



Edison Group, Inc Windows Server 2003 NetBench Performance Report

Windows Performance Analysis

For

Microsoft

January 31, 2006

Printed in the United States of America.

Copyright © 2005 Edison Group, Inc. New York. Prepared for Microsoft. Edison Group offers no warranty either expressed or implied on the information contained herein and shall be held harmless for errors resulting from its use. Microsoft is distributing this document and offers no warranty either expressed or implied on the information contained herein and shall be held harmless for errors resulting from its use.

All products are trademarks of their respective owners.

First Publication: January 2006

Produced by: Andrew Podensenin, Senior Analyst

Table of Contents

Performance Analysis Methodology	1
Test Lab Design	1
STRESS-TESTING SOFTWARE.....	1
STRESS-TESTING NODES AND CLIENT TUNING PARAMETERS	2
FILE SERVER HARDWARE AND TUNING PARAMETERS	3
NETWORKING TOPOLOGY	6
Stress Test	6
CONFIGURATION OF THE TEST MIXES.....	6
PRELIMINARY TESTS AND ASSUMPTIONS	7
ACCURACY AND VARIABILITY OF RESULTS	7
TEST RESULTS	7
ADDITIONAL INFORMATION.....	9

Executive Summary

Edison Group was hired by Microsoft to measure the file server performance of Windows 2003 64-bit server. The stress test was performed with the industry-standard NetBench 7.03 software with multiple client nodes running in a non-blocking network infrastructure. The test was performed at Edison Group labs in New York, NY. The server, the client, and the networking hardware were rented from a third party. The server and the client operating systems were purchased from a mail-order store. Microsoft employees were not permitted to access the Edison Group labs during the test. The server used passed Windows 2003 Enterprise Server compatibility test and was tuned in accordance with the Performance Tuning Guidelines from Microsoft¹. The standard test suite was extended to allow 200 clients to run. The observed peak throughput was 4,123 Mbit/s for 64-bit Windows 2003 Server SP1, and 3,994 Mbit/s for 64-bit Windows 2003 Server R2. The maximum performance point was reached, as indicated herein by the shape of the performance curve. The stress test was performed three times to ensure the stability and reproducibility of results. In order to obtain the more accurate number, the stress test was repeated around the global maximum of the curve with a smaller step.

Performance Analysis Methodology

Performance was measured by the standard NetBench test suite. NetBench benchmark suite measures file server performance as the number of bytes transferred to and from the server per second. NetBench suite tests a file server by generating a controlled number of requests from the managed load-generating nodes. Each node can emulate several file-accessing clients. The NetBench test site was configured to incrementally generate the load on the system until the maximum performance is reached.

Test Lab Design

Stress-Testing Software

NetBench version 7.03² was used for load generation. This is the most recent stable version of NetBench. Standard NetBench tests were used with modification of the mixes to allow for over 60 clients. Additional client directories for these clients were created on the server.

¹ Performance Tuning Guidelines for Windows Server 2003. Microsoft Corporation. October 2003.
<http://www.microsoft.com/windowsserver2003/evaluation/performance/tuning.mspx>

² <http://www.veritest.com/benchmarks/netbench/default.asp>

Stress-Testing Nodes and Client Tuning Parameters

A total of 100 stress-testing nodes were used to generate the load. The nodes had the following hardware configuration:

- Generic workstation with Biostar M7VIG 400 mainboard
- Sempron 2200+ (1.5 GHz)
- 512 MB of AENEON PC3200 DDR1 RAM
- 40 GB 7,200 rpm ATA 133 Maxtor hard drive
- On-board VIA VT6103 10/100 Mbit NIC

Windows XP SP2 was installed on all the nodes with poset-SP2 fixes. The following modifications were made to the standard configuration of Windows XP:

DormantFileLimit

HKLM\system\CurrentControlSet\Services\lanmanworkstation\parameters\
(REG_DWORD)

NOTE: By default this registry key is not created. (Windows XP client computers only.)

Specifies the maximum number of files that should be left open on a share after the application has closed the file. Recommended 100 (decimal).

ScavengerTimeLimit

HKLM\system\CurrentControlSet\Services\lanmanworkstation\parameters\
(REG_DWORD)

Windows XP client computers only. The amount of time in seconds that the redirector waits before it starts scavenging dormant file handles (cached file handles that are not currently used by any application). Recommended 100 (decimal).

DisableByteRangeLockingOnReadOnlyFiles

HKLM\System\CurrentControlSet\Services\LanmanWorkStation\Parameters\
(REG_DWORD)

Windows XP client computers only. Some distributed applications that lock portions of a read-only file as synchronization across clients require that file-handle caching and collapsing behavior be turned off for all read-only files. This parameter can be set if such applications will not be run on the system and if collapsing behavior can be enabled on the client computer. Recommended 0x1.

TcpAckFrequency

TcpAckFrequency applies only to .XP Clients. The recommended setting for TcpAckFrequency is between one third and one half of TcpWindowSize.

For FastEthernet cards:

HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\Interfaces

For each FastEthernet adapter, add:

TcpAckFrequency (REG_DWORD) = 5 (decimal)

By default this entry is not in the registry. If only acking data and not any control packets, ack once every 5 packets, instead of the default of 2. This helps in reducing packet processing costs for the network stack in the case of large writes (uploads) from the client into the server.

One NetBench controller station was used with the hardware and OS identical to those of a stress-testing node.

File Server Hardware and Tuning Parameters

Windows 2003 64-bit Enterprise server was installed using default settings as a workgroup server with a file server role. The on-board storage was configured as RAID 1 with 30 GB OS volume (drive C) and a 42 GB application volume (drive D). The external disk enclosure was configured as four RAID 0 volumes, two per channel on the dual-channel RAID controller with 12 72G 10K RPM drives. All the drives were formatted with an NTFS partition. Each set of 25 clients was configured to access a separate logical volume of the array.

The following changes were made to the server:

- Server set to maximize throughput for file sharing and to boost the background applications. NICs were set to the following:
 - NIC Speed – 1 Gbit
 - Spooler service was disabled
- The RAID system was formatted with 128 KB allocation unit size. Raid Controller settings:
 - Rebuild priority – Low
 - Expand priority – Low
 - Accelerator ratio – 50% read, 50% write

The following tuning parameters were applied:

NtfsDisable8dot3NameCreation

HKLM\System\CurrentControlSet\Control\FileSystem\ (REG_DWORD)

Recommended 0x1. Default is 0. This parameter determines whether NTFS generates a short name in the 8.3 (DOS) naming convention for long filenames and for filenames that contain characters from the extended character set. If the value of this entry is 0, files can

potentially have two names: the name that the user specifies, and the short name that NTFS generates. If the name the user specifies conforms to the 8.3 naming convention, NTFS does not generate a short name.

Changing this value does not change the contents of a file, but it avoids the short-name attribute creation for the file, also changing the way NTFS displays and manages the file.

Disablelastaccess

HKLM\System\CurrentControlSet\Control\FileSystem\ . (REG_DWORD)

Recommended 0x1. By default, this registry key is not created. If you have an NTFS volume with a large number of folders or files, and a program briefly accesses each of these in turn, the I/O bandwidth used to generate the Last Access Time updates can be a significant percentage of the overall I/O bandwidth. To increase the speed of access to a folder or file, you can set **disablelastaccess** to disable updating the Last Access Time. After you use this command and restart the computer, the Last Access Time is no longer updated. If you create a new file, the Last Access Time remains the same as the File Creation Time.

NoAliasingOnFileSystem

HKLM\System\CurrentControlSet\Services\LanManServer\Parameters\
(REG_DWORD)

By default, this registry is not created. Recommended 0x1. If the clients of a server do not access a file by both short and long names, and there are not multiple shares on that server at that point to the same file set, you can set this key to inform the clients that they can intelligently cache the namespace for performance improvement.

TreatHostAsStableStorage

HKLM\system\CurrentControlSet\Services\LanmanServer\Parameters (REG_DWORD)
= 1

The SMB_COM_FLUSH command makes sure that all data and allocation information for the corresponding file has been written to stable storage. A response is not sent to the client until the writes are complete. However, two factors may contribute to unnecessarily poor server performance:

Many client programs flush files more frequently than is required. Client programs should flush files only when this is necessary for data consistency. Too many data flushes slow the performance of the file server because the flushes interfere with the server's ability to optimally schedule disk activity. Programs that are running on the server do not know that it is sufficient to leave the data in the memory of the file server; if the file server is sufficiently reliable, it may be acceptable to keep file data stored there. In this case, the memory on the file server is considered *stable storage*, but the programs are not aware of it.

NumTcbTablePartitions

HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\.(REG_DWORD)

Recommended 0x10 (for a 4P or larger file server). By default this key is not created.

This parameter controls the number of TCB table partitions. The TCB table can be partitioned to improve scalability on multiprocessor systems by reducing contention on the TCB table. If the workload is network-intensive, this needs to be set to the square of the number of processors on the machine.

TcpAckFrequency

TcpAckFrequency applies only to Windows Server 2003. The recommended setting for **TcpAckFrequency** is between one-third and one-half of **TcpWindowSize**.

For Gigabit cards:

HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\Interfaces

For each Gigabit adapter, add:

TcpAckFrequency (REG_DWORD) = 13 (decimal)

By default this entry is not in the registry. If only acking data and not any control packets, ack once every 13 packets, instead of the default of two. This helps in reducing packet processing costs for the network stack, in the case of large writes (uploads) from the client into the server.

Interrupt Affinity

Interrupt affinity means binding of interrupts from a specific device to specific processor(s) in a multiprocessor server. This enforces running the ISR and DPC routines on the said processor(s). Because network connections and file server sessions all stay on the same network adapter, binding interrupts from the network adapter to a processor allows for processing incoming packets (SMB requests, data) on a specific set of processors, improving locality and scalability. You cannot configure affinity on single-processor computers.

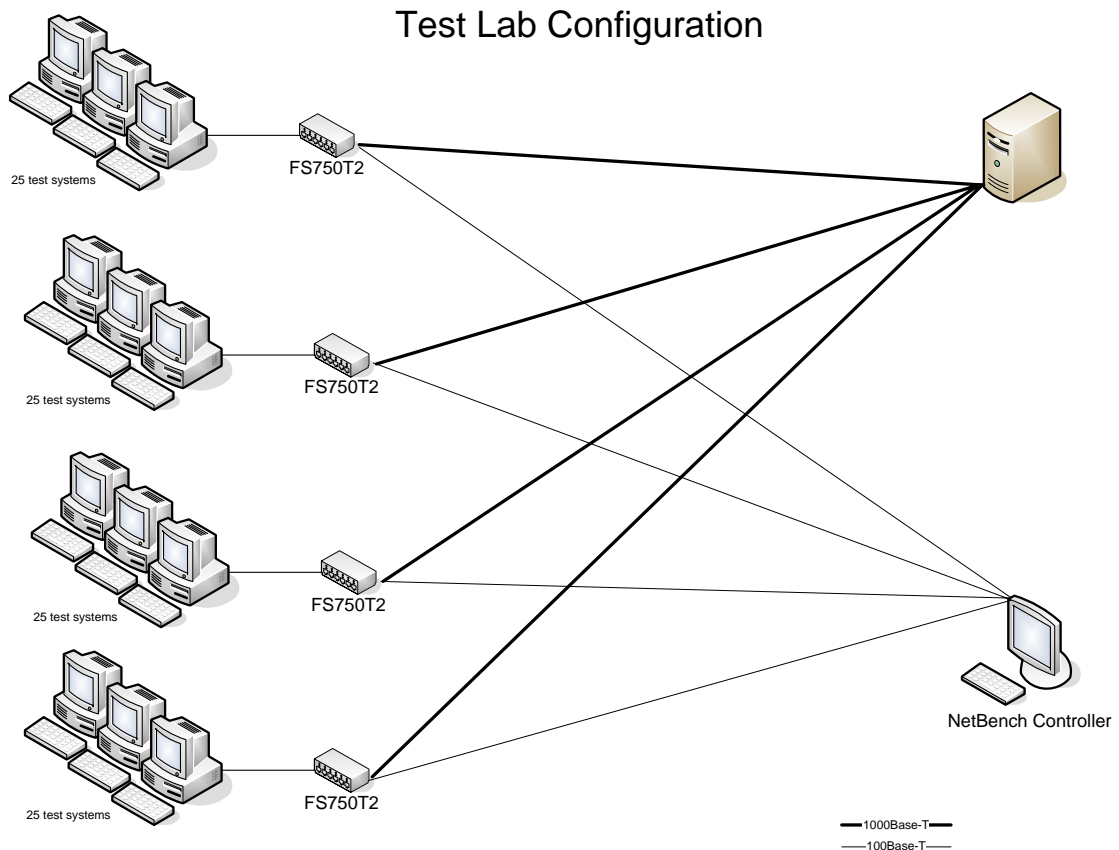
The Interrupt-Affinity Filter (IntFiltr) tool allows you to change the CPU-affinity of the interrupts in a system.

Using this utility, you can direct any device's interrupts to a specific processor or set of processors (as opposed to always sending interrupts to any of the CPUs in the system). Note that different devices can have different interrupt-affinity settings. This utility will work on any server running Windows Server 2003, regardless of what processor or interrupt controller is used.

The Interrupt-Affinity Filter was used to bind the interrupts from Gigabit NICs to two CPUs each.

Networking Topology

The network was composed of four Netgear FS750T2 48-port 10/100 Mbit switches with two Gigabit uplinks. The server and the NetBench controller were connected to each of the four switches. The most recent firmware was installed on all the switches. The complete network diagram of the test lab is presented below.



Stress Test

Configuration of the test mixes

The following mix test definitions and duration information were used:

- Virtual Clients per node – 2
- Ramp Up – 1 iteration
- Ramp Down – 1 iteration
- Length – 10 iterations
- Delay – 0 seconds

- Think time – 0 seconds
- Workspace – 20 MB
- Number of mixes – 9 (1, 8, 16, 24, 32, 40, 48, 56, and 64 clients)
- Maximum number of clients – 64

Preliminary Tests and Assumptions

Prior to running each test suite, the NetBench software was reinstalled and the data volume was reformatted to avoid any residual data from previous test runs. Upon completion of each successive test, the file server, the controller, and the stress-generating nodes were rebooted.

Accuracy and Variability of Results

All the tests were performed in tight coordination with Microsoft technical support, which assisted in the fine-tuning of the server and the client. With this support we were able to get the maximum performance out both Windows Server 2003 SP1 and Windows Server 2003 R2. Though R2 exhibited slightly lower performance numbers, the difference between the two versions was very slight.

Each measurement was taken three times, with approximately 40 points difference between the minimum and the maximum results at the peak performance points. Therefore, the obtained results are 99.5% accurate and can be reported as 4,120±20 and 3993±20. It is probable that additional tuning will be possible as Windows Server 2003 R2 matures as a platform.

The points appearing to be the global maxima in each case are higher than the neighboring points by more than 20 units. Additionally, the visual (and numerical) interpolation of the obtained graphs demonstrates that we have a curve with one global maximum located at the coordinates stated in our report.

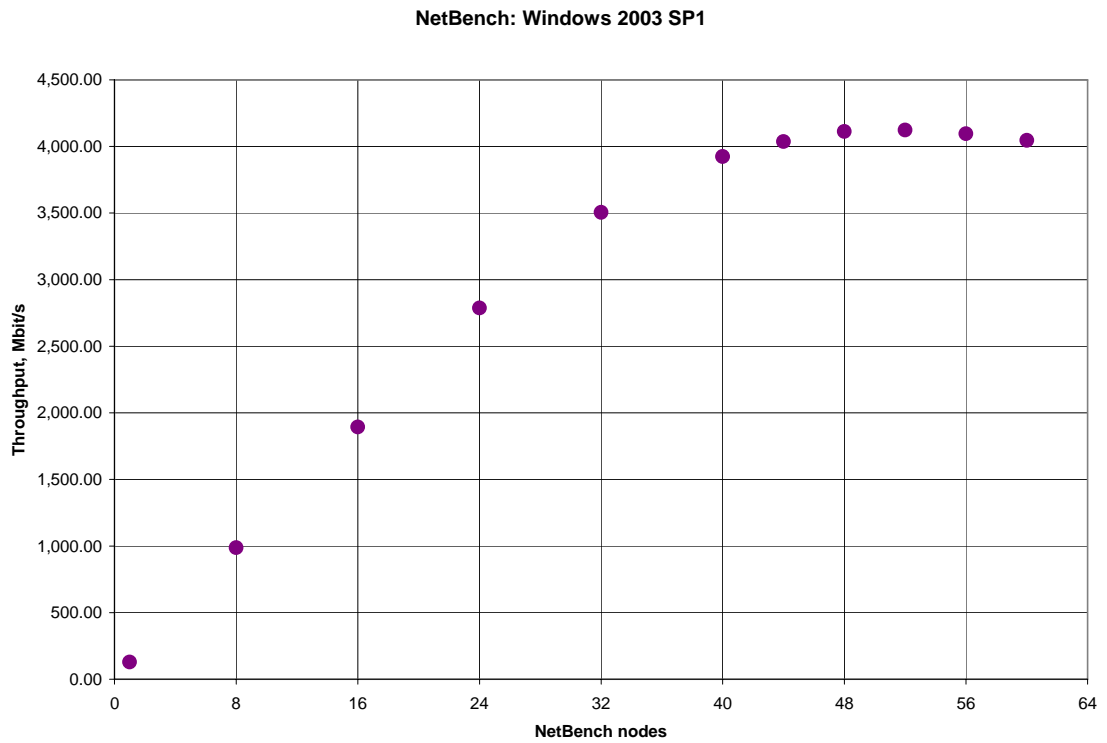
Test Results

The following results were obtained for 64-bit Windows 2003 Server SP1.

Nodes	Throughput, Mbit/s
1	129.54
8	987.12
16	1,893.99
24	2,786.90
32	3,505.36
40	3,924.79
44	4,037.36

48	4,112.86
52	4,123.73
56	4,095.39
60	4,045.60

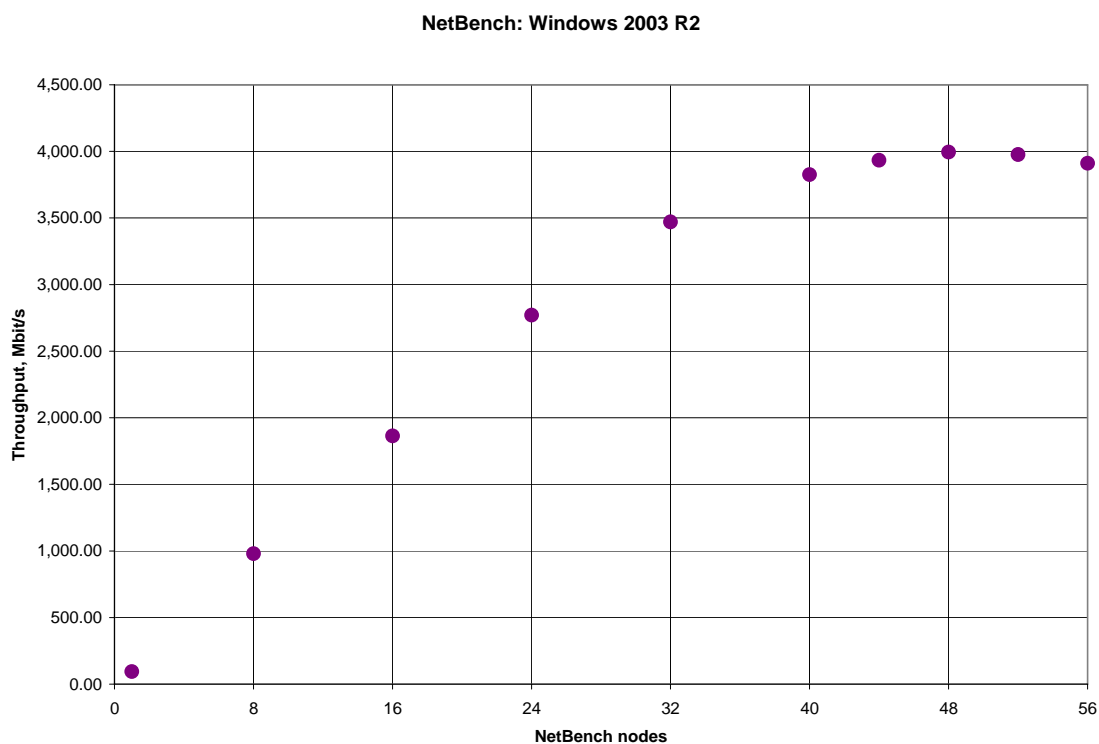
The results are displayed on the graph below.



The following results were obtained for 64-bit Windows 2003 Server R2.

Nodes	Throughput, Mbit/s
1	94.02
8	980.20
16	1,864.05
24	2,770.73
32	3,470.38
40	3,825.58
44	3,932.94
48	3,993.63
52	3,975.94
56	3,911.00

The results are displayed on the graph below.



Additional Information

Additional information is available by writing to:

Edison Group

89 Fifth Avenue

New York, NY 10003

info@TheEdison.com

<http://www.theedison.com>

