



# Performance Considerations for Implementing Microsoft Internet Information Server (IIS) on Netfinity 5500 M10 Servers

*Implementing IIS on Netfinity 5500 M10 Servers*

## Executive Summary

To be competitive in today's business environment, companies must be able to effectively utilize online technologies via the Internet and intranet. IBM's Netfinity® 5500 M10 server hardware combined with Microsoft® Internet Information Server (IIS) software provides businesses with a competitive edge in the rapidly evolving Internet and intranet business environment.

This document provides performance issues to be considered when implementing IIS on the Netfinity 5500 M10. Specifically, it describes these factors and the effect they have on performance:

- Installing a single Intel® 400-MHz Pentium®II Xeon™ processor in the Netfinity 5500 M10
- Installing dual 400-MHz Pentium II Xeon processors
- Implementing IIS with no optimization
- Optimizing IIS
- Using IBM's Web Server Accelerator to optimize network performance

This document is designed to serve as a guideline when using IIS and the Netfinity 5500 M10 in a networked environment. It explains the tests that were run using Ziff-Davis's WebBench™ 2.0 to determine the throughput of these configurations (based on requests per second) and offers analysis of the results.

## Overview of the Netfinity 5500 M10

The IBM Netfinity 5500 M10 server delivers power, scalability, manageability, and high levels of performance to business-critical applications at an affordable price. As a Web server, the Netfinity 5500 M10 provides both:

- **High availability** so that users get what they want, when they want it. The 5500 M10 includes the IBM Netfinity Predictive Failure Analysis (PFA) for processors, voltage regulator modules (VRMs), memory, fans, power supplies, and hard drive options to warn users before problems occur. In addition, exclusive Light-Path diagnostics, hot-pluggable PCI card slots, drive bays, fans, and power supplies allow customers to replace or upgrade components without powering down the server.

- **Performance** so that users get what they need quickly. The 5500 M10 server excels in network server applications using either uniprocessing or two-way symmetrical multiprocessing (SMP). It employs two Intel 400-MHz Pentium II Xeon processors with integrated 512-KB or 1-MB Level 2 (L2) cache.

## Test Network Configuration

The test network consisted of a total of 80 clients connected to two subnets. Each client was connected to a 10/100-Mbps hub (10 clients per hub). The hubs were connected to two 100-Mbps switches (four hubs per switch). Finally, the two switches were connected to a single, dual-homed Netfinity 5500 M10 server.

Figure 1 shows the test network configuration:

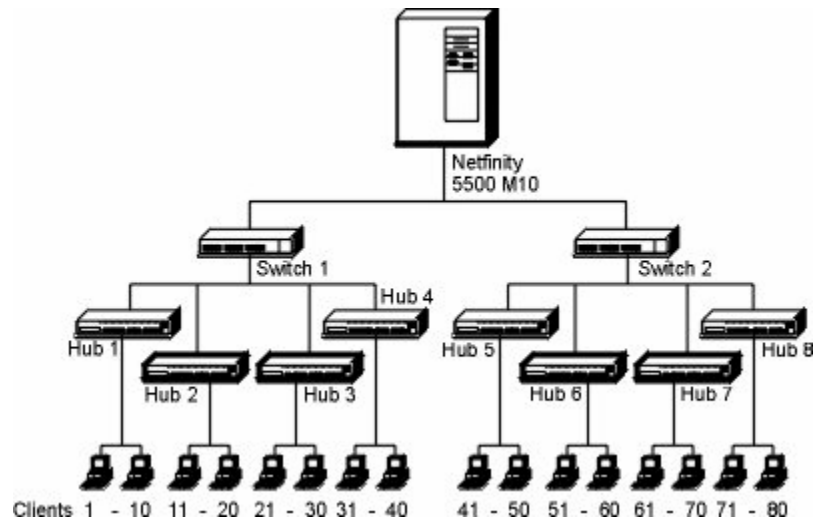
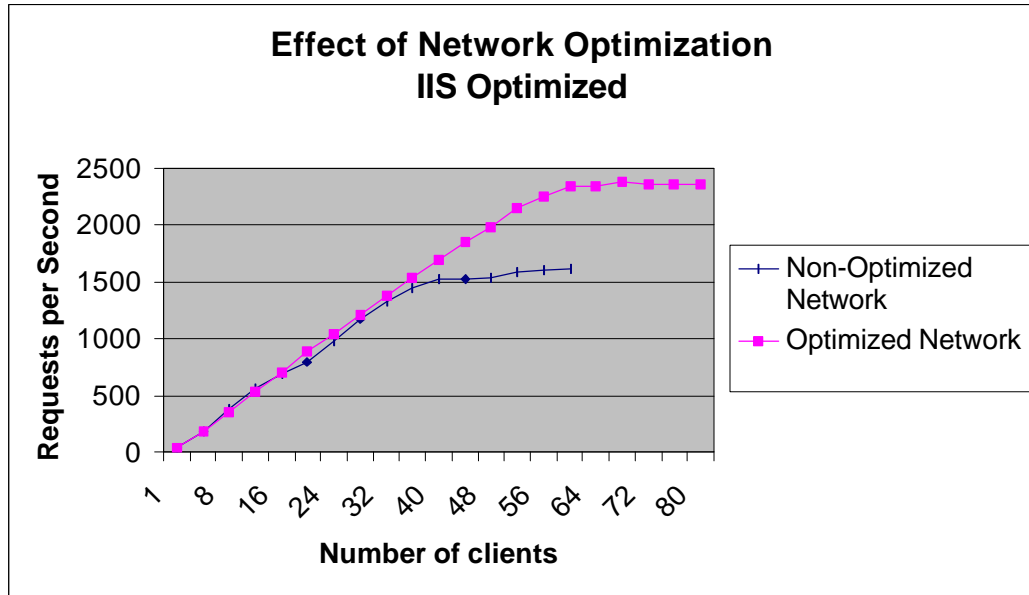


Figure 1. Test Network Configuration

**Note:** Initial testing was performed on 40 clients divided into two separate subnets of 20 clients each. Each subnet contained two Ethernet 10/100-Mbps hubs joined with a crossover cable and connected to the Netfinity 5500 M10 server. However, this configuration proved inadequate for testing purposes; the network became saturated before the Netfinity 5500 M10. During performance runs, the collision light on the hubs stayed on constantly and the hubs dropped to 10-Mbps mode.

Readings on the network usage (performed using Microsoft's Performance Monitor) indicated network usage above 70 percent. As the network became saturated, client performance varied greatly. Rather than all clients suffering degraded performance, some clients continued to perform at high levels while other were completely starved.

The effect of optimizing the network was quite dramatic. The following figure shows requests per second for a dual Xeon 5500 M10 with IIS Optimized. The Non-Optimized Network line represents the original network configuration, and the Optimized Network line represents the network configuration after adding additional hubs and switches.



The following software and components were used in the server and clients:

Feature	Netfinity 5500 M10 Server	Clients
Processor Type	400 MHz Xeon	133 MHz Pentium
Memory	512 MB ECC SDRAM	80 MB
L2 Cache	1 MB	256 KB
RAID Level	Non-Array	Non-Array
Disk Drive	Two IBM 9.1-GB Wide Ultra SCSI drives (7200 rpm)	1-GB IDE disk drive
Network Interface Adapter/ Network Interface Card	Two 100-Mbps Ethernet Adapters	Two 100-Mbps Ethernet Adapters set to maximum network throughput
Network Operating System	Windows NT Server 4.0 with Service Pack 3	Windows NT Workstation 4.0 with Service Pack 3
System Partition Size	4 GB	1 GB
Web Server	Microsoft Internet Information Server 4.0	N/A
Test Software	WebBench 2.0	N/A

## Testing Procedures Used

The performance of the Netfinity 5500 M10 was tested using WebBench Version 2.0 from Ziff-Davis Inc. (available on the Web at <http://www.zdnet.com/zdbop/webbench/home.htm>). WebBench is a Web server testing tool that measures the ability of the server to deliver both static and dynamic HTML pages. WebBench measures performance by sending HTTP GET requests to the server from each of the clients in the test network. It records and displays data at the point when the server becomes saturated.

**Note:** The WebBench tests used to measure performance for dynamic HTML pages were not run for this document because Microsoft recommends that ISAPI applications be used in place of CGI scripts for dynamic Web content. For more information about the Microsoft Internet Server API, see the overview located at <http://www.microsoft.com/win32dev/apiext/isapimrg.htm>.

In addition, Microsoft's direction for dynamic content is to use .asp pages and COM objects. Performance of such an arrangement is application- and implementation-dependent. No standard benchmark tests currently exist for this arrangement.

WebBench produces two different results:

- **Requests per second**, which is a basic measure of client/server interaction. WebBench determines the overall requests per second by first dividing the total number of successful requests by the number of seconds it takes for the requests to complete for each client. Then all client scores are added together to determine the overall score.
- **Throughput**, which is the number of bytes per second being sent by the server to the clients. WebBench determines throughput by first dividing the total number of bytes transferred to a client by the number of seconds it takes for the request to complete. Then all client scores are added together to determine the overall throughput.

File size distribution is an important aspect of these measurements. If a greater percentage of client requests is for large files, initial throughput measurements will be higher and requests per second measurements will be lower. However, if a greater percentage of client requests is for small files, requests per second measurements will be higher but throughput measurements will be lower.

For these tests, the standard WebBench file size distribution was used (nine files arranged in size from 512 bytes to 128 KB). The most frequently retrieved file was 4 KB, and 92 percent of the files retrieved were less than 32 KB.

In addition, these standard timing parameters were used:

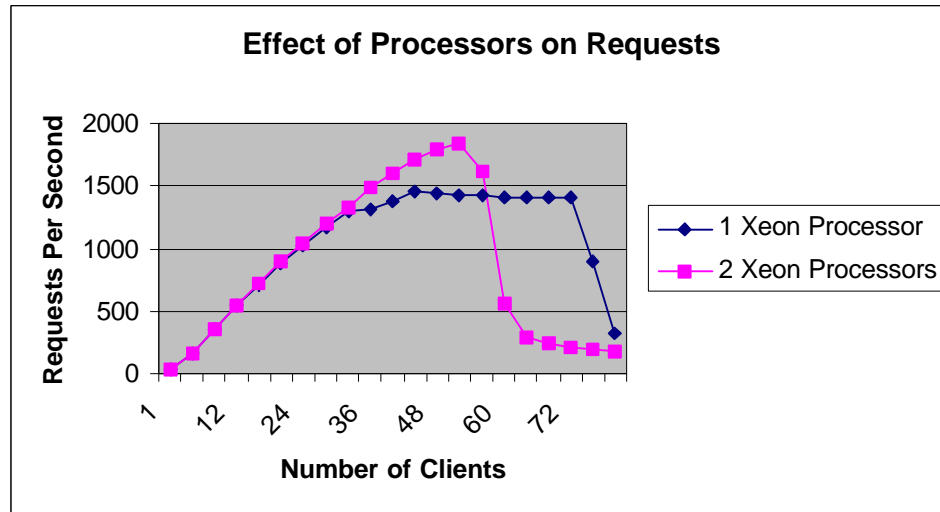
Client ramp up	30 seconds
Client ramp down	30 seconds
Client test duration	300 seconds
Delay between requests	0 seconds

## Results

This section describes the results obtained through our testing.

### ***IIS without Optimizations***

A baseline for performance was first established with all default settings for IIS. Separate runs were made for one Xeon processor and two Xeon processors.



In both test runs, the number of requests per second drops suddenly (at approximately 74 clients for the single processor and 56 clients for the dual processors). Using Performance Monitor to evaluate the server shows that at the point where the requests per second drops suddenly, disk utilization on the server is approaching 100 percent. It is interesting to note that for a certain number of clients, a single-processor system actually outperforms a dual-processor system because the lack of computing power slows the rate at which requests are received (decreasing disk utilization on the server).

The solution to the disk utilization problem depends on the server's primary function. If the server is primarily serving static Web pages, as was the case for our tests, optimizing IIS can resolve the bottleneck because the pages are loaded into memory (see "Optimizing Internet Information Server" for more information). However, if the server is primarily providing a download service of large files, loading all of the files into memory is not a viable option. In this case, a better solution may be to use faster hard drives and implement the integrated RAID technology on the server.

### ***Optimizing Internet Information Server***

Optimizing IIS can improve performance. When IIS is optimized, the Web pages are loaded into memory. The server then responds to requests for Web pages by serving them from memory rather than accessing the disk.

Microsoft's Web site contains a large amount of information concerning IIS optimization. For the purposes of this test, the following optimization parameters were used (based on *Internet Information Server 4.0 Tuning Parameters for High-Volume Sites*, which is published on Microsoft's Web site):

- Changed the server setting to AppServer, which instructs Windows NT to aggressively trim file cache.
- Installed the hotfix utility, which is available at:

[ftp://ftp.microsoft.com/bussys/iis/iis-public/fixes/usa/proxy](http://ftp.microsoft.com/bussys/iis/iis-public/fixes/usa/proxy)

Then, under the **Configuration** button of the **Application Settings**, removed any unused mappings.

- Disabled logging for the Web site, which frees system resources and improves performance.
- Set the receive buffers for the Network Interface Card (NIC) to maximum on the server, which minimizes the number of dropped packets on the receiving end.
- Set the MaxUserPort and TcpWindowSize TCP registry entries to ensure sufficient user ports and enlarge the TCP window size.
- Set the ObjectCacheTTL registry entry to 600. This affects the frequency with which the cache scavenger runs.
- Set the OpenFileInCache registry entry to 8000. This determines the number of files served from memory rather than disk.

Tests were conducted both before IIS optimization and after IIS optimization to determine what difference optimization had on the number of requests. The initial comparison was made with dual processors installed.

Next, with IIS optimized, the single processor run was compared with the dual processor run.

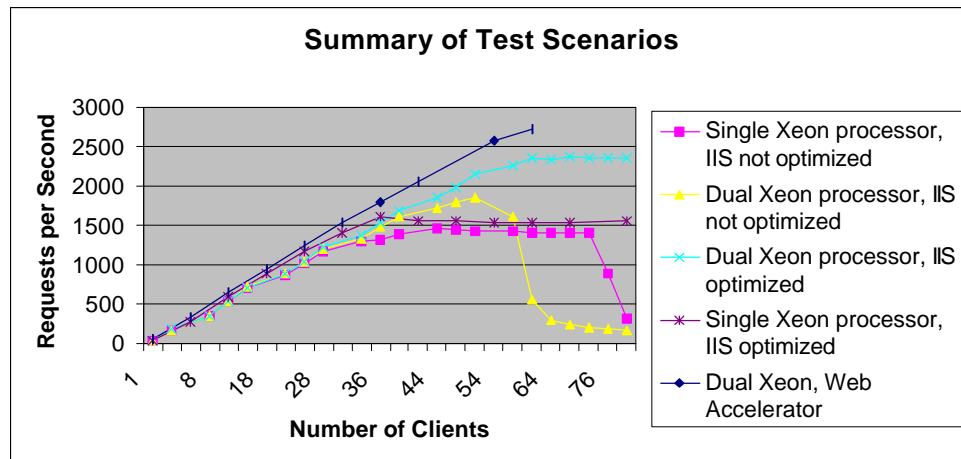
### **Using IBM's Netfinity Web Server Accelerator**

IBM's Netfinity Web Server Accelerator is standard caching reverse proxy software designed to substantially reduce the server response time for static Web page requests. Web Server Accelerator stores static Web content in server cache, and intercepts client requests for this content at the system level. Any changes that are made to the Web content are detected by Web Server Accelerator, which updates cache to ensure clients have access to the latest content. For more information about IBM Web Server Accelerator, view the IBM Netfinity Web Server Accelerator white paper, which is available in PDF format at [ftp://ftp.pc.ibm.com/pub/pccbbs/pc\\_servers/webaccel.pdf](http://ftp.pc.ibm.com/pub/pccbbs/pc_servers/webaccel.pdf).

We conducted tests using Netfinity Web Server Accelerator. Tests were conducted on a dual-processor server with IIS installed but not optimized to determine if the installation of Web Server Accelerator would lead to any additional increase in the requests per second.

## **Summary**

The following figure shows a summary of the requests per second achieved, based on each of the test scenarios.



Based on the results of our testing, it is definitely beneficial to either optimize IIS or use IBM's Web Server Accelerator. The Web Server Accelerator provides better performance than IIS optimization alone, even when IIS serves all of the static pages out of memory. By separating static Web page serving from dynamic Web page server, the Web Server Accelerator allows IIS to be completely optimized for dynamic pages.

The choice of dual or single processor depends on the anticipated requests per second as well as on the budget available to purchase a second processor. You may not need the extra power of the second processor, because the single processor provides nearly 1500 requests per second, which is approximately 129,600,000 requests per day. However, you can increase the requests per second by approximately 1.6 times by installing a second processor.

## **Where to Go for More Information**

Visit these Web sites for more information about the topics discussed in this document.

- For more information about Internet Information Server, visit the Microsoft Web site at <http://www.microsoft.com>.
- For more information about the Netfinity 5500 M10, visit the IBM Web site at <http://www.pc.ibm.com/us/accessories/netfinity/index.html>.
- For more information about WebBench, visit the Ziff-Davis Web site at <http://www.zdnet.com/zdbop/webbench/home.htm>.



© International Business Machines Corporation 1999

IBM Personal Computer Company  
Department LO6A  
3039 Cornwallis Road  
Research Triangle Park NC 27709

All rights reserved

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. IBM reserves the right to change specifications or other product information without notice.

IBM Netfinity servers and PC servers are assembled in the U.S., Great Britain, Japan, Australia and Brazil and are comprised of U.S. and non-U.S. parts.

IBM, and Netfinity are trademarks of International Business Machines Corporation in the United States and/or other countries.

BackOffice, Microsoft, Windows, Windows NT and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

Intel, Pentium II, and Pentium II Xeon are trademarks of Intel Corporation.

Lotus and Domino are trademarks of Lotus Development Corporation.

Tivoli is a trademark of Tivoli Systems, Inc., in the United States or other countries or both.

WebBench is a trademark of Ziff-Davis Inc.

Other company, product and service names may be trademarks or service marks of other companies.

This publication may include typographical errors and technical inaccuracies. The content is provided AS IS, without express or implied warranties of any kind.