Using VMware ESX Server with IBM WebSphere Application Server

Authors: High Performance On Demand Solutions and VMware
Web address: ibm.com/websphere/developer/zones/hvws
Technical Contact: Marty Yarnall yarnall@us.ibm.com
Lance Berc lance@vmware.com
Qing Bo Wang wangqbo@cn.ibm.com

Management Contact: Noshir Wadia noshir@us.ibm.com
Kai Young kaiyoung@us.ibm.com

Date: July 24, 2006
Status: Version 1.0

Abstract: This paper looks at the best practices when running VMware ESX Server and IBM WebSphere Application Server on a large symmetric multiprocessor (SMP) server to achieve the best application performance (throughput and response time). Our tests concluded that the use of VMware software is an excellent option for those wanting to run WebSphere Application Server in a virtualized SMP environment.

Note: Before using this information, read the information in "Notices" on the last page.

Contents

Introduction

IBM's High Performance On Demand Solutions (HiPODS) team joined VMware®, Inc. in early 2006 to continue a 2004 study[1] on applying VMware's virtualization technology, ESX Server, to the IBM® WebSphere® Application Server. The 2004 paper reported that ESX Server is sound technology that has significant potential in an on demand environment and is an effective tool for production-level virtualization.

This paper looks at a server consolidation, taking under utilized hardware running disparate Web-based applications and combining them on to one physical server while still maintaining the best application performance (throughput and response time). We attempt to answer these questions:

- What are the performance expectations when consolidating multiple Web-based applications in a production ESX Server environment?
- Are VMware Virtual Symmetric Multi-Processing (SMP) guest machines better than uniprocessor guest machines for Web-based applications?
- How many guest machines can be run on a physical server and still get good performance?

IBM and VMware are working together to provide greater benefits to end users by promoting greater interoperability between the solutions that each bring to the marketplace. This paper presents the best practices we derived from our tests. It should provide a base for others to start building their own solution to meet their unique needs. The information and conclusions are based on current knowledge and can change as new technologies, products, solutions, and experiences become available.

SMP, Linux®, and WebSphere Application Server

As large SMP configurations running Linux become increasingly popular, customers are looking to consolidate multiple applications onto a single server to increase server utilization and reduce the number of physical servers they have to manage.

For WebSphere Application Server configurations, there are several different approaches to server consolidation available including running multiple independent WebSphere Application Server instances on a single server, or running multiple application servers with WebSphere Application Server Network Deployment providing central administration, load balancing and failover across the servers. WebSphere Application Server Extended Deployment can also be leveraged to provide even greater performance optimizations

For this set of experiments, we compared a baseline running multiple WebSphere Application Servers on a single server to running each application server on a VMware virtual machine (VM) in a single server. This allowed for the simplest comparison configurations. Each server was separately administered, and each application executed independently, with no load balancing performed across the applications. A shared, non-virtualized, database was used by all applications.

For the baseline measurements in this paper, we chose to run multiple applications using separate application server instances. Running multiple application servers works well when a single server does not use the maximum performance of the server, that is, it cannot maximally utilize the CPU, memory, and other resources.

Another approach is to use virtualization to run multiple guests machines, each with a single instance of the application server. This approach was used for the VMware measurements in this paper. This approach provides greater isolation between the applications. For performance comparison purposes, these experiments ran WebSphere Application Server instances in each guest machine, however, virtualization also provides an opportunity to consolidate different types of applications.

Overall testing

To provide the most meaningful performance comparisons, independent copies of the IBM® Trade Performance Benchmark Sample for WebSphere Application Server (otherwise known as Trade6) were run in each application server. The client workload was simulated using the Web Performance Tool's (WPT) stress application. Results were determined by measuring the transaction throughput of the Trade6 running on the baseline system versus running in an ESX Server virtual machine environment. The optimal transaction throughput was created by using the number of clients that resulted in consistent response times with a CPU utilization of ~90% across the linear portion (no CPU over-commit) of the measurement graph.

Using the Trade6 and stress applications, we created a three-tier architecture that could easily be used for testing baseline and VMware client machines:

- Tier 1 was the WPT stress application, which ran on its own separate server and served both the baseline and virtual machine client machines

- Tier 2 was WebSphere Application Server and Trade6. This tier was installed on the baseline server and each virtual machine client machine to create a *system under test* (SUT).

  - For the baseline SUT machine, this tier was duplicated vertically creating multiple application servers or JVMs on a single physical server simulating the consolidation of applications. Figure 1 shows the three-tier testing architecture as seen in the baseline configuration

  - On the VMware SUT machine, each virtual machine contained one Tier 2 resulting in a scale-out effect on the single physical server. Figure 2 shows the same three-tiers as seen on the VMware virtual machine.

- Tier 3 was the standalone DB2® database used by all Trade6 applications. The database tier was not virtualized.

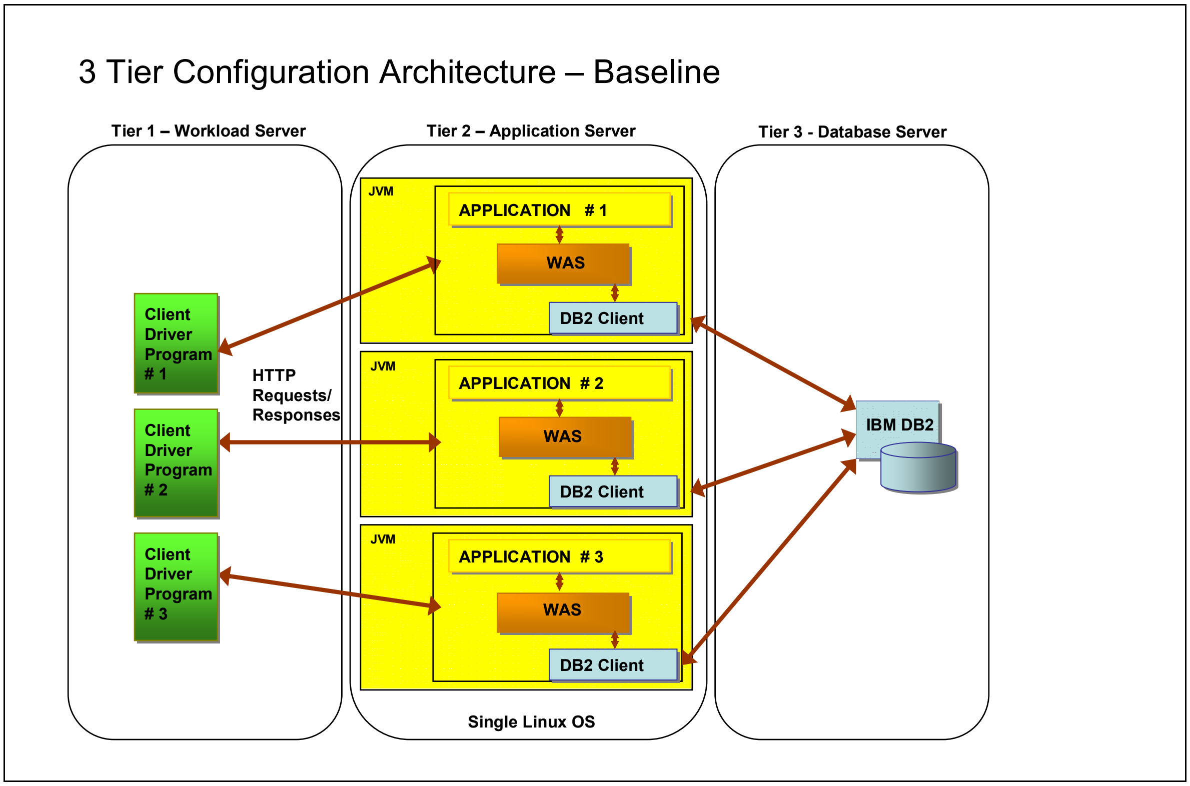

Figure 1. Three-tier testing configuration used on the baseline SUT

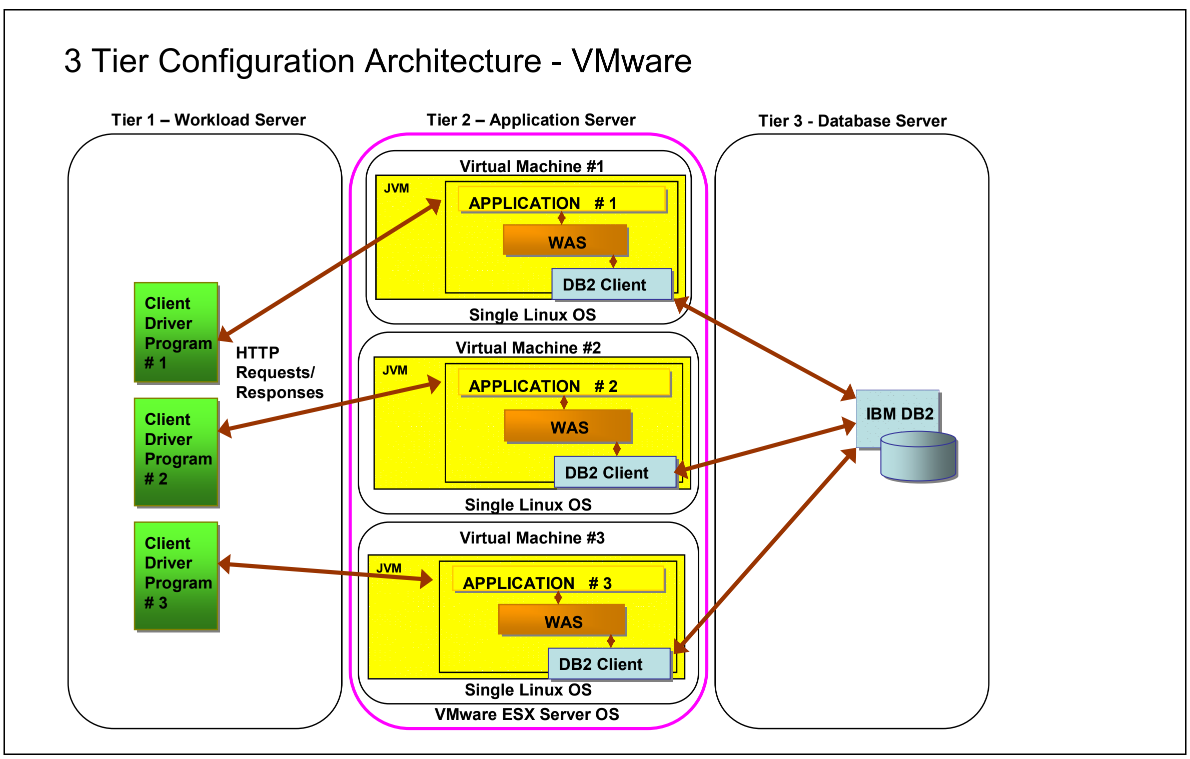

Figure 2. Three-tier testing configuration used on the VMware SUT

Testing strategy

The testing goal was to achieve the greatest throughput measured in ‘pages returned per second’ while keeping the average response time constant. The testing environment was monitored to ensure that the only system bottleneck was the SUT and that no other bottlenecks were encountered in the network, client driver server, or DB2 server.

Two sets of tests were conducted.

- 8-way SUT: The first set measured the baseline and a VMware environment using all eight CPUs and 16 GB of memory on the SMP server, which this paper calls an *8-way*.

- 4-way SUT: The second set measured SMP server using four CPUs and 8 GB of memory, referred to as a *4-way*. The 4-way was accomplished by removing the cable that interconnects the SMP expansion modules on the 8-way server; this cut in half the available number of CPUs and memory.

VMware ESX Server offers two options in virtual guest machines when using an SMP server: uniprocessor (one CPU) or SMP (two CPUs). Both types were measured. To maintain comparability between the baseline and VMware measurement, we attempted to keep client response times the same. In the simulated Web clients, this was accomplished by setting total virtual clients up to 32, with each client using four threads.

We installed the ESX Server and WebSphere software “out of the box” with no special tuning parameters. Our intent was to measure performance under typical user conditions to help guide expectations and capacity planning under normal field conditions.

No unsuccessful connections, including failed connections, early server closes, request write failures, or timeouts occurred during the measurement runs.

Test environment

VMware ESX Server (SUT)

Machine: IBM eServer® xSeries® 445 8870-C6X
CPU: (8) Intel® Xeon® MP 3.0 GHz
Memory: 16 GB
Swap space: 1 GB
Virtualization layer: VMware ESX v2.5.3

Guest operating system (SUT – replicated on the ESX Server):
Guest operating system: Red Hat Linux AS 3.4
Web application server: IBM WebSphere Application Server v6.02
Web test application: Trade 6

Baseline server (SUT)

Machine: IBM eServer xSeries 445 8870-C6X
CPU: (8) Intel Xeon MP 3.0 GHz
Memory: 16 GB
Swap space: 1 GB
Operating system: Red Hat Linux AS 3.4
Web application server: IBM WebSphere Application Server v6.02
Web test application: Trade 6

Workload driver server

Machine: IBM eServer xSeries 346 8840-11U
CPU: (2) Intel Xeon 3.0 GHz
Memory: 16 GB
Operating system: Red Hat Linux AS 3.4
Database: DB2 v8.0
Test driver: Web Performance Tools (WPT) 1.9

Database server

Machine: IBM eServer xSeries 440 8687-3RX
CPU: (4) Intel Xeon 1.6 GHz
Memory: 4 GB
Operating system: Microsoft® Windows® 2003 Server
Database: DB2 v8.0

Testing expectations

Our testing objective was to measure the throughput of WebSphere Application Server running in a virtualized VMware environment verses running the same applications on the native hardware. Because each VMware virtual machine uses only a fraction of the physical resources (in this case CPU) we expect each virtual machine to produce a fraction of the throughput of the physical system; in a perfect world each uniprocessor virtual machine would produce ¼ the physical throughput on a 4-way system and $1/8^{th}$ on an 8-way machine. Anything less is virtualization overhead and anything more shows lack of native scaling.

Also, for the over-commit scenarios (where the number of processors defined to virtual machine guest machines is greater than the number of physical processors) in a perfect world, 100% of all CPUs would be used for WebSphere Application Server processing resulting in a constant throughput with linearly increasing response time. Any reduction in throughput or jumps in response times are the result of ESX Server scheduling overhead when the system is saturated and helps form a planning expectation when peak loads occur.

Testing results

The charts in this section show Trade6 throughput (pages returned per second) as a *bar* and uses the left-axis labels. Client response times are shown as a *line* and use the right-axis labels.

Baseline results

Figure 3 shows Trade6 throughput and client response times when running WebSphere Application Server in a baseline (native - no virtualization) environment.

Both the 4-way (dark blue on chart) and 8-way (light blue on chart) performance was best when running with three application servers (x-axis). Both a decrease in throughput and an increase in response time occurred when adding more application servers showing that we had saturated the server's performance.

The 4-way throughput was approximately 240 pages per second. The 8-way throughput was approximately 365 pages per second. Comparing these gives this configuration of WebSphere Application Server a 4-way to 8-way scaling factor of 1.5.

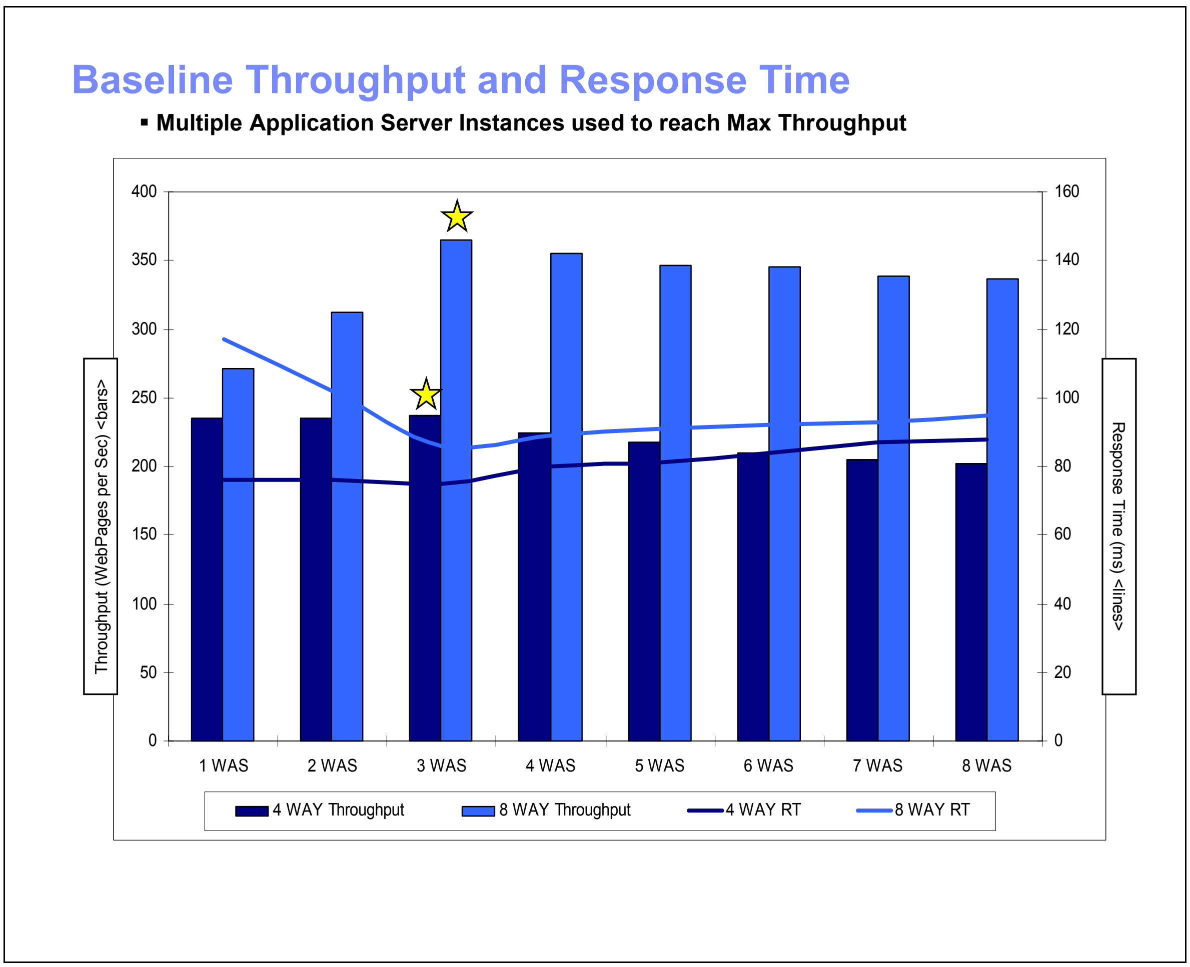

Figure 3: Throughput and response time results of baseline environment

VMware results: 4-way

Figure 4 shows Trade6 throughput and response time when using a 4-way server running WebSphere Application Server in a VMware environment. Uniprocessor guest machines are shown in orange and SMP guest machines are shown in maroon.

Each uniprocessor guest machine had one WebSphere Application Server. The uniprocessor guest machines best performance was running four guest machines (a total of four CPUs as shown on the x-axis). Total throughput from the four application servers was approximately 200 pages per second. Each SMP had one WebSphere Application Server. The SMP guest machines best performance was running two guest machines (a total of four CPUs). Each virtual machine produced slightly more than fifty page views and the total throughput grew linearly until all CPUs were in use. Total throughput from the two application servers was approximately 180 pages per second.

Overcommitting physical CPUs is a common and accepted practice when virtualizing servers. The advantage of over-committing is to slice the application's performance into smaller pieces, therefore providing a good performance-on-demand. This is also a common practice for application service providers and other hosted environments. Overcommitting does not give optimal performance for a WebSphere Application Server production server with CPU rates greater than 90%. As expected, running guest machines with total CPUs greater than the available physical CPUs lowers throughput and increases response times. However, the throughput decrease was relatively slight showing that the ESX Server resource scheduler does a good job allocating resources when the system is saturated.

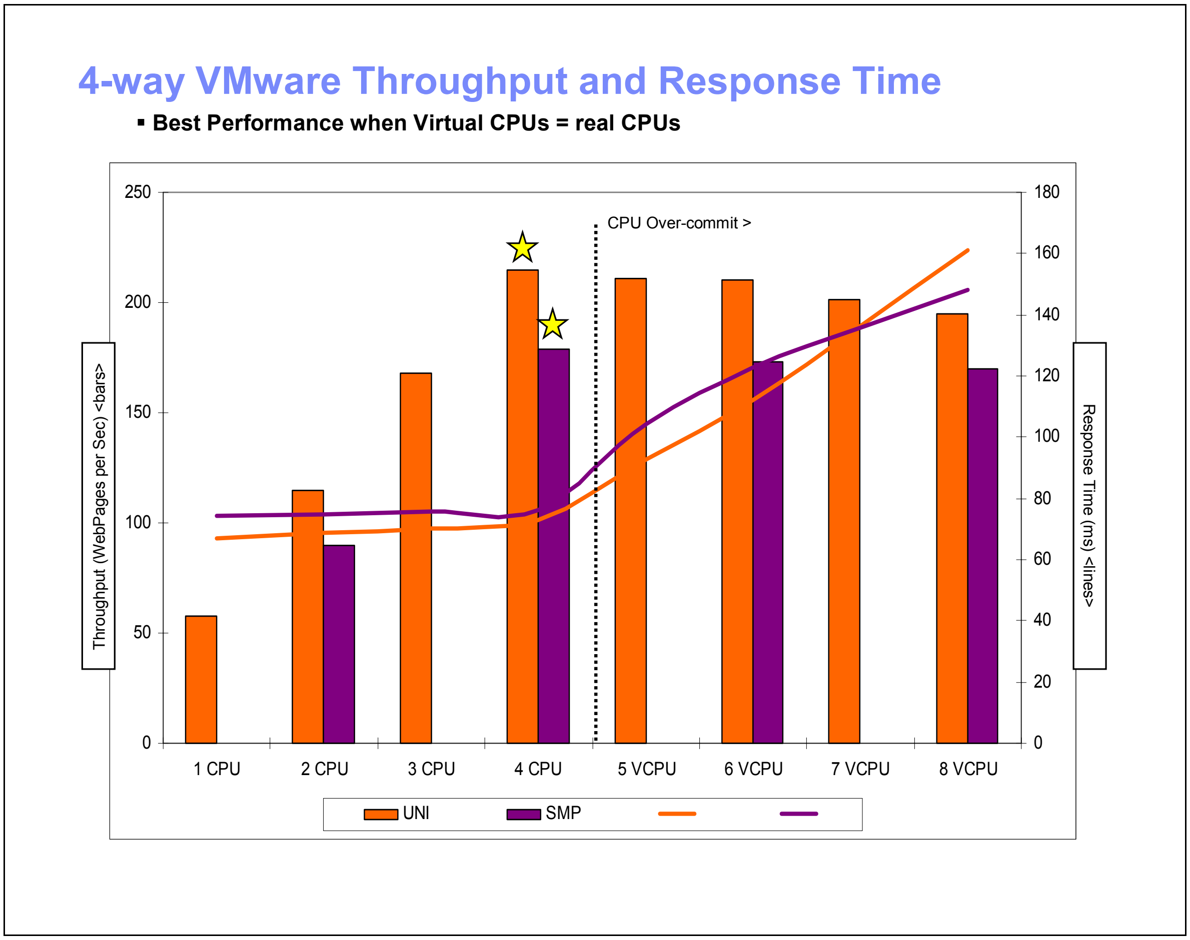

Figure 4: Throughput and response time results for VMware ESX Server on 4-way

VMware ESX Server results: 8-way

Figure 5 shows Trade6 throughput and response time when using an 8-way server running WebSphere Application Server in a VMware environment.

The best performance for uniprocessor was running eight guest machines (for a total of eight CPUs). Total throughput from the eight application servers was approximately 370 pages per second. This is a scaling factor of 1.9 compared to the 4-way uniprocessor.

The SMP guest machines best performance was running four guest machines (for a total of eight CPUs). Total throughput from the four application servers was approximately 320 pages per second. This is a scaling factor of 1.8 compared to the 4-way SMP.

As expected, virtual machines produced linear throughput growth until all CPUs were in use, and then had slight loss due to scheduling overhead as additional virtual machines were added and the CPUs were overcommitted.

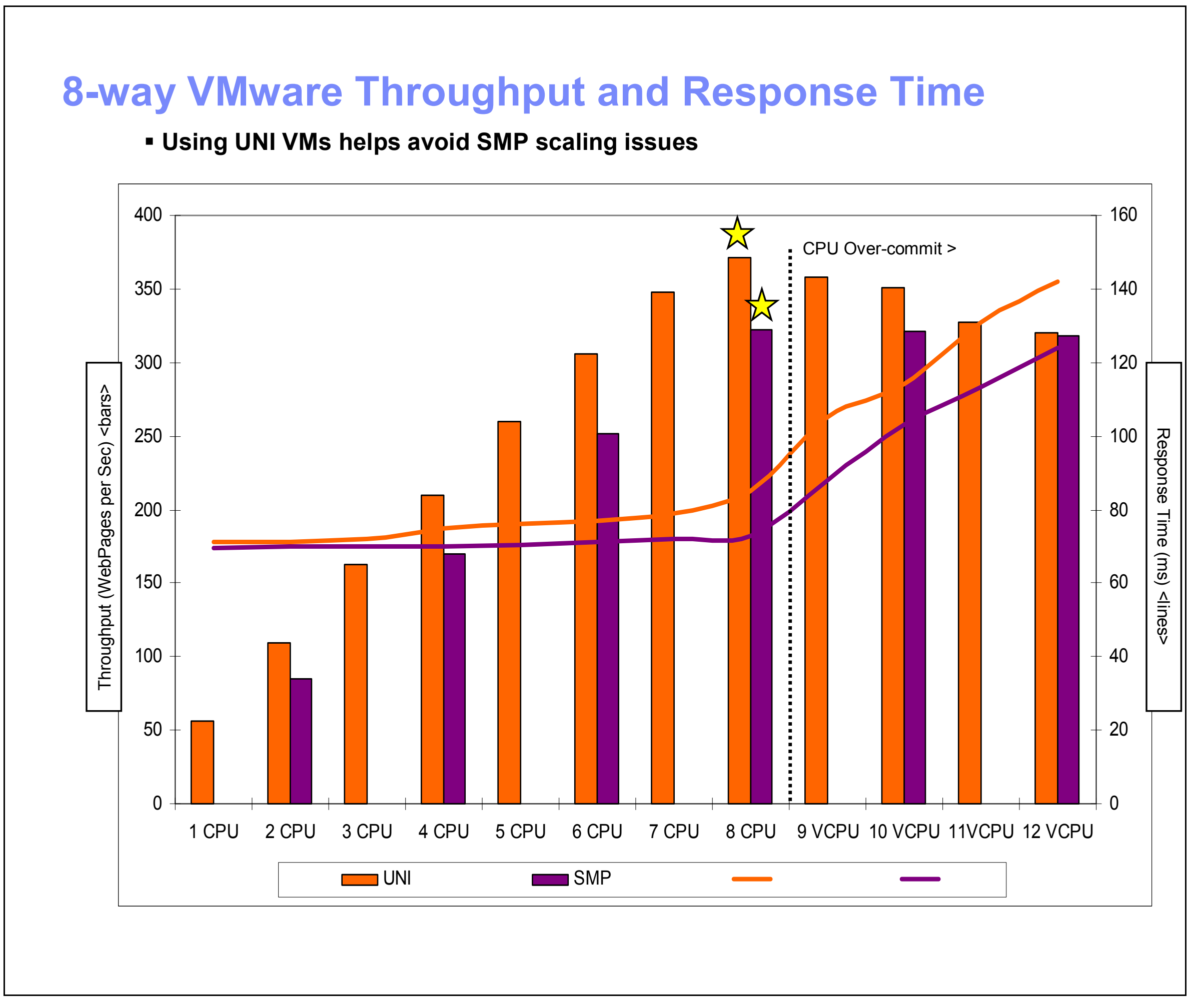

Figure 5: Throughput and response time results for VMware ESX Server on 8-way

Summary of results

Figure 6 shows the best results for the baseline configuration, VMware uniprocessor and VMware SMP guest machines on the 4-way and 8-way. On the 4-way, VMware shows expected throughputs based on some overhead for virtualization. On the 8-way, VMware shows performance comparable to the baseline.

When compared to the baseline, four uniprocessor virtual machines had ~10% overhead on a 4-way, and eight uniprocessor virtual machines had virtually no overhead on an 8-way system. VMware SMP virtual machines had more total overhead, but a single VMware SMP virtual machine is able to drive more throughput than a single uniprocessor virtual machine.

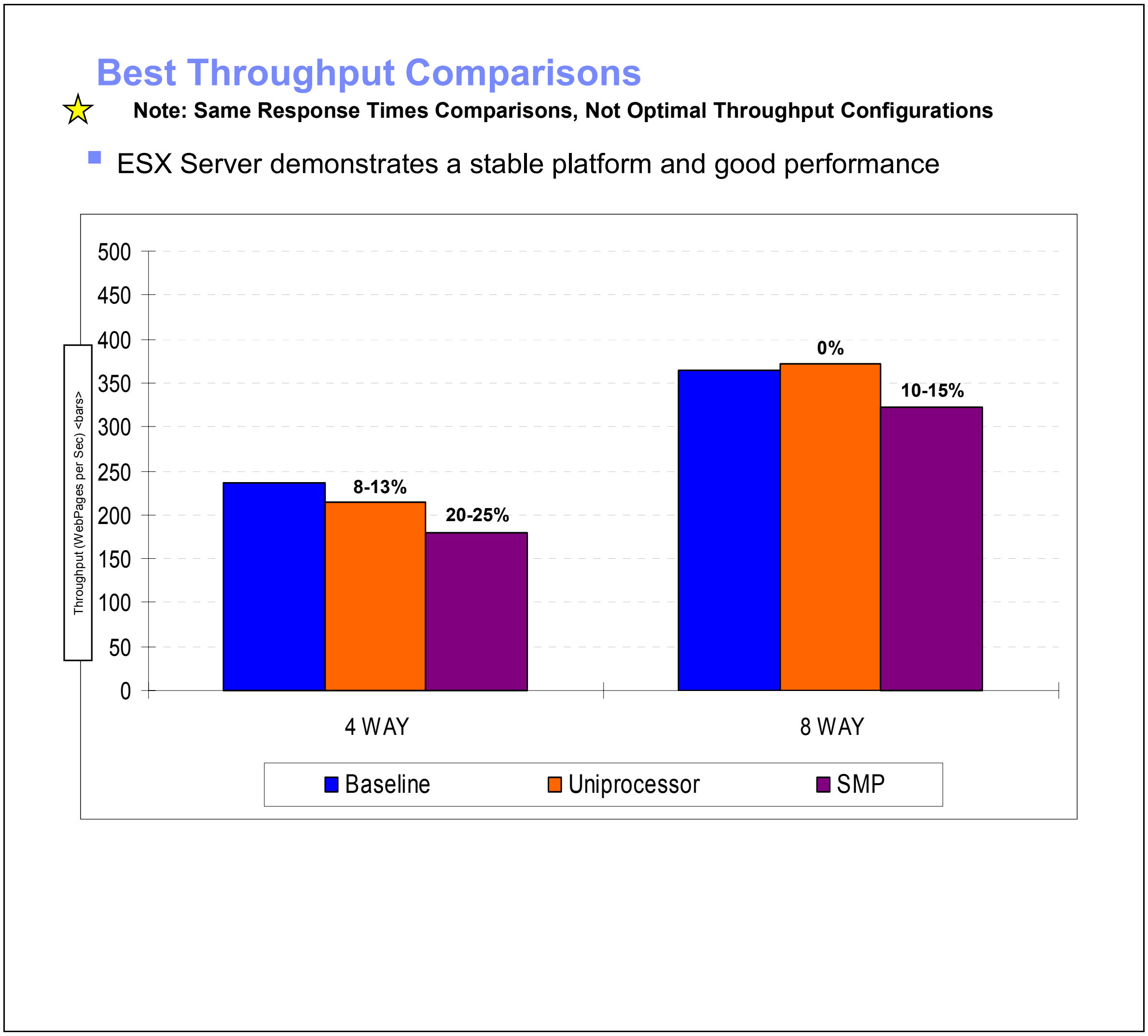

Figure 6: Best throughput and response time comparisons

Conclusion

Based on the results, we can conclude:

- VMware virtualization is an excellent choice for Web-based application consolidation.
- Keeping the number of virtual machines equal to or less than the number of physical CPUs in the server produces the best performance: good throughput plus good response times.
- On large SMP servers (an 8-way in this case), an interesting option is to run WebSphere applications in uniprocessor virtual guests. This allows for very good server utilization, good application performance, and a very flexible VMware ESX Server configuration.

Appendixes

Complete chart data

Baseline								
4-way	1 WAS	2 WAS	3 WAS	4 WAS	5 WAS	6 WAS	7 WAS	8 WAS
Throughput	235	235	237	224	218	210	205	202
Response Time	76	76	75	80	81	84	87	88
WPT Threads/Clients	4/18	4/18	4/18	4/18	4/18	4/18	4/18	4/18
8-way	1 WAS	2 WAS	3 WAS	4 WAS	5 WAS	6 WAS	7 WAS	8 WAS
Throughput	271	312	365	355	346	345	339	337
Response Time	117	102	86	89	91	92	93	95
WPT Threads/Clients	4/32	4/32	4/32	4/32	4/32	4/32	4/32	4/32

VMware Uni												
4-way	1 CPU	2 CPU	3 CPU	4 CPU	5 CPU	6 CPU	7 CPU	8 CPU				
Throughput	57	108	155	201	200	189	193	193				
Response Time	71	76	77	78	98	125	143	164				
WPT Threads/Clients	4/4	4/8	4/12	4/16	4/20	4/24	4/28	4/32				
8-way	1 CPU	2 CPU	3 CPU	4 CPU	5 CPU	6 CPU	7 CPU	8 CPU	9 CPU	10 CPU	11 CPU	12 CPU
Throughput	56	109	163	210	260	306	348	371	358	351	327	320
Response Time	71	71	72	75	76	77	79	85	105	114	131	142
WPT Threads/Clients	4/4	4/8	4/12	4/16	4/20	4/24	4/28	4/32	4/36	4/40	4/44	4/48

VMware SMP												
4-way	1 CPU	2 CPU	3 CPU	4 CPU	5 CPU	6 CPU	7 CPU	8 CPU				
Throughput		90		179		173		170				
Response Time		75		76		123		148				
WPT Threads/Clients		4/6		4/12		4/18		4/24				
8-way	1 CPU	2 CPU	3 CPU	4 CPU	5 CPU	6 CPU	7 CPU	8 CPU	9 CPU	10 CPU	11 CPU	12 CPU
Throughput		85		170		252		320		321		318
Response Time		70		70		71		73		103		124
WPT Threads/Clients		4/6		4/12		4/18		4/24		4/30		4/36

Tool information

IBM Trade Performance Benchmark Sample for WebSphere Application Server (otherwise known as Trade 6) is the fourth generation of the WebSphere end-to-end benchmark and performance sample application. This package provides a suite of IBM-developed workloads for characterizing performance of Java™ 2 Platform, Enterprise Edition (J2EE™) application servers. The Trade application models an electronic stock brokerage providing Web and Web services-based online securities trading. Trade provides a real-world, business application mix of transactional enterprise beans, message-driven beans, servlets, JavaServer Pages™ (JSP™) files, Java database connectivity (JDBC™) and Java Message Service (JMS) data access, adjustable to emulate various work environments.

The Web Performance Tools (WPT) is a set of applications allowing a user to test a Web server, a Web site, and/or a Web application. For this testing, we used the 'stress' application, which is a high-performance, simple, threaded HTTP engine capable of simulating hundreds or even thousands of HTTP clients, using a highly configurable set of directives in a human readable and easily modified configuration file. The Web Performance Tools has been retired because it overlaps with the functionality provided by the IBM® Rational® Suite TestStudio

About VMware, Inc.

VMware was founded in 1998 to bring mainframe-class virtual machine technology to industry-standard computers. VMware delivered its first product, VMware Workstation, in 1999 and entered the server market in 2001 with VMware GSX Server and VMware ESX Server. With the launch of VMware VirtualCenter in 2003, and the groundbreaking VMotion technology, the company established itself as the thought leader in the newly emerging virtual infrastructure marketplace.

Many large companies use VMware solutions to simplify their IT, fully leverage their existing computing investments and respond faster to changing business demands. VMware is based in Palo Alto, California. For more information about VMware, see: www.vmware.com

References

1. HiPODS, *VMware ESX Server 2: Performance and Scalability Evaluation*, www.ibm.com/developerworks/websphere/zones/hipods/library.html, (January 2004).

2. V.K.Anand, and W.C.Jamison, *A middle performance characterization of Linux using IBM WebSphere Application Server*, IBM Systems Journal 44, No. 2, 353–367 (April 2005).

Acknowledgements

The major contributors to this document are Lance Berc and Axel Streichardt from VMware, and Marty Yarnall, Qing Bo Wang, and Harold Hall from IBM.

We would also like to thank Kristen Edwards from VMware, and Kai Young, Noshir Wadia, and Ruth Willenborg from IBM.

Notices

Trademarks

The following are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

DB2
eServer
IBM
Rational
WebSphere
xSeries

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Intel and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

Special Notice

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While IBM may have reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document were determined in various controlled laboratory environments and are for reference purposes only. Customers should not adapt these performance numbers to their own environments as system performance standards. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment