Oracle Database Scalability in VMware® ESX

VMware ESX 3.5

Database applications running on individual physical servers represent a large consolidation opportunity. However enterprises considering such consolidation want guidance as to how well databases scale using virtualization.

Database workloads are often provisioned on systems with much higher CPU and memory configurations than required by their average utilization. This is most commonly driven by the need for "headroom"—because database workloads are hard to migrate to a larger system, administrators often select a system much larger than immediately required in order to ensure that there will be sufficient capacity for peak portions of the workload and to allow for future growth. As a result, the utilization of dedicated database systems is often less than 20 percent of available CPU. For example, when studying over 2,000 four-way Oracle database instances, the average utilization was found to be 4.92 percent (from data captured by VMware Capacity Planner, June 2006). In addition, today's multisocket, multicore systems, with high memory capacity and bandwidth, multiple high-speed network connections, and numerous high-performance storage options provide computing resources that are difficult to fully utilize with a single database instance. As a result, there is ample opportunity to consolidate several database instances onto a single server.

The tests presented in this paper demonstrate that when scaling out (that is, increasing the number of virtual machines on one physical server) the performance of the Oracle database workloads in each virtual machine remain similar to the performance of the Oracle database workload in a single virtual machine on the same host while ESX CPU utilization scales in a nearly linear fashion. We tested with a 1GB database, which could be cached in the Oracle System Global Area, and with a 100GB database, which could not fit in the cache.

This paper covers the following topics:

ESX Support for Database Workloads

VMware ESX allows hardware to be partitioned, providing applications such as databases enough resources to keep utilization high while using the remaining resources for other workloads. Along with this resource partitioning, ESX provides the scalability required for database workloads. The following are some of the many features that make ESX ideal for consolidating database systems on a single computing platform:

- **High-performance I/O:** ESX can drive over 100,000 database I/O accesses per second—more than enough to accommodate the requirements of even the largest databases.

- **CPU scalability:** ESX can make full use of the increasingly large number of cores in today's high-performance servers and offers two types of CPU scalability:
  - Scaling out by supporting multiple virtual machines on a single physical host
  - Scaling up by supporting up to four virtual processors in each guest virtual machine
- **Memory scalability:** Oracle databases benefit greatly from large amounts of memory. ESX 3.5 allows each virtual machine to be configured with up to 64GB of memory. Because consolidation of workloads allows higher processor utilization, the average memory requirement per processor is higher—often about twice that of nonvirtualized systems. To accommodate these growing requirements, the memory scalability curve has been pushed considerably, with ESX 3.5 now supporting up to 256GB of physical memory.
- **Large pages:** Oracle databases have for some time used large memory pages in the CPU's memory management unit (MMU) to optimize memory performance. This large-page feature can be enabled in many operating systems. ESX 3.5 provides large-page support, allowing the database to fully utilize this feature.

ESX 3.5 also offers other benefits, such as:

- NUMA optimizations in which guest operating systems are preferentially allocated physical memory on the node where the guest is running
- Paravirtualization using virtual machine interface (VMI)
- Transparent page sharing
- The flexibility to add hardware resources based on demand rather than overcommitting resources during deployment

These and many other features enable ESX 3.5 to provide a perfect platform for consolidating physical servers running Oracle database.

Tests with 1GB Database

The first phase of our tests used a small database that could be cached in the Oracle System Global Area.

Experimental Setup for Tests with 1GB Database

The tests described in this paper were conducted using DVD Store database test suite version 2 (DS2) from Dell, Inc. DS2 is an online e-commerce test application with a back-end database component, a Web application layer, and driver programs. For more information about DS2, see http://www.delltechcenter.com/page/DVD+Store.

In the first phase of these tests, we wanted to demonstrate the CPU scale-out capabilities of ESX with each virtual machine running at 100 percent CPU utilization. To accomplish this, the Oracle database application had to read with near zero latency the data required to process user queries. Therefore, from among the three database sizes DS2 can be configured to use (10MB, 1GB, and 100GB), we selected the 1GB version. This allowed the entire database to be cached in the Oracle System Global Area (SGA), thus avoiding any I/O latency arising from reads and writes to the physical disks.

The server used was a Sun Fire X4600 M2 server with 16 CPU cores and 256GB of memory running VMware ESX 3.5. Between one and eight workload virtual machines, each with two virtual CPUs and 4GB of memory, were running 64-bit Red Hat Enterprise Linux 4, Update 4 and Oracle 10g R2 (10.2.0.1).

Load was generated by the Oracle 10g R2 32-bit Windows client on a system natively running Windows Server 2003 Release 2 Enterprise Edition with SP2.

For complete details of the test environment, see "Test Environment" on page 9.

Performance Measurement for Tests with 1GB Database

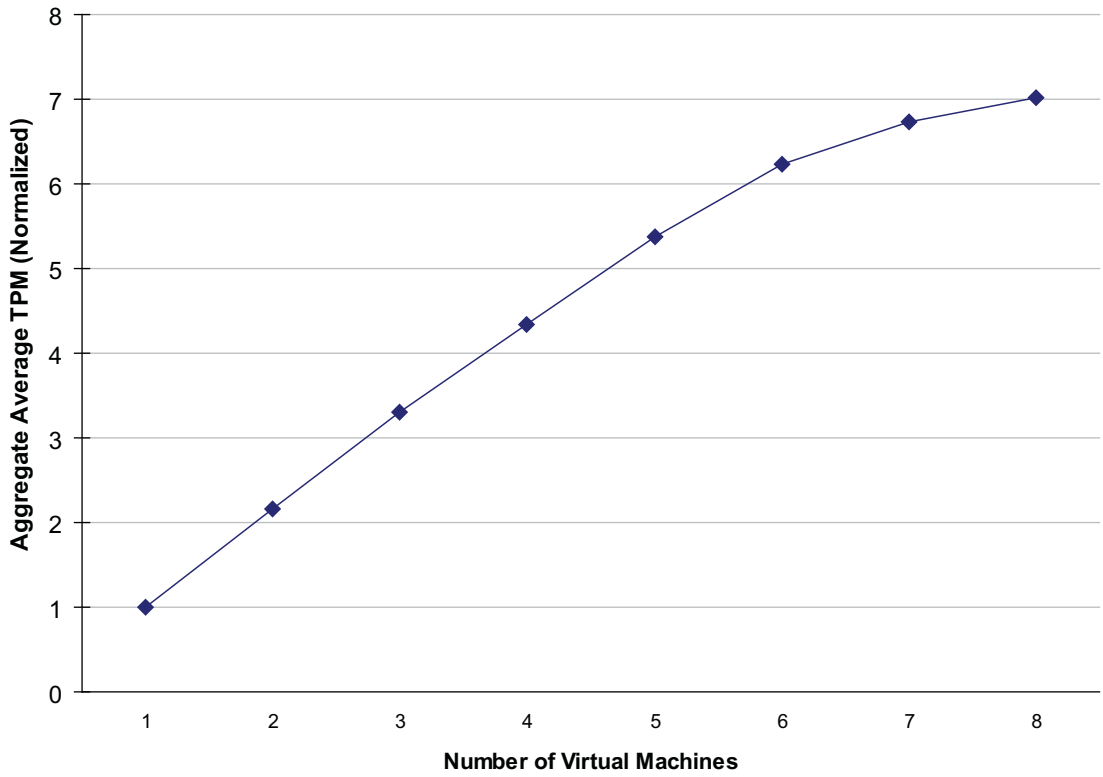
We used the performance of a single virtual machine configured with two virtual processors as the reference score. We then compared the performance of additional identically-configured virtual machines to this reference score as they were added to the ESX system. We ensured that the virtual CPUs of each of these virtual machines were fully saturated during the tests. Once they reached a steady state, we collected the transactions per minute and response times of the DS2 workload in each virtual machine as well as the average CPU utilization across all physical CPUs during a test cycle.

This allows us to show variations in transactions per minute (TPM) from the baseline performance as fractions of the reference score when additional load is added to the ESX system. These fractions are plotted as the normalized values shown in Figure 1 and Figure 2.

Scaling Performance in Tests with 1GB Database

Figure 1 shows the normalized aggregate transactions per minute of all the virtual machines powered on during a test cycle, while Figure 2 and Figure 3 show the individual performance of virtual machines during a test cycle. As seen from Figure 1, the aggregate TPM scales in a nearly linear fashion as virtual machines are added. The scaling tapers off only as the last virtual machines are added, because at that stage the resources (especially CPU) are nearing saturation.

Figure 1. Aggregate Average Transactions per Minute



In Figure 2 and Figure 3, for a given data point on the X axis the total number of vertical bars corresponds to the total number of virtual machines powered on during that test. For each data point the leftmost bar corresponds to the first virtual machine, the next bar to the second virtual machine, and so on. The bars in both graphs show the average of two runs, with the lines at the top of the vertical bars showing the minimum and maximum values.

As shown in Figure 2, the normalized values of the number of transactions per minute performed by each workload virtual machine as additional workload virtual machines are powered on compare favorably to the baseline performance.

Figure 2. Average Transactions per Minute

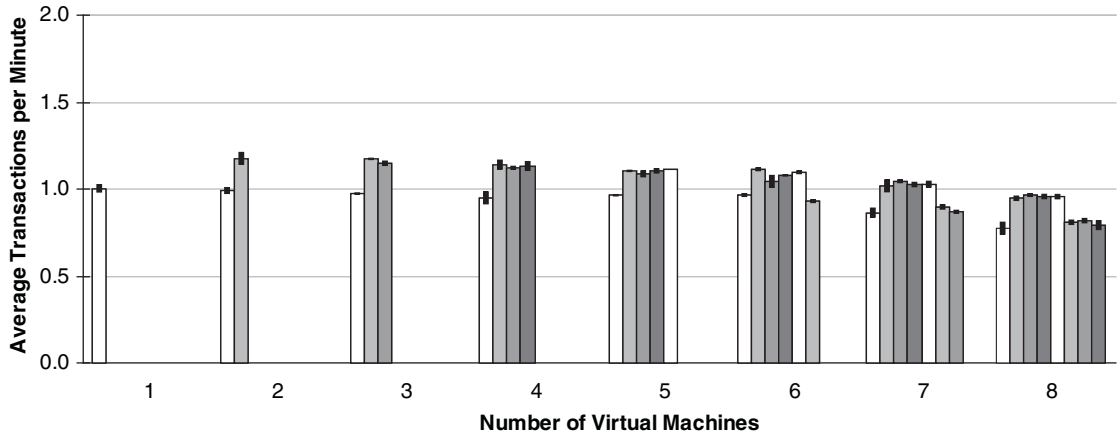
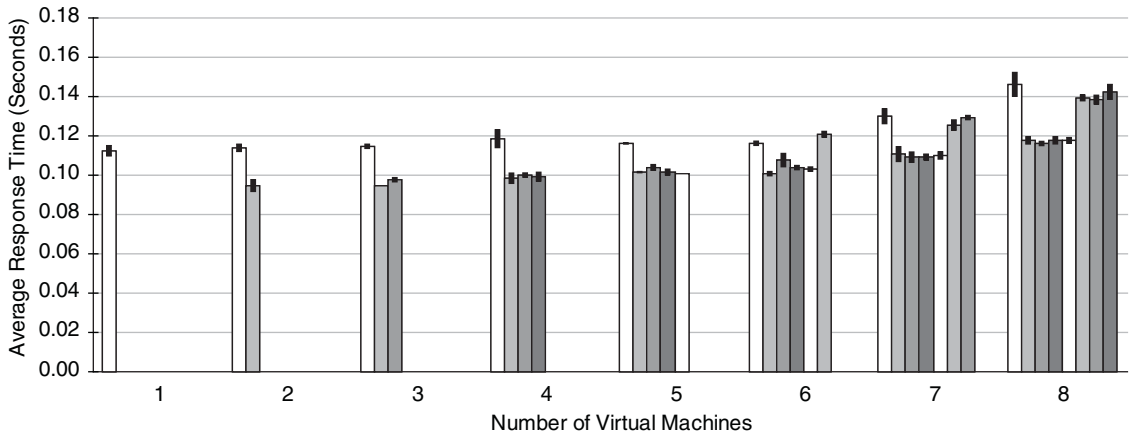


Figure 3 shows that as the number of virtual machines (and the corresponding load) on the ESX system increases, the database query response time shows only minimal degradation. With seven virtual machines, the response time of the lowest performing virtual machine is only 15 percent above the single virtual machine baseline, and even with eight virtual machines (and full CPU commitment), the response time does not increase by more than 21 percent. When it takes longer to process a transaction, the number of transactions per minute is lower. Therefore, virtual machines with higher transactions per minute in Figure 2 correspond to virtual machines with lower response time in Figure 3.

Figure 3. Average Response Time (Seconds)

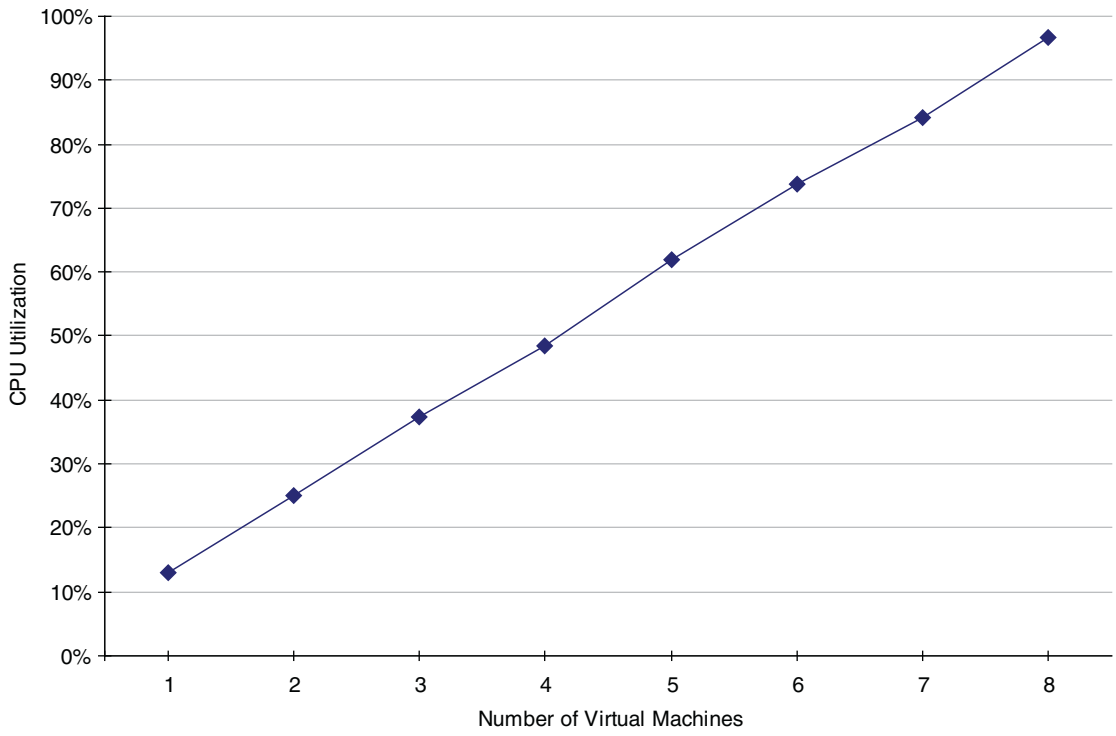


The performance of the virtual machines in these tests fell into two distinct groups, one group slightly higher, the other slightly lower. Each virtual machine's performance corresponded to the NUMA node on which that virtual machine was running, a variance believed to be caused by the architecture. In addition, the eight virtual machine test fully committed the 16 physical CPUs. This distributed the overhead of processing interrupts for disk and network I/O across all CPUs instead of restricting them to free CPUs, as was done when fewer virtual machines were running.

Our goal was to see how resource utilization scales as the number of virtual machines, and thus the overall load on the ESX host, increases. We also wanted to see the performance of the ESX scheduler when the CPU resources were fully committed. Figure 2 and Figure 3 show that ESX efficiently allocates available resources to virtual machines as the demand increases.

Figure 4 shows the average CPU utilization as seen by ESX. This includes the CPU cost of the virtual machines running the Oracle database as well as all virtualization overheads. The figure shows that as the number of virtual machines is increased there is a nearly linear scaling of CPU utilization on the ESX system, which indicates the efficiency of the ESX resource scheduler.

Figure 4. CPU Utilization



Tests with 100GB Database

The second phase of our tests used a larger database that could not be cached in the Oracle System Global Area.

Experimental Setup for Tests with 100GB Database

For the second phase of tests, we configured DS2 to use the 100GB database. This database was larger than the Oracle SGA, hence the data was not entirely cached during the test runs.

For the second phase of our tests, we used the same Sun Fire X4600 M2 server used in first phase. In this phase, each virtual machine was configured with 32GB of memory. As before, Red Hat Enterprise Linux 4, update 4 (64 bit) was installed as the guest operating system. Oracle 10g R2 (10.2.0.1) was installed as the database.

Each virtual machine was assigned seven LUNs. The database files associated with the DS2 benchmark were created on these LUNs. The complete storage layout of each virtual machine is detailed in Table 1.

Table 1. Storage Layout for DVD Store Data Files

Tablespaces	Number of Virtual Disks	Virtual Disk Size (GB)	Number of LUNs	RAID Type	Number of Physical Spindles in the RAID	RAID Group Number
System	1	6	1	0	5	0
Redo	1	5	1	0	5	0
Data	1	60	1	0	6	1
	1	60	1	0	6	2
	1	6	1	0	6	1 and 2
Index	1	30	1	0	6	3
	1	30	1	0	7	4

The tablespaces for the DS2 database were placed on separate virtual disks, which were hosted on separate LUNs as shown in Table 1. All the LUNs were assigned to virtual machines as raw device mapped (RDM) disks using physical mode. Each LUN was created on a RAID 0 disk group.

Despite the lack of redundancy, we used RAID 0 to maximize the disk throughput available for virtual machines running the Oracle 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.

For complete details of the test environment, see "Test Environment" on page 9.

Performance Measurement for Tests with 100GB Database

To compare the performance of ESX 3.5 to native performance, we used the performance of the DS2 database in a native environment as the reference score. We compared the performance of the DS2 database in a virtual machine with a configuration similar to the native configuration to this reference score. We ensured that the CPUs of both native and virtual machines were fully saturated. In both cases, to measure the performance, we collected the transactions per minute and average response time of each transaction in the DS2 workload at steady state.

In the scale-out study, we used the performance of a single virtual machine as the reference score to which we compared the performance of additional virtual machines with the same configuration as the first virtual machine when they were added to the ESX system. We ensured that the CPUs of each virtual machine were fully saturated during a test cycle. At steady state we collected the transactions per minute and response time of each transaction in the DS2 workload in each virtual machine that was powered on. Simultaneously, we measured the total I/O accesses per second for each virtual machine and the average CPU utilization across all physical CPUs using the ESX command line utility `esxtop`. For details on `esxtop`, see the appendix on performance monitoring utilities in the *Resource Management Guide* (http://www.vmware.com/pdf/vi3_35/esx_3/r35u2/vi3_35_25_u2_resource_mgmt.pdf).

We then calculated the aggregate transactions per minute, aggregate I/O accesses per second and average response time of all transactions across all virtual machines that were powered on during a test cycle. We then normalized the aggregate transactions per minute as a ratio of the reference score. All the values are plotted as graphs and explained in detail in the following sections.

Performance Comparison of Oracle Database in a Virtual Machine and in a Native Environment

To compare the performance of the Oracle database in a virtual machine on ESX 3.5 to that of the same database in a native environment, we installed Oracle 10g R2 (10.2.0.1) on native Red Hat Enterprise Linux 4 update 4 (64 bit) on the same Sun Fire x4600 M2 server on which we installed ESX. We configured the native operating system to use only two physical processors and 32GB of memory (the same as the virtual machine configuration). Figure 5 and Figure 6 compare the performance of Oracle 10g R2 in a virtual machine on ESX 3.5 to its performance in a native environment. In both cases, the CPU was saturated, running at 100 percent utilization. As the graphs show, the performance of Oracle 10g R2 in a virtual machine on ESX 3.5 is about 94 percent of native.

Figure 5. Normalized Transaction in Seconds (Higher is Better)

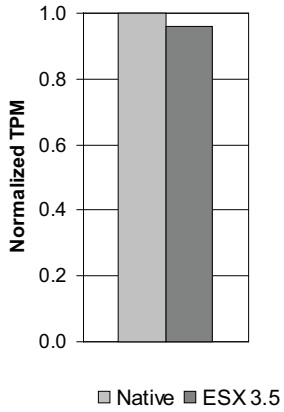
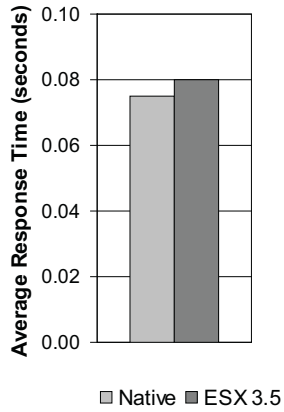


Figure 6. Average Response Time (Lower is Better)



Scaling Performance in Tests with 100GB Database

Figure 7 shows the normalized aggregate transactions per minute of all the virtual machines powered on during a test cycle. As shown in Figure 7, the aggregate transaction per minute increases in a nearly linear fashion as more virtual machines are added. Each virtual machine's performance directly depends on the performance of the NUMA node on which it runs. Performance of NUMA nodes on the Sun Fire X4600 M2 server differs slightly from one node to another, a variance believed to be caused by the architecture. The small dip in aggregate transactions per minute in Figure 7 (between three and four virtual machines in the graph) can be attributed to the variance in the NUMA node behavior.

Figure 7. Normalized Aggregate Transactions per Minute

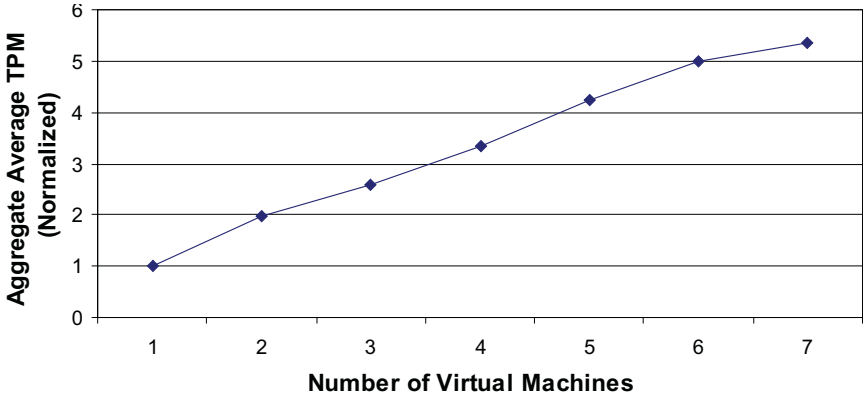


Figure 8 shows the average response time of each transaction across all virtual machines powered on during a test cycle in seconds. As shown in the graph, the response time increases from 80 to 100 milliseconds as the number of virtual machines increases from one to seven with all the virtual machines running at 100 percent CPU utilization.

Figure 8. Average Response Time (Seconds)

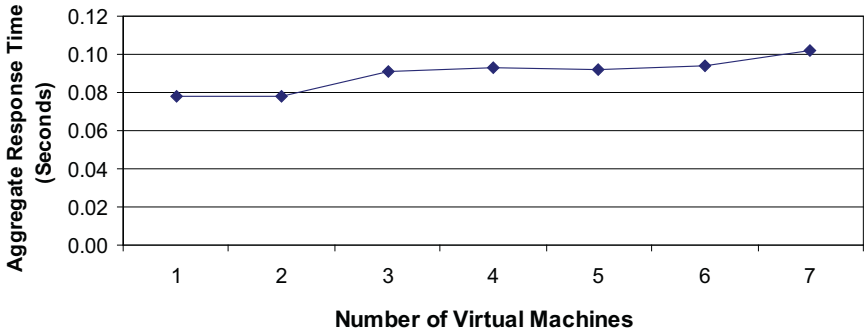


Figure 9 shows the average CPU utilization of all physical cores in the server. This CPU utilization includes the cost of virtual machines running the Oracle database as well as all related virtualization overhead. As shown in the graph, CPU utilization scales in nearly linear fashion as the number of virtual machines running on ESX increases, which indicates the effectiveness of the ESX resource scheduler.

Figure 9. CPU Utilization

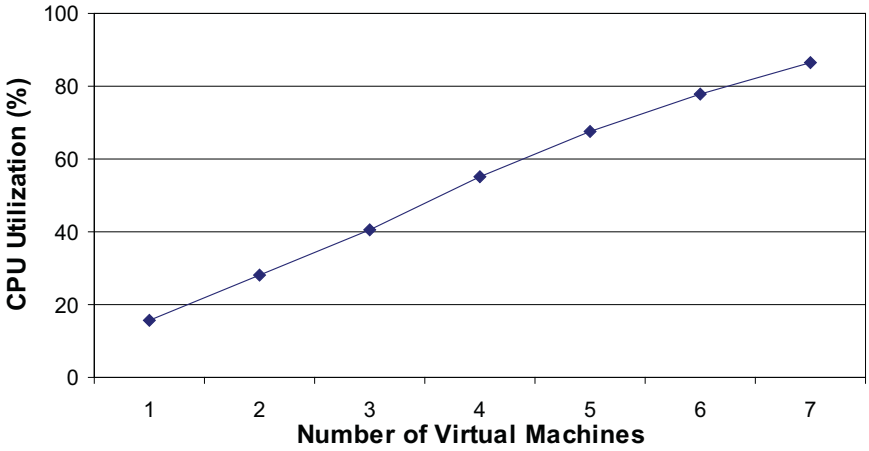
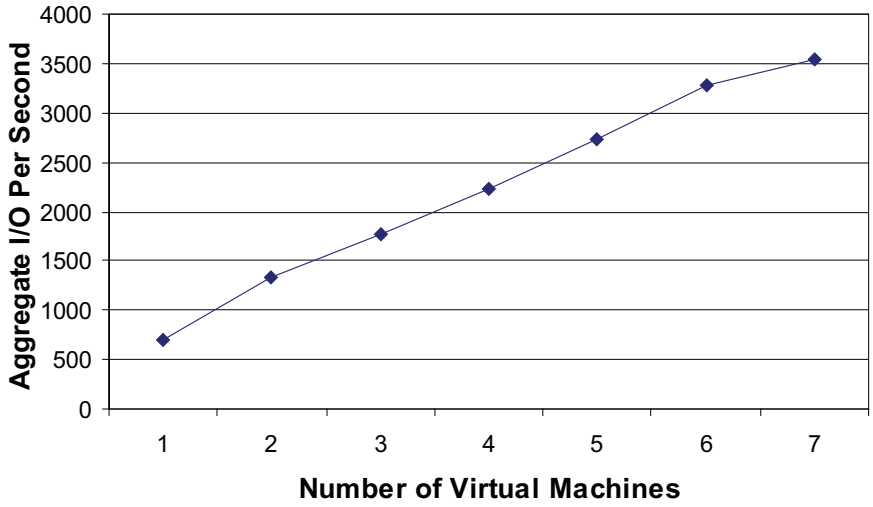


Figure 10 shows the aggregate I/O operations per second of all virtual machines powered on during a test cycle. This includes read operations from and write operations to data, index, redo, and system disks for each virtual machine. As shown in the graph, I/O operations scale almost linearly as the number of virtual machines increases. The total I/O access per second for a given virtual machine depends directly on the total transactions per minute, which in turn is affected by the performance of the NUMA node on which the virtual machine is running. This is validated by the behavior seen in Figure 10, which is consistent with that in Figure 7.

Figure 10. Aggregate I/O per Second



Test Environment

This section provides details about the hardware and software environment used to run these tests.

Server

Server Hardware

Sun Fire X4600 M2

Processors: Eight 2.8GHz AMD Opteron 8220 dual-core CPUs (16 cores total)
L2 cache: 1MB per core

Memory: 256GB

Internal storage: 2.5-inch SAS internal disk

Network interface controllers: Two onboard Intel 82546EB dual-port Gigabit Ethernet controllers (four ports total)

HBA: Two QLogic QLE2462 adapters (each dual-port, 4Gbps Fibre Channel)

Hypervisor

VMware ESX 3.5.0 Build 60217

Virtual Machine Configurations

Number of virtual machines: Seven

Virtual CPUs: Two per virtual machine

Memory for tests with 1GB database: 4GB
Memory for tests with 100GB database: 32GB

Virtual hard drive: 50GB for operating system and Oracle application

Guest operating system: Red Hat Enterprise Linux 4, Update 4, 64 bit
Operating system kernel: 2.6.9-42.ELlargesmp

Virtual NIC: 1

Guest network driver: Vmxnet

Application for Tests with 1GB Database

Database: Oracle 10g R2 (10.2.0.1) RHEL4 x86_64

SGA: 1.6GB

PGA: 555MB

Application for Tests with 100GB Database

Database: Oracle 10g R2 (10.2.0.1) RHEL4 x86_64

SGA: 29GB (was configured to use large pages)

PGA: 1GB

Storage

SAN Storage

EMC CLARiiON CX340 SAN; DPE and DAE with 15 disk drives each

Total disk drives: 30

Disk drive specification: 146GB 10K RPM disks

Read cache: 2GB (per SP)

Write cache: 1GB

Flare operating system revision: 03.24.040.5.011

Configuration per virtual machine for tests with 100GB database: Three LUNs for data, two LUNs for index, one LUN for redo, and one LUN for system

LUN Configuration Per Virtual Machine for Tests with 1GB Database

LUN size: One LUN, 6GB

LUN Configuration Per Virtual Machine for Tests with 100GB Database

Three LUNs for data, two LUNs for index, one LUN for redo, and one LUN for system

LUN sizes:

- Data: Two LUNs, 60GB each; one LUN, 6GB
- Index: Two LUNs, 30GB each
- Redo: One LUN, 5GB
- System: One LUN, 6GB

Fibre Channel Switch

EMC DS-8B2 (Brocade Silkworm 3200)

Fabric operating system: v5.0.1b

Link speed: 4Gb/second

Client

Hardware

Dell PowerEdge 2950

Processors: Two Intel Xeon 5100-series dual-core processors (four cores total)

Memory: 4GB

Network interface cards: Two Broadcom BCM5708C NetXtreme II Gigabit Ethernet adapters

Operating System

Windows Server 2003 Release 2 Enterprise Edition with SP2

Application

Oracle 10g R2 32-bit Windows client

Conclusion

In this paper we demonstrate that when running multiple virtual machines with Oracle database workloads on VMware ESX 3.5, the individual performance remains close to that of the Oracle database workload in a single virtual machine, while CPU utilization scales in a nearly linear fashion.

The first phase of our tests was conducted with a 1GB DS2 database that was cached in Oracle's SGA in each virtual machine. The physical CPUs on the server were fully saturated and I/O latencies arising from reads and writes to the physical disks were avoided. In the second phase we used a 100GB DS2 database. In both phases, as more virtual machines running Oracle database were added to the ESX host, performance of each virtual machine remained close to that of a single virtual machine. At saturation, CPU utilization was directly proportional to the number of virtual machines powered on. In the second phase we also observed that performance of the Oracle database in a virtual machine on ESX 3.5 was 94 percent of native.

The tests described in this paper confirm that ESX is very efficient in scaling overall resource utilization with a very minimal effect on performance. This scalability is one of the factors that make VMware ESX the perfect platform on which to consolidate demanding, mission-critical workloads such as Oracle databases.

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

Revision 20080806 Item: PS-064-PRD-02-01