WHITE PAPER

Performance and Scalability of Microsoft® SQL Server® on VMware vSphere™ 4

vmware®

Table of Contents

Introduction

VMware vSphere™ 4 contains numerous performance related enhancements that make it easy to virtualize a resource heavy database with minimal impact to performance. The improved resource management capabilities in vSphere facilitate more effective consolidation of multiple SQL Server virtual machines (VMs) on a single host without compromising performance or scalability. Greater consolidation can significantly reduce the cost of physical infrastructure and of licensing SQL Server, even in smaller-scale environments.

This paper describes a detailed performance analysis of Microsoft SQL Server 2008 running on vSphere. The performance test places a significant load on the CPU, memory, storage, and network subsystems. The results demonstrate efficient and highly scalable performance for an enterprise database workload running on a virtual platform.

To demonstrate the performance and scalability of the vSphere platform, the test:

- Measures performance of SQL Server 2008 in an 8 virtual CPU (vCPU), 58GB virtual machine using a high end OLTP workload derived from TPC-E[1].
- Scales the workload, database, and virtual machine resources from 1 vCPU to 8 vCPUs (scale up tests).
- Consolidates multiple 2 vCPU virtual machines from 1 to 8 virtual machines, effectively overcommitting the physical CPUs (scale out tests).
- Quantifies the performance gains from some of the key new features in vSphere.

The following metrics are used to quantify performance:

- Single virtual machine OLTP throughput relative to native (physical machine) performance in the same configuration.
- Aggregate throughput in a consolidation environment.

Highlights

The results show vSphere can virtualize large SQL Server deployments with performance comparable to that on a physical environment. The experiments show that:

- SQL Server running in a 2 vCPU virtual machine performed at 92 percent of a physical system booted with 2 CPUs.
- An 8 vCPU virtual machine achieved 86 percent of physical machine performance.

The statistics in Table 1 demonstrate the resource intensive nature of the workload.

Table 1. Comparison of workload profiles for physical machine and virtual machine in 8 CPU configuration

Metric	Physical Machine	Virtual Machine
Throughput in transactions per second*	3557	3060
Average response time of all transactions**	234 ms	255 ms
Disk I/O throughput (IOPS)	29 K	25.5 K
Disk I/O latencies	9 ms	8 ms
Network packet rate receive Network packet rate send	10 K/sec 16 K/sec	8.5 K/sec 8 K/sec
Network bandwidth receive Network bandwidth send	11.8 Mb/sec 123 Mb/sec	10 Mb/sec 105 Mb/sec

[1] See disclaimer on page 13.

*Workload consists of a mix of 10 transactions. Metric reported is the aggregate of all transactions.

**Average of the response times of all 10 transactions in the workload.

In the consolidation experiments, the workload was run on multiple 2 vCPU SQL Server virtual machines:

- Aggregate throughput is shown to scale linearly until physical CPUs are fully utilized.
- When physical CPUs are overcommitted, vSphere evenly distributes resources to virtual machines, which ensures predictable performance under heavy load.

Performance Test Environment

In this section, the workload's characteristics, hardware and software configurations, and the testing methodology are described. The same components were used for both the physical machine and virtual machine experiments.

Workload Description

The workload used in these experiments is modeled on the TPC-E benchmark. This workload will be referred to as the Brokerage workload. The Brokerage workload is a non-comparable implementation of the TPC-E business model[2].

- The TPC-E benchmark uses a database to model a brokerage firm with customers who generate transactions related to trades, account inquiries, and market research. The brokerage firm in turn interacts with financial markets to execute orders on behalf of the customers and updates relevant account information.
- The benchmark is "scalable," meaning that the number of customers defined for the brokerage firm can be varied to represent the workloads of different-sized businesses. The benchmark defines the required mix of transactions the benchmark must maintain. The TPC-E metric is given in transactions per second (tps). It specifically refers to the number of Trade-Result transactions the server can sustain over a period of time.

The Brokerage workload consists of ten transactions that have a defined ratio of execution. Of these transaction types, four update the database, and the others are read-only. The I/O load is quite heavy and consists of small access sizes. The disk I/O accesses consist of random reads and writes with a read- to-write ratio of 7:1.

The workload is sized by number of customers defined for the brokerage firm. To put the storage requirements in context, at the 185,000-customer scale in the 8 CPU experiments, approximately 1.5 terabytes of storage were used for the database.

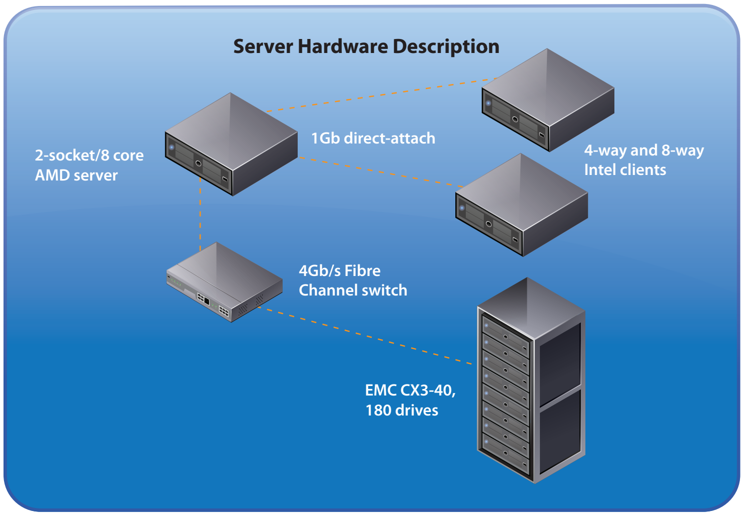
In terms of the impact on the system, this workload spends considerable execution time in the operating system kernel context which incurs more virtualization overhead than user-mode code. This workload also was observed to have a large cache resident working set and is very sensitive to the hardware translation lookaside buffer (TLB). Efficient virtualization of kernel level activity within the guest operating system code and intelligent scheduling decisions are critical to performance with this workload.

[2] See disclaimer on page 13.

Hardware Configuration

The experiments were run in the VMware performance laboratory. The test bed consisted of a server running the database software (in both native and virtual environments), a storage backend with sufficient spindles to support the storage bandwidth requirements, and a client machine running the benchmark driver. Figure 1 is a schematic of the test bed configuration:

Figure 1. Test bed configuration


Server Hardware Description
2-socket/8 core AMD server
1Gb direct-attach
4-way and 8-way Intel clients
4Gb/s Fibre Channel switch
EMC CX3-40, 180 drives

Server Hardware Description

Table 2. Server hardware

Dell Power Edge 2970
Dual Socket Quad-Core Server
2.69GHz AMD Opteron Processor 2384
64GB memory

Storage Hardware and Configuration

Table 3. Storage hardware

EMC CLARiiON CX3-40 array
180 15K RPM disk drives
LUN configuration for 8 vCPU virtual machine : 32 data + 1 log LUN in both Native and ESX
LUN configuration for multiple virtual machines : 4 data + 1 log LUN in each VM

Client Hardware

The single-virtual machine tests were run with a single client. An additional client was used for the multi-virtual machine tests because a single client benchmark driver could drive only four virtual machines. Table 4 gives the hardware configuration of the two client systems:

Table 4. Client hardware

Dell Power Edge 1950	HP Proliant DL580 G5
Single-socket, quad-core server	Dual-socket, quad-core server
2.66GHz Intel Xeon X5355 processor	2.9GHz Intel Xeon X7350 processor
16.0GB memory	64GB memory

Software Configuration

Software Versions

The operating system software, Microsoft SQL Server 2008, and benchmark driver programs were identical in both native and virtual environments.

Table 5. Software Versions

Software	Version
VMware ESX 4.0	RTM build #164009
Operating system (guest and native)	Microsoft® Windows Server® 2008 Enterprise (Build 60001: Service Pack 1)
Database management software	Microsoft® SQL Server® 2008 Enterprise Edition 10.0.1600.22

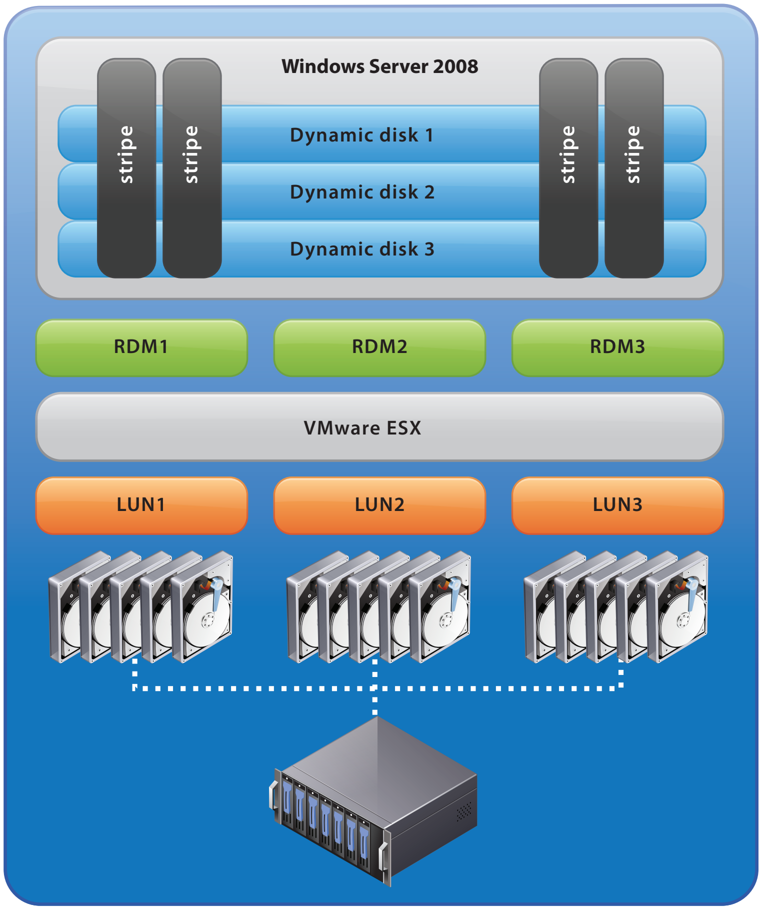
Storage Layout

Measures were taken to ensure there were no hot spots in the storage layout and that the database was uniformly laid out over all available spindles. Multiple, five-disk wide, RAID-0 LUNs were created on the EMC CLARiiON CX3-40. Within Windows, these LUNs were converted to dynamic disks with striped "volumes" created across them. Each of these volumes was used as a datafile for the database.

The same database was used for experiments comparing native and virtual environments to ensure that the configuration of the storage subsystem was identical for both experiments. This was achieved by importing the storage LUNS as Raw Disk Mapping (RDM) devices in ESX. A schematic of the storage layout is in Figure 2.

Figure 2. Storage layout

Benchmark Methodology



Single Virtual Machine Scale Up Tests

The tests include native and virtual experiments at 1, 2, 4, and 8 CPUs. For the native experiments, the bcdedit Windows utility was used to specify the number of processors to be booted. The size of the database, the memory size, and SQL Server buffer cache were carefully scaled to ensure the workload profile was unchanged, while getting the best performance from the system for each data point. All the scale up experiments ran at 100 percent CPU utilization in native and virtual configurations. For example, at 2 vCPUs, in the guest, the workload would saturate both virtual CPUs. Table 6 describes the configuration used for each experiment.

Table 6. Brokerage workload configuration for scale up tests

Number of CPUs	Database Scale (Number of Customers)	Number of User Connections	Virtual Machine/ Native Memory	SQL Server Buffer Cache Size
1	30,000	65	9GB	7GB
2	60,000	120	16GB	14GB
4	115,000	220	32GB	28GB
8	185,000	440	58GB	53.5GB

Multiple Virtual Machine Scale Out Tests

The experiment includes identical 2 vCPU virtual machines with the same load applied to each. In order to provide low disk latencies for the high volume of IOPS from each virtual machine, the data disks for individual VMs are configured on exclusive spindles and two virtual machines shared one log disk. Table 7 describes the configuration used for each virtual machine in the multi-virtual machine experiments.

Table 7. Brokerage workload configuration for each 2 vCPU virtual machine

Database Scale	Number of User Connections	Virtual Machine Memory	SQL Server Buffer Cache
20000	35	7GB	5GB

Experiments with vSphere 4: Features and Enhancements

The 4 vCPU configuration mentioned above in Table 6 was used to obtain the results with ESX features and enhancements.

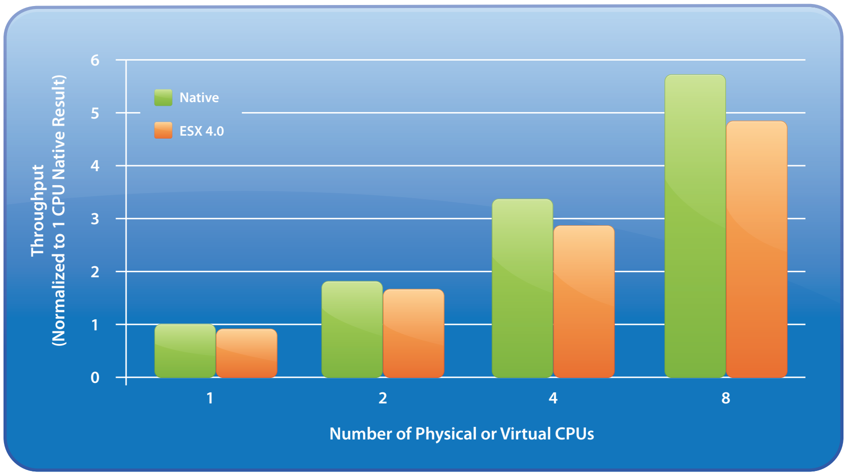
Performance Results

The sections below describe the performance results of experiments with the Brokerage workload on SQL Server in a native and virtual environment. Single and multiple virtual machine results are examined, and then results are detailed that show improvements due to specific ESX 4.0 features.

Single Virtual Machine Performance Relative to Native

How vSphere 4 performs and scales relative to native are shown in Figure 3 below. The results are normalized to the throughput observed in a 1 CPU native configuration:

Figure 3. Scale-up performance in vSphere 4 compared with native



The graph demonstrates the 1 and 2 vCPU virtual machines performing at 92 percent of native. The 4 and 8 vCPU virtual machines achieve 88 and 86 percent of the non-virtual throughput, respectively. At 1, 2, and 4 vCPUs on the 8 CPU server, ESX is able to effectively offload certain tasks such as I/O processing to idle cores. Having idle processors also gives ESX resource management more flexibility in making virtual CPU scheduling decisions. However, even with 8 vCPUs on a fully committed system, vSphere still delivers excellent performance relative to the native system.

The scaling in the graph represents the throughput as all aspects of the system are scaled such as number of CPUs, size of the benchmark database, and SQL Server buffer cache memory. Table 8 shows ESX scaling comparably to the native configuration's ability to scale performance.

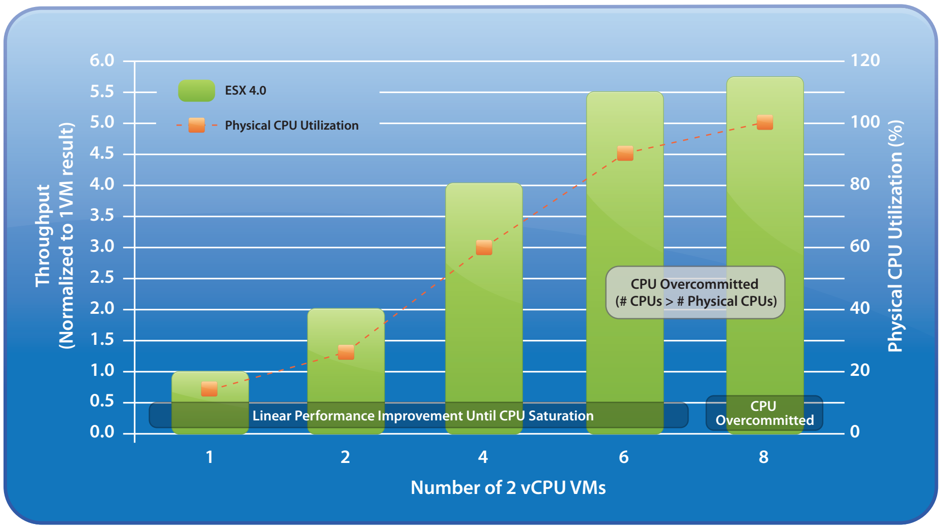
Table 8. Scale up performance

Comparison	Performance Gain
Native 8 CPU vs. 4 CPU	1.71
vSphere 8 vCPU vs. 4 vCPU	1.67

Multiple Virtual Machine Performance and Scalability

These experiments demonstrate that multiple heavy SQL Server virtual machines can be consolidated to achieve scalable aggregate throughput with minimal performance impact to individual virtual machines. Figure 4 shows the total benchmark throughput as eight 2 vCPU SQL Server virtual machines are added to the Brokerage workload onto the single 8-way host:

Figure 4. Consolidation of multiple SQL Server virtual machines



Each 2 vCPU virtual machine consumes about 15 percent of the total physical CPUs, 5GB of memory in the SQL Server buffer cache, and performs about 3600 I/Os per second (IOPS).

As the graph demonstrates, the throughput increases linearly as up to four virtual machines (8 vCPUs) are added. As the physical CPUs were overcommitted by increasing the number of virtual machines from four to six (a factor of 1.5), the aggregate throughput increases by a factor of 1.4 .

Adding eight virtual machines to this saturates the physical CPUs on this host. ESX 4.0 now schedules 16 vCPUs onto eight physical CPUs, yet the benchmark aggregate throughput increases a further 5 percent as the ESX scheduler is able to deliver more throughput using the few idle cycles left over in the 6 vCPU configuration. Figure 5 shows the ability of ESX to fairly distribute resources in the 8 vCPU configuration:

Figure 5. Overcommit fairness for 8 virtual machines

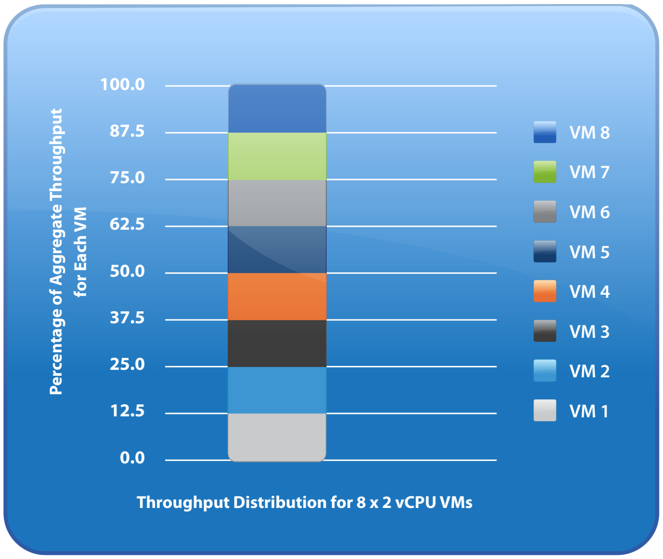


Table 9 highlights the resource intensive nature of the eight virtual machines that were used for the scale out experiments:

Table 9. Aggregate system metrics for eight SQL Server virtual machines

Aggregate Throughput in Transactions per Second	Host CPU Utilization	Disk I/O Throughput (IOPS)	Network Packet Rate	Network Bandwidth
2760	100%	23K	8 K/s receive 7.5 K/s send	9 Mb/sec receive 98 Mb/sec send

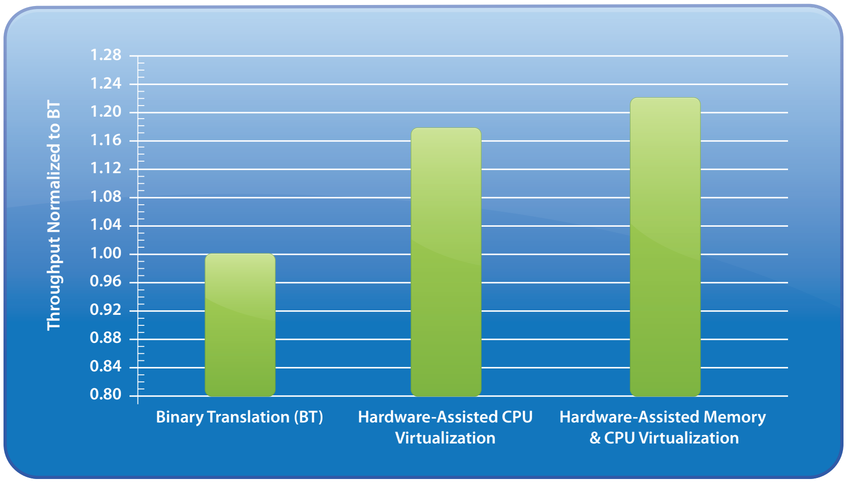
Performance Impact of Individual Features

In this section, we quantify the performance gains from key existing ESX features such as choice in virtual machine monitor and large page support.

Virtual Machine Monitor

Modern x86 processors from AMD and Intel have hardware support for CPU instruction set and memory management unit (MMU) virtualization. More details on these technologies can be found in [1], [2], and [3]. ESX 3.5 and vSphere effectively leverage these features to reduce the virtualization overhead and increase performance as compared to native for most workloads. Figure 6 details the improvements due to AMD's hardware MMU as supported by VMware. Results are shown as compared to VMware's all-software approach, binary translation (BT), and a mixed mode of AMD-V and vSphere's software MMU.

Figure 6. Benefits of hardware assistance for CPU and memory virtualization



In these experiments, enabling AMD-V results in an 18 percent improvement in performance compared with BT. Most of the overhead in binary translation of this workload comes from the cost of emulating privileged, high resolution timer-related calls used by SQL Server. RVI provides another 4 percent improvement in performance. When running on processors that include support for AMD-V RVI, ESX chooses RVI as the default monitor type for Windows Server 2008 guests.

Large Pages

Hardware assist for MMU virtualization typically improves the performance for many workloads. However, it can introduce overhead arising from increased latency in the processing of TLB misses. This cost can be eliminated or mitigated with the use of large pages in ESX ([2], [3]). Large pages can therefore reduce overhead due to the hardware MMU in addition to the performance gains they provide to many applications.

Database applications such as this test environment significantly benefit from large pages. Table 10 demonstrates the performance gains that can be seen from configuring large pages in the guest and ESX.

Table 10. Performance benefits of large pages with RVI monitor

SQL Server 2008 Memory Page Size	Physical Memory Page Size on ESX	Performance Gain Relative to Small Pages in Guest and ESX
Small	Small	Baseline
Large	Small	5%
Small*	Large*	13%
Large	Large	19%

*ESX 3.5 and vSphere do a best-effort allocation of all guest pages onto large physical pages in order to ensure good performance by default for your application on the hardware-assisted MMU monitor.

As seen in the table, the Brokerage workload gains significantly from having large pages in both SQL Server and ESX. More information on large pages and its impact to a variety of workloads is available at [4].

Performance Impact of vSphere Enhancements

In this section, key enhancements in vSphere are briefly described and gains are quantified for the Brokerage workload. Many of these improvements derive from changes in the storage and networking stacks and in the CPU and memory resource scheduler. Most of these enhancements are designed to deliver best performance out-of-the-box and should require little tuning.

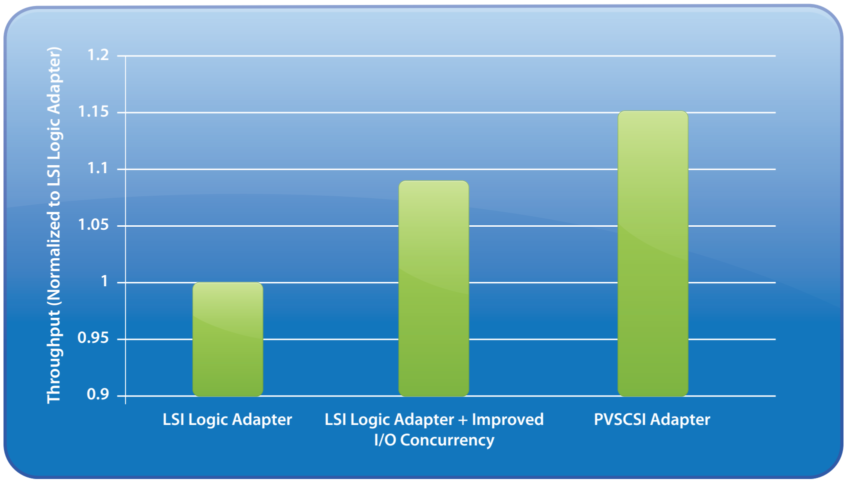
Virtual Machine Monitor

vSphere introduced support for eight vCPUs in a virtual machine. As shown in Table 9, SMP scaling for this workload in the physical and virtual environment closely matches. In both cases, approximately 70 percent improvement in throughput is observed going from four to eight CPUs.

Storage Stack Enhancements

There are a number of improvements in the vSphere storage layer that significantly improve the CPU efficiency of performing I/O operations and allow virtual machines to handle the demands of the most intensive enterprise applications. Figure 7 summarizes the performance impact from vSphere storage enhancements:

Figure 7. Performance improvement from storage stack enhancements



A brief description of these enhancements and features is given below:

- **PVSCSI: A New Paravirtualized SCSI Adapter and Driver**

  The PVSCSI driver is installed in the guest operating system as part of VMware Tools and shares I/O request and completion queues with the hypervisor. The tighter coupling enables the hypervisor to poll for I/O requests from guest and complete requests using an adaptive interrupt coalescing mechanism. This batching of I/O requests and completion of interrupts significantly improves the CPU efficiency for handling large volumes of I/Os per second (IOPs).

  A 6 percent improvement in throughput with the PVSCSI driver and adapter is observed over the LSI Logic parallel adapter and guest driver.

- **I/O Concurrency Improvements**

  In previous releases of ESX, I/O requests issued by the guest are routed to the VMkernel via the virtual machine monitor (VMM). Once the requests reach the VMkernel, they execute asynchronously. The execution model in ESX 4.0 allows the VMM to asynchronously handle the I/O requests allowing vCPUs in the guest to execute other tasks immediately after initiating an I/O request. This improved I/O concurrency model is designed around two ring structures per adapter which are shared by the VMM and the VMkernel.

  The workload benefited significantly from this optimization and saw a 9 percent improvement in throughput.

- **Virtual Interrupt Coalescing**

  In order to further reduce the CPU cost of I/O in high-IOPS workloads, ESX 4.0 implements virtual interrupt coalescing to allow for batching of I/O completions to happen at the guest level. This is similar to the physical interrupt coalescing that is provided in many Fibre Channel host bus adapters.

  With this optimization, the disk I/O interrupt rate within a Windows Server 2008 guest was half of that within the native operating system in the 8 vCPU configuration.

- **Interrupt Delivery Optimizations**

  In ESX 3.5, virtual interrupts for Windows guests were always delivered to vCPU0. vSphere posts interrupts to the vCPU that initiated the I/O, allowing for better scalability in situations where vCPU0 could become a bottleneck.

  ESX 4.0 uniformly distributes virtual I/O interrupts to all vCPUs in the guest for the Brokerage workload.

Network Layer

Network transmit coalescing between the client and server benefited this benchmark by two percent on vSphere. Transmit coalescing was available for the enhanced vmxnet networking adapter since ESX 3.5 and it was further enhanced in vSphere. Dynamic transmit coalescing was also added to the E1000g adapter.

In these experiments, the parameter **Net.vmxnetThroughputWeight**, available through the Advanced Software Settings configuration tab in the VI Client, was changed from the default value 0 to 128, thus favoring transmit throughput over response time.

Setting this parameter could increase network transmit latencies and must be tested for its impact before being set. No increase in transaction response times were observed for the Brokerage workload due to this setting.

Resource Management

vSphere includes several enhancements to the CPU scheduler that help significantly in the single virtual machine as well as multiple virtual machine performance and scalability of this workload.

- **Further Relaxed Co-scheduling & Removal of Cell**

  Relaxed co-scheduling of vCPUs is allowed in ESX 3.5 but has been further improved in ESX 4.0. In previous versions of ESX, the scheduler would acquire a lock on a group of physical CPUs within which vCPUs of a virtual machine were to be scheduled.

  In ESX 4.0 this has been replaced with finer-grained locking, reducing scheduling overheads in cases where frequent scheduling decisions are needed. Both these enhancements greatly help the scalability of multi-vCPU virtual machines.

- **Enhanced Topology / Load Aware Scheduling**
  A goal of the vSphere CPU resource scheduler is to make CPU migration decisions in order to maintain processor cache affinity while maximizing last level cache capacity.

  Cache miss rates can be minimized by always scheduling a vCPU on the same core. However, this can result in delays in scheduling certain tasks as well as lower CPU utilization. By making intelligent migration decisions, the scheduler in ESX 4.0 strikes a good balance between high CPU utilization and low cache miss rates.

  The scheduler also takes into account the processor cache architecture. This is especially important in light of the differences between the various processor architectures in the market. Migration algorithms have also been enhanced to take into account the load on the vCPUs and physical CPUs.

Conclusion

The experiments in this paper demonstrate that vSphere is optimized out-of-the-box to handle the CPU, memory, and I/O requirements of most common SQL Server database configurations. The paper also clearly quantified the performance gains from the improvements in vSphere. These improvements can result in better performance, higher consolidation ratios, and lower total cost for each workload.

The results show that performance is not a barrier for configuring large multi-CPU SQL Server instances in virtual machines or consolidating multiple virtual machines on a single host to achieve impressive aggregate throughput. The small difference observed in performance between the equivalent deployments on physical and virtual machines—without employing sophisticated tuning at the vSphere layer—indicate that the benefits of virtualization are easily available to most SQL Server installations.

Disclaimers

All data is based on in-lab results with the RTM release of vSphere 4. Our workload was a fair-use implementation of the TPC-E business model; these results are not TPC-E compliant and are not comparable to official TPC-E results. TPC Benchmark and TPC-E are trademarks of the Transaction Processing Performance Council.

The throughput here is not meant to indicate the absolute performance of Microsoft SQL Server 2008, nor to compare its performance to another DBMS. SQL Server was used to place a DBMS workload on ESX, and observe and optimize the performance of ESX.

The goal of the experiment was to show the relative-to-native performance of ESX, and its ability to handle a heavy database workload. It was not meant to measure the absolute performance of the hardware and software components used in the study.

The throughput of the workload used does not constitute a TPC benchmark result.

Acknowledgements

The author would like to thank Aravind Pavuluri, Reza Taheri, Priti Mishra, Chethan Kumar, Jeff Buell, Nikhil Bhatia, Seongbeom Kim, Fei Guo, Will Monin, Scott Drummonds, Kaushik Banerjee, and Krishna Raja for their detailed reviews and contributions to sections of this paper.

About the Author

Priya Sethuraman is a Senior Performance Engineer in VMware's Core Performance Group where she focuses on optimizing the performance of applications in a virtualized environment. She is passionate about performance and the opportunity to make software run efficiently on a large system. She presented her work on "Enterprise Application Performance and Scalability on ESX" at VMworld Europe 2009.

Priya received her MS in Computer Science from University of Illinois Urbana-Champaign. Prior to joining VMware, she had extensive experience in database performance on UNIX platforms.

References

[1] AMD Virtualization (AMD-V) Technology. http://www.amd.com/us-en/0,,3715_15781_15785,00.html. Retrieved April 14, 2009.

[2] Performance Evaluation of AMD RVI hardware Assist. http://www.vmware.com/pdf/RVI_performance.pdf.

[3] Performance Evaluation of Intel EPT Hardware Assist. http://www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf.

[4] Large Page Performance. Technical report, VMware, 2008. http://www.vmware.com/resources/techresources/1039.

[5] Improving Performance with Interrupt Coalescing for Virtual Machine Disk I/O in VMware ESX Server. Irfan Ahmad, Ajay Gulati, and Ali Mashtizadeh, "Second International Workshop on Virtualization Performance: Analysis, Characterization, and Tools (VPACT'09)", held with IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2009.

VMware, Inc. 3401 Hillview Ave Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com