

SQL Server Workload Consolidation

VMware ESX 3.5

Database workloads are very diverse. While most database servers are lightly loaded, larger database workloads can be resource-intensive, exhibiting high I/O rates or consuming large amounts of memory. With improvements in virtualization technology and hardware, even servers running large database workloads run well in virtual machines. Servers running Microsoft's SQL Server, among the top database server platforms in the industry today, are no exception.

An important consideration in consolidation scenarios is application performance when packing multiple virtual machines on a single hardware platform. Application performance in virtual machines should continue to meet or exceed required service levels. That is to say, the virtual platform should:

- Be scalable.
- Ensure that all virtual machines get resources in proportion to their load levels up to specified resource limits.
- Provide performance isolation for each virtual machine running on a host.
- Ensure that the overall load of a host will have minimal impact on the performance of applications running in individual virtual machines hosted on that host.

This paper demonstrates the ability of VMware® ESX 3.5 to scale while guaranteeing fairness and isolation under a demanding SQL Server load. Experiments discussed in this paper involve virtual machines demanding system resources as their numbers increase, virtual machines with varying degrees of load contending for finite system resources, and runaway virtual machines attempting to demand additional system resources due to increasing load. In all cases ESX 3.5 meets or exceeds the most demanding needs.

Executive Summary

Experiments described in this paper were conducted with multiple virtual machines running on ESX 3.5, with each virtual machine running Microsoft SQL Server 2005 64bit with Service Pack 2. We used the Dell DVD Store 2 (DS2) database test suite to stress the SQL databases. DS2 was configured to use a large database, consisting of approximately 90GB of data and indexes. Each virtual machine used one virtual disk for database files and one virtual disk for log files. Both virtual disks were placed on the same VMFS datastore. The datastore and the underlying LUN were not shared with any other virtual machine. All virtual machines were configured identically.

There were four main focus areas for this study:

- 1 Demonstrate ESX scalable performance in consolidated environments.
- 2 Illustrate the fairness of resource allocation in a consolidated environment.
- 3 Show that virtual machines are allocated resources proportional to their workloads (up to specified resource limits).
- 4 Validate performance isolation in a multi-virtual machine scenario.

Based on the test data presented in this paper, we can conclude that:

- As the number of virtual machines is increased ESX offers linear scaling of performance until the physical CPUs are saturated.
- Fairness in resource sharing makes ESX a robust platform for hosting and consolidating virtual machines running SQL Server workloads.
- While resources are available, the CPU load on the host has minimal impact on the performance of applications running in virtual machines on that host.
- Virtual machines get the resources they need when resources are not overcommitted. Resources not used by idle virtual machines are dynamically allocated to other virtual machines that need resources without affecting performance.

Experiment Setup

This section describes the configuration used to run the tests presented in this paper.

Test Application

The DS2 application (available from <http://linux.dell.com/dvdstore>) was used to conduct all the experiments. DS2 provides a complete online e-commerce test application, with a backend database component, a web application layer, and driver programs. Separate driver programs are available to drive the database directly or through the web application. In this investigation, we used the DS2 workload to directly test database performance in a virtualized environment.

The client component of the test kit was executed on a remote physical machine. Database servers were hosted in virtual machines.

Test Parameters

Parameter values were passed to DS2 by means of a configuration file. The parameters used in these tests are specified in [Table 1](#).

Table 1. Test Parameters

Parameter	Description	Value
n_threads	Number of concurrent users	varies
ramp_rate	Startup rate (users/sec)	10
run_time	Runtime during which statistics are kept (min)	10
db_size_str	Database size	L

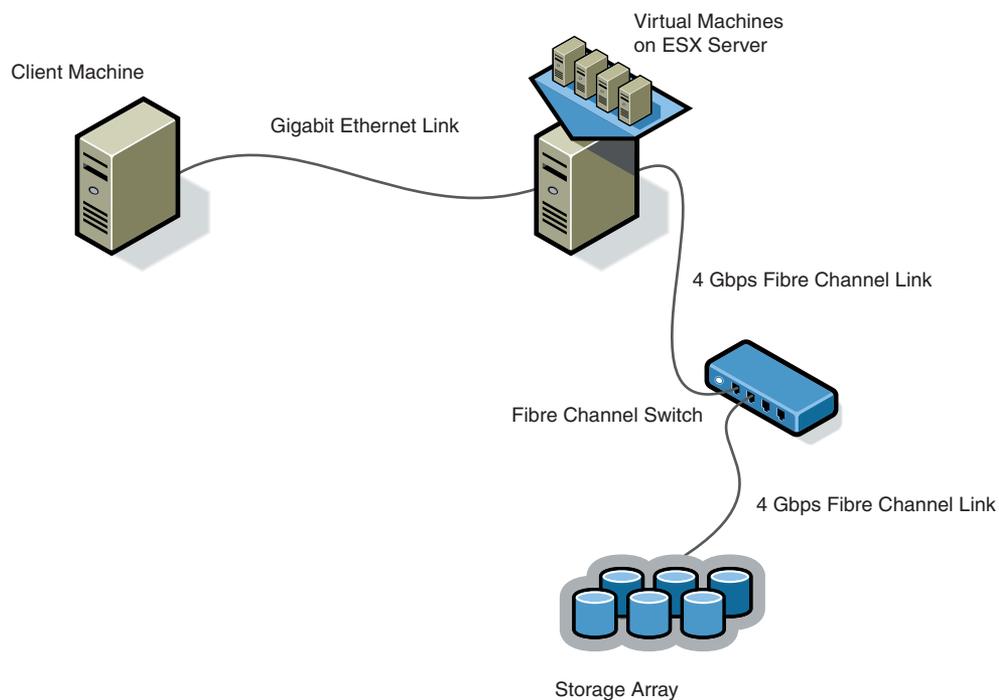
Table 1. Test Parameters

Parameter	Description	Value
warmup_time	Warmup time before statistics are kept (min)	2
think_time	Think time (sec)	0.1
pct_newcustomers	Percent of customers who are new	30
n_searches	Average number of searches per order	3
search_batch_size	Average number of items returned in each search	4
n_line_items	Average number of items per order	5

Test Environment

A dual-socket, quad-core Dell PowerEdge 2950 and a 4-way HP DL580 server were used as the test server and the client machine, respectively, while an EMC CX3-40 SAN was used to meet all storage needs. The Gigabit Ethernet link between the test machines was via a crossover cable. The test server and the storage array were linked via a Fibre Channel switch. Detailed specifications of all components are listed at the end of the paper. To ensure accuracy of the performance data all components and links were dedicated to these experiments. Parameter settings for the SAN, HBA, and Fibre Channel switch were at default values.

The test server, storage array, and client machine used to drive the workload were set up as shown in [Figure 1](#).

Figure 1. Test Setup

Virtual Machines

Microsoft Windows Server 2003, Enterprise x64 Edition, SP2 was used as the operating system in each virtual machine running SQL Server. Each virtual machine was configured with 5GB of memory and two virtual CPUs. In addition to the operating system disk, all virtual machines were configured with two virtual disks that were used for the SQL database.

Software Configuration

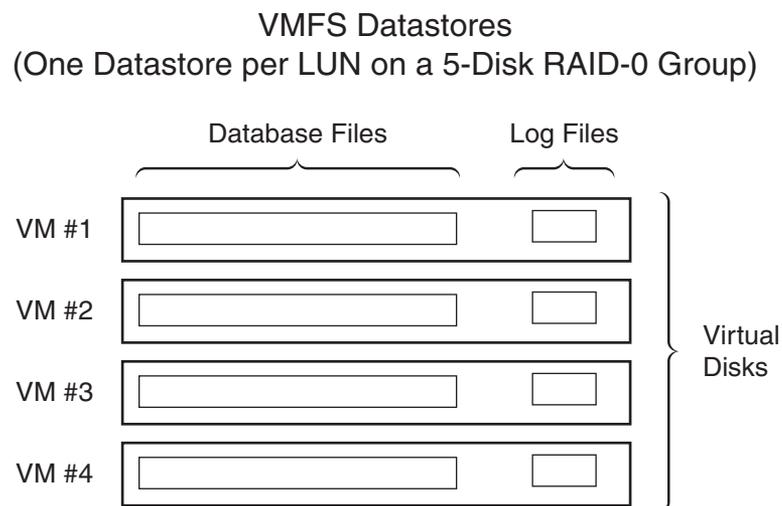
ESX 3.5 was used in these tests. Configuration was left at default settings. The SQL Server recovery interval was altered from its default to five minutes and the recovery model from Full to Simple. This results in logging performance which is more predictable. CPU statistics collection was turned off, reducing the overhead inside the virtual machine without loss of information.

Storage Layout

Each virtual machine used a separate VMFS datastore for creating two virtual disks that hosted the database files. Each VMFS datastore was created on a separate five-disk RAID-0 LUN. Database tables and associated indexes were located in one of the virtual disks while the other virtual disk was used for the log. This configuration was identical for all virtual machines. [Figure 2](#) gives a graphical representation of the storage configuration of each virtual machine.

Despite the lack of redundancy, we used RAID-0 to maximize the disk throughput available for virtual machines running the SQL Server database. This prevented the storage subsystem from becoming a bottleneck during the tests. However, production database deployments would likely use some level of redundancy in the storage layout, thus preferring RAID-5 or RAID-10 to RAID-0. Use of RAID levels with redundancy will have negligible impact on the database performance described in this paper if the RAID is configured with an appropriate number of disks.

Figure 2. Virtual Storage Layout



The test database consists of approximately 90GB of data and indexes. The initial row counts of all tables are listed in [Table 2](#). Several tables increase in size during the test, necessitating the re-initialization of test databases before every test.

Table 2. Initial Data Records

Table Name	# of Rows
Customers	200 million
Cust_Hist	600 million
Orders	120 million
Orderlines	600 million
Products	1 million
Inventory	1 million
Reorder	0 initially
Categories	16

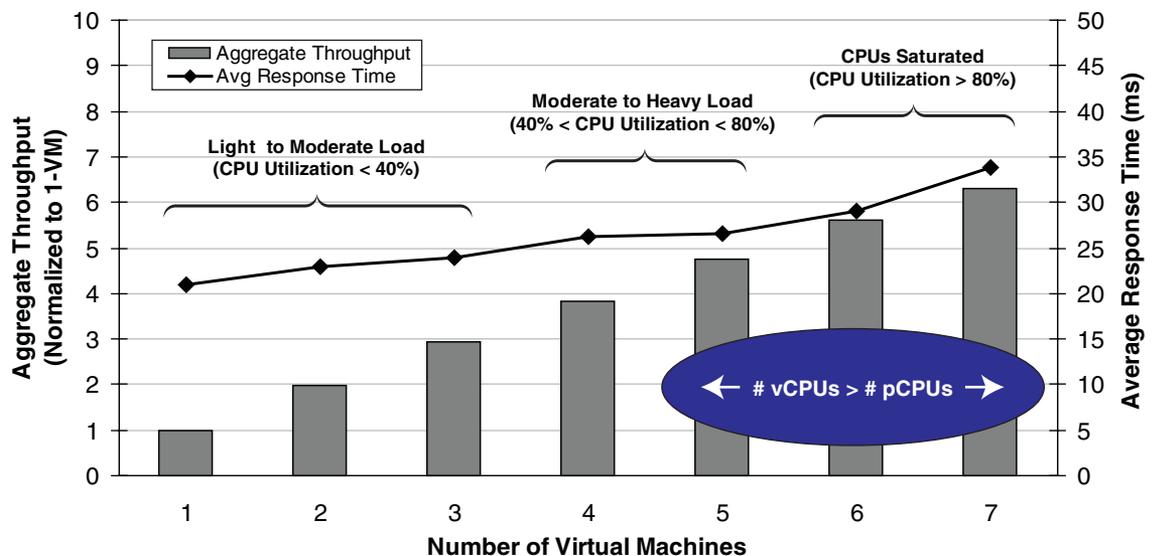
Results

This section presents the results of our experiments. Orders per minute (OPM) was used as the throughput metric to compare performance in each case. Average response time of each operation was used to compare latency in each case. Both OPM and average response time are reported by the DS2 application. All CPU utilization data was collected in ESX using esxtop.

Scaling Performance

In the first phase of our experiments we started with a single virtual machine running SQL Server 2005 and the DS2 database. We used the DS2 test suite to stress the database running in the virtual machine and recorded its performance. This was used as a baseline for our scalability experiments. We then gradually increased the number of virtual machines running on the ESX host from one to seven. Each virtual machine was identical to others in all respects and ran SQL Server 2005 and the DS2 database. We used separate instances of the DS2 driver program to stress each virtual machine and recorded the application performance in each case. Results are shown in [Figure 3](#).

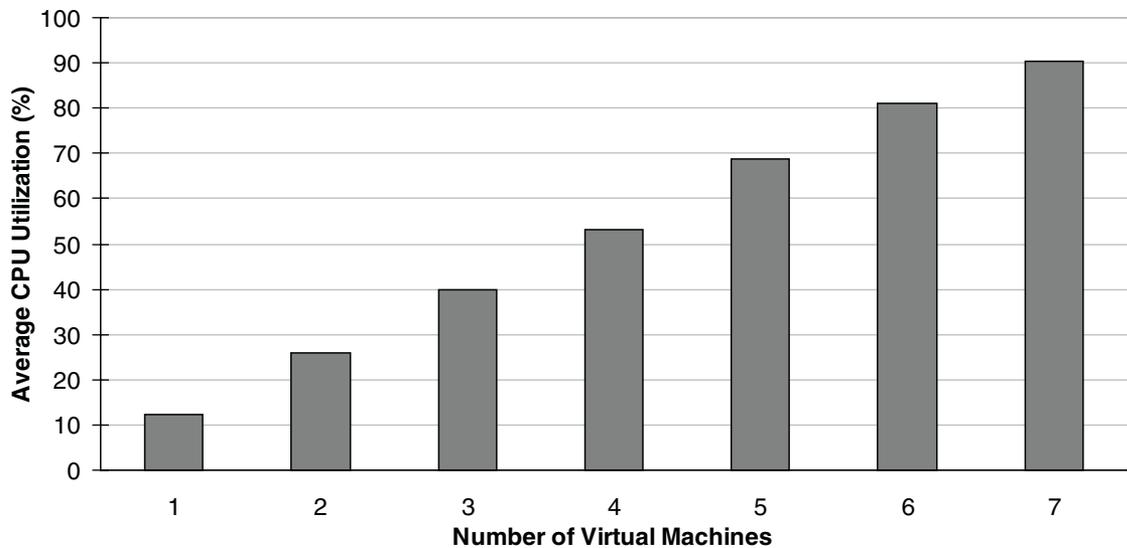
Figure 3. Throughput Scalability



[Figure 3](#) plots the aggregate performance of a given number of virtual machines. Performance of one virtual machine was used as the baseline to normalize aggregate performance when running multiple virtual machines. As shown in the graph, the aggregate performance scales linearly. Though the number of virtual processors exceeded the number of physical cores in the host once we reached five virtual machines, we continue to observe linear scalability up to seven virtual machines. At this stage, additional virtual machines saturated the physical CPUs.

We also measured application response time of individual virtual machines as their number was increased. We plotted the average response time of all the virtual machines as a function of the number of virtual machines running at the time. This is also shown in [Figure 3](#). As seen in the figure, with light to moderate CPU load the increase in application response time was barely noticeable. When CPU load increased, application response time also increased, but at a much slower rate. As the system reached CPU saturation the application response time increased at a faster rate, indicating contention among virtual machines for system resources.

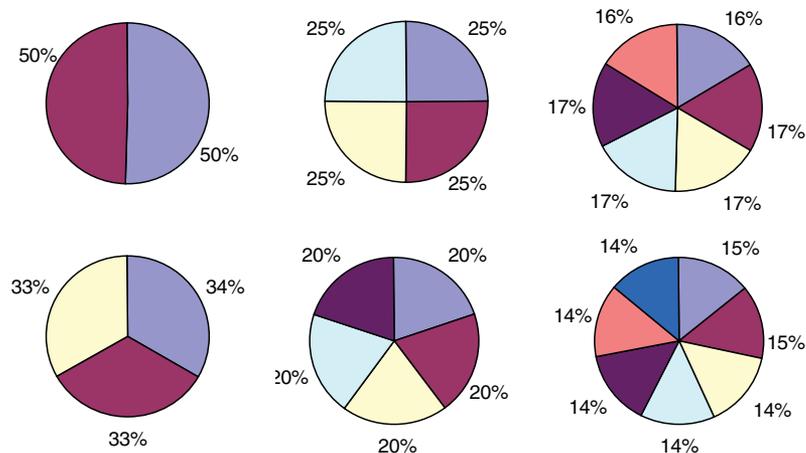
[Figure 4](#) shows overall CPU utilization as the number of virtual machines was increased. The graph illustrates that as virtual machines were added CPU utilization scaled linearly until the CPUs reach saturation.

Figure 4. CPU Utilization

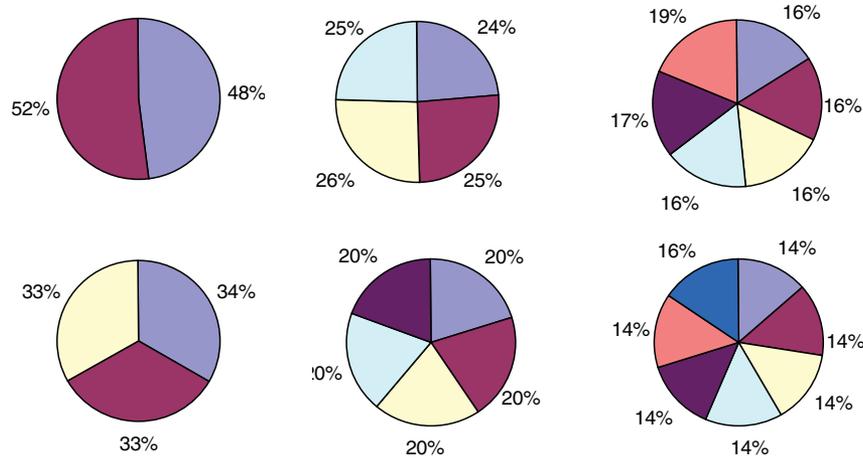
Fairness

Fair sharing of system resources is an important factor in evaluating the efficacy of consolidation. Fairness can be quantified by comparing the throughput and response time metrics of the workload running in the virtual machines. Ideally, if resources are not overcommitted, the throughput levels and average response time of the operations should be the same for identically configured virtual machines running identical workloads.

The pie charts in [Figure 5](#) show the ratios of the application throughput measured in individual virtual machines as the number of virtual machines running on the host was increased. Each wedge in the pie chart represents the throughput of one virtual machine as a percentage of the total throughput. The data presented in the charts show that the throughput of the virtual machines remained nearly identical regardless of the number of virtual machines running on the ESX 3.5 host.

Figure 5. Uniformity of Throughput across Virtual Machines

[Figure 6](#) shows the measured average response time of the DS2 application in each virtual machine relative to the other virtual machines. Each wedge represents the application response time of one virtual machine as a percentage of the sum of the response times of all virtual machines. The data presented in the charts shows that the average response time of each virtual machine is nearly the same regardless of the number of virtual machines running on ESX 3.5 host.

Figure 6. Evenness of Average Response Time across Virtual Machines

Performance Under Uneven Load

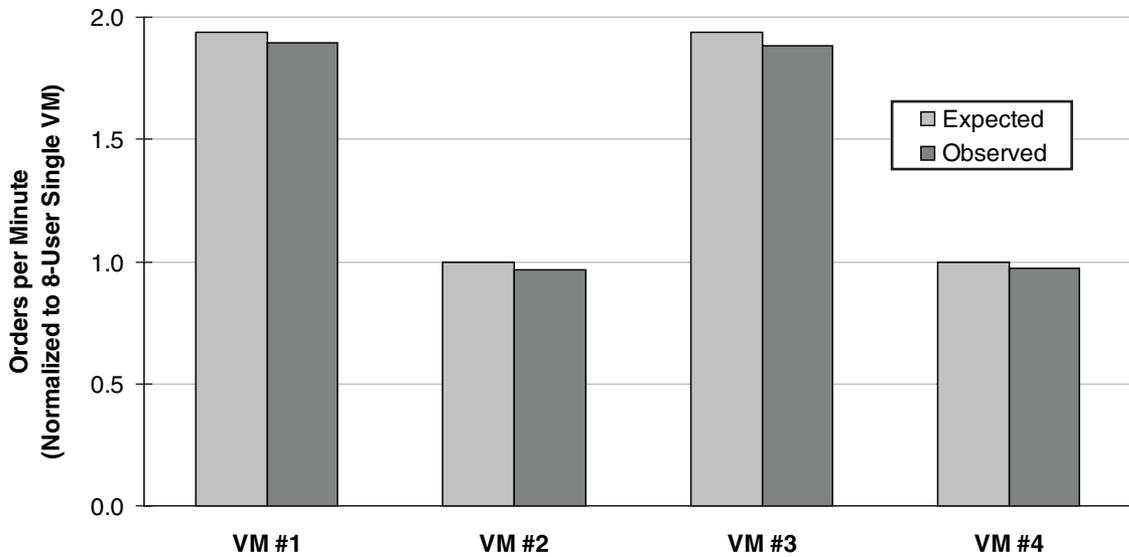
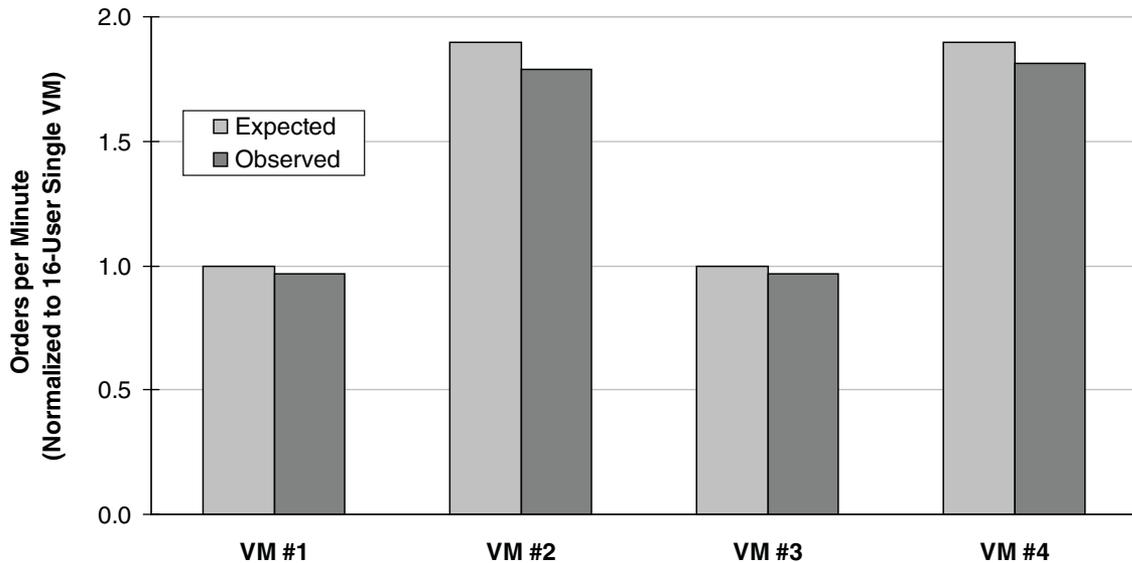
In the second phase of our experiments we wanted to study the ability of the ESX resource scheduler to ensure fairness in resource allocation and isolation of performance when virtual machines are running at different load levels. To achieve this we ran two sets of virtual machines each running at different load level. All virtual machines ran SQL Server 2005 database.

The specific test cases are enumerated in [Table 3](#).

Table 3. Test Cases with Uneven Load

Test	Load on virtual machines	Notes
A	Single virtual machine: 8, 16 and 32 users	Baseline
B	VM #1, VM #3: 16 users each VM #2, VM #4: 8 users each	- Show that virtual machines deliver throughput in proportion to their user load levels. - Data from Test Case A was used as the baseline.
C	VM #1, VM #3: 16 users each VM #2, VM #4: 32 users each	- Show that increasing load on VM #2 & VM #4 does not affect the performance of VM #1 & VM #3 - Both pairs of virtual machines continue to get the resources they request to deliver throughput in proportion to the load levels

[Figure 7](#) and [Figure 8](#) compare the expected throughput with actual throughput of individual virtual machines observed after running Test cases B and C, respectively. The expected throughput is based on the results of running the DS2 application against a lone virtual machine with 8, 16, and 32 user connections.

Figure 7. Expected vs. Observed Throughput Levels (Test B)**Figure 8.** Expected vs. Observed Throughput Levels (Test C)

During test B, the loads on the CPUs of the individual virtual machines were light to moderate (< 50%). In this case the observed performance of ESX was well within 96% of the expected performance. When running Test C, virtual machines with 32 user connections became heavily loaded ($\approx 80\%$) while virtual machines with 16 users remained moderately loaded (< 50%). In this case, observed performance of moderately loaded virtual machines (#1 and #3) was within 96% of the expected performance, whereas observed performance of heavily loaded virtual machines (#2 and #4) was about 94% of expected performance. We believe the slight drop in the observed performance to be due to sharing of resources by virtual machines and extra CPU cycles spent in scheduling resources to virtual machines.

Experimental results shown in [Figure 7](#) and [Figure 8](#) indicate that the performance of virtual machines remains nearly identical regardless of whether they are the only virtual machine running on an ESX host or they co-exist with other virtual machines. ESX isolates the virtual machines contending for resources from each other.

The pie charts in [Figure 9](#) and [Figure 10](#) show the relative number of users and the application throughput of each virtual machine in Tests B and C, respectively. As seen in the charts, the fraction of application throughput of each virtual machine matches the fraction of users connected to the database in that virtual machine. These results demonstrate that ESX does an excellent job of allocating the system resources required to process a given workload in a virtual machine and ensures that resource allocation remains proportional to the load on the virtual machine.

Figure 9. Load-Level and Throughput Balance (Test B)

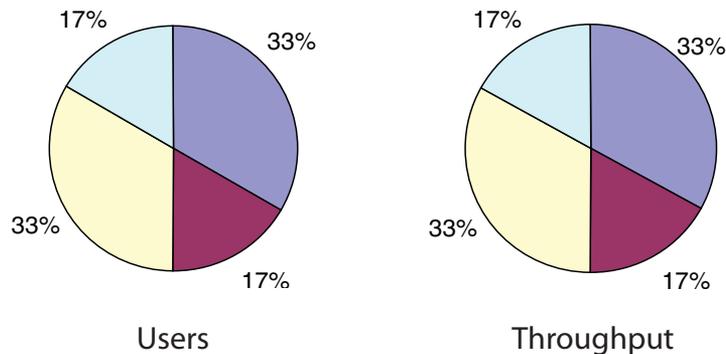
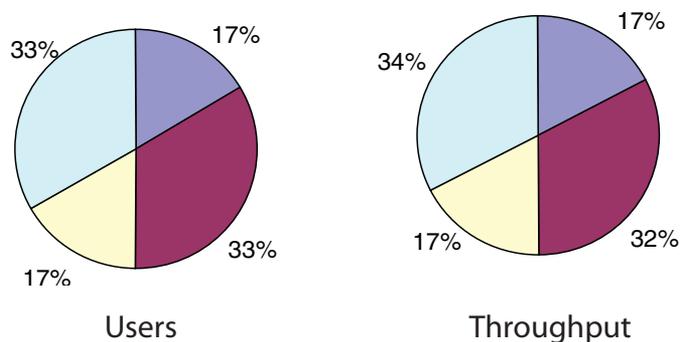


Figure 10. Load-Level and Throughput Balance (Test C)



Results of Tests B and C indicate that ESX not only isolates virtual machines running in a consolidated environment, but ensures that virtual machines get the resources they need to process their workloads.

Runaway Virtual Machine Performance

In this section we describe the results of our experiments to study the efficiency of ESX in handling variations in resource requirements of virtual machines. Resource requirements of virtual machines can vary when the workloads in the virtual machines vary over time. To achieve this, we kept the load in three virtual machines constant while we increased the load in the fourth virtual machine progressively. We call this fourth virtual machine a “runaway virtual machine.” Assuming that consolidation sizing was done with sufficient headroom, ESX resource scheduling should ensure that low-load virtual machines are not starved by the demands of the virtual machine whose load keeps increasing. The specific test cases are listed in [Table 4](#).

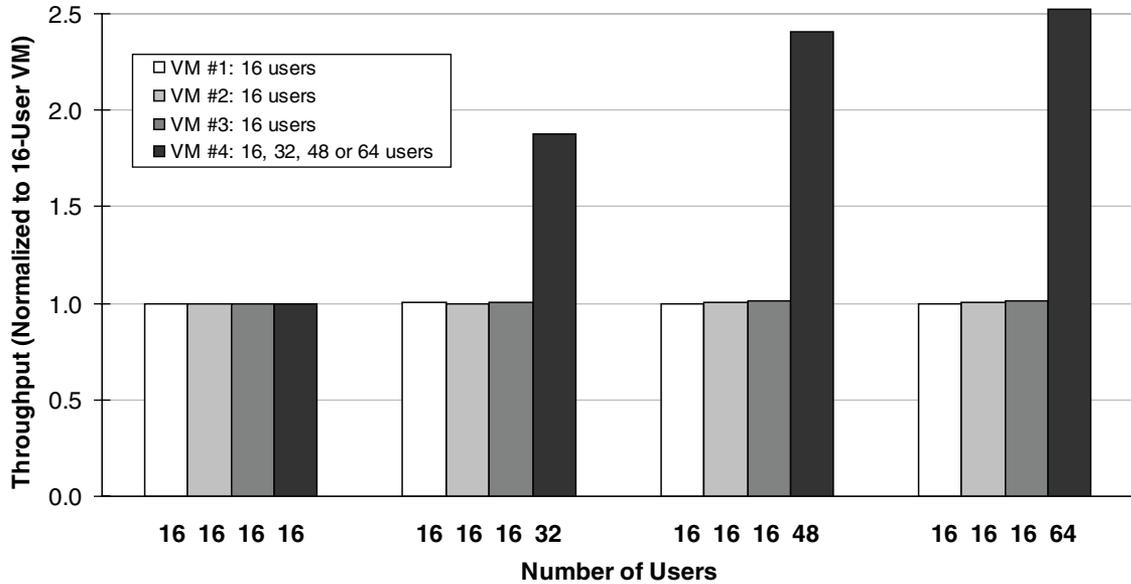
Table 4. Runaway Virtual Machine Test Cases

Test	Load on virtual machines	Notes
D	VM #1, VM #2, VM #3, VM #4: 16 users each	Multi-VM baseline
E	VM #1: variable load in increments of 16 users	Single-VM baseline
F	VM #1, VM #2, VM #3: 16 users each VM #4: variable load in increments of 16 users	<ul style="list-style-type: none"> - Show that the performance of the application in VM #1, VM #2, & VM #3 is not affected by runaway performance in VM #4 - Compare lightly-loaded virtual machines to Test D - Compare runaway virtual machine to Test E

We conducted the experiments in three steps. First we ran the DS2 test suite against four virtual machines, keeping the workload identical in all of them (Test D in [Table 4](#)). We then shut down three of the virtual machines and ran the DS2 test suite against the remaining virtual machine, varying the workload by incrementally increasing the workload in steps of 16 users (Test E in [Table 4](#)). Finally we powered on all four virtual machines and ran the DS2 test suite against all of them. This time we kept the workload constant in three of the virtual machines, but varied the workload in the remaining virtual machine in steps of 16 users (Test F in [Table 4](#)).

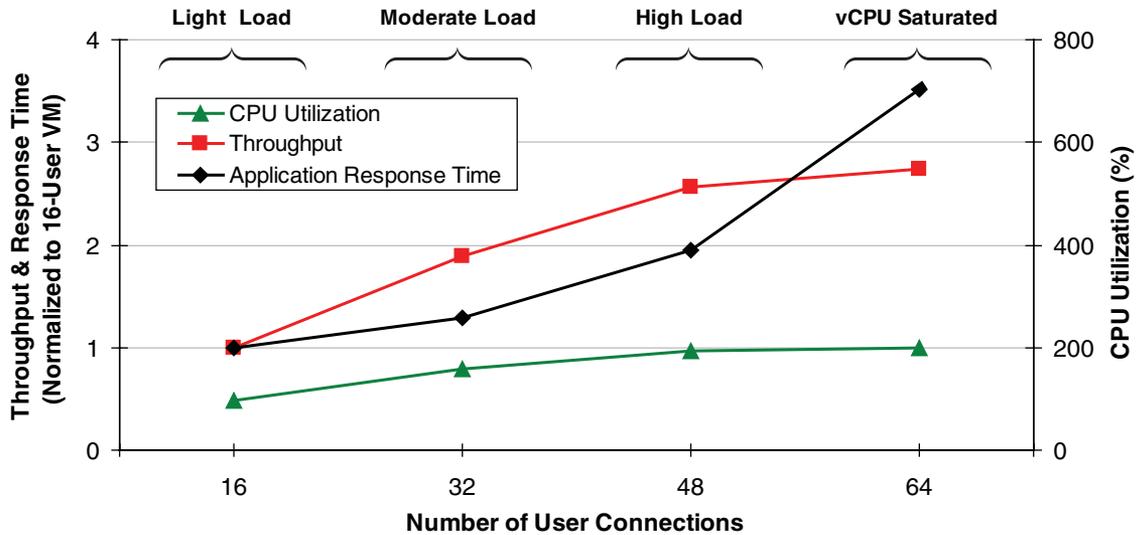
[Figure 11](#) compares the performance of all virtual machines in Test F. As seen in the graph, the performance of the three virtual machines whose workload remains constant is unaffected by the variation of the workload in the fourth virtual machine.

Figure 11. Effect of Increasing Load on Application Performance



We also wanted to compare the effect of consolidation on the performance of a virtual machine with variable workload. To understand this first let us look at a single virtual machine running SQL Server database and the DS2 workload. [Figure 12](#) shows the results of Test E.

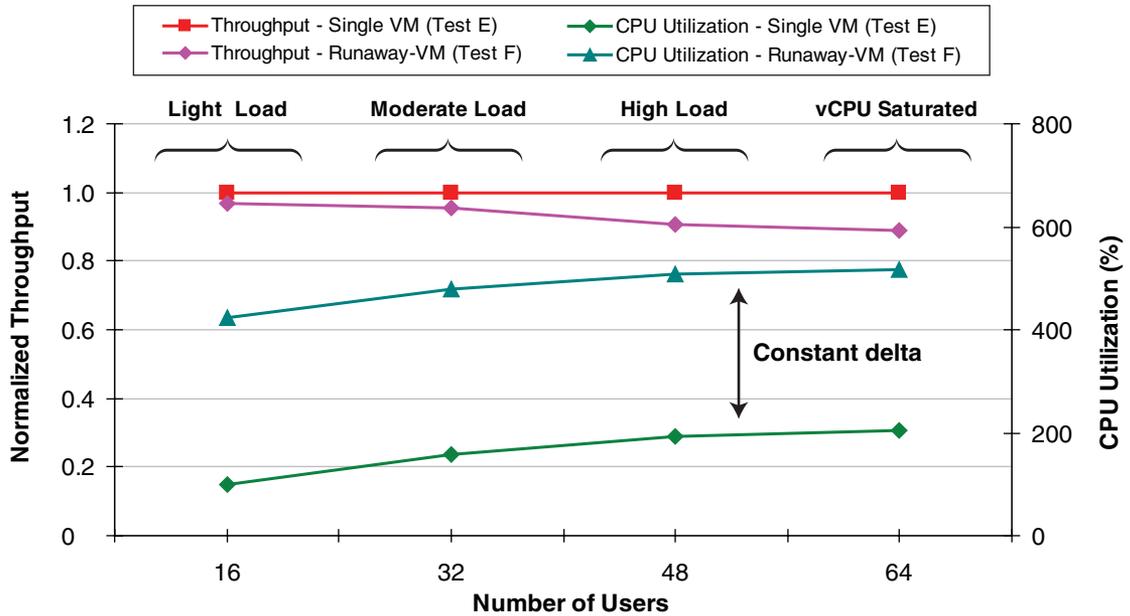
Figure 12. Effect of Workload Variations on Application Performance in Virtual Machines (Single-VM Environment)



The CPU Utilization scale in [Figure 12](#) goes to 800% to represent the full capacity of eight cores. The throughput scales linearly until the number of users reaches 48. The response time of the application increases at a lower rate. During this period, CPU utilization also increases and reaches saturation at 48 users. At this stage, increasing the number of users only increases the application response time while throughput flattens out. With 64 user connections, overall CPU utilization reaches 200%, indicating that the two physical CPU cores allocated to the virtual machine are fully saturated.

We ran a similar test on a virtual machine running in a consolidated environment. This represents Test F in [Table 4](#). We then compared the performance of a virtual machine with a variable workload from Test F with that of a single virtual machine with the identical variable workload in Test E. [Figure 13](#) shows the results of our comparison.

Figure 13. Effect of Workload Variations on Application Performance in Virtual Machines (Multi-VM Environment)



The CPU Utilization scale in [Figure 13](#) goes to 800% to represent the full capacity of eight cores. As seen in the figure, when the runaway virtual machine is lightly to moderately loaded, its performance is about 95% of the single virtual machine from Test E. The 5% drop could be attributed to sharing of hardware resources (CPU cores, caches, memory) and to virtualization and scheduling overhead. As the workload increases, resource contention and other overheads take more CPU cycles away from the virtual machine that could otherwise have been used to process the workload. When the vCPUs become saturated, there is about a 10% performance drop seen in the runaway-virtual machine compared with the single virtual machine. While running a system at CPU saturation is not very common in enterprise data centers, it is shown here to illustrate that customers can still expect good performance with a resource-constrained virtual machine in a consolidated environment. In such scenarios one can expect performance to be within 90% of a virtual machine that has enough resources available.

The CPU utilization of the ESX host is shown relative to the sum of the capacities of the individual cores in the system (in this case 800%, as there are eight cores). It is very important to note that the increase in CPU utilization in both Tests E and F is proportional, indicating that the CPU resources allocated to the runaway virtual machine and the single virtual machine are very similar; the constant delta is due to the additional presence of the three lightly-loaded virtual machines in Test F.

Conclusion

Experiments discussed in this paper demonstrate the highly scalable virtualization platform provided by ESX 3.5 for consolidating servers running SQL Server databases. Results shown indicate the efficiency of ESX in distributing the processing power available on modern large multi-way servers among multiple virtual machines, resulting in a very high utilization of the systems. While doing so, ESX provides each virtual machine a near-native experience through isolation and guaranteed resource allocation. ESX seamlessly handles variations in demand for resources when the application load in the virtual machines fluctuate, while at the same time preventing resource hogging by a few virtual machines causing starvation for others in a shared environment. Just like native platforms, applications continue to meet or exceed their service levels on a virtualized platform using ESX 3.5, but at much lower cost. All these traits make ESX 3.5 an excellent choice for consolidating servers running SQL Server databases.

Test Environment

Test Server

Model: Dell PE2950
Processors: Two quad-core Intel Xeon X5450 3GHz processors (1333MHz FSB, 12MB L2 cache)
RAM: 32GB
Internal Storage: Four 146GB SAS hard drives
Network: Two Broadcom NICs; Two Intel® PRO 1000PT PCIe dual-port server NICs
HBA: QLE2462-SP dual-port PCI-e x4 4Gb Fibre Channel

Client Driver

Model: HP ProLiant DL580 G2
Processors: Four dual-core Intel Xeon 2.2 GHz processors (400MHz FSB, 512KB L2 cache, 2MB L3 cache)
RAM: 16GB
Storage Controller: Compaq Computer Corporation Smart Array 5i/532 (rev 01)
Network: Four Intel 8254NXX Gigabit Ethernet controllers (rev 01)

SAN Storage

Model: EMC Clariion CX3-40 SAN
Disks: Total forty 146GB 15K RPM disks
Flare OS revision: 03.26.040.5.011

Virtualization Product Version

VMware ESX 3.5, release build 64607

SQL Server Version

Microsoft SQL Server 2005 Enterprise Edition (64 bit) with Service Pack 2

Test Virtual Machines

CPU: 2-way SMP
Memory: 4GB
Guest OS: Windows 2003 Enterprise x64 Edition, SP2

If you have comments about this documentation, submit your feedback to: docfeedback@vmware.com

VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 www.vmware.com

Copyright © 2008 VMware, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,944,699, 6,961,806, 6,961,941, 7,069,413, 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149,843, 7,155,558, 7,222,221, 7,260,815, 7,260,820, 7,269,683, 7,275,136, 7,277,998, 7,277,999, 7,278,030, 7,281,102, 7,290,253, 7,356,679, 7,409,487, 7,412,492, 7,412,702, 7,424,710, 7,428,636, 7,433,951, 7,434,002, and 7,447,854; patents pending. VMware, the VMware "boxes" logo and design, Virtual SMP, and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Revision 20081118; Item: EN-000160-00
