the actual operational boundaries of an environment and simulates the real world usage of the applications, the OS, and the HW components, respectively. In general, performance testing can be divided into load, stress, and scalability testing.

    - *Load Testing* determines the systems behavior under various workloads. The objective of load testing is to quantify how the systems components react as the workload is gradually increased.

    - *Stress Testing* determines either the breaking point of a system or the area where performance becomes unacceptable. In other words, stress testing reveals the maximum service level that a system can achieve.

    - Scalability Testing evaluates the effects of adding additional hardware (physical resources) and/or software (logical resources) components to a system to distribute the work differently among the system components.

Testing Tools

Utilizing a sound, flexible, and portable testing tool is paramount to conduct comprehensive testing projects. Fortuitous' Valeo benchmarking suite allows IT departments to understand how the major OS abstractions depend on the speed and availability of the underlying hardware platform. The individual benchmarks that comprise Valeo can be used as a benchmarking, load testing, stress testing, and scalability

testing tools, respectively. In other words, Valeo incorporates all the testing disciplines in one product that can be used on UNIX and Linux based systems. Valeo was designed and implemented in a way that it supports a multithreaded programming model (for scalability testing), reflects the state of the current hardware and software technology, and supports SMP, N-Tier, and cluster/GRID environments.

Valeo consists of 27 individual micro-benchmarks that stress and load test the major system's components such as the CPU, the memory, the IO, and the communication subsystems, respectively. The micro-benchmarks can also be used to study the scalability behavior of the infrastructure and are divided up into throughput and response time related modules. The micro benchmarks can be executed either via scripts (as a unit), or can be invoked individually from the command line. Further, Valeo provides 4 individual macro benchmarks that stress test the CPU, the memory, the IO, and the NW subsystems. The 4 macro benchmarks can be individually customized to reflect the actual workload (application) behavior at a customer site. The combination of the micro and macro benchmarks allow quantifying the performance capacity of an IT infrastructure, conduct cross-platform comparisons, identify subsystems that perform sub-par, optimize applications, as well as to track and identify the impact that new HW or an OS upgrades have on overall systems performance and stability. Please see www.fortuitous.com for more information on Valeo.

Summary

IT systems testing is a process of technical investigation that is intended to reveal quality-related information about the IT environment with respect to the context in which it is intended to operate. The intention of the IT testing process is to identify errors, quantify the capacity potential of the environment, and to determine if the systems meet user expectations. Testing adds value and credibility to any IT project, improves user satisfaction, and minimizes the IT hours necessary to troubleshoot and fire fight systems issues in a production environment. Next to the testing process, a high quality testing tool is required to accomplish the goals of IT systems testing.

References

- *E.Dustin: Effective Software Testing. Addison Wesley, 2002*
- *E. Dustin, et al: Automated Software Testing. Addison Wesley, 1999*
- *Myers, Glenford., The Art of Software Testing. John Wiley and Sons, 1979*
- *M. Fewster, Dorothy Graham: Software Test Automation. Addison Wesley, 1999*
- *R. Black: Managing the Testing Process. Second Edition, John Wiley and Sons*