VeriTest.®

April 2003

1001 Aviation Parkway, Suite 400 • Morrisville, NC 27560 • 919-380-2800 • Fax 919-380-2899
320 B Lakeside Drive • Foster City, CA 94404 • 650-513-8000• Fax 650-513-8099
www.veritest.com • info@veritest.com

Microsoft Windows Server 2003 with Internet Information Services (IIS) 6.0 vs. Linux Competitive Web Server Performance Comparison

Test report prepared under contract from Microsoft

Executive summary

Microsoft commissioned VeriTest, a division of Lionbridge Technologies, Inc., to conduct a series of tests comparing the Web serving performance of the following server operating system configurations running on a variety of server hardware and processor configurations:

- Windows Server 2003 Enterprise Edition RC2 (subsequently referred to as Windows Server 2003 in the remainder of this report)
- Red Hat Linux Advanced Server 2.1
- Red Hat Linux 8.0 Professional

For these tests, Hewlett Packard supplied three server systems as follows:

- HP ProLiant DL760 server configured with four 900MHz Pentium III Xeon processors, 4GB of RAM and four Intel PRO/1000 MF Server Adapters.
- HP ProLiant DL760 server configured with eight 900MHz Pentium III Xeon processors, 4GB of RAM and eight Intel PRO/1000 MF Server Adapters.
- HP ProLiant DL380 G2 server configured with two 1.4GHz Pentium III processors, 2GB of RAM and two Intel PRO/1000 MF Server Adapters.

Please refer to the Test Methodology section and Appendix A for complete details regarding the server systems used for these tests.

Key findings

- Windows Server 2003 with Internet Information Services (IIS) 6.0 delivered higher Web server throughput compared to Apache 1.3.23 running on Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 on the configurations we tested.
- Windows Server 2003 with IIS 6.0 delivered 300 percent better peak performance in the static Web Server performance testing using eight processors on the DL760 compared to Apache 1.3.23 running under Red Hat Linux Advanced Server 2.1
- Windows Server 2003 with IIS 6.0 delivered 51% better peak performance in the static Web Server Performance testing using four processors on the DL760 server compared to TUX running under Red Hat Linux Advanced Server 2.1.
- Windows Server 2003 with IIS 6.0 delivered 101% better peak performance in the dynamic E-commerce Web Server performance testing on the DL380 compared to Apache 1.3.23 and TUX running under Red Hat Linux Advanced Server 2.1 and 58% better peak performance compared to Apache 2.0.40 running under Red Hat Linux 8.0 Professional.

For the Web server performance tests, we used Ziff Davis Media's WebBench 4.1 benchmarking software. WebBench measures Web server performance by using large numbers of physical test clients to generate an HTTP based workload against a Web server under test. These test clients make a series of HTTP 1.0 requests for different combinations of static and dynamic based content. As the Web server under test responds to the client requests, each WebBench client records the number of HTTP requests made and the amount of data moved during the test. Once a test completes, WebBench reports test results in Requests Per Second and throughput in bytes per second.

The Web server performance testing consisted of executing a number of tests using a variety of standard and customized WebBench test suites against each server described above configured with each of the operating systems described above using the following processor combinations:

DL380 server configured with 2 processors
DL760 servers configured with 1, 2, 4 and 8 processors.

The following list describes the different types of tests we performed to measure Web server performance. Each item in the list describes a specific combination of content requested from the Web server. Unless otherwise stated, we conducted all tests listed below on each server using IIS 6.0, Apache and TUX.

Static test suite requesting 100 percent static content
Combination of 80 percent static content and 20 percent CGI-based dynamic content
Combination of 76 percent static content, 16 percent CGI-based dynamic content and 8 percent Secure Socket Layer (SSL) 3.0 based static and CGI-based dynamic content

TUX is a kernel-based HTTP server available with Linux kernels 2.4 and later. TUX improves Web server performance by caching static Web content at the kernel level.

We had planned to conduct tests using TUX with Red Hat Linux 8.0 Professional on the DL380 server. However, every test we attempted failed approximately half way through the test resulting in the DL380 server refusing to accept new HTTP connections from the WebBench test clients. As a result, the only test results reported using Red Hat Linux 8.0 Professional were generated using Apache 2.0.40.

Comparison of ISAPI vs. CGI Performance on Windows Server 2003

In addition to providing support for CGI, Windows Server 2003 supports the ISAPI interface as well for creating dynamic content. Using the ISAPI interface allows for improved performance when executing dynamic content using IIS 6.0 and is the method preferred over CGI for creating dynamic content on Windows Server 2003. While testing using ISAPI content was not a direct part of this testing, it's worth noting the performance improvements that are possible when using the ISAPI interface compared to CGI on Windows Server 2003.

Using the CGI interface requires that the Web server spawn a new process on the server to execute the CGI-based dynamic content. This is a very expensive process in terms of server system resources and, relatively speaking, requires a great deal of time to perform the operation leading to significantly lower test results compared to a static test. On the other hand, ISAPI modules are dynamic link libraries (DLL's) that are loaded into the process space of the Web server. As a result, executing ISAPI-based content is significantly faster compared to executing the same CGI-based content resulting in significantly better test results compared to using CGI-based dynamic content.

WebBench 4.1 provides default test suites using both CGI and ISAPI modules for use with Windows Server 2003. The dynamic content associated with both test suites is simple executable code that reads a number of Web server environment variables and places their values into an HTML response string of 1024 bytes and returns this data to the requesting WebBench client. WebBench does not provide an ISAPI module for use with Apache so a direct comparison of ISAPI performance using IIS 6.0 and Apache was beyond the scope of this testing.

However, for all the dynamic content test results presented in this report, we're including a set of ISAPI based test results generated on the identical hardware using Windows Server 2003 to show the possibilities for improved performance using ISAPI compared to CGI with IIS 6.0 on Windows Server 2003.

Please refer to the Test Methodology section for complete details of the WebBench test suites used during the testing and how we conducted the Web server performance tests.

Web Server Performance Test Results

This section summarizes the Web server performance results. The charts below display the peak requests per second values generated during each type of Web server performance test. Please refer to the Test Methodology section of this report for complete details on the WebBench test suites used to generate these test results.

Static Content Results

We conducted these tests by configuring the WebBench test clients to make 100 percent of their requests for static content. Figure 1 shows the peak static requests per second values generated on both the DL380 and DL760 servers using all operating system and processor combinations. We found that regardless of the server employed or the number of processors, Windows Server 2003 generated better peak Web serving performance using static content compared to both Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional using our test configurations. This was particularly true when testing with the DL760 server configured with four and eight processors.

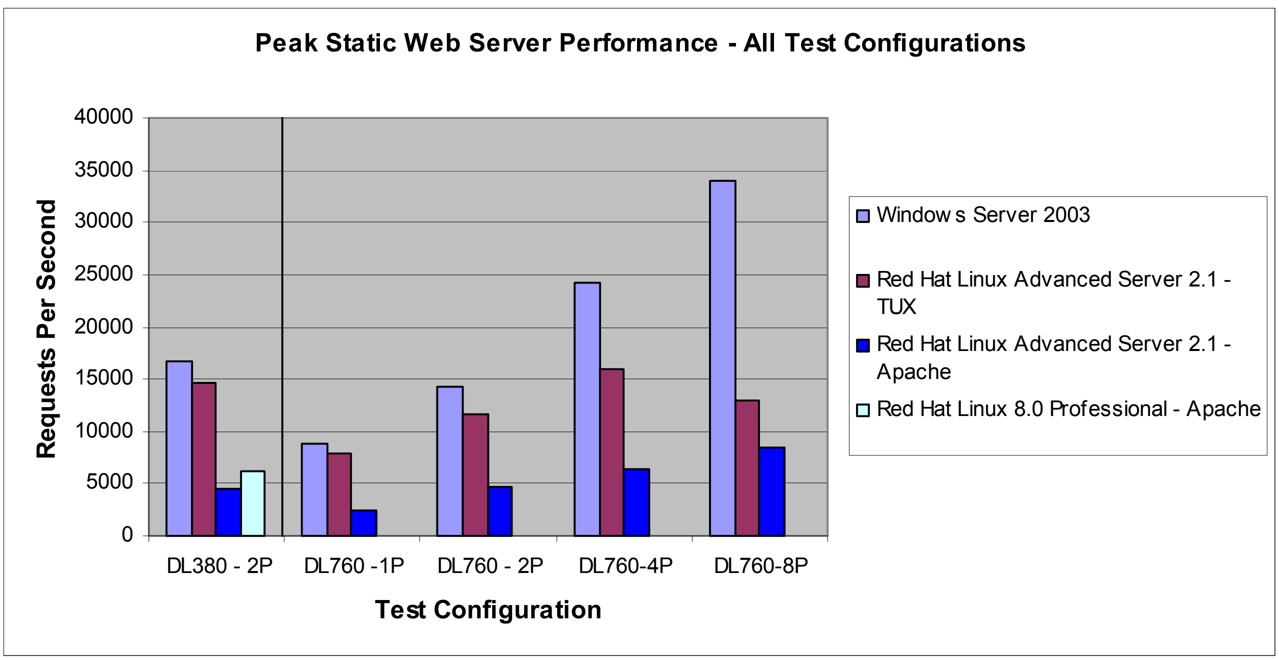

Figure 1. Peak Static Web Server Performance On All Test Configurations

We encountered an issue during testing when using TUX alone under Red Hat Linux Advanced Server 2.1 to deliver static content. As figure 1 shows, we observed good scaling of the static Web Server performance on the DL760 server regardless of the Web server employed until we tested using TUX with Red Hat Linux Advanced Server 2.1 using eight processors. When testing with eight processors, the static test results actually decreased approximately 23 percent compared to the results generated using TUX with Red Hat Linux Advanced Server 2.1 using four processors.

We double checked the TUX and operating system configuration and ran additional tests to try to resolve this issue, but were ultimately unsuccessful. We submitted a formal support request with Red Hat Technical Support on April 8th, 2003 regarding this issue with TUX and provided problem and configuration details. On April 9th, 2003 we received an indication that our request had been escalated to the senior technical support staff. On April 30th we had still not received a response from Red Hat Technical Support that offered any options for resolving this issue. By this time, we were required to return the servers used during the testing to Hewlett-Packard. This meant that even if Red Hat Technical Support had responded with options for resolving this issue, we would no longer have the hardware necessary to conduct additional testing. Therefore, we published the existing numbers shown in this report.

Figure 2 below shows the actual peak WebBench static Web server performance results in requests per second generated on both the DL380 and DL760 server using all operating system and processor combinations. Additionally, figure 2 shows the percentage improvement in Static Web server performance when testing with Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. These results clearly show that, in our test configurations, significant performance improvements are possible when serving static Web content using Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 using Apache. Lesser, yet still noticeable improvements, are possible when serving static Web content using Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 using TUX.

Operating System	DL380 - 2P	DL760 -1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 and IIS 6.0	16783	8861	14214	24293	33991
Red Hat Linux Advanced Server 2.1 and TUX	14741	7880	11676	16035	13007
Red Hat Linux Advanced Server 2.1 and Apache 1.3.23	4467	2387	4639	6465	8496
Red Hat Linux 8.0 Professional and Apache 2.0.40	6260	N/A	N/A	N/A	N/A
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using TUX and Apache 1.3.23	14%	12%	22%	51%	161%
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using Apache 1.3.23	276%	271%	206%	276%	300%
Percent Improvement with Windows Server 2003 over Red Hat Linux 8.0 Professional using Apache 2.0.40	168%	N/A	N/A	N/A	N/A

Figure 2. Peak Static Web Server Performance and Percentage Improvement of Windows Server 2003 in Static Web Server Performance Data On All Test Configurations

Dynamic Content Performance Test Results

We conducted these tests by configuring the WebBench test clients to make 80 percent of their requests for static content and 20 percent for a simple CGI executable. For the CGI tests that employed the TUX web server, we configured TUX to serve only static content and configured Apache to serve the CGI based content. The TUX and Apache Web servers then worked as a team to handle the mixture of static and CGI based content requested during these tests.

In addition to providing support for CGI, Windows Server 2003 supports the ISAPI interface as well for creating dynamic content. Using the ISAPI interface allows for improved performance when executing dynamic content using IIS 6.0. While testing using ISAPI content was not a direct part of this testing, it’s worth noting the performance improvements that are possible when using the ISAPI interface compared to CGI.

Figure 3 shows the peak request per second values generated on both the DL380 and DL760 server using all operating system and processor combinations we tested. We found that regardless of the server employed or the number of processors, in our test configurations, Windows Server 2003 generated better peak Web serving performance using the CGI-based content compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. This was particularly true on the DL380 server and when using two and four processor configurations on the DL760 server. We also found that, for the most part, using TUX to serve the static portion of the requested content improved the overall CGI-based test results compared to using only Apache to serve both the static and CGI-based content.

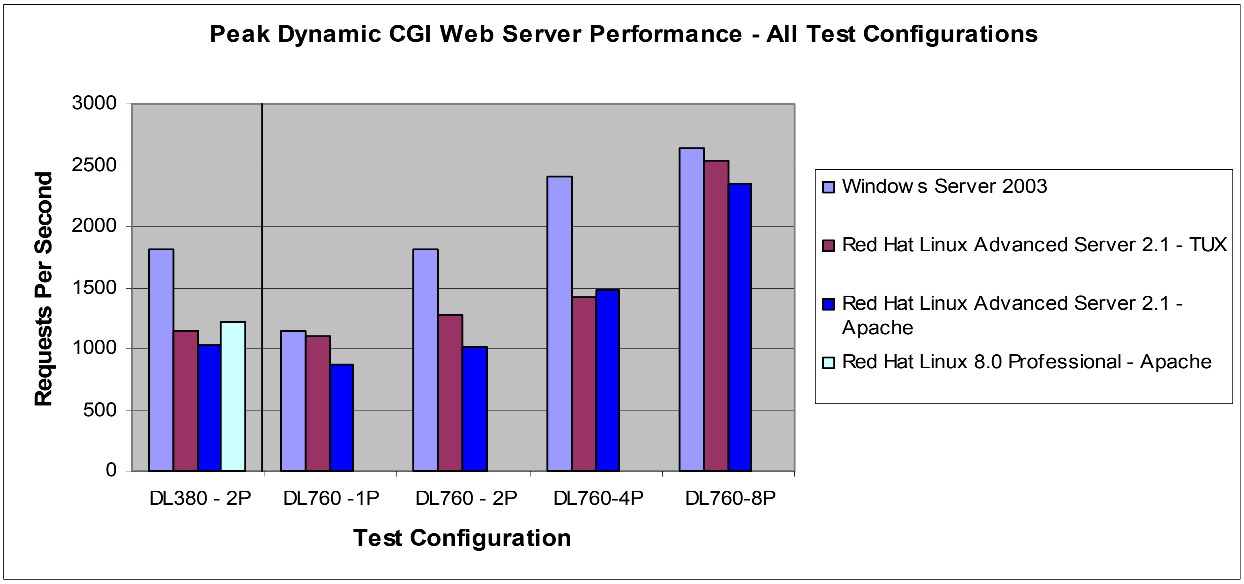

Figure 3. Peak Dynamic CGI-based Web Server Performance On All Test Configurations

Figure 4 below shows the actual peak WebBench CGI Web server performance results in requests per second generated on both the DL380 and DL760 server using all operating system and processor combinations. Additionally, figure 4 shows the percentage improvement in CGI-based Web server performance when testing with Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional using all operating system and processor combinations. These results clearly show that, in our test configurations, significant performance improvements are possible when serving a combination of static and CGI-based Web content when moving from either Red Hat Linux Advanced Server 2.1 or Red Hat Linux 8.0 Professional to Windows Server 2003.

Operating System	DL380 - 2P	DL760 -1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 and IIS 6.0	1814	1146	1805	2413	2639
Red Hat Linux Advanced Server 2.1 and TUX	1151	1103	1269	1418	2534
Red Hat Linux Advanced Server 2.1 and Apache 1.3.23	1031	870	1019	1476	2352
Red Hat Linux 8.0 Professional and Apache 2.0.40	1211	N/A	N/A	N/A	N/A
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using TUX and Apache 1.3.23	58%	4%	42%	70%	4%
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using Apache 1.3.23	76%	32%	77%	63%	12%
Percent Improvement with Windows Server 2003 over Red Hat Linux 8.0 Professional	50%	N/A	N/A	N/A	N/A

using Apache 2.0.40					

Figure 4. Peak Dynamic CGI Web Server Performance and Percentage Improvement of Windows Server 2003 in CGI Web Server Performance Data On All Test Configurations

Figure 5 compares the dynamic CGI-based test results from this report using Windows Server 2003 and a set of test results using the ISAPI interface under Windows Server 2003 generated on the identical server hardware as the other testing in this report

It's clear from the graph that using the ISAPI interface for generating dynamic content can significantly improve the overall Web server performance when serving dynamic content.

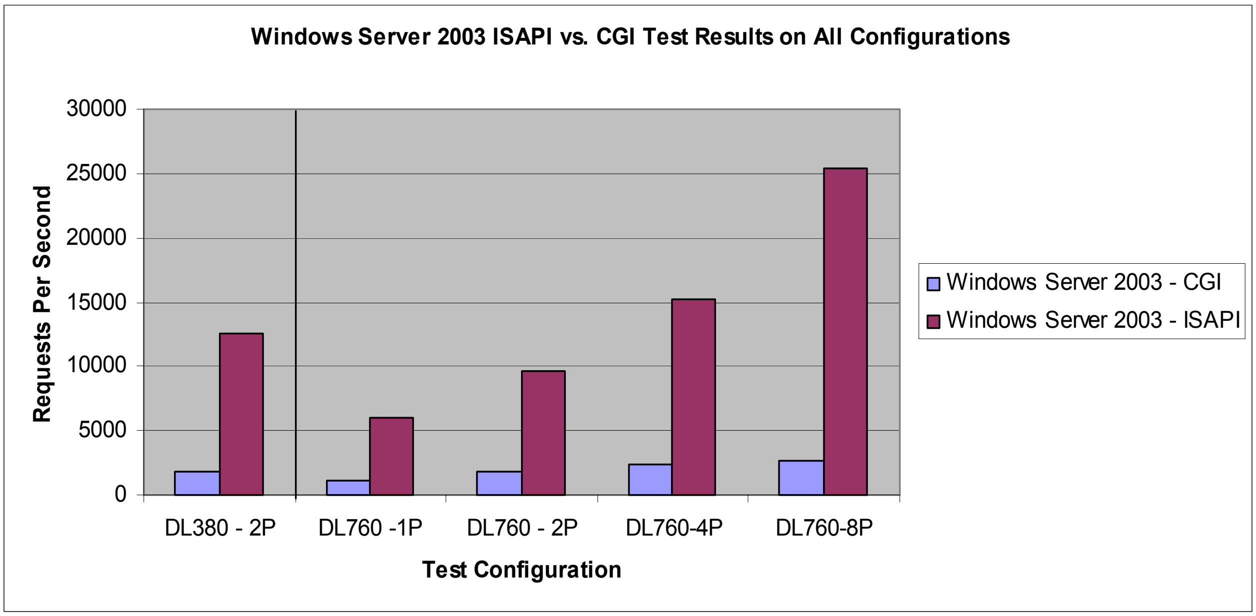

Figure 5. Windows Server 2003 ISAPI vs. CGI Test Results on All Configurations

Figure 6 shows the actual peak dynamic CGI and ISAPI test results and the percentage increase in performance when using the ISAPI interface compared to the CGI interface.

Operating System	DL380 - 2P	DL760 - 1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 - CGI	1814	1146	1805	2413	2639
Windows Server 2003 - ISAPI	12551	6054	9685	15270	25329
Percentage Improvement Using ISAPI Compared to CGI on Windows Server 2003	592%	428%	437%	533%	860%

Figure 6. Peak Dynamic ISAPI and CGI Web Server Performance and Percentage Improvement When using ISAPI vs. CGI on Windows Server 2003

E-Commerce Performance Test Results

We conducted these tests by configuring the WebBench test clients to make 76 percent of their requests for static content, 16 percent for a simple CGI executable and the remaining 8 percent for static and CGI-based content using SSL 3.0 for secure Web server communications. For the SSL/CGI tests that employed the TUX web server, we configured TUX to serve only non-secure static content and configured Apache to serve the secure static as well as the secure and non-secure CGI based content. The TUX and Apache Web servers then worked as a team to handle the mixture of secure and non-secure static and CGI based content requested during these tests.

Figure 7 shows the peak request per second values generated on both the DL380 and DL760 server using all operating system and processor combinations we tested. We found that regardless of the server employed or the number of processors, using our test configurations, Windows Server 2003 generated better peak Web serving performance using the SSL/CGI-based content compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. This was particularly true on the DL380 server and when using two and four processor configurations on the DL760 server. We also found that, for the most part, using TUX to serve the non-secure static portion of the requested content improved the overall SSL/CGI test results compared to using only Apache to serve both the static and CGI based content.

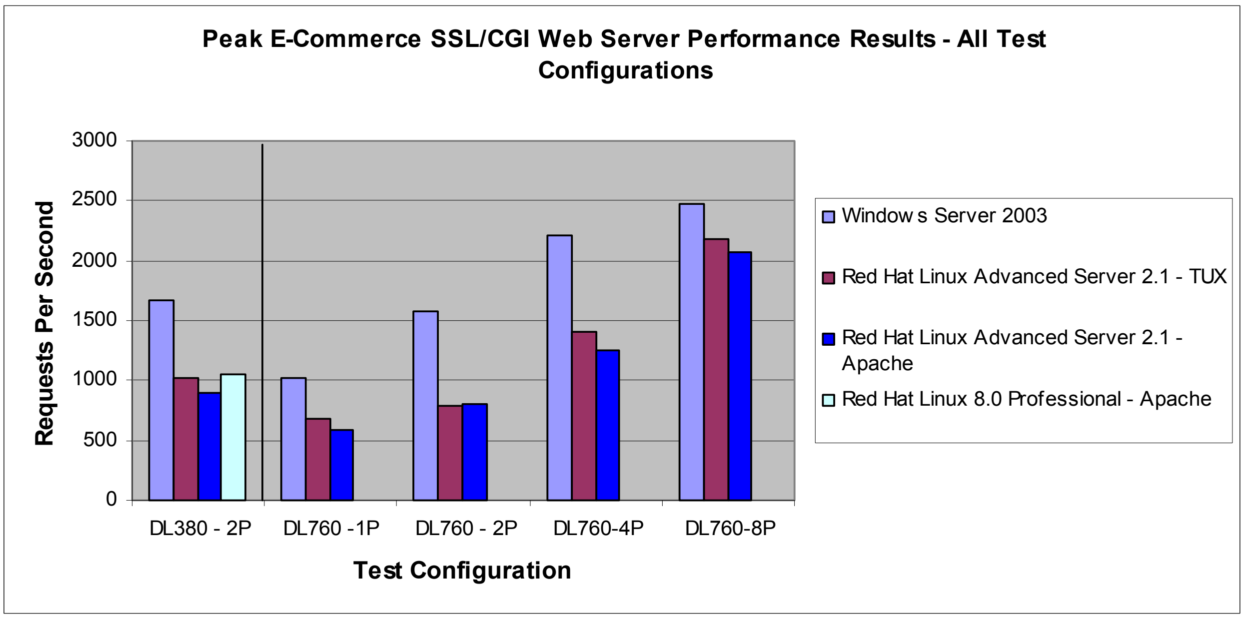

Figure 7. Peak E-Commerce SSL/CGI-based Web Server Performance On All Test Configurations

Figure 8 below shows the actual peak WebBench SSL/CGI Web server performance results in requests per second generated on both the DL380 and DL760 server using all operating system and processor combinations. Figure 8 also shows the percentage improvement in SSL/CGI-based Web server performance when testing with Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. These results clearly show that, in our test configurations, significant performance improvements are possible when serving a combination of static and SSL/CGI-based Web content when moving from either Red Hat Linux Advanced Server 2.1 or Red Hat Linux 8.0 Professional to Windows Server 2003.

Operating System	DL380 - 2P	DL760 -1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 and IIS 6.0	1668	1020	1580	2214	2480
Red Hat Linux Advanced Server 2.1 and TUX	1019	673	785	1407	2181
Red Hat Linux Advanced Server 2.1 and Apache 1.3.23	900	589	808	1247	2066
Red Hat Linux 8.0 Professional and Apache 2.0.40	1058	N/A	N/A	N/A	N/A
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using TUX and Apache 1.3.23	64%	52%	101%	57%	14%

Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using Apache 1.3.23	85%	73%	96%	78%	20%
Percent Improvement with Windows Server 2003 over Red Hat Linux 8.0 Professional using Apache 2.0.40	58%	N/A	N/A	N/A	N/A

Figure 8. Peak E-Commerce SSL/CGI Web Server Performance and Percentage Improvement of Windows Server 2003 in SSL/CGI Web Server Performance Data On All Test Configurations

Figure 9 compares the dynamic E-Commerce SSL/CGI-based test results from this report using Windows Server 2003 and a set of test results using the dynamic E-Commerce SSL/ISAPI interface under Windows Server 2003 generated on the identical server hardware as the other testing in this report.

It's clear from the graph that using the ISAPI interface for generating dynamic content significantly improves the overall Web server performance when serving dynamic content.

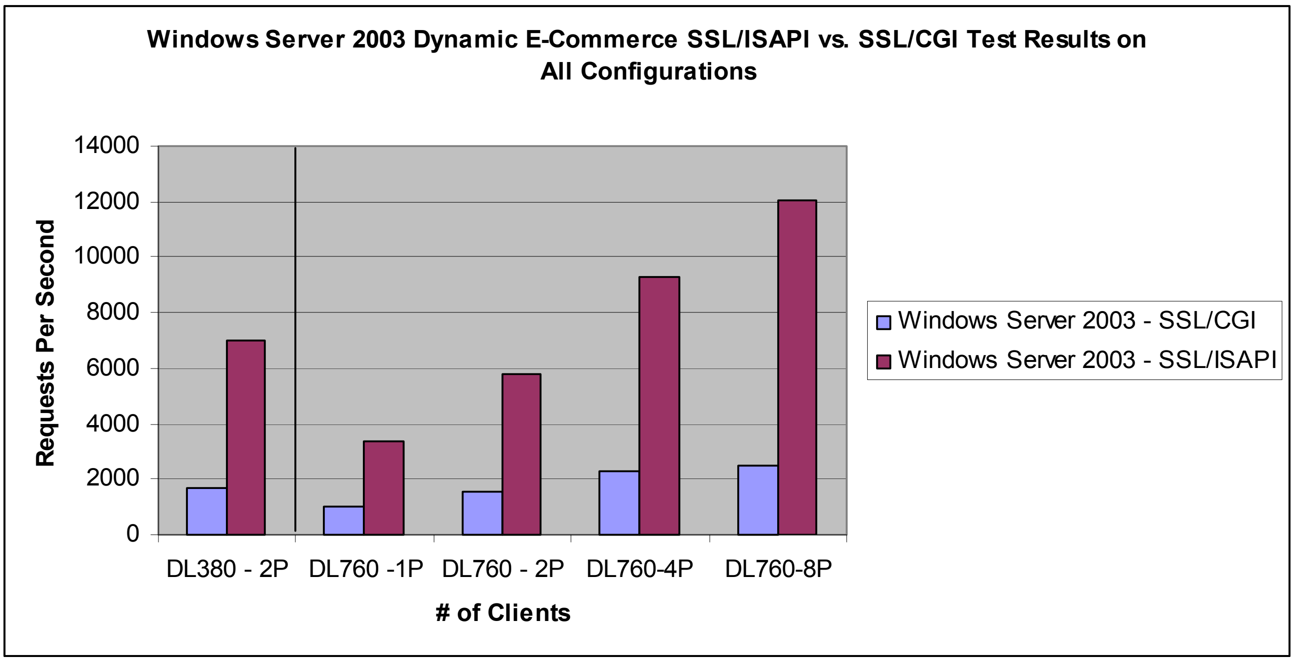

Figure 9. Windows Server 2003 SSL/ISAPI vs. SSL/CGI Test Results on All Configurations

Figure 10 shows the actual peak dynamic CGI and ISAPI test results and the percentage increase in performance when using the ISAPI interface compared to the CGI interface.

Operating System	DL380 - 2P	DL760 - 1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 - SSL/CGI	1668	1020	1580	2268	2480
Windows Server 2003 - SSL/ISAPI	6999	3338	5768	9285	12079
Percentage Improvement Using SSL/ISAPI Compared to SSL/CGI on Windows Server 2003	320%	227%	265%	309%	387%

Figure 10. Peak Dynamic E-Commerce SSL/ISAPI and SSL/CGI Web Server Performance and Percentage Improvement When using SSL/ISAPI vs. SSL/CGI on Windows Server 2003

Testing methodology

Microsoft commissioned VeriTest, a division of Lionbridge Technologies, Inc., to conduct a series of tests comparing the Web serving performance of the following server operating system configurations running on a variety of server hardware and processor configurations:

Windows Server 2003 Enterprise Edition Release Candidate 2(RC2)
Red Hat Linux Advanced Server 2.1
Red Hat Linux 8.0 Professional

Hewlett-Packard supplied the server hardware for these tests. Specifically, we used the following systems:

HP ProLiant DL760 G2 server configured with four 900MHz Pentium III Xeon processors, 4GB of RAM and four Intel PRO/1000 MF Server Adapters. This system contained an embedded SmartArray 5i RAID controller connected to four 36.4GB 15,000RPM Ultra3 SCSI disk drives. Additionally, we installed a second RAID subsystem consisting of a total of 28 18.2GB 15,000 RPM Ultra3 SCSI disk drives connected to a SmartArray 5300 RAID controller.

HP ProLiant DL760 G2 server configured with eight 900MHz Pentium III Xeon processors, 4GB of RAM and eight Intel PRO/1000 MF Server Adapters. This system contained an embedded SmartArray 5i RAID controller connected to four 36.4GB 15,000RPM Ultra3 SCSI disk drives. Additionally, we installed a second RAID subsystem consisting of a total of 28 18.2GB 15,000 RPM Ultra3 SCSI disk drives connected to a SmartArray 5300 RAID controller.

HP ProLiant DL380 G2 server configured with two 1.4GHz Pentium III processors, 2GB of RAM and two Intel PRO/1000 MF Server Adapters. This system contained an embedded SmartArray 5i RAID controller connected to six 36.4GB 15,000RPM Ultra3 SCSI disk drives.

VeriTest provided the network test client hardware for these tests. Specifically, we used the following systems:

240 client systems configured with a single 850Mhz Pentium III processor, 256MB of RAM, 10GB IDE hard drive and single 100 Mbps Ethernet adapter.

Test Network Configuration

For the Web server performance tests, we created two distinct test networks each using 120 physical clients. We connected the first 120-node network to the HP DL760 server containing four processors and four network adapters. We connected all 120 clients through four Extreme Networks Summit48 switches (30 clients per switch) using 100 Mbps, full duplex connections. We configured the 120 clients into four distinct subnets each containing 30 clients. We used the Gigabit ports on the Summit48 switch to connect each subnet of 30 clients to one of the four Intel PRO/1000 MF Gigabit Server Adapters installed in the HP DL760 server.

We connected the second 120-node network to the HP DL760 server containing eight processors and eight network adapters. We connected all 120 clients through four Extreme Networks Summit48 switches (30 clients per switch) using 100 Mbps, full duplex connections. We configured the 120 clients into eight distinct subnets each containing 15 clients. We used the Gigabit ports on the Summit48 switch to connect each subnet of 15 clients to one of the eight Intel PRO/1000 MF Gigabit Server Adapters installed in the HP DL760 server.

Because the HP DL380 server contained only two network adapters, we used two of the 30-client network segments configured in the first 120-client network described above for all tests involving the DL380 server. We connected each 30-client network segment through a separate Extreme Networks Summit48 switch using

100 Mbps, full duplex connections. We used the Gigabit ports on each Summit48 switch to connect each subnet of 30 clients to one of the two Intel PRO/1000 MF Gigabit Server Adapters installed in the HP DL380 server. Please refer to Appendix D of this report for visual representations of the network configurations used for these tests.

Web Server Performance Testing

For the Web server performance tests, we used Ziff Davis Media's WebBench 4.1 benchmarking software. WebBench measures Web server performance by using large numbers of physical test clients to generate an HTTP based workload against a Web server under test. These test clients make a series of HTTP 1.0 requests for different combinations of static and dynamic based content. As the Web server under test responds to the client requests, each client records the number of HTTP requests made and the amount of data moved during the test. Once a test completes, WebBench reports test results in requests per second and throughput in bytes per second.

Web servers are generally capable of handling HTTP requests for both static and dynamically generated content using both secure and non-secure connections. Figure 11 below shows the Web servers used for all test configurations. TUX is a kernel-based HTTP server available with Linux kernels 2.4 and later.

Operating System	Web Server
Windows Server 2003	IIS 6.0
Red Hat Linux Advanced Server 2.1	Apache 1.3.23, TUX 2.1
Red Hat Linux 8.0 Professional	Apache 2.0.40, TUX 2.1

Figure 11. Web server and version information for all test configurations

The following list describes the different types of tests we performed to measure Web server performance. Each item in the list describes a specific combination of content requested from the Web server.

- Static test suite requesting 100 percent static content
- Combination of 80 percent static content and 20 percent CGI-based dynamic content
- Combination of 76 percent static content, 16 percent CGI-based dynamic content and 8 percent Secure Socket Layer (SSL) 3.0 based static and CGI-based dynamic content

To test the DL380 system, we used the standard WebBench 4.1 static and dynamic test suites to generate the loads described above. The standard WebBench test suites use a total of 60 physical clients. Each test suite starts using a single load-generating client and slowly increases the load on the Web server by adding test clients in increments of four until a total of 60 clients have participated in the test. Each of the standard test suites described above started one WebBench engine running one thread to generate the load during the test.

Because the DL760 systems contained more memory and processing power compared to the DL380 system, we created a new set of test suites using the workloads from the standard test suites to test the DL760 systems. Like the standard test suites, these new test suites started with a single test client but increased the load on the Web server by adding test clients in groups of eight until a total of 120 clients had participated in the test. These new test suites used identical workloads compared to standard test suites, but were designed to put roughly twice the load on the server compared to the standard test suites. Like the standard WebBench test suites, each of the new test suites created to test the DL760 systems starts one WebBench engine running one thread on each physical test client to generate the load during the test.

The Web server performance testing consisted of running the test suites described above against the DL760 servers using 1, 2, 4 and 8 processor configurations running each of the following operating systems:

- Windows Server 2003

Red Hat Linux Advanced Server 2.1

Additionally, we ran each of the three standard test suites described above against the DL380 server using a two-processor configuration against the following operating systems:

Windows Server 2003
Red Hat Linux Advanced Server 2.1
Red Hat Linux 8.0 Professional

When testing the DL760 servers running Windows Server 2003, we modified the boot.ini file on the DL760 server containing four processors and four network segments to allow us to start the server using 1, 2, or 4 processors. For the one processor testing on the DL760 server, we loaded the appropriate uni-processor kernel and hardware abstraction library (HAL) from the Windows Server 2003 media sent by Microsoft for these tests.

When testing the DL760 servers running Red Hat Linux Advanced Server 2.1, we used the Enterprise SMP kernel (2.4.9-e.3enterprise) when testing with 2, 4 and 8 processors and the single processor kernel (2.4.9-e.3) when testing using a single processor. When conducting testing using two processors, we used the Linux boot option “maxcpus=2” to restrict the operating system to use only two processors.

Please refer to Appendix B of this report for details on how we installed and configured each of the operating systems listed above for the Web Server performance testing.

During the Web server performance testing, it became apparent that while the test suites described above were more than sufficient to determine the peak Web serving performance of most tested server configurations running Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional, they were not capable of saturating either the DL380 or the DL760 systems under certain combinations of processors and content type when running Windows Server 2003 and Red Hat Linux Advanced Server 2.1 on the DL380 server configured with two processors. Specifically, these scenarios are as follows:

DL760 configured with four processors serving 100 percent static content using IIS 6.0
DL760 configured with eight processors serving 100 percent static content using IIS 6.0
DL380 configured with two processors serving 100 percent static content using IIS 6.0
DL380 configured with two processors serving 100 percent static content using TUX

To determine the peak Web serving performance under the above scenarios, we created an additional set of three test suites that placed substantially more load on the Web server at all client load points. These test suites ran a single WebBench engine on each physical test client, but had each WebBench engine run different numbers of threads when requesting specific types of content from the Web server under test. These test suites are described in figure 12 below.

Server	# of Processors	Content Type	Client Thread Configuration
DL380	2	Static	4 threads per engine at all client load points
DL760	4	Static	4 threads per engine at all client load points
DL760	8	Static	10 threads per engine at client loads of 1 – 88, 15 threads per engine at client loads of 96 - 120

Figure 12. Custom multi-threaded WebBench Test Suites

To allow a direct comparison of test results across platforms, we ran the single threaded test suites described above on all configurations tested. We then used the multi-threaded test suites described in figure 12 to find the peak Web serving performance only for those scenarios where the single threaded test suites were not sufficient to determine the peak Web serving performance.

For all testing, we executed each of the test suites twice for each specific configuration to ensure the accuracy and repeatability of the test results. We then computed the average of these two test runs at each client load point to determine the results presented in this report.

For all Web server performance testing, the 240 network test clients ran Windows XP and Service Pack 1.

For the Web server performance testing using Windows Server 2003, we performed a series of operating system and testbed client tunings as specified by documentation provided from Microsoft.

Additionally, we spent considerable time investigating and testing potential performance tuning options for improving the Web server performance on the Red Hat Linux platforms tested using both TUX and Apache. While investigating tuning options, we looked at a number of items including previous competitive tests comparing Windows operating systems to Linux, as well as a variety of books and Web sites with information about tuning the performance of TUX and Apache. We gathered what appeared to be the most likely candidates to maximize the performance of these Web servers and then spent several days running a series of tests designed to determine which, if any of these tuning options actually made a difference in our Web server performance testing.

Please refer to Appendix C of this report for complete details of the tuning conducted for the Web server performance testing.

Test results

This section shows the results of the Web and File serving performance we conducted. Please refer to the Testing Methodology section for complete information on the tests we performed.

Web Server Performance Test Results

This section contains the detailed results of the Web server performance testing we conducted using Windows Server 2003, Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional on both the DL380 and DL760 servers.

In general, we found that Windows Server 2003 provided better overall peak Web serving performance compared to both Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional on all test configurations. This was true regardless of the specific content type used during the test. Again, when testing on the DL380 server, we used only the dual processor configuration. When testing on the DL760 server, we conducted tests using configurations of one, two, four and eight processors.

Static Content Results

We conducted these tests by configuring the WebBench test clients to make 100 percent of their requests for static content. Figure 13 shows the peak static requests per second values generated on both the DL380 and DL760 servers using all operating system and processor combinations. We found that regardless of the server employed or the number of processors, Windows Server 2003 generated better peak Web serving performance using static content compared to both Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional using our test configurations. This was particularly true when testing with the DL760 server configured with four and eight processors.

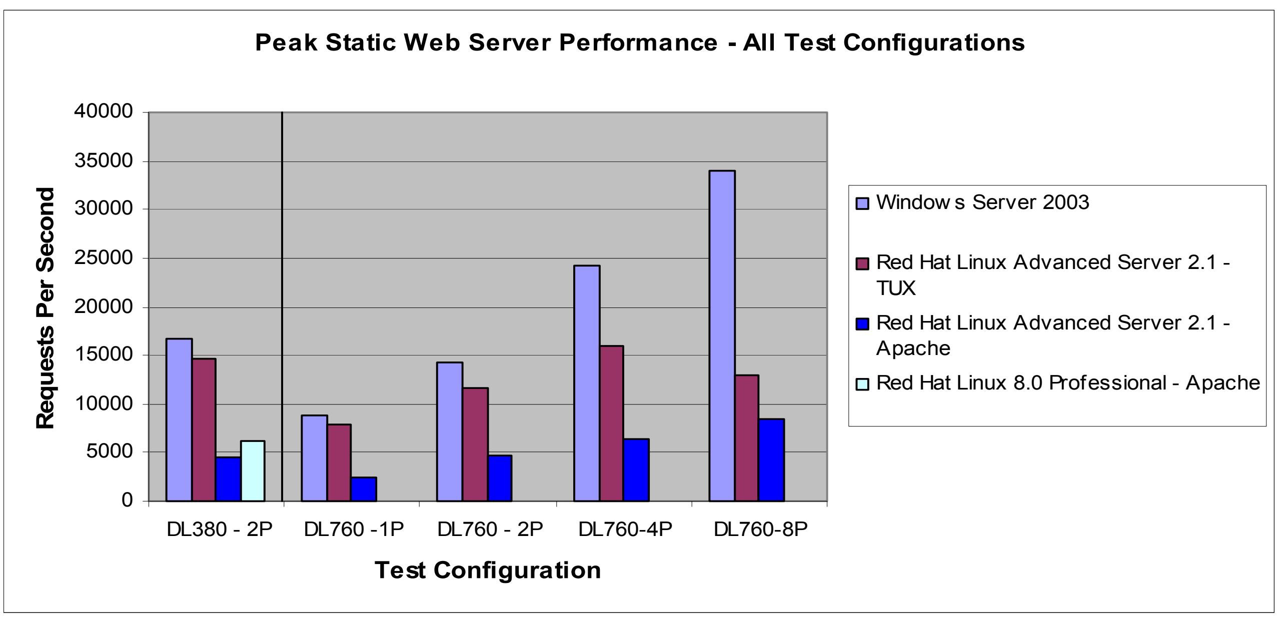

Figure 13. Peak Static Web Server Performance On All Test Configurations

We encountered an issue during testing when using TUX alone under Red Hat Linux Advanced Server 2.1 to deliver static content. As figure 13 shows, we observed good scaling of the static Web Server performance on the DL760 server regardless of the Web server employed until we tested using TUX with Red Hat Linux

Advanced Server 2.1 using eight processors. When testing with eight processors, the static test results actually decreased approximately 23 percent compared to the results generated using TUX with Red Hat Linux Advanced Server 2.1 using four processors.

We double checked the TUX and operating system configuration and ran additional tests to try and resolve this issue, but were ultimately unsuccessful. We submitted a formal support request with Red Hat Technical Support on April 8th, 2003 regarding this issue with TUX and provided problem and configuration details. On April 9th, 2003 we received an indication that out request had been escalated to the senior technical support staff. On April 30th we had still not received a response from Red Hat Technical Support that offered any options for resolving this issue. By this time, we were required to return the servers used during the testing to Hewlett-Packard. This meant that even if Red Hat Technical Support had responded with options for resolving this issue, we would no longer have the hardware necessary to conduct additional testing. Therefore, we published the existing numbers shown in this report.

Figure 14 below shows the actual peak WebBench static Web server performance results in requests per second generated on both the DL380 and DL760 server using all operating system and processor combinations. Additionally, figure 15 shows the percentage improvement in Static Web server performance when testing with Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. These results clearly show that, in our test configurations, significant performance improvements are possible when serving static Web content using Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 using Apache. Lesser, yet still noticeable improvements, are possible when serving static Web content using Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 using TUX.

Operating System	DL380 - 2P	DL760 -1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 and IIS 6.0	16783	8861	14214	24293	33991
Red Hat Linux Advanced Server 2.1 and TUX	14741	7880	11676	16035	13007
Red Hat Linux Advanced Server 2.1 and Apache 1.3.23	4467	2387	4639	6465	8496
Red Hat Linux 8.0 Professional and Apache 2.0.40	6260	N/A	N/A	N/A	N/A
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using TUX and Apache 1.3.23	14%	12%	22%	51%	161%
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using Apache 1.3.23	276%	271%	206%	276%	300%
Percent Improvement with Windows Server 2003 over Red Hat Linux 8.0 Professional using Apache 2.0.40	168%	N/A	N/A	N/A	N/A

Figure 14. Peak Static Web Server Performance and Percentage Improvement of Windows Server 2003 in Static Web Server Performance Data On All Test Configurations

Figure 15 below shows the results of the static content testing on the DL380 server platform for all operating systems tested using both the standard, single threaded static test suite and the multi-threaded static test suite using four threads on each physical test client. Analyzing the results, it is clear that the increasing trend of the result curve using the single-threaded static test suite with Windows Server 2003 and with TUX under Red Hat Linux Advanced Server 2.1 shows that we had not encountered the true peak capabilities of either Web server using the single threaded static test suite. Using the multi-threaded static test suite allows us to completely saturate both Web servers and find this peak.

When using the multi-threaded static test suite, Windows Server 2003 generated a peak of 16,783 requests per second. This is an increase of approximately 22 percent compared to the peak results generated using the single threaded static test suite. When using the multi-threaded static test suite, TUX under Red Hat Linux

Advanced Server 2.1 generated a peak of 14,741 requests per second. This is an increase of approximately 2 percent compared to the peak results generated using the single threaded static test suite.

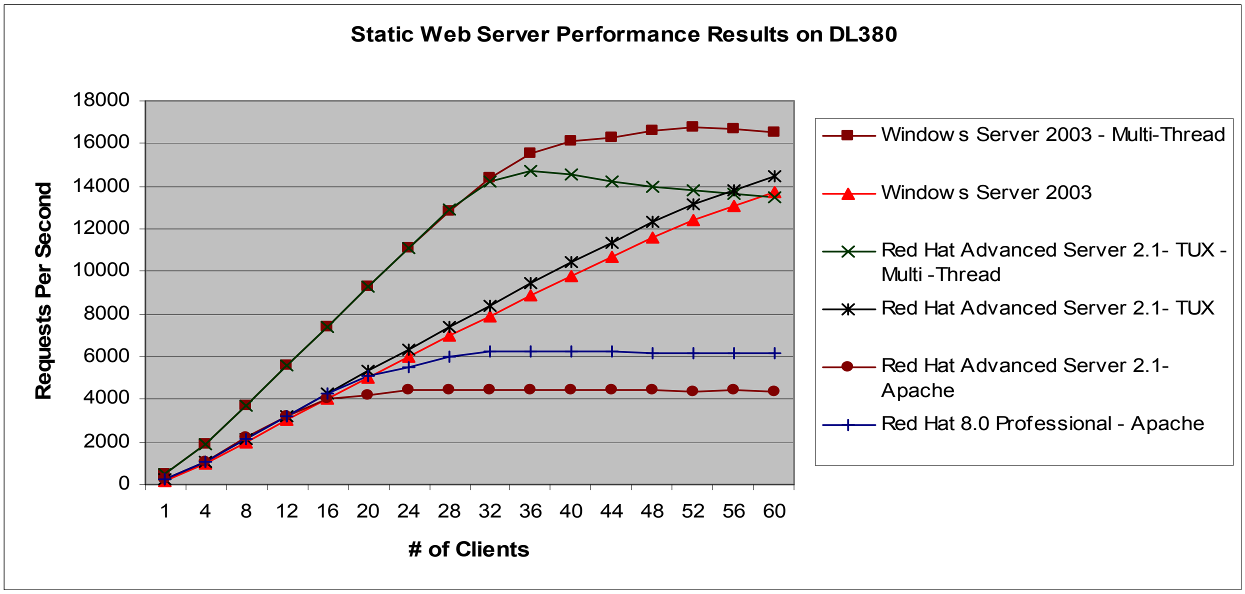

Figure 15. Static Web Server Performance Results on DL380 server configuration

Figures 16 - 19 below display the full set of WebBench data for the static Web server performance results on the DL760 server platform for all Operating Systems and processor configurations using 1, 2, 4 and 8 processors. These results show that, in our test configurations, in addition to providing better peak static Web server performance compared to TUX under Red Hat Linux Advanced Server 2.1, Windows Server 2003 provides significantly better static Web serving performance at lower, medium and high client loads compared to Apache on Red Hat Linux Advanced Server 2.1.

Figures 18 and 19 below show the results of the static content testing on the DL760 server platform using four and eight processors respectively for all operating systems tested. Analyzing the results, it is clear that the increasing trend of the result curve using the single-threaded static test suite with both four and eight processors with Windows Server 2003 shows that we had not encountered the true peak capabilities of the Web server. Using the multi-threaded static test suite allows us to completely saturate the IIS 6.0 Web server and find this peak with both four and eight processors.

When using the multi-threaded static test suite on the DL760 configured with four processors, Windows Server 2003 generated a peak of 24,293 requests per second compared to a peak of 20,886 requests per second using the single-threaded static test suite. This is an increase of approximately 16 percent compared to the peak results generated using the single threaded static test suite.

When using the multi-threaded static test suite on the DL760 configured with eight processors, Windows Server 2003 generated a peak of 33,991 requests per second compared to a peak of 23,387 requests per second using the single-threaded static test suite. This is an increase of approximately 45 percent compared to the peak results generated using the single threaded static test suite.

The test result curve in figure 20 showing the full WebBench multi-threaded static test results for the HP DL760 server using eight processors looks considerably different compared to the results generated by the single threaded version of the test suite. This is because the multi-threaded test suite used for this test utilized a total of 10 threads per each WebBench client through loads of up to 88 clients and then uses 15 threads per WebBench client at loads after 88 clients. This has the effect of suddenly increasing the level of the overall load placed on IIS 6.0 by the WebBench clients and results in a dramatic improvement in the number of requests sent by the WebBench clients and processed by IIS 6.0 during the test mixes.

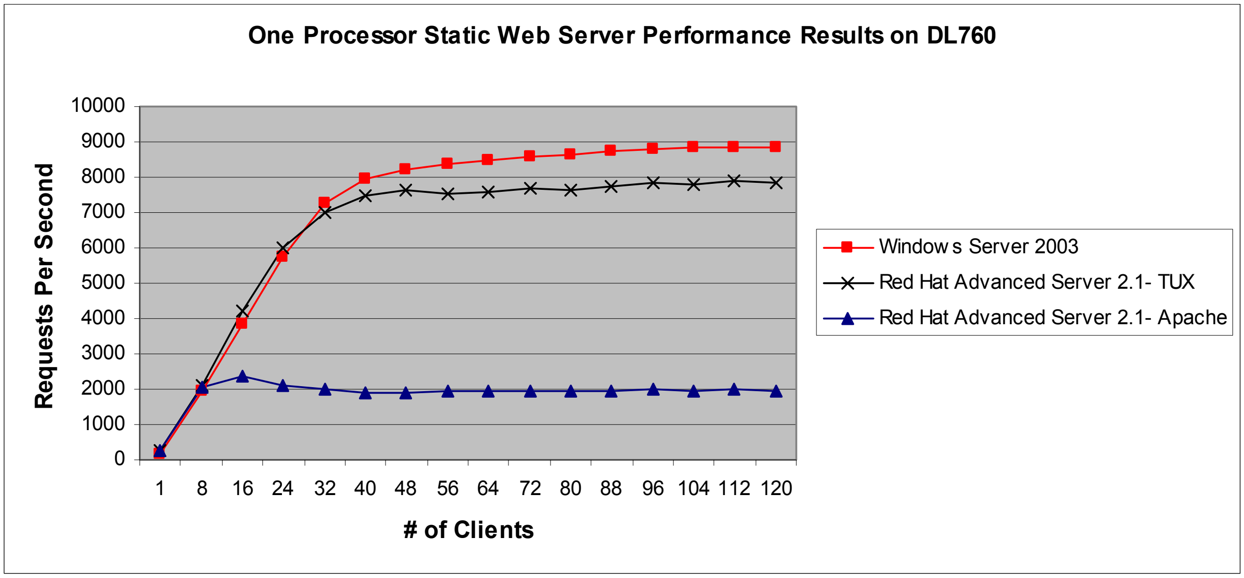

Figure 16. One processor Static Web Server Performance Results on DL760

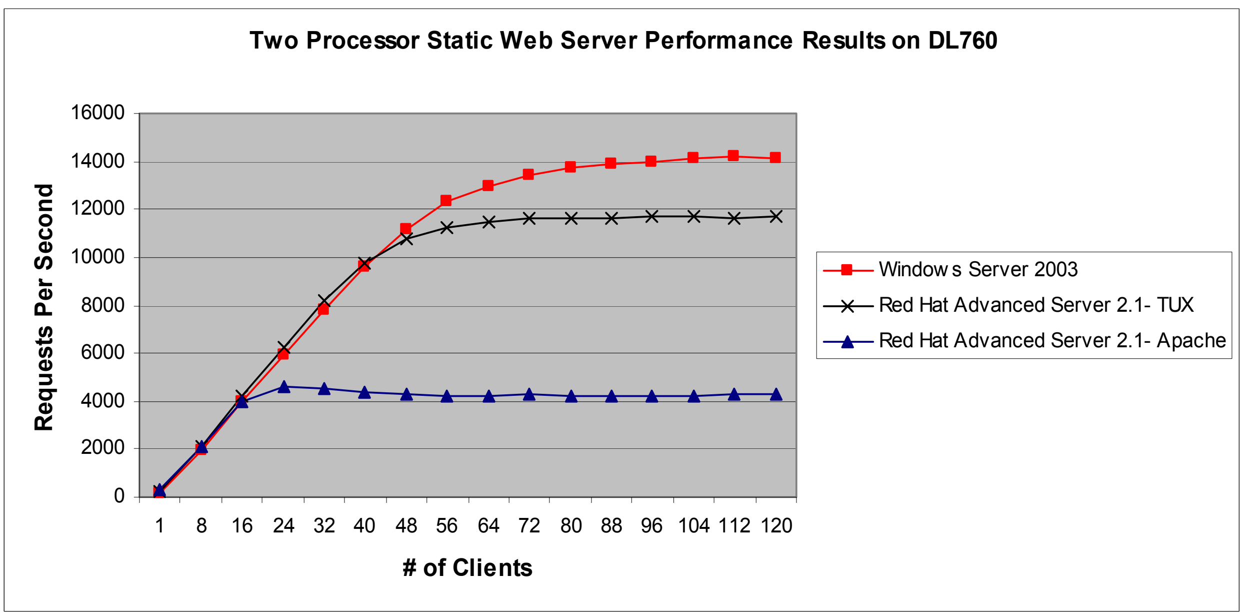

Figure 17. Two Processor Static Web Server Performance Results on DL760

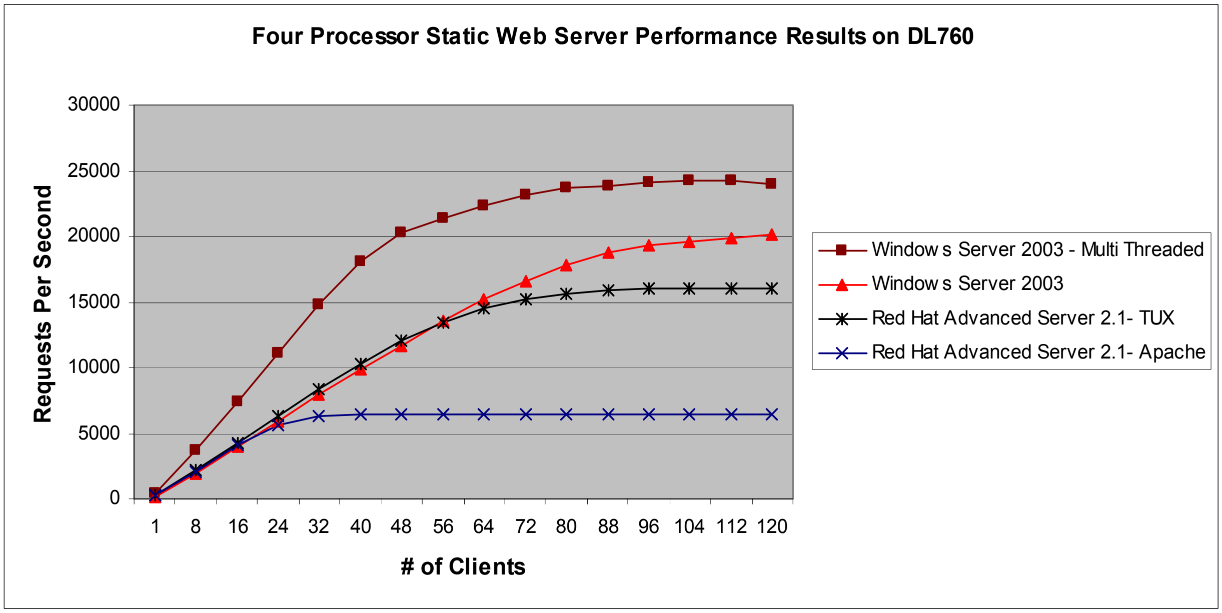

Figure 18. Four Processor Static Web Server Performance Results on DL760

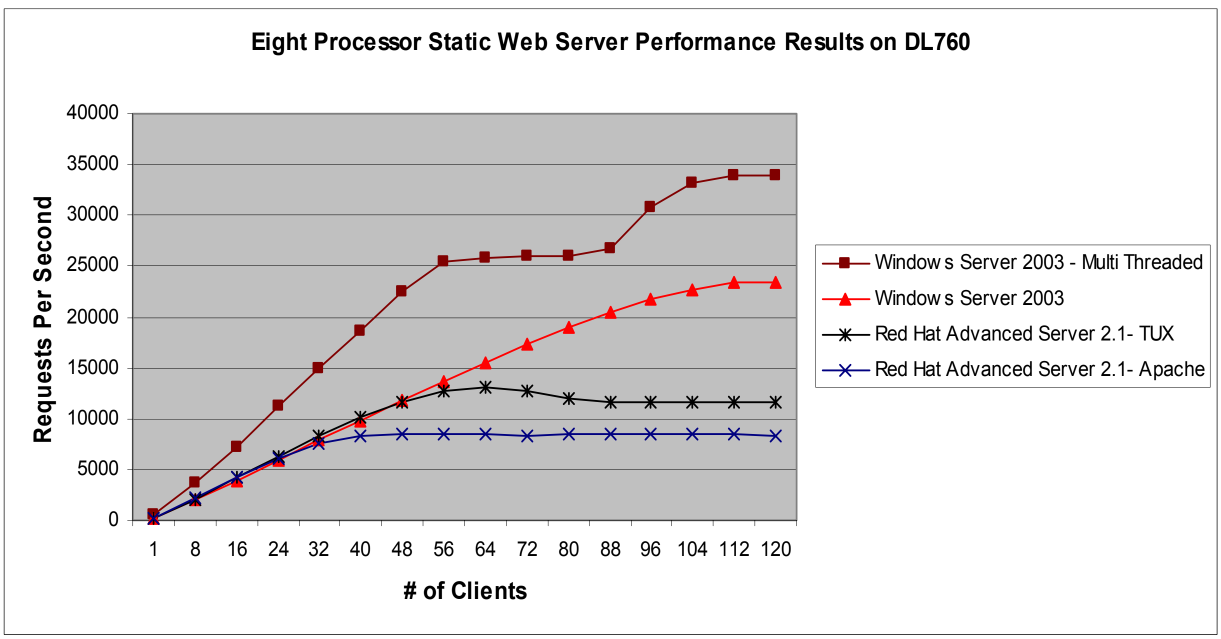

Figure 19. Eight Processor Static Web Server Performance Results on DL760

Dynamic Content Performance Test Results

We conducted these tests by configuring the WebBench test clients to make 80 percent of their requests for static content and 20 percent for a simple CGI executable. For the CGI tests that employed the TUX web server, we configured TUX to serve only static content and configured Apache to serve the CGI based content. The TUX and Apache Web servers then worked as a team to handle the mixture of static and CGI based content requested during these tests.

In addition to providing support for CGI, Windows Server 2003 supports the ISAPI interface as well for creating dynamic content. Using the ISAPI interface allows for improved performance when executing dynamic content using IIS 6.0. While testing using ISAPI content was not a direct part of this testing, it's worth noting the performance improvements that are possible when using the ISAPI interface compared to CGI.

Figure 20 shows the peak request per second values generated on both the DL380 and DL760 server using all operating system and processor combinations we tested. We found that regardless of the server employed or the number of processors, in our test configurations, Windows Server 2003 generated better peak Web serving performance using the dynamic CGI-based content compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. This was particularly true on the DL380 server and when using two and four processor configurations on the DL760 server. We also found that, for the most part, using TUX to serve the static portion of the requested content improved the overall CGI-based test results compared to using only Apache to serve both the static and CGI-based content.

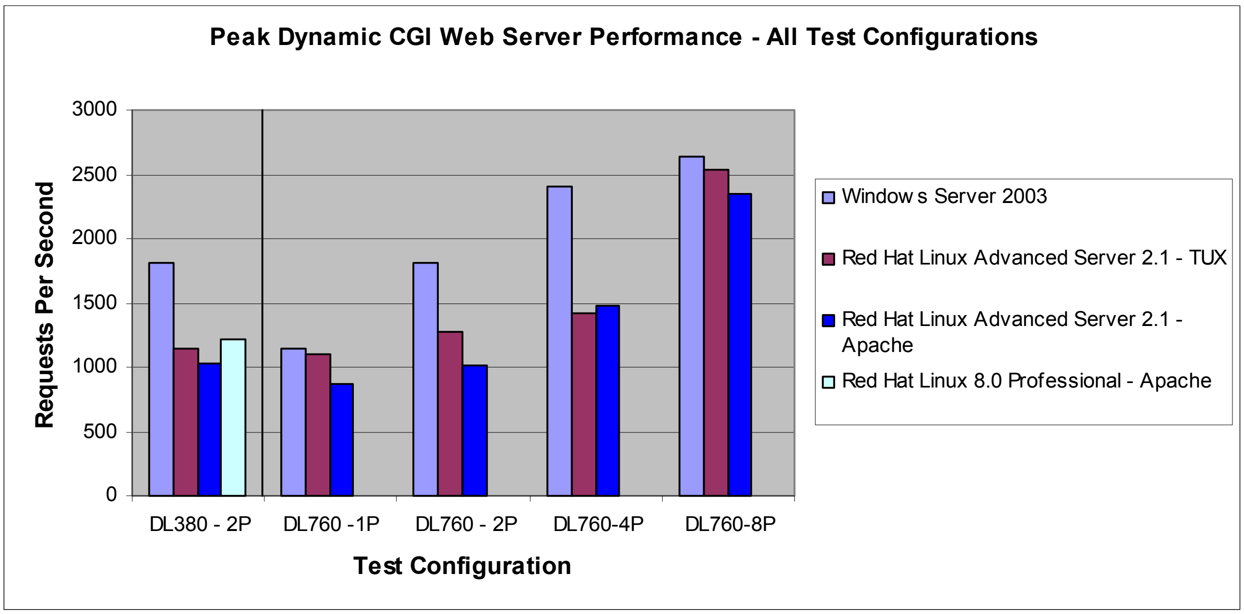

Figure 20. Peak Dynamic CGI-based Web Server Performance On All Test Configurations

Figure 21 below shows the actual peak WebBench CGI Web server performance results in requests per second generated on both the DL380 and DL760 server using all operating system and processor combinations. Additionally, figure 21 shows the percentage improvement in CGI-based Web server performance when testing with Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional using all operating system and processor combinations. These results clearly show that, in our test configurations, significant performance improvements are possible when serving a combination of static and dynamic CGI-based Web content when moving from either Red Hat Linux Advanced Server 2.1 or Red Hat Linux 8.0 Professional to Windows Server 2003.

Operating System	DL380 - 2P	DL760 -1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 and IIS 6.0	1814	1146	1805	2413	2639
Red Hat Linux Advanced Server 2.1 and TUX	1151	1103	1269	1418	2534
Red Hat Linux Advanced Server 2.1 and Apache 1.3.23	1031	870	1019	1476	2352
Red Hat Linux 8.0 Professional and Apache 2.0.40	1211	N/A	N/A	N/A	N/A
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1	58%	4%	42%	70%	4%

using TUX and Apache 1.3.23					
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using Apache 1.3.23	76%	32%	77%	63%	12%
Percent Improvement with Windows Server 2003 over Red Hat Linux 8.0 Professional using Apache 2.0.40	50%	N/A	N/A	N/A	N/A

Figure 21. Peak Dynamic CGI Web Server Performance and Percentage Improvement of Windows Server 2003 in CGI Web Server Performance Data On All Test Configurations

Figure 22 compares the dynamic CGI-based test results from this report using Windows Server 2003 and a set of test results using the ISAPI interface under Windows Server 2003 generated on the identical server hardware as the other testing in this report

It's clear from the graph that using the ISAPI interface for generating dynamic content can significantly improve the overall Web server performance when serving dynamic content.

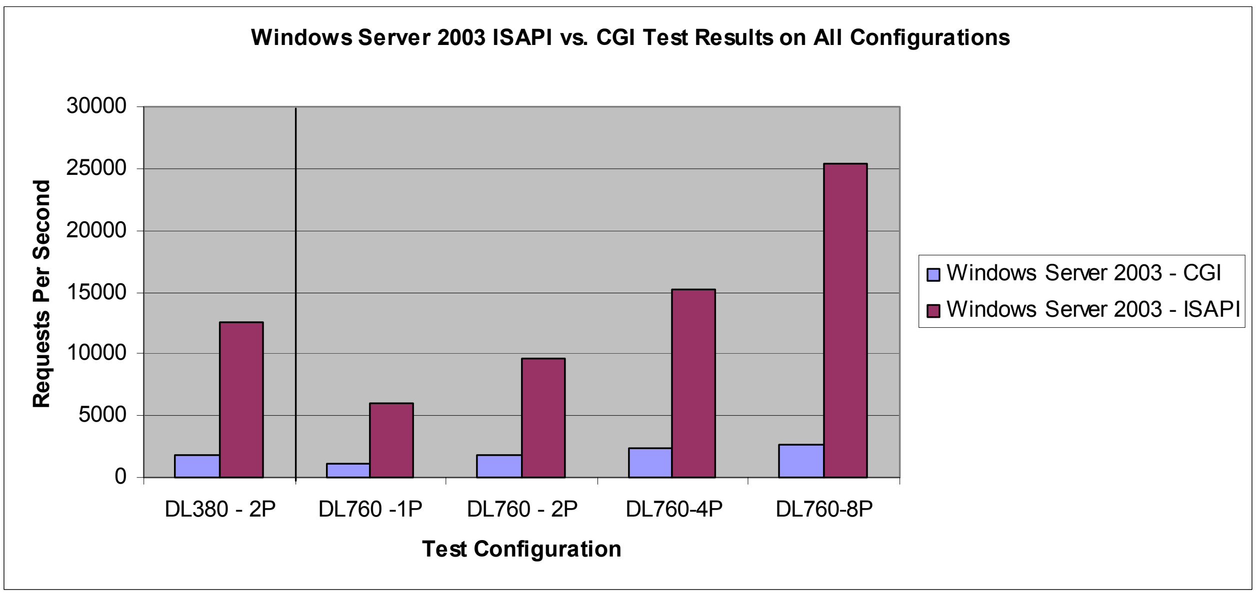

Figure 22. Windows Server 2003 ISAPI vs. CGI Test Results on All Configurations

Figure 23 shows the actual peak dynamic CGI and ISAPI test results and the percentage increase in performance when using the ISAPI interface compared to the CGI interface.

Operating System	DL380 - 2P	DL760 - 1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 - CGI	1814	1146	1805	2413	2639
Windows Server 2003 - ISAPI	12551	6054	9685	15270	25329
Percentage Improvement Using ISAPI Compared to CGI on Windows Server 2003	592%	428%	437%	533%	860%

Figure 23. Peak Dynamic ISAPI and CGI Web Server Performance and Percentage Improvement When using ISAPI vs. CGI on Windows Server 2003

Figure 24 below shows the results of the CGI content testing on the DL380 server platform for all operating systems tested using the standard, single threaded CGI test suite. These results show that using the single-threaded CGI test suite, Windows Server 2003 delivered the best overall Web serving performance of all platforms tested. Additionally, figure 24 shows a complete set of ISAPI based test results generated on the DL380 that shows the increase in the Web server performance using ISAPI compared to CGI using IIS 6.0 under Windows Server 2003.

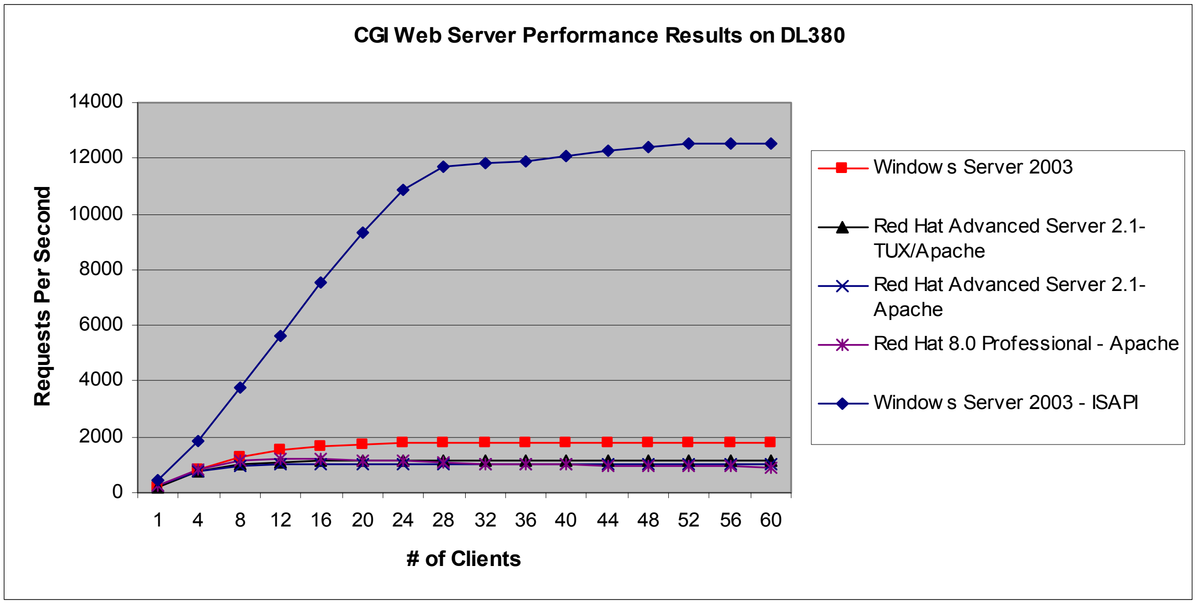

Figure 24. CGI-based Web Server Performance Results on DL380

Figures 25 - 28 below display the full set of WebBench data for the CGI Web server performance results on the DL760 server platform for all operating systems and processor configurations using 1, 2, 4 and 8 processors. These results show that, in the majority our test configurations, Windows Server 2003 using IIS 6.0 provides better dynamic CGI Web serving performance at lower, medium and high client loads compared to Red Hat Linux Advanced Server 2.1 using both Apache only and a combination of TUX to handle the static requests and Apache to handle the CGI requests.

Additionally, figures 25 - 28 shows a complete set of ISAPI based test results generated on the DL760 at all processor configurations that shows the increase in the Web server performance using ISAPI compared to CGI using IIS 6.0 under Windows Server 2003.

The test result curve in figure 28 showing the full WebBench ISAPI test results for the HP DL760 server using eight processors looks considerably different compared to the results generated by the single threaded version of the test suite. This is because these results were generated using a multi-threaded test suite similar to that used to generate the peak static results shown in this report. This multi-threaded ISAPI based test suite utilized a total of 10 threads per each WebBench client through loads of up to 88 clients and then uses 15 threads per WebBench client at loads after 88 clients. This has the effect of suddenly increasing the level of the overall load placed on IIS 6.0 by the WebBench clients and results in a dramatic improvement in the number of requests sent by the WebBench clients and processed by IIS 6.0 during the test mixes.

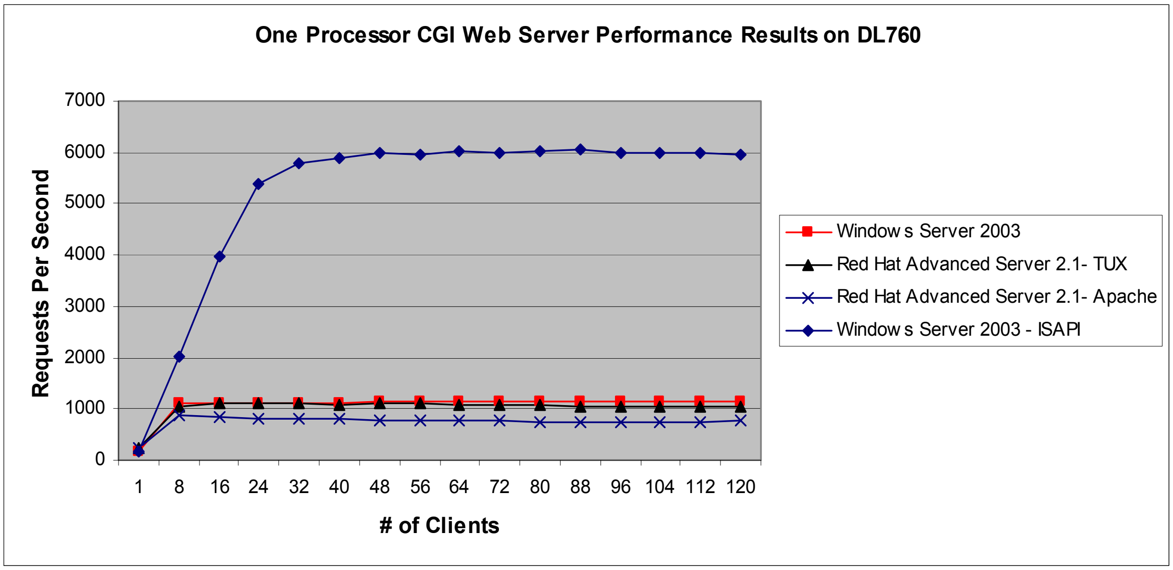

Figure 25. One Processor CGI-based Web Server Performance Results on DL760

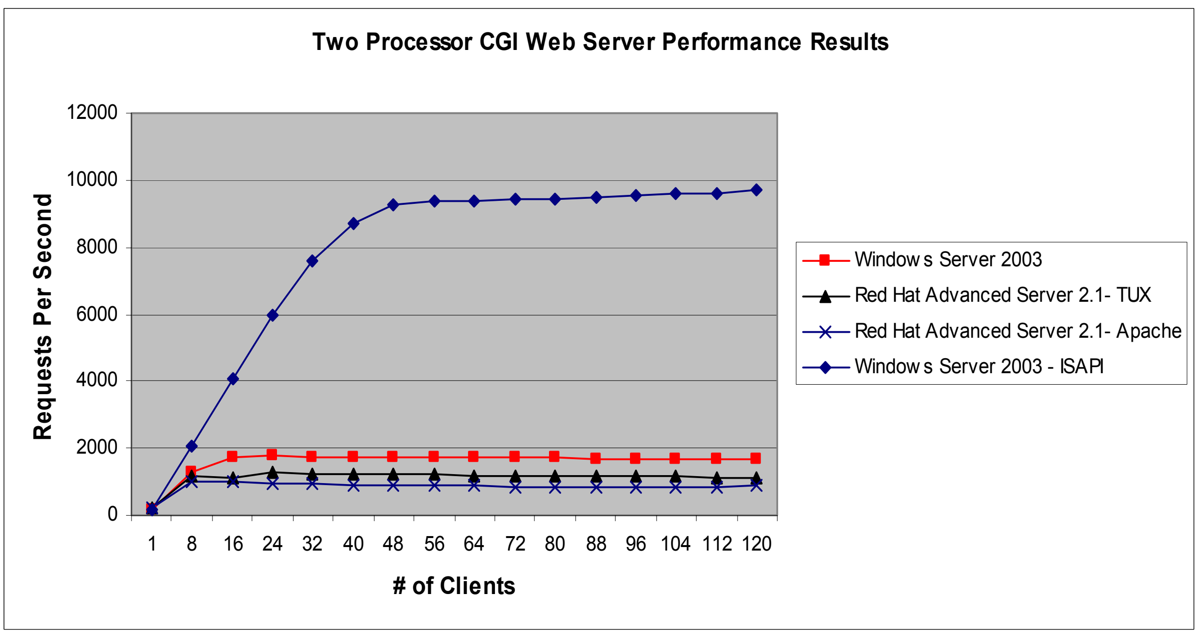

Figure 26. Two Processor CGI-based Web Server Performance Results on DL760

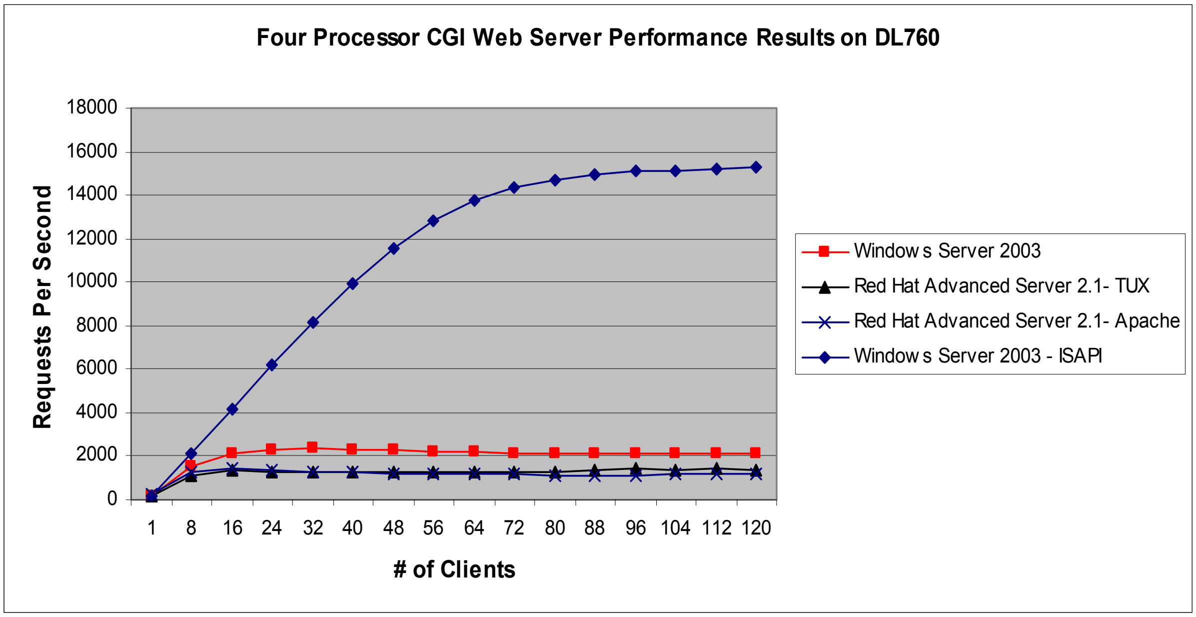

Figure 27. Four Processor CGI-based Web Server Performance Results on DL760

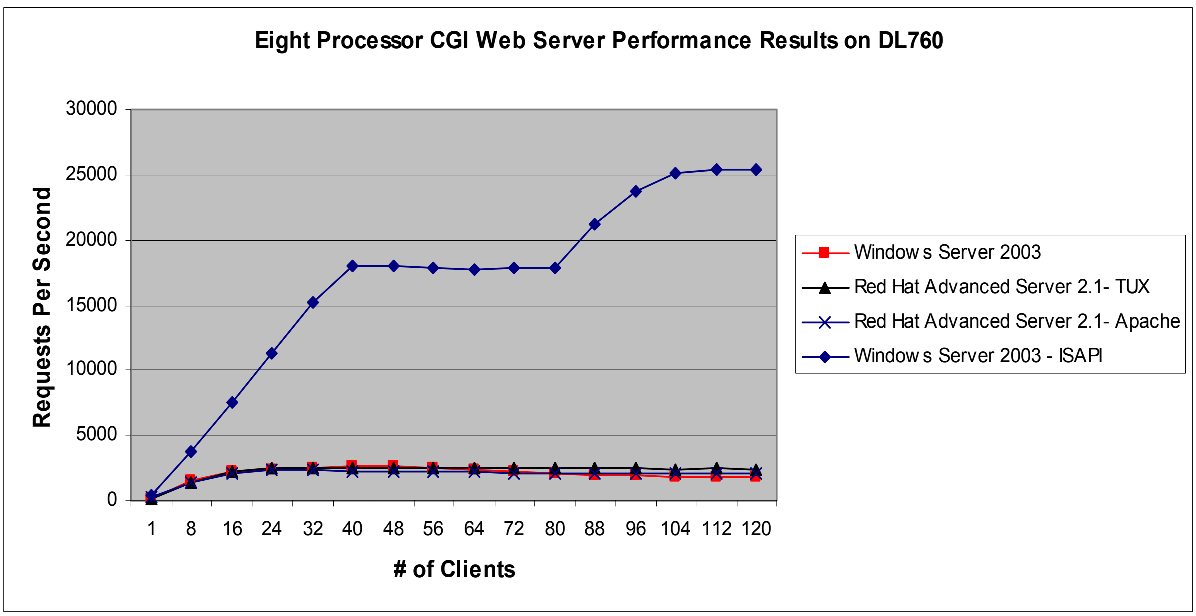

Figure 28. Eight Processor CGI-based Web Server Performance Results on DL760

E-Commerce Performance Test Results

We conducted these tests by configuring the WebBench test clients to make 76 percent of their requests for static content, 16 percent for a simple CGI executable and the remaining 8 percent for static and CGI-based content using SSL 3.0 for secure Web server communications. For the SSL/CGI tests that employed the TUX web server, we configured TUX to serve only non-secure static content and configured Apache to serve the secure static as well as the secure and non-secure CGI based content. The TUX and Apache Web servers then worked as a team to handle the mixture of secure and non-secure static and CGI based content requested during these tests.

Figure 29 shows the peak request per second values generated on both the DL380 and DL760 server using all operating system and processor combinations we tested. We found that regardless of the server employed or the number of processors, using our test configurations, Windows Server 2003 generated better peak Web serving performance using the SSL/CGI-based content compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. This was particularly true on the DL380 server and when using two and four processor configurations on the DL760 server. We also found that, for the most part, using TUX to serve the non-secure static portion of the requested content improved the overall SSL/CGI test results compared to using only Apache to serve both the static and CGI based content.

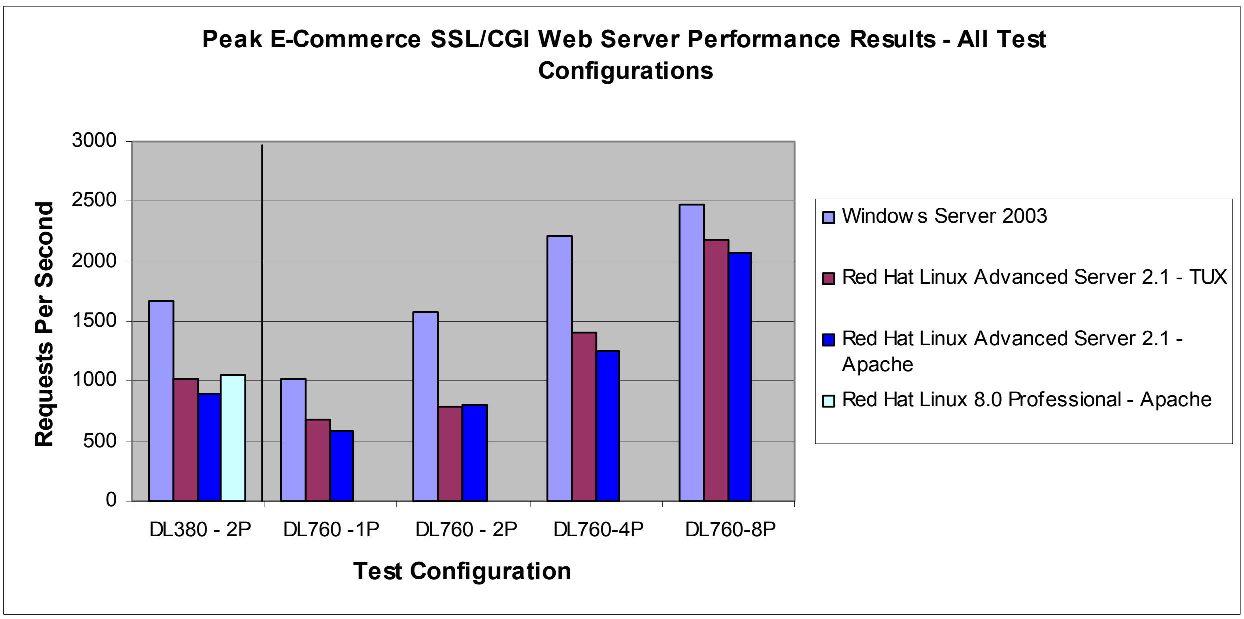

Figure 29. Peak E-Commerce SSL/CGI-based Web Server Performance On All Test Configurations

Figure 30 below shows the actual peak WebBench SSL/CGI Web server performance results in requests per second generated on both the DL380 and DL760 server using all operating system and processor combinations. Figure 30 also shows the percentage improvement in SSL/CGI-based Web server performance when testing with Windows Server 2003 compared to Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional. These results clearly show that, in our test configurations, significant performance improvements are possible when serving a combination of static and SSL/CGI-based Web content when moving from either Red Hat Linux Advanced Server 2.1 or Red Hat Linux 8.0 Professional to Windows Server 2003.

Operating System	DL380 - 2P	DL760 -1P	DL760 - 2P	DL760-4P	DL760-8P
Windows Server 2003 and IIS 6.0	1668	1020	1580	2214	2480
Red Hat Linux Advanced Server 2.1 and TUX	1019	673	785	1407	2181
Red Hat Linux Advanced Server 2.1 and Apache 1.3.23	900	589	808	1247	2066
Red Hat Linux 8.0 Professional and Apache 2.0.40	1058	N/A	N/A	N/A	N/A
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using TUX and Apache 1.3.23	64%	52%	101%	57%	14%
Percent Improvement with Windows Server 2003 over Red Hat Linux Advanced Server 2.1 using Apache 1.3.23	85%	73%	96%	78%	20%
Percent Improvement with Windows Server 2003 over Red Hat Linux 8.0 Professional using Apache 2.0.40	58%	N/A	N/A	N/A	N/A

Figure 30. Peak E-Commerce SSL/CGI Web Server Performance and Percentage Improvement of Windows Server 2003 in SSL/CGI Web Server Performance Data On All Test Configurations

Figure 31 compares the dynamic E-Commerce SSL/CGI-based test results from this report using Windows Server 2003 and a set of test results using the dynamic E-Commerce SSL/ISAPI interface under Windows Server 2003 generated on the identical server hardware as the other testing in this report.

It's clear from the graph that using the ISAPI interface for generating dynamic content significantly improves the overall Web server performance when serving dynamic content.

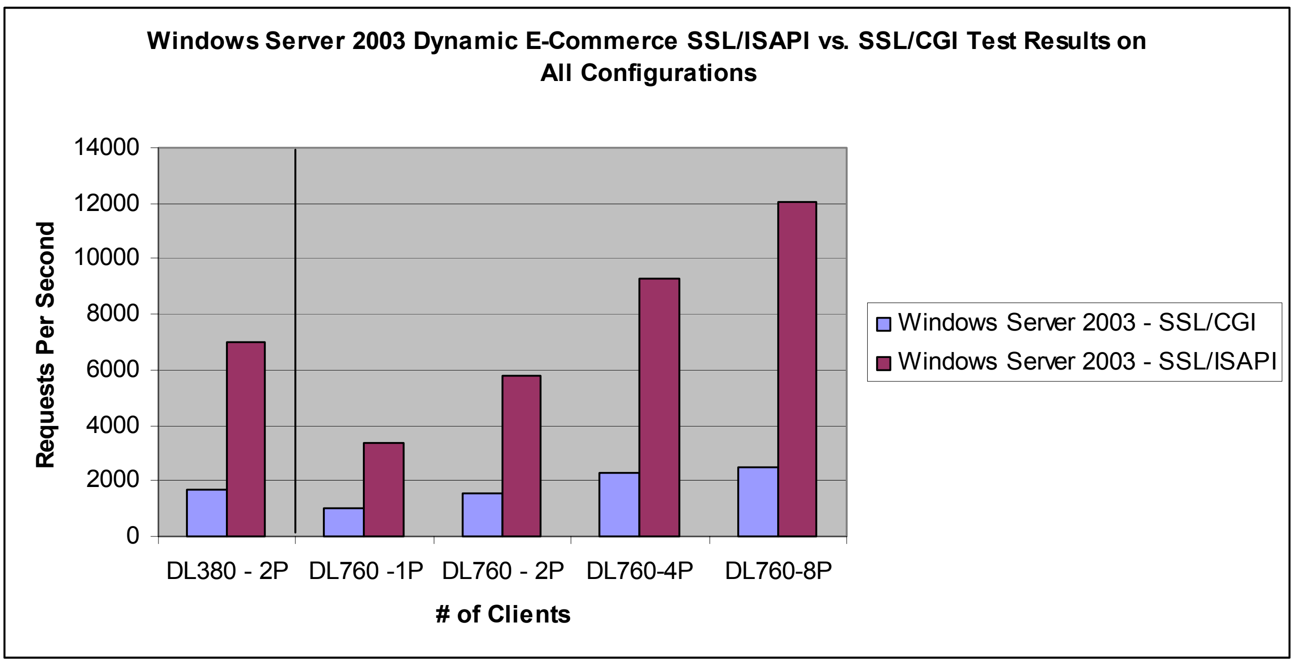

Figure 31. Windows Server 2003 SSL/ISAPI vs. SSL/CGI Test Results on All Configurations

Figure 32 shows the actual peak dynamic CGI and ISAPI test results and the percentage increase in performance when using the ISAPI interface compared to the CGI interface.

Operating System	DL380 - 2P	DL760 - 1P	DL760 - 2P	DL760- 4P	DL760- 8P
Windows Server 2003 - SSL/CGI	1668	1020	1580	2268	2480
Windows Server 2003 - SSL/ISAPI	6999	3338	5768	9285	12079
Percentage Improvement Using SSL/ISAPI Compared to SSL/CGI on Windows Server 2003	320%	227%	265%	309%	387%

Figure 32. Peak Dynamic E-Commerce SSL/ISAPI and SSL/CGI Web Server Performance and Percentage Improvement When using SSL/ISAPI vs. SSL/CGI on Windows Server 2003

Figure 33 below shows the results of the SSL/CGI content testing on the DL380 server platform for all operating systems tested using the standard, single threaded CGI test suite. These results show that using the single-threaded SSL/CGI test suite, Windows Server 2003 delivered the best overall Web serving performance of all platforms tested. Additionally, figures 31 shows a complete set of SSL/ISAPI based test results generated on the DL380 at all processor configurations that shows the increase in the Web server performance using SSL/ISAPI compared to SSL/CGI using IIS 6.0 under Windows Server 2003.

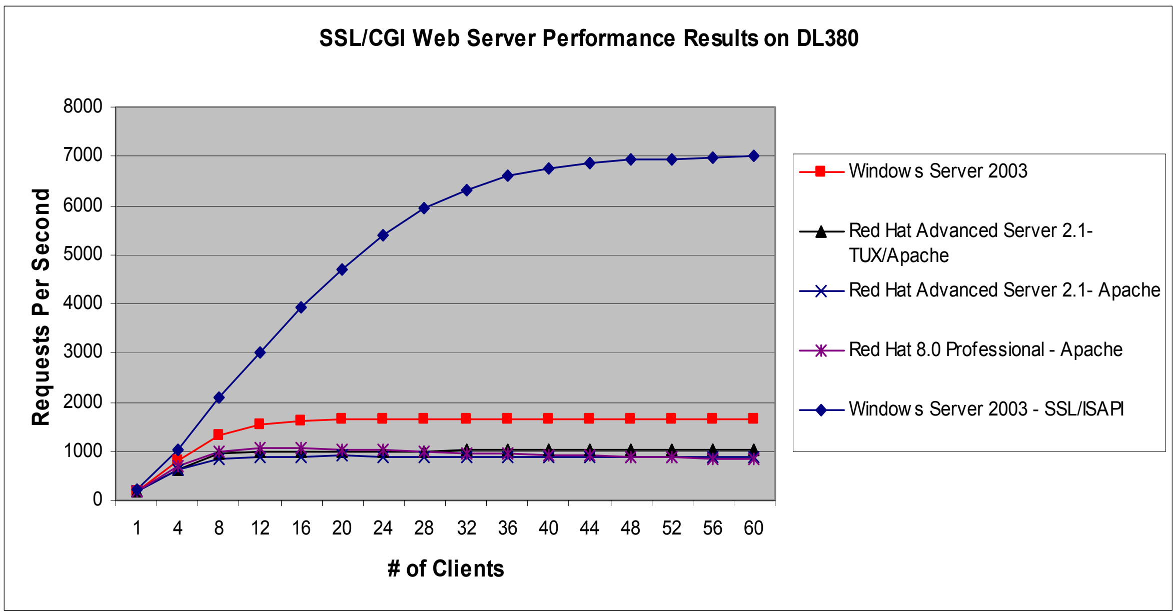

Figure 33. SSL/CGI-based Web Server Performance Results on DL380

Figures 34 - 37 below display the full set of WebBench data for the SSL/CGI Web server performance results on the DL760 server platform for all operating systems and processor configurations using 1, 2, 4 and 8 processors. These results show that, in the majority of our test configurations, Windows Server 2003 using IIS 6.0 provides better dynamic SSL/CGI Web serving performance at lower, medium and high client loads compared to Red Hat Linux Advanced Server 2.1 using both Apache only and a combination of TUX to handle the non-secure static requests and Apache to handle the secure and non-secure SSL and CGI requests.

Additionally, figures 34 - 37 shows a complete set of SSL/ISAPI based test results generated on the DL760 at all processor configurations that shows the increase in the Web server performance using SSL/ISAPI compared to SSL/CGI using IIS 6.0 under Windows Server 2003.

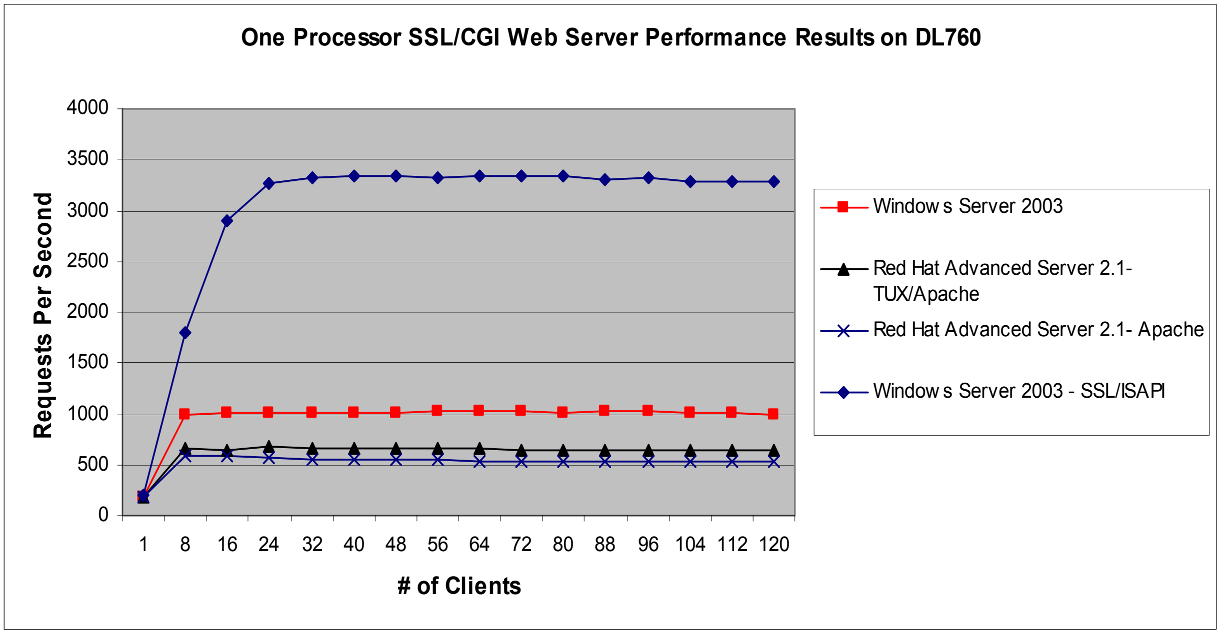

Figure 34. One Processor SSL/CGI-based Web Server Performance Results on DL760

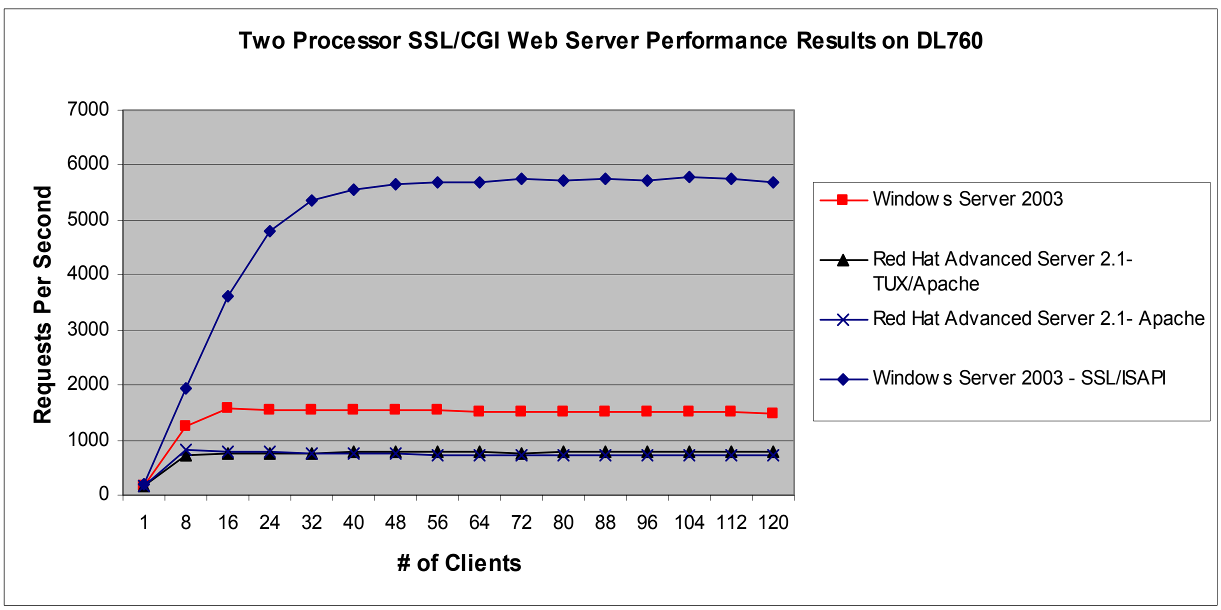

Figure 35. Two Processor SSL/CGI-based Web Server Performance Results on DL760

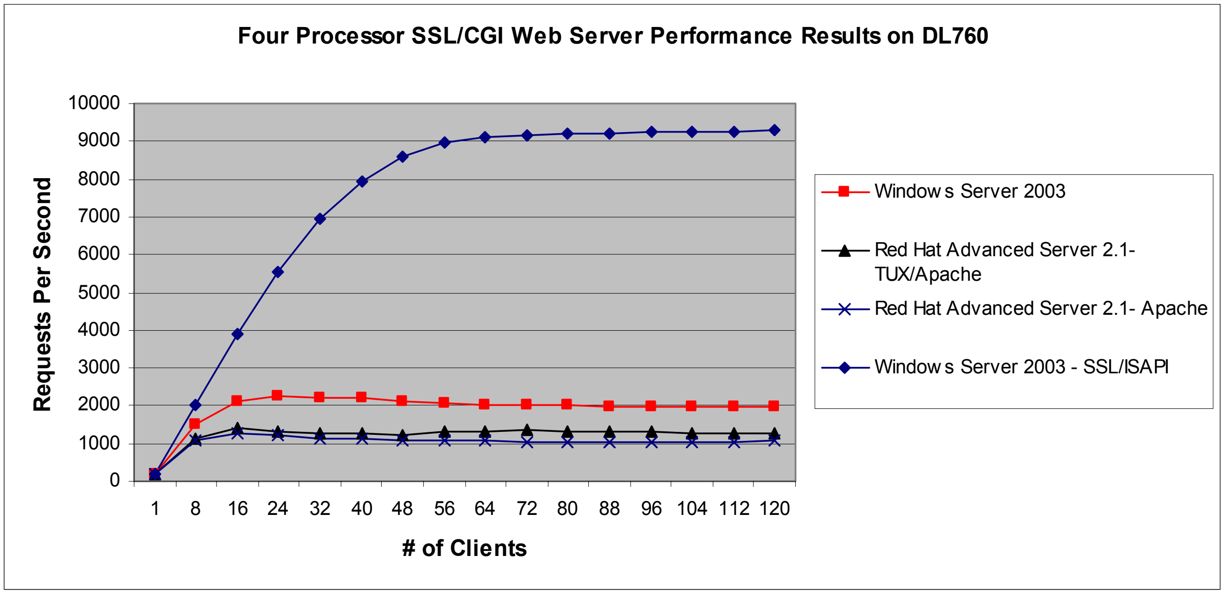

Figure 36. Four Processor SSL/CGI-based Web Server Performance Results on DL760

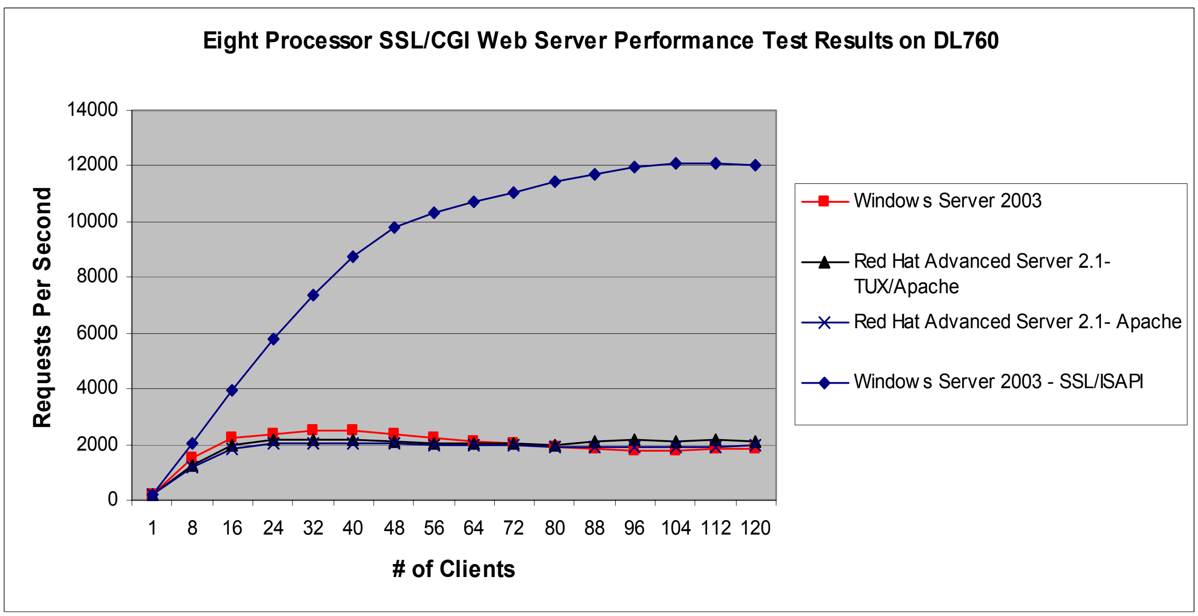

Figure 37. Eight Processor SSL/CGI-based Web Server Performance Results on DL760

Appendix A. Test Server and Network Client Configuration Information

Network Testbed Clients	
Machine Type	Dell PowerEdge 350
BIOS	Intel
Processor(s)	Intel PIII 850MHz
Hard Drive	10GB IDE
Memory	256MB
L2 Cache	256K
Motherboard	Intel
Network Adapter(s)	Intel Pro100 Management Adapter
Video Card	NVIDIA GeForce2 MX
OS	Windows XP/SP1

Figure 38. Network Testbed Client Disclosure Information

DL760 – 8P Configuration Information	
Machine Type	Compaq DL760
BIOS	Compaq
Hard Drive	4 x 36GB 15,000 RPM Ultra3 SCSI
Processor(s)	8 x Intel 900Mhz Pentium III Xeon
Memory	4GB
L2 Cache	2MB
Motherboard	Intel
Network Adapter(s)	8 x Intel PRO 1000 MF Server Adapters
Video Card	ATI 3D RAGE II PCI
OS	Windows Server 2003, Red Hat Linux Advanced Server 2.1, Red Hat Linux 8.0 Professional

Figure 39. DL760 – 8P Server Disclosure Information

DL760 – 1, 2 , and 4P Configuration Information	
Machine Type	HP Proliant DL760
BIOS	Compaq
Hard Drive	4 x 36GB 15,000 RPM Ultra3 SCSI
Processor(s)	4 x Intel 900Mhz Pentium III Xeon
Memory	4GB
L2 Cache	2MB
Motherboard	Intel
Network Adapter(s)	4 x Intel PRO 1000 MF Server Adapters
Video Card	ATI 3D RAGE II PCI
OS	Windows Server 2003, Red Hat Linux Advanced Server 2.1, Red Hat Linux 8.0 Professional

Figure 40. DL760 – 1P, 2P, and 4P Server Disclosure Information

DL380 – 2P Configuration Information	
Machine Type	HP ProLiant DL380 G2
BIOS	Compaq
Processor(s)	2 x Pentium III 1.4 GHz
Hard Drive	6 x 36GB 15,000 RPM Ultra3 SCSI
Memory	2GB
L2 Cache	512K
Motherboard	Intel
Network Adapter(s)	2 x Intel PRO 1000 MF Server Adapters
Video Card	ATI 3D RAGE II PCI
OS	Windows Server 2003, Red Hat Linux Advanced Server 2.1, Red Hat Linux 8.0 Professional

Figure 41. DL380-2P Server Disclosure Information

Appendix B. Operating System Installation and Configuration

This section describes the basic steps we performed to install each of the operating systems used during these tests. Regardless of the operating system used, we configured the RAID subsystems on each server the same way for all testing using HP's SmartStart 6.0 utility and selecting the defaults as shown in figure 42 below.

RAID Controller Parameter	Value
Expanded Priority	Low
Rebuild Priority	Low
Cache Ratio	50% READ / 50% WRITE
Stripe Size	128K

Figure 42. Default RAID Controller Parameters

For the DL760 server configured with eight processors, we configured the 28 drives connected to the SmartArray 5300 controller into four logical RAID 0 data volumes of approximately 121 GB each. Each logical volume was created using the default RAID controller parameters listed in figure 42. During installing the specific operating system, we used the appropriate disk management utilities to create two volumes on each of the four 121GB logical RAID 0 volumes for a total of eight volumes of approximately 60GB each. Figure 43 below shows the file system parameters used for each of the operating systems tested on the DL760 server configured with eight processors.

Operating System	# of Volumes	Volume Size	Format Type	Block Size
Windows Server 2003	8	60GB	NTFS	64K bytes
Red Hat Linux Advanced Server 2.1	8	60GB	ext3	default

Figure 43. File system parameters for DL760 server configured with eight processors

For the DL760 server configured with four processors, we configured the 28 drives connected to the SmartArray 5300 controller into four logical RAID 0 data volumes of approximately 120 GB each. Each logical volume was created using the default RAID controller parameters listed in figure 42. After installing the specific operating system, we used the disk management utilities to create one volume on each of the four 120GB logical RAID 0 volumes for a total of four volumes of approximately 120 GB each. Figure 44 below shows the file system parameters used for each of the operating systems tested on the DL760 server configured with four processors.

Operating System	# of Volumes	Volume Size	Format Type	Block Size
Windows Server 2003	4	120GB	NTFS	64K bytes
Red Hat Linux Advanced Server 2.1	4	120GB	ext3	default

Figure 44. File system parameters for DL760 server configured with four processors

Additionally, for the DL760 servers, we configured one of the four physical drives connected to the embedded SmartArray 5i as a volume of approximately 36GB using default RAID controller parameters. The operating system was installed on this single 36GB volume.

For the DL380 server, we configured one of the six physical drives into a logical volume of approximately 36GB using the default RAID controller parameters. The operating system was installed on this volume. We then configured four drives connected to the SmartArray 5i controller into a single logical RAID 0 data volume of approximately 140 GB using the default RAID controller parameters described above. After installing the specific operating system, we used the disk management utilities to create four basic volumes on the single logical RAID 0 volume each approximately 36GB. Figure 45 below shows the file system parameters used for each of the operating systems tested on the DL380 server.

Operating System	# of Volumes	Volume Size	Format Type	Block Size
Windows Server 2003	4	36GB	NTFS	32K bytes
Red Hat Linux Advanced Server 2.1	4	36GB	ext3	default
Red Hat Linux 8.0 Professional	4	36GB	ext3	default

Figure 45. File system parameters for DL380 server

For all Windows Server 2003 configurations tested, we increased the size of the NTFS log file to 64K bytes for each data volume using the following command:

```
Chkdsk /x <drive>: /l:65536
```

The following sections describe the specific steps we took to install the operating systems used in these tests.

Windows Server 2003

Microsoft provided a fully functional copy of Windows Server 2003 Enterprise Edition Release Candidate 2 (RC2) for these tests. To install this operating system, we performed the following steps:

- Using SmartStart 6.0, selected Microsoft Windows Server 2003 as the operating system to install and began the installation process
- During installation, configured the network parameters to match the client testbed segments.
- Installed the intfltr.sys processor affinity module and optimally configured the affinity to map processors to NIC's for all multi-processor tests.
- Configured the RAID subsystem as described above.

Red Hat Linux Advanced Server 2.1

Red Hat Linux Advanced Server 2.1 is the enterprise solution offering from Red Hat. This software is designed for the enterprise for use with large departmental and datacenter deployments. There is generally a long release cycle between versions of this operating system and it is billed as being a very stable product that is tuned specifically for improved performance on SMP systems using up to eight processors and 16GB of RAM. For the Web Server performance testing, we installed the Red Hat Linux Advanced Server 2.1 product as follows selecting default installation options except where noted below:

- Configured the RAID subsystem as described above.
- Rebooted server to start installation process
- Selected "custom" installation option and accepted all pre-selected items
- Selected to install tools for software development
- Selected to install tools for kernel development
- Selected the kernel-enterprise and kernel-smp packages to load kernel sources
- Downloaded and installed the latest Linux version of the Intel PRO/1000 Gigabit NIC drivers available from Intel Web site (driver version 4.4.19). Used the default settings per recommendations in the README file.
- During installation, configured the network parameters to match the client testbed segments.
- Configured processor affinity to bind the interrupts from the NIC's to specific processors where appropriate. This was only performed on configurations that utilized multiple processors.

We checked the Red Hat Web site for available updates and errata for the Red Hat Linux Advanced Server 2.1 product and found no bug fixes or enhancements related to Apache or TUX performance issues. As a result, we applied no additional patches and made no additional modifications to the Red Hat Linux Advanced Server 2.1 distribution used for these tests.

To maximize Web Server performance on the DL760 server running Red Hat Linux Advanced Server 2.1, we configured the processor affinity feature available through the operating system to bind individual IRQ values associated with individual network interface cards (NIC's) in the server under test to a specific processor in the server under test. When configuring processor affinity for use with Red Hat Linux Advanced Server 2.1 on the DL760 server configured with four processors and four network segments, we were only able to associate a total of three specific IRQ's with the four NIC's in the server.

To try and alleviate this, we located the four NIC's in the DL760 server using a variety of different slot combinations spread over the two PCI busses in the DL760 server. In the end we were not able to associate more than three distinct IRQ's with the four NIC's when conducting Web Server performance tests with Red

Hat Linux Advanced Server 2.1 on the DL760 server configured with four processors. This meant that two of the four processors in the DL760 server serviced only individual NICs while the remaining two processors in the DL760 server combined to service the remaining two NIC's that shared the same IRQ.

We did not encounter this situation on the DL760 server configured with eight NIC's or the DL380 server configured with two NIC's. In both of these configurations, a unique IRQ value was associated with each of the NIC's in the server. This allowed us to map each of the NIC's in the server to a separate processor for optimal use of the processor affinity feature.

Red Hat Linux 8.0 Professional

Red Hat Linux 8.0 Professional is the Red Hat offering for Small Office Home Office (SOHO) users as well as other non-enterprise installations. For the Web Server performance testing, we installed the Red Hat Linux 8.0 Professional product as follows selecting default installation options except where noted below:

Rebooted server to start installation process
Configured the RAID subsystem as described above.
Selected to install tools for software development
Selected to install tools for kernel development
Selected to install Editors
Selected to install Administration Tools
Selected to install System Tools
Selected to install Server Configuration Tools
During installation, configured the network parameters to match the client testbed segments.
Used default NIC driver and settings per recommendations in the Intel driver README
Configured processor affinity to bind the interrupts from the NIC's to specific processors where appropriate. This was only performed on configurations that utilized multiple processors.

We checked the Red Hat Web site for available updates and errata for the Red Hat Linux 8.0 Professional product. We found a number of security related updates, but found no bug fixes or enhancements related to Apache or TUX performance issues. As a result, we applied no additional patches and made no additional modifications to the Red Hat Linux 8.0 Professional distribution used for these tests.

Client Operating System Tunings for both Linux and Windows Server configurations

We made the following registry changes on the testbed client systems running Windows XP Professional when conducting the Web server performance testing:

Set HKLM\System\CurrentControlSet\Services\tcpip\Parameters\MaxHashTableSize to 65535.
Set HKLM\System\CurrentControlSet\Services\tcpip\Parameters\MaxUserPort to 65534.
Set HKLM\System\CurrentControlSet\Services\tcpip\Parameters\TcpWindowSize to 65536.

Appendix C. Web Server Performance Tunings

For the Web server performance testing, we performed a series of operating system and Web server tunings as specified by documentation provided from Microsoft. Additionally, we spent considerable time investigating and testing potential performance tuning options for both the Apache and TUX Web servers.

While investigating tuning options for the Apache and TUX Web servers, we looked at a number of items including previous competitive tests comparing Windows operating systems to Linux, as well as a variety of books and Web sites with information about tuning the performance of Apache and TUX. We gathered what appeared to be the most likely candidates to maximize the performance of the Apache and TUX Web servers and then spent several days running a series of tests designed to determine which, if any of these tuning options actually made a difference in Web server performance.

Our own investigation showed that, with some minor tweaks, the default configuration values set for Apache and TUX generated the best overall performance. In a number of cases, changing default parameters actually decreased the overall performance. Additionally, we found that using the MMapFile directive to map static content into system memory at the time of Web server startup significantly increased the performance of the Apache Web servers when serving only static content.

During our investigation, we found that any of our tests that involved a combination of static and CGI-based dynamic content performed considerably worse if the static content requested during the test had been previously mapped in to memory using the MMapFile directive. As a result, we did not use the MMapFile directive to map the static content into memory prior to starting the testing when running either the CGI or SSL/CGI based test suites.

Additionally, for the dynamic SSL/CGI tests, we configured each Web server to accept SSL-based requests for secure content. This included creating and installing a digital certificate for the Web servers under test. For the Apache servers on both Red Hat Linux Advanced Server 2.1 and Red Hat Linux 8.0 Professional, we used the OpenSSL certificate tools that come with Apache and simply built and used the default test certificates using a 1024-bit key. For IIS 6.0, we used Windows Server 2003 Certificate Services to create a certificate using a 1024-bit key and installed it into the IIS 6.0.

During the SSL handshaking process, the WebBench client and the Web server negotiate a set of parameters to facilitate the passing of encrypted data based on a list of available cipher suites that are common to both the WebBench client and the Web server under test. Generally, the strongest encryption type supported by both client and Web server is chosen. Figure 46 shows these parameters for all SSL based testing.

Operating System	Key Exchange Algorithm	Encryption	Message Digest
Windows Server 2003	RSA	RC4 (128-bit)	MD5
Red Hat Linux Advanced Server 2.1	Diffie-Helman	3DES (168-bit)	SHA1
Red Hat Linux 8.0 Professional	Diffie-Helman	3DES (168-bit)	SHA1

Figure 46. SSL Encryption Information For All Test Configurations

Windows Server 2003 Enterprise Edition RC2

Web server performance testing under Windows Server 2003 Enterprise Edition consisted of making the following registry modifications to the server systems under test:

Set HKLM\System\CurrentControlSet\Services\InetInfo\Parameters\MaxCachedFileSize to 1048576 bytes.
Set HKLM\System\CurrentControlSet\Services\HTTP\Parameters\UriMaxUriBytes to 1048576 bytes.
Set HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccess to 1.
Set HKLM\System\CurrentControlSet\Services\tcpip\Parameters\MaxHashTableSize to 65535.

We made the following changes to the default configuration of Internet Information Server (IIS) 6.0 for use with testing that involved non-secure static and CGI-based content:

Set the CentralBinaryLoggingEnabled option to “TRUE” in the IIS Metabase.
Using the Microsoft Management Console, removed script and execute access from the document root directory that contained only static content.
Using the Microsoft Management Console, disabled the “Index This Resource” property for the main Web server.
Using the Microsoft Management Console, disabled access logging for the web server.

Created a virtual directory called “cgi-bin” to store the WebBench ISAPI and CGI based dynamic content for all tests.
Set the Application Protection property to “Low (IIS Process)” for the “cgi-bin” virtual directory.

We made the following changes to the default configuration of Internet Information Server (IIS) 6.0 for use with all tests that involved secure static and CGI-based content:

Set the CentralBinaryLoggingEnabled option to “TRUE” in the IIS Metabase.
Using the Microsoft Management Console, removed script and execute access from the document root directory that contained only static content.
Using the Microsoft Management Console, disabled the “Index This Resource” property for the main Web server.
Using the Microsoft Management Console, disabled access logging for the web server.
Created a virtual directory called “cgi-bin” to store the WebBench ISAPI and CGI based dynamic content for all tests.
Set the Application Protection property to “Low (IIS Process)” for the “cgi-bin” virtual directory.
Using IIS 6.0, created a certificate request using a 1024-bit key. Submitted this certificate request to a system configured with Windows Server 2003 Certificate Services and generated a digital certificate. Installed the resulting certificate into IIS 6.0 for use with SSL/CGI-based testing.

Red Hat Linux Advanced Server 2.1

Red Hat Linux Advanced Server 2.1 ships with version 1.3.23 of the Apache Web server. As a result of our investigation, we made the following changes to the default Red Hat Linux Advanced Server 2.1 Apache Web server configuration when testing the performance of the Apache Web server using only non-secure static data:

Enabled the mmap_static_module in the Apache configuration file httpd.conf. This allowed us to directly map the static content used by WebBench directly into system memory.
Set MaxRequestsPerChild to zero in the httpd.conf file
Set the document root directory to point to a directory on a fast RAID 0 volume.
Disabled Access Logging
Cut and pasted the URL’s for over 6000 static WebBench files into the Apache configuration file httpd.conf. Used the MMapFile directive to load each of the static content files into system memory when the Web server is started.

We made the following changes to the default Apache Web server configuration when testing CGI and SSL/CGI based test suites:

Enabled the mod_ssl module in the Apache configuration file httpd.conf. This allowed us to use Apache to satisfy SSL based requests.
Set MaxRequestsPerChild to zero in the httpd.conf file
Set the document root directory to point to a directory on a fast RAID 0 volume.
Set the cgi-bin directory to point to a directory on a fast RAID 0 volume.
Disabled Access Logging
Used the OpenSSL tools that ship with Apache to create a self signed digital certificate using a 1024 bit key.

Additionally, our testing involved making requests of the TUX Web server for static content as well as various combinations of static content mixed with CGI and SSL/CGI content as previously described. TUX is equipped to directly handle static content. Any other content such as CGI-based dynamic modules or SSL-based requests made of the TUX Web server is passed along to another Web server for processing.

In our tests we configured the TUX and Apache Web server to work together to process CGI and SSL/CGI based requests made during the testing. This requires configuring the Apache Web server to listen on a TCP port other than the standard port 80 normally reserved for making HTTP connections to a Web server. This also requires TUX to be configured to pass HTTP requests for content it cannot fulfill to the Apache Web server.

TUX comes with two primary configuration files, /etc/sysconfig/tux and /etc/tux.mime.types, that allow users to configure a small number of TUX variables. A larger number of other TUX configuration parameters can be configured by modifying values in the /proc/sys/net/tux file system.

When testing using TUX on the DL760 and DL380 servers, we made the following modifications to the default Apache and TUX configurations for all tests involving Apache and TUX. Additionally, we made the modifications to the file system described below only when testing a combination of both the Apache and TUX Web servers.

Apache Configuration:

Set MaxRequestsPerChild directive to zero in the Apache httpd.conf file
Set the Apache Web server document root directory to point to a directory on a fast RAID 0 volume.
Set the Apache Web server cgi-bin directory to point to a directory on a fast RAID 0 volume.
Disabled Access Logging
Used the OpenSSL tools that ship with Apache to create a self signed digital certificate using a 1024 bit key.
Configured the Apache Web server to listen on TCP port 81

TUX Configuration:

Edited the file /etc/tux.mime.types and removed the comment from the line: #TUX/redirect pl php and modified the line to read TUX/redirect cgi. This indicated that requests for content with a ".cgi" extension should be referred to the Apache Web server.
We bound each of the TUX threads running on the server to a specific IP address by modifying a series of TUX "listen" parameters in the /proc/net/tux/thread#/listen/0 file system to include the IP address and port 80.
Set /proc/sys/net/logging to zero to disable TUX Web server logging.
Set /proc/sys/net/tux/clientport to 81. This configures the TUX Web server to pass unknown requests on to another Web server listening on TCP port 81. In this case, the other Web server was Apache.

File System Modifications for TUX Testing:

Set /proc/sys/net/core/optmem_max to 10,000,000
Set /proc/sys/net/core/rmem_default to 10,000,000
Set /proc/sys/net/core/rmem_max to 10,000,000
Set /proc/sys/net/core/wmem_default to 10,000,000
Set /proc/sys/net/core/wmem_max to 10,000,000
Set /proc/sys/net/core/hot_list_length to 10,000
Set /proc/sys/net/ipv4/max_tw_buckets to 2,000,000
Set /proc/sys/net/ipv4/tcp_rmem to "30,000,000 30,000,000 30,000,000"
Set /proc/sys/net/ipv4/tcp_wmem to "30,000,000 30,000,000 30,000,000"
Set /proc/sys/net/ipv4/tcp_mem to "30,000,000 30,000,000 30,000,000"
Set /proc/sys/net/ipv4/tcp_timestamps to 0

To ensure that the configuration was correct, we used Internet Explorer to request the WebBench CGI-based dynamic content with and without using SSL. This CGI-based dynamic content returns information regarding the Web server that handled the request and the port on which the request was made. In both cases, the Apache Web server handled the request as expected.

Additionally, when conducting tests using TUX that request only non-secure static content, we did not start the Apache Web server. We found that even though TUX was handling all the requests during these tests, there were still a number of Apache processes consuming CPU resources.

Red Hat Linux 8.0 Professional

Red Hat Linux 8.0 Professional ships with version 2.0.31 of the Apache Web server. Unlike Apache 1.3.19, this version of Apache did not come with the capability to use SSL built into the binary distribution. We therefore, installed the source code for Apache 2.0.40 from the Red Hat 8.0 Professional distribution CD's and built a version of Apache 2.0.40 that included the SSL modules that allowed us to make SSL-based requests of the Web server. We then used this version of Apache for all Web server performance testing using Red Hat Linux 8.0 Professional.

We had planned to conduct tests using TUX with Red Hat Linux 8.0 Professional on the DL380 server. However, every test we attempted failed approximately half way through the test resulting in the DL380 server refusing to accept new HTTP connections from the WebBench test clients. As a result, there are no test results for the TUX Web server on the DL380 server in this report.

We made the following changes to the default Red Hat Linux 8.0 Professional Apache Web server configuration when testing the performance of the Apache Web server using only non-secure static data:

- Enabled the mod_file_cache module in the Apache configuration file httpd.conf. This allowed us to directly map the static content used by WebBench directly into system memory.
- Set MaxRequestsPerChild to zero in the httpd.conf file
- Set the document root directory to point to a directory on a fast RAID 0 volume.
- Disabled Access Logging.
- Cut and pasted the URL's for over 6000 static WebBench files into the Apache configuration file httpd.conf. Used the MMapFile directive to load each of the static content files into system memory when the Web server is started.

We made the following changes to the default Apache Web server configuration when testing CGI and SSL/CGI based test suites:

- Enabled the mod_ssl module in the Apache configuration file httpd.conf. This allowed us to use Apache to satisfy SSL based requests.
- Set MaxRequestsPerChild to zero in the httpd.conf file
- Set the document root directory to point to a directory on a fast RAID 0 volume.
- Set the cgi-bin directory to point to a directory on a fast RAID 0 volume.
- Disabled Access Logging
- Used the OpenSSL tools that ship with Apache to create a self signed digital certificate using a 1024 bit key.

Appendix D. Test Network Diagrams

Figures 47 - 49 below show the testbed configurations for testing the servers described above for all processor configurations.

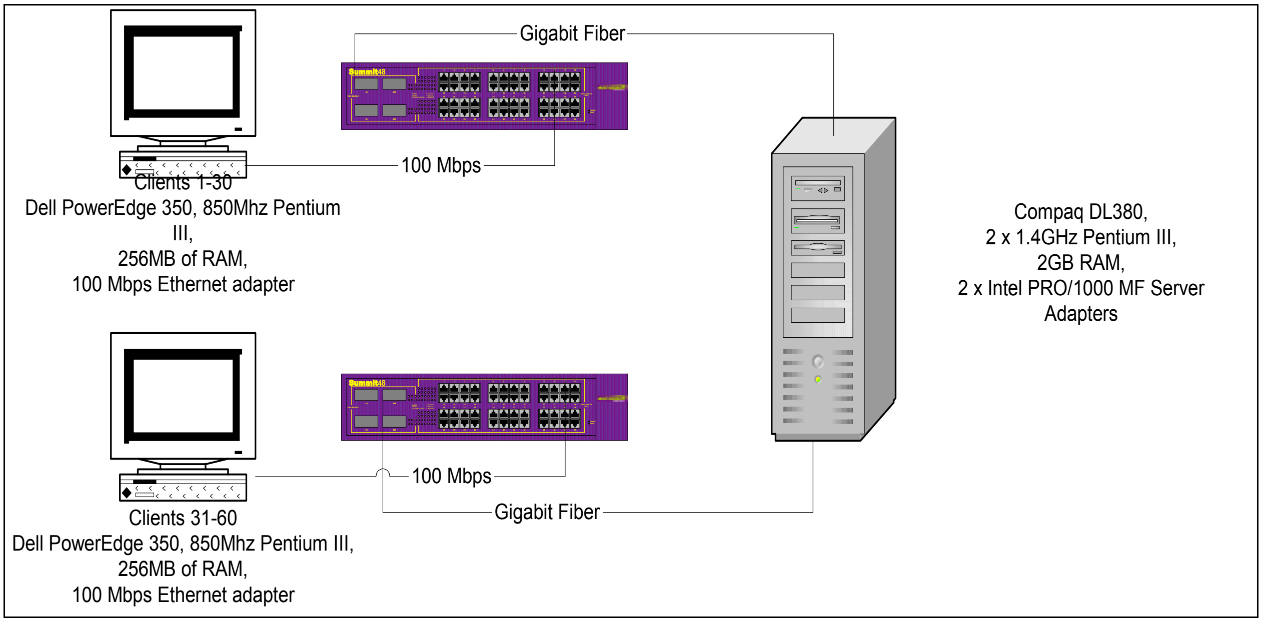

Figure 47. DL380 Test Configuration

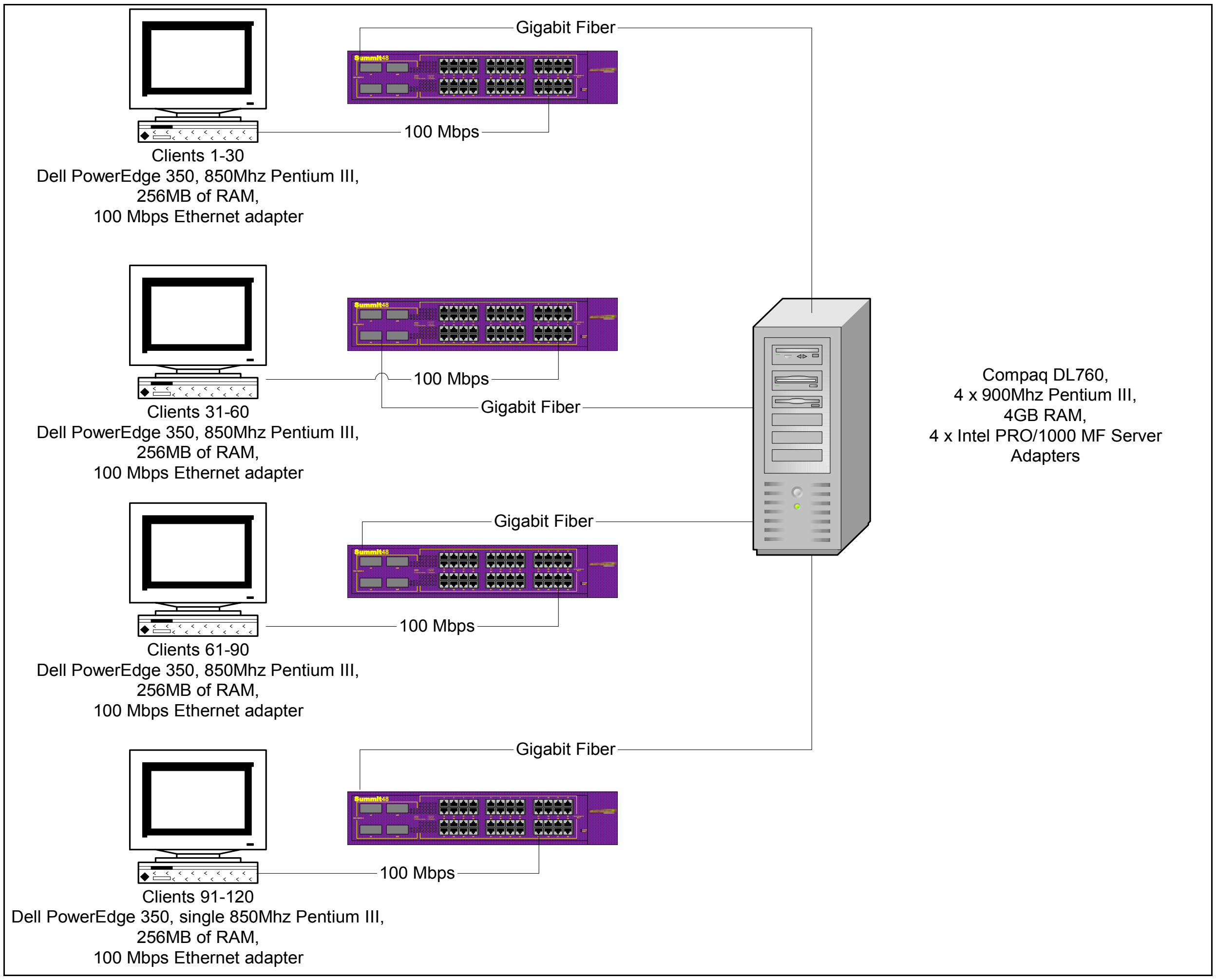

Figure 48. DL760 Test Configuration using 1, 2 and 4 Processors

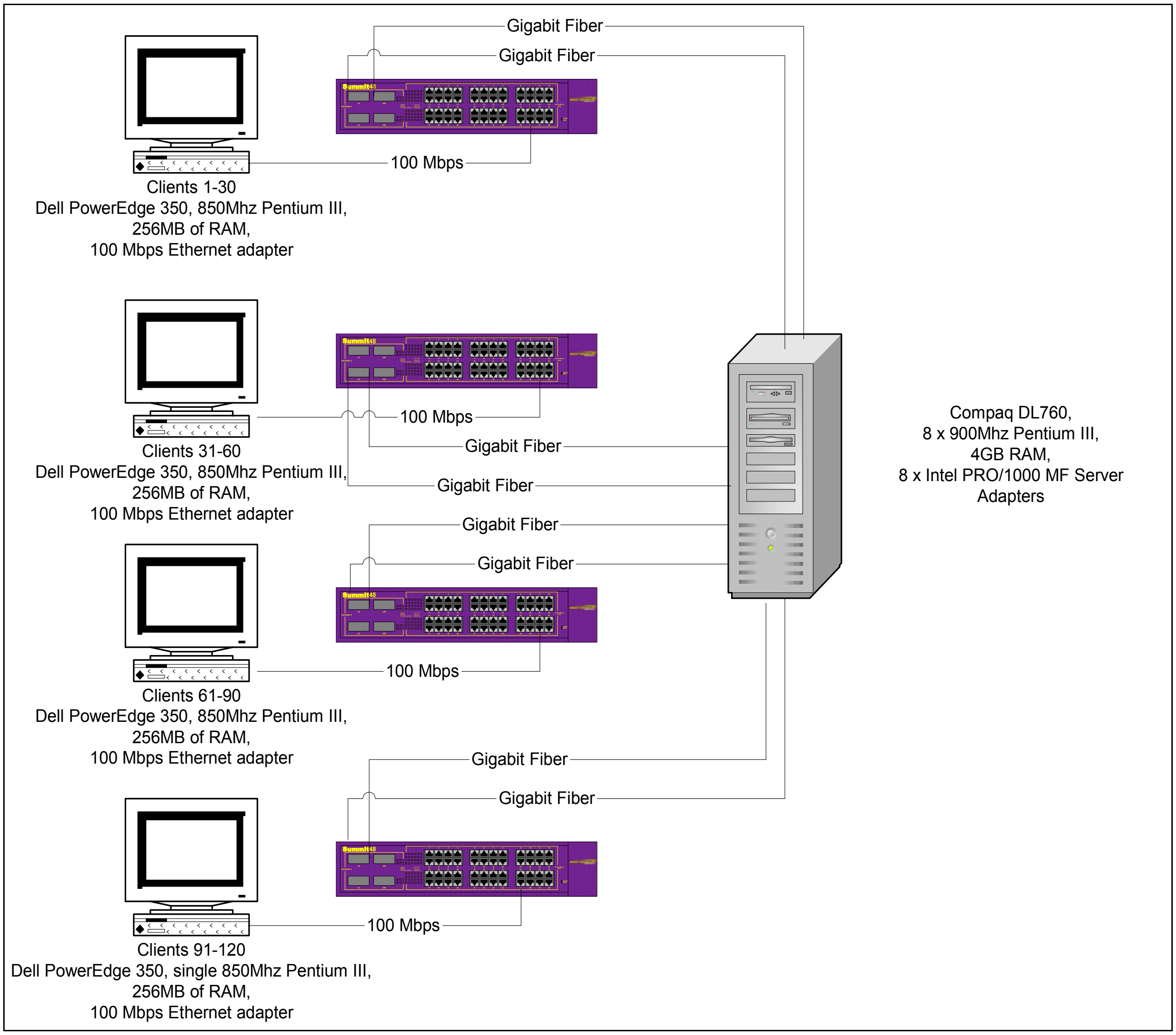

Figure 49. DL760 Test Configuration using 8 Processors

VeriTest (www.veritest.com), the testing division of Lionbridge Technologies, Inc., provides outsourced testing solutions that maximize revenue and reduce costs for our clients. For companies who use high-tech products as well as those who produce them, smoothly functioning technology is essential to business success. VeriTest helps our clients identify and correct technology problems in their products and in their line of business applications by providing the widest range of testing services available.

VeriTest created the suite of industry-standard benchmark software that includes WebBench, NetBench, Winstone, and WinBench. We've distributed over 20 million copies of these tools, which are in use at every one of the 2001 Fortune 100 companies. Our Internet BenchMark service provides the definitive ratings for Internet Service Providers in the US, Canada, and the UK.

Under our former names of ZD Labs and eTesting Labs, and as part of VeriTest since July of 2002, we have delivered rigorous, objective, independent testing and analysis for over a decade. With the most knowledgeable staff in the business, testing facilities around the world, and almost 1,600 dedicated network PCs, VeriTest offers our clients the expertise and equipment necessary to meet all their testing needs.

For more information email us at info@veritest.com or call us at 919-380-2800.