Partner Solution
for Microsoft® Systems Architecture
Internet Data Center

Version 1.5

Chapter 12
Testing Your Implementation

Prescriptive Architecture Guide

Abstract

This chapter outlines the steps necessary to verify that a deployment of a base Microsoft® Systems Architecture (MSA) Internet Data Center (IDC) is correctly done and performs to the required base specifications. The procedures in this chapter may be modified to accommodate any departures from the base configuration as needed, or a base environment may be built and tests may be performed on it before deviations are made.

CONTENTS

INTRODUCTION 1
Items to Be Tested 1

TEST EXECUTION 2
Build Verification Tests 2
Load Tests 2
Resource Requirements 3
Exerciser Application 3
Load-Generating Tools 4
Load Script 4
Load Client Machines 4
Load Client Network Connection Into the Internet Data Center 4
Other Considerations 6

PERFORMANCE AND SCALE OUT TEST RESULTS 7
Performance with Four IIS Machines 7
Scaling Out IIS Machines 8

APPENDIXES 10
Appendix 12.1 – ACT Script 10
Appendix 12.2 – Build Verification Test 10
Appendix 12.3 – Performance and Scale Out Test Results 10

INTRODUCTION

Testing should be done throughout the implementation process. Even before an application is installed, it makes sense to verify that the basic platform is capable of achieving its design capabilities.

The series of tests discussed in this chapter are provided to you as a guideline for your own build verification test (BVT) and performance capability tests. If these tests as written are carried out on an environment configured as described in earlier chapters of this guide, they all should be successful and performance numbers should come close to any published performance numbers. However, you may also use the information in this chapter as an aid to help you develop tests that are relevant to your own implementation, including its deviations from the one described in the prescriptive architecture.

Items to Be Tested

The test cases and performance numbers cited in this chapter apply to configurations that contain only the hardware and software configured and installed as described in this guide. These include network devices, firewalls, all servers, Microsoft SQL Server™ 2000 cluster, Microsoft operations manager, and NetIQ AppManager.

The load tests are designed to confirm the system's high availability under load and to provide an end-to-end base system capacity you can compare against the known baseline established during load testing at Microsoft. The tests include a simple scale-out test of 1 to 33 IIS servers to determine scaling linearity.

TEST EXECUTION

To validate a base implementation you should run two types of tests, a build verification test (BVT) and performance capability tests. The BVT is a way to verify that the various components of your implementation were installed and configured as documented. Although not perfect, it is an effective double check that everything is as it should be by running a suite of performance tests.

The sequential testing phases are summarized below:

- Conduct network validation testing
- Conduct security validation tests
- Conduct no-load availability tests
- Conduct no-load database cluster failover tests
- Conduct management validation tests
- Conduct availability tests under load
- Conduct database cluster failover tests under load
- Conduct end-to-end load capacity and web scale-out tests

Build Verification Tests

The build verification test (BVT) contains a series of audit steps and test procedures that ensure your team has installed and configured the Internet Data Center environment as documented. The test includes steps to verify that your systems are compliant with Microsoft's Strategic Technology Protection Program (STPP). It also includes basic failover tests for all redundant system components.

The BVT used for this Prescriptive Architecture Guidance (PAG) is included in Appendix 12.2.

Load Tests

Once you have verified that the basic environment is in place and works, it is time to run load tests. These tests will tell you what the capacity of your system is—as well as expose configuration errors that are otherwise hidden. The tests simulate actual extreme live conditions that your system may experience, permitting you to see problems before they are exposed in production.

The tests are arranged so that you first generate load with simple objectives. If problems with your implementation exist, most will be exposed at this point. The next few tests repeat previous failover tests, but instead of running with a single tester's load they run with a realistic or stressful load level, simulating failovers in the "worst situations." Finally a simple web scale-out test allows you to determine the linearity of your front end and to verify the performance numbers of your back-end servers when subjected to very high loads.

To run load tests that can be compared directly against the results included at the end of this chapter, you will need the same "exerciser application" used in the original testing. In addition, you will need some kind of Web load-generating tool, such as Microsoft's application center test (ACT) tool that is bundled with Visual Studio .NET or Mercury Interactive's LoadRunner tool. Both load tools were used in testing this PAG.

The load-generating tools are used to simulate thousands of very impatient web users, all accessing information from your site. By directing the load tools to target the exerciser application, diagnostic and repeatable measurements can be made of the Internet Data Center environment under load conditions similar to, or worse than, the conditions it will be exposed to in real life.

Your own performance numbers may not match the ones reported here even if you implement what appears to be exactly the same system. Because of the complexity of the environment, minute differences in hardware characteristics and other factors will introduce differences in your actual system response characteristics. Your performance will be close, but not the same.

The load tests used for this PAG are included in Appendix 12.2.

Resource Requirements

The only resource required to run BVT tests is a PC that can be used to simulate an Internet user and that can also reach the VPN servers in the Internet Data Center. The PC needs to be loaded with a Microsoft Terminal Server client and with an Internet Explorer browser.

Additional resources are needed to run load tests. These are discussed in the following sections.

Exerciser Application

The exerciser application used for the tests described in this chapter was a small pseudo-bookstore application named "Nile." It is a well-understood application that is frequently used in benchmarking dynamic web application environments. Documentation about Nile, its use in other benchmarking studies, and its source code can be found on the Microsoft MSDN web site by navigating to the MSDN Code Center and then to the "Sample Applications and Reference Architectures" area.

The version of Nile code used was one that used ASP, VBasic, and COM+ objects, and ran entirely on the IIS machines, not using the application servers.

Load-Generating Tools

The first load-generating tool used was the Microsoft application center test (ACT) tool, which is bundled with Visual Studio .NET. The script used for this testing is included in Appendix 12.1.

Another optional load-generating tool used in testing this PAG was Mercury Interactive's LoadRunner. This is a more comprehensive load-testing tool, but is not required to run the tests listed here.

Load Script

The load script emulates three types of visitors to the "Nile Bookstore," browsers who simply look at books, buyers who browse for books and then purchase some of them, and changers who change a part of their account information. For these tests the roles were distributed so that about 70% of the users were browsers, 30% of the users were buyers, and 10% of all browsers and buyers change information about their account. The amount of browsing is randomly distributed between one and five catalog pages. Each pass through the script constitutes a single user session.

Load Client Machines

Loads need to be generated from a set of test clients. They can be almost any moderately powerful PC or server. Often it is effective to use retired rack-mounted servers for the job, since load generating software does not require the highest powered hardware to generate huge loads. For the tests run in this PAG, 18 otherwise obsolete servers were used; in this case generic 2-processor Pentium II web servers with 500 Mbytes of RAM, 4 GB SCSI hard drives and dual NICs. They generated saturation levels of load at less than half of their capacity.

Load Client Network Connection Into the Internet Data Center

The BVT and load clients need to be on their own VLAN or subnet and configured to appear to the Internet Data Center environment as if they were "outside" machines on the Internet. The following diagram shows the network topography used for the testing of this PAG.

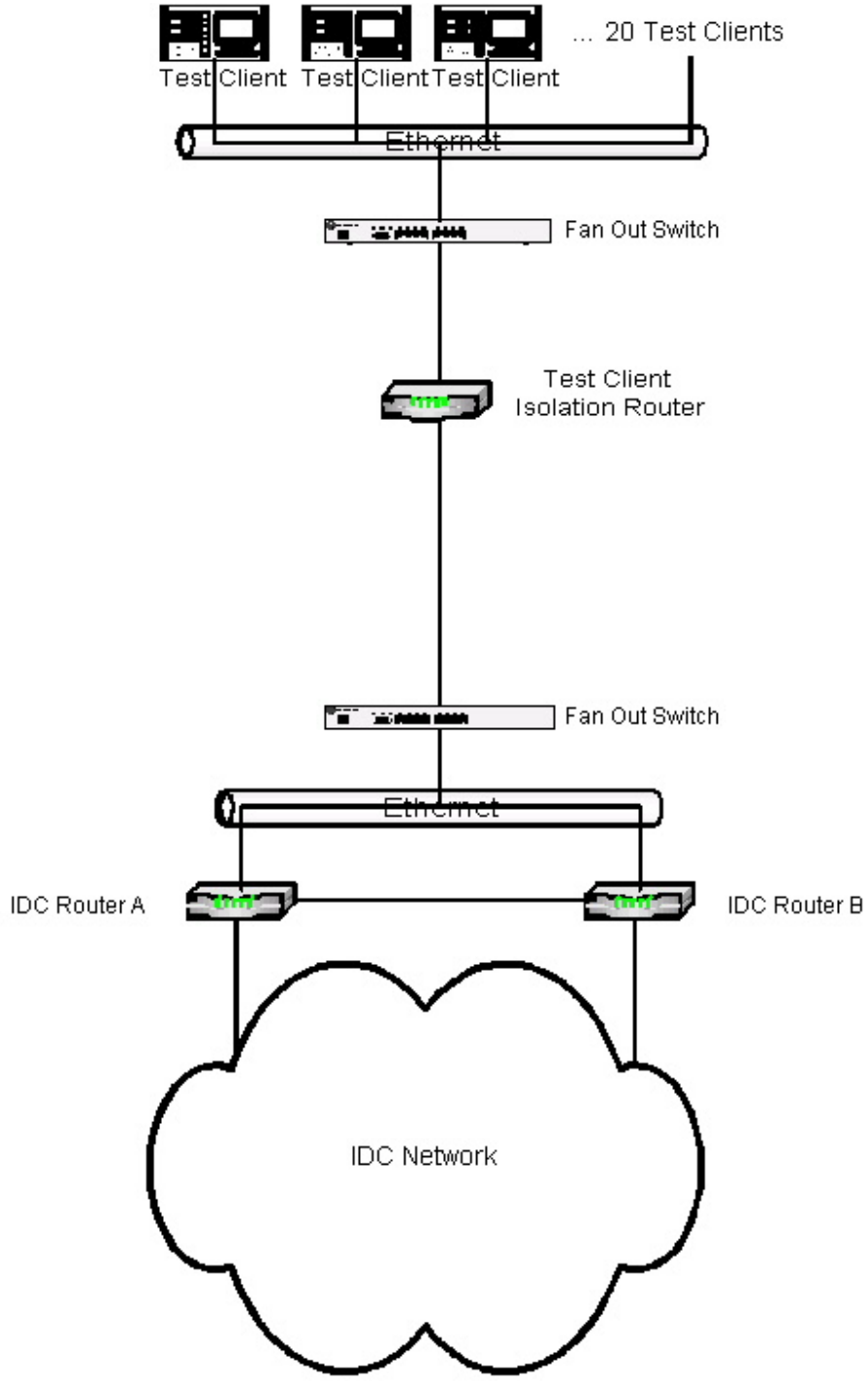


Figure 12.1 Network Topography for PAG Testing

Other Considerations

The actual testing carried out for this PAG went far beyond the scope of the tests listed here. It included detailed cases designed to thoroughly explore possible issues arising from the integration of the components of this PAG. Because testing was done to this level for the PAG, you do not have to subject the parts of your system that match this PAG to the same depth and breadth of testing.

If you are departing from this PAG, you should design new test cases to validate your configuration, test the integration of the non-PAG components, and generate appropriately different end-to-end load scripts tailored to the various uses of the actual applications that will be run on the system.

You should also follow the processes documented in Chapter 9, "Designing Your Testing Process," of the *MSA Internet Data Center Reference Architecture Guide*. This includes producing a Test Plan, establishing a triage process for test and development staff, acquiring and implementing a bug tracking system (and associated processes in test and development to manage bugs and their correction), and establishing a well-understood and agreed-upon release criterion to determine when your Internet Data Center is ready to be put into production.

PERFORMANCE AND SCALE OUT TEST RESULTS

Performance tests were carried out on an instantiation of this PAG using the setup described previously. This section summarizes the results of these tests. Detailed results are in Appendix 12.3.

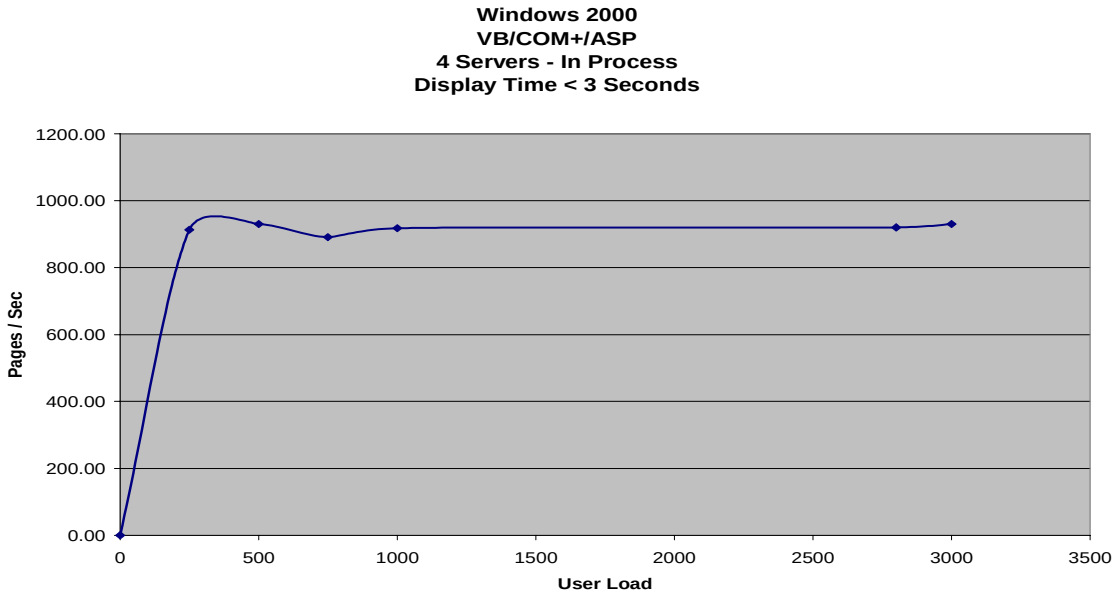
These tests were patterned after the Doculabs report on Windows 2000, @Bench Test Report: Performance and Scalability of Windows 2000 by Marianne Pendleton and Gautam Desai . This report can be found on Microsoft's MSDN web site by navigating to the MSDN Library and following the links to "Enterprise Development", then continue drilling down to "Enterprise Development Sample Applications", "Nile.com", "Nile.com 1.0" and finally "Architectural Decisions for Dynamic Web Applications: Performance, Scalability, and Reliability". A link to this report is on this page.

For all load tests there was no pause between pages. Servers were considered "saturated" when the time-to-last-byte of any page exceeded three seconds.

Performance with Four IIS Machines

Figure 12.2 illustrates the number of pages displayed per second as a function of the simultaneous user load. It shows how the number of pages per second rises rapidly with the number of users and then stays constant—in the case of four IIS machines at around 1400 pages/second. What cannot be seen in this plot is the fact that, as the number of users increases, the time-to-last-byte increases also. The maximum number of users that can be accommodated by four IIS machines in the base configuration of the Internet Data Center is about 4250. This compares almost exactly with the results reported in the Doculabs report.

Figure 12.2 Windows 2000 VB/COM+/ASP performance with four servers – in-process display time less than three seconds



In Figure 12.3, it is obvious that IIS and DNS servers' total CPU percentage is at 100%. This is common behavior for an IIS machine and does not reflect its true load. Other counters for other back-end machines show stable, flat numbers for increasing load, indicating that when using the Nile application, four web servers does not stress the rest of the Internet Data Center environment at all.

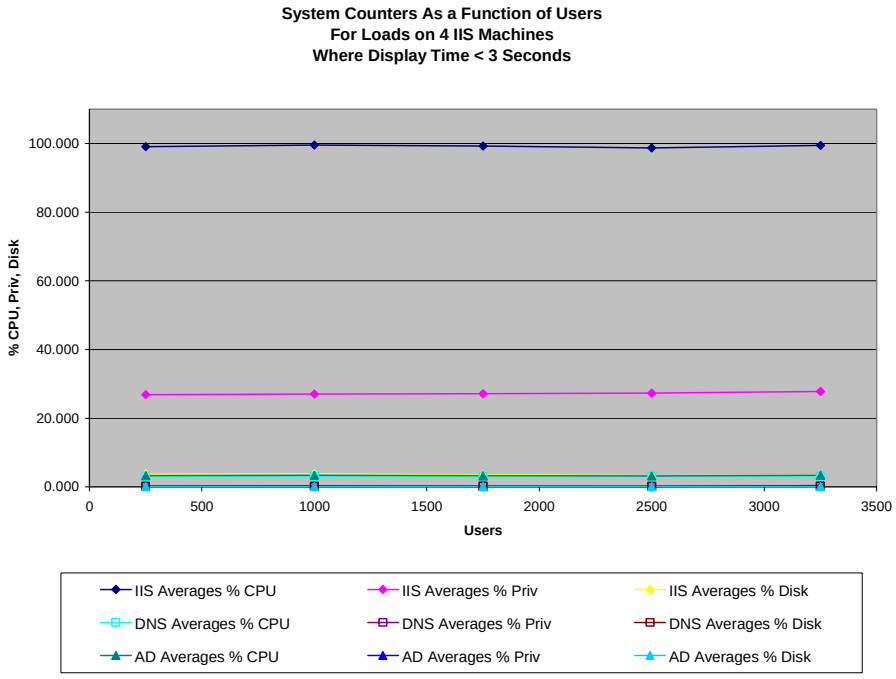


Figure 12.3 System counters as a function of users for loads on four IIS machines where display time is less than three seconds

Scaling Out IIS Machines

Figure 12.4 shows the saturation points where the time-to-last-byte for pages begins to exceed 3 seconds for 1, 2, 4, 6, 8, 16 and 33 IIS machines. The increase in capacity is almost linear and doubling the number of IIS machines nearly doubles performance. The peak value of 11,520 users retrieving 2484 pages per second corresponds to 8,942,400 page views per hour.

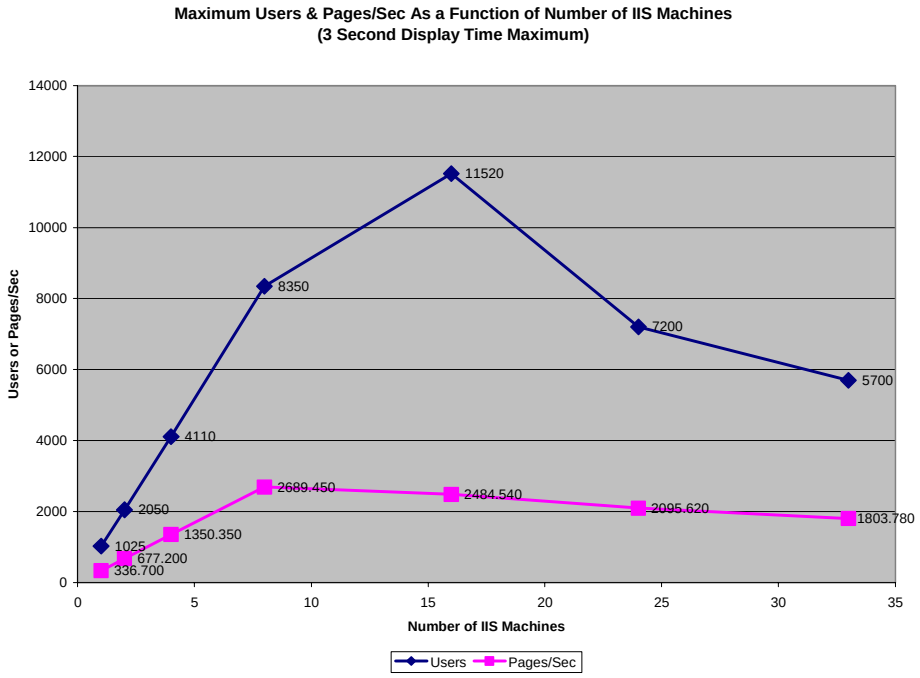


Figure 12.4 Maximum users & pages/sec as a function of number of IIS machines (three-second display time maximum)

APPENDIXES

This section provides details of the appendix files that are provided as part of the Internet Data Center architecture documentation

Appendix 12.1 – ACT Script

This appendix provides a script that can be used with the load-generating tool Microsoft's Application Center Test (ACT), which is bundled with Visual Studio .NET. The appendix file is *ACT Script.txt*.

Appendix 12.2 – Build Verification Test

This appendix is the build verification test (BVT) used for the Internet Data Center. It contains a series of audit steps and test procedures that ensure your team has installed and configured the Internet Data Center environment as documented. The appendix file is *IDC-Build Verification Test.xls.*

Appendix 12.3 – Performance and Scale Out Test Results

This appendix provides the detailed IDC test results. The appendix file is *C-Performance and Scale Out Test Results.xls*.