A Test Integration Approach for Network Performance Modeling

Nita Goel, Banafcheh Youssefi, Ramalingham Swaminathan, Baljinder Sandhu
Cistel Technology, 210 Colonnade Road, Suite 204, Nepean, ON
(Nbansal,bani,rswami,baljinder)@cistel.com

1. Introduction

This abstract presents an incremental integration testing solution that will introduce a set of methods and tools to customers for monitoring the performance of their network and their applications from an end-user's viewpoint. An improved level of delivered services and a predictable operating environment intends to be provided to customers.

2. Background

The following computational environment (Figure 1) describes the working relationships of the three groups: Development, Testing and Production.

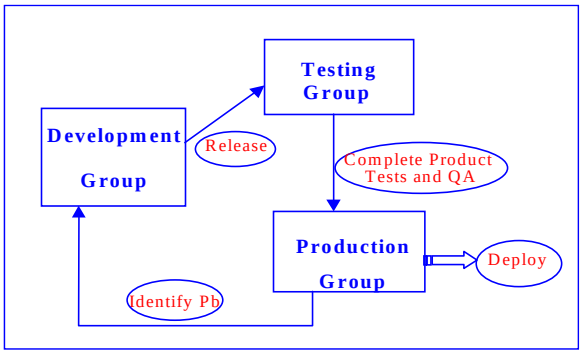

Figure 1. Computational Environment Organization

The production group is responsible for the deployment and operations of the applications that run the business processes as well as the operation of the infrastructure that supports them. In the case of failure of any of the software operations, the problem is identified and most of the time resolved by the operation group. If not resolved, the problem is referred to the development group. Depending on the problem, the development group can decide to fix it and send a patch back to the Production group. Otherwise, the fix is delayed for a deployment with the next release. Upon the completion of the development, the software products are sent to the testing group to run module testing, component and application testing. The testing group also performs quality and integration testing at the product level in the lab. Upon approval of the application, the software is passed to the production group for deployment on the network.

An inclusive integration testing facility has to be provided to enable the measurement of the running applications and the management of their quality of service (QOS). The focus of the solution is to monitor the computing infrastructure, which includes the measurement of the response time service level thresholds as well as the actions to take when objectives are not met.

3. Issues

In order to understand if the current network performance levels are typical, if the behavior is aberrant, or if change has occurred, the production group needs to understand the normal behavior of its network and its applications. For this purpose the network load as well as the applications behavior need to be characterized. The following key elements are developed:

- Selection of the list of resources to monitor
- Identification of the list of parameters to monitor on each resource/application
- Definition of the baseline intervals according to the nature of the resource observed Daily (hour intervals), weekly (day or hour intervals) and monthly are typical cycles
- Building the monitoring models
- Deployment and set-up of the monitoring models

4. Modeling Methods

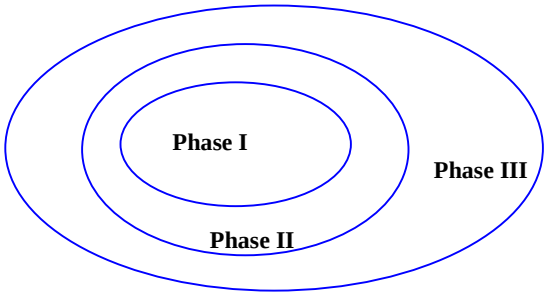

Figure 2. Major phases of the development

Network performance can meet performance targets (aka Quality of Service (QOS)) by careful design of the network, by collecting network statistics for a healthy network (called baselining),

and by regularly monitoring the network to ensure the current network performance falls within the baseline.

The work involved in achieving these objectives is divided into the following phases (Figure 2):

4.1 Phase I: Sample Integration Testing Model: The focus is on building a test integration model for a subset of the network infrastructure. The purpose of this phase is to build a baselining model of a subset of the network and to show the feasibility of the performance baselining. This phase includes the characterization of the typical load of the selected infrastructure. This requires monitoring various components of the selected subset of the network from which a baselining performance picture will be drawn. The baseline will describe the typical computational figures of the selected subset.

4.2 Phase II: Network Integration Testing Model: works on designing and building a monitoring model of the whole network. Phase II also works on designing and calibrating a prediction model to forecast the capacity of the network. This phase reuses the modeling approach of Phase I by broadening assumptions about the network devices to be included in the integration testing.

4.3 Phase III: Application Integration Testing Model: works on designing and building the model that assesses the impact of deploying any new software component in the network. In particular, the focus is on the measurement of the applications from the user's point of view. The model identifies applications bottlenecks and congestion points and provides recommendations on what actions to take to achieve better service levels objectives.

5. Conclusions

Once the experience with Phase I – III is assessed, the following tasks are performed:

- Increase/tune the number of network elements tested
- Increase/tune the number of servers tested
- Increase/tune the number of clients/application tested
- Increase the likelihood of operational problems being found (by improving the software model to focus on problems with large operational impact)
- Methods and setup to determine if the ND (Network Device) meets minimum performance criteria for placement into the production network
- Methods and setup to determine if the ND has consistent and reliable behavior under loaded traffic conditions
- Establish Pass/Fail decision-making criteria for deployment of NDs into the production network
- Methods to add network elements in order to reduce capacity problems

6. References

[1] P. Oppenheimer: *"Top-down Network Design"*. Cisco Press book.

[2] I. Turek, "SMS 2 Exam 70-086", The Corliolis Group.

[3] M.W. Murhammer et al., "TCP/IP Tutorial and Technical Overview", 6th, ISBN 0-13-020130-8

[4] "CSC Year 2000 Test Lab", obtained from Ron Church (Dec. 1999)

[5] Microsoft Visual Basic 5.0 Developer's Workshop, 4th ed.