VMware vSphere® 6 and Oracle Database Scalability Study

Scaling Monster Virtual Machines

TECHNICAL WHITE PAPER

Table of Contents

Executive Summary

Many customers run their production Oracle databases within VMware vSphere® virtual machines to support critical and demanding applications. The performance of these Oracle databases can be key to the success of daily operations. VMware conducted a series of tests to demonstrate the performance and scalability of large virtual machines on vSphere 6 running Oracle Database 12c. Using a transactional workload, it was found that a single large "Monster" virtual machine can efficiently use an entire current-generation four-socket server. These results demonstrate that even very large and demanding Oracle database workloads can be successfully run with excellent performance with vSphere.

Introduction

vSphere 6 introduces the ability to run virtual machines (VMs) ranging up to 128 virtual CPUs (vCPUs) and 4TB of RAM. This doubles the number of vCPUs supported from the previous version and increases the amount of RAM by four times. This new capability provides the potential for customers to run larger workloads than ever before in a virtual machine.

Oracle Database 12c is the latest release of Oracle's flagship product that has long been at the center of operations of organizations of all sizes. One of the most compelling features of Oracle Database has been its ability to scale to support applications with extremely large performance requirements.

Running Oracle Database in a vSphere VM has provided customers with the database performance they need in combination with the increased capabilities provided by vSphere including features such as vMotion, vSphere HA, and DRS. Using these features can improve the uptime and availability of Oracle databases while still achieving excellent performance. With vSphere 6, it is now possible to run Oracle databases that need to scale up to 128 CPUs while still taking advantage of all the other features and benefits of virtualization.

A series of tests were run with a virtual machine hosting an Oracle 12c database. A transactional workload was used to measure the performance of this large "Monster" VM on vSphere 6. The Oracle 12c database VM was scaled from 15 vCPUs all the way up to 120 vCPUs, and the maximum achieved throughput was measured.

The results show that performance scaled up well as the VM approached the maximum number of vCPUs that are supported with vSphere 6.

Test Environment

The tests were run on a Dell PowerEdge R920 with 4x Intel Xeon E7-4890 v2 (Ivy-Bridge-EX) and 1 TB of RAM. Each of these processors has 15 cores, bringing the total for the server to 60. With Hyper-Threading enabled, there are a total of 120 logical processors (or execution threads) in the server. Hyper-Threading was enabled for all tests conducted.

Storage was provided by an EMC VNX5800 configured with 30 SAS SSDs and 70 SAS HDDs. The storage was connected via dual QLogic 8Gb Fibre Channel adapters. Network connectivity was over dual 1Gb Intel NICs. An additional two servers were used to host virtual machines that were used as client driver systems. Figure 1 shows the test-bed setup.

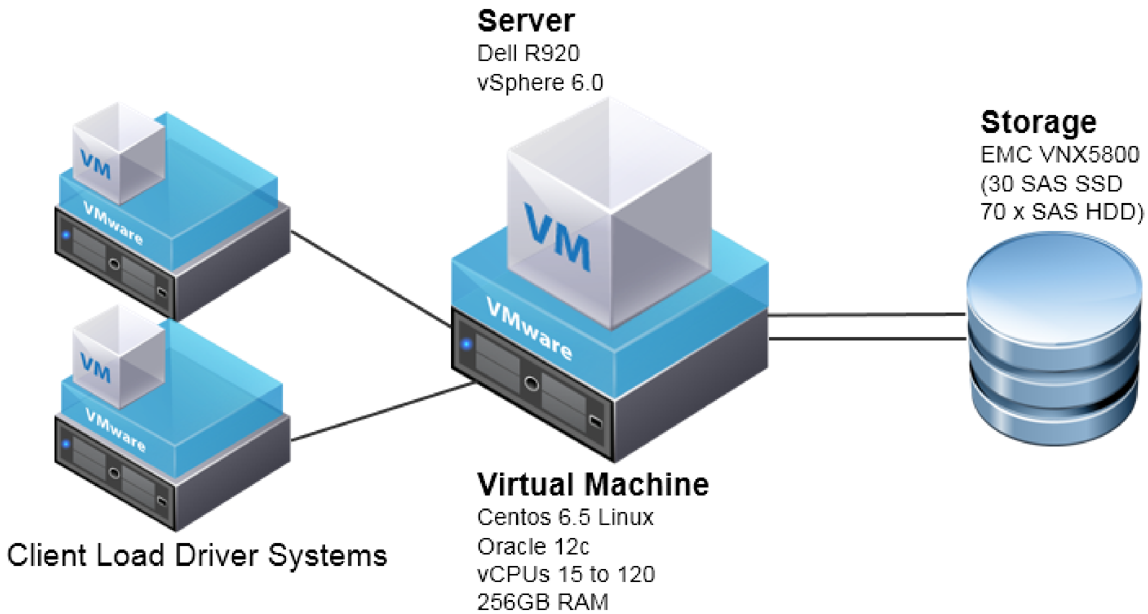

Figure 1. Test-bed setup

Virtual Machine Configuration

The virtual machine under test was configured with various numbers of CPUs for the tests, but the rest of the configuration was kept constant. The virtual machine had 256GB of virtual memory assigned, a vmxnet3 network adapter, and four PVSCSI virtual storage adapters assigned.

Storage for the testing was configured on the EMC VNX5800 to make maximum use of the high speed SSDs for the random access portion of the database. Four LUNs were created on the storage array that consisted of five SSDs each. These were assigned to be used for the data of the Oracle instances. Four additional RAID 1/0 LUNs were created using 4 x 15K RPM HDDs each. These LUNs were used for the sequential access logs of the database. The instances were spread across these LUNs evenly with two virtual disks created on each LUN. The virtual disks were connected to the VM by spreading them across the four PVSCSI virtual storage adapters evenly.

The virtual machine was installed with CentOS 6.5 Linux and Oracle 12c Database. The installation guide for Oracle 12c was followed [1] and the Oracle-provided RPM was used for installation. Linux was configured to use hugepages (large pages) and the I/O scheduler was set to no-op. The Oracle installation RPM disables NUMA for Linux using a kernel parameter, and this was the setting used for final tests after determining that it provided the best performance for these tests. It is recommended to try Oracle and Linux NUMA settings to see if they provide any performance advantage for your workload. In all tests, NUMA was enabled in the server BIOS at the hardware level and was available for vSphere to use.

Test Workload

The open source DVD Store 2.1 workload was used for testing [2]. It simulates an online store that allows users to log in, browse, and then purchase from a wide selection of DVDs. It makes use of many database features such as transactions, foreign keys, full text search, stored procedures, and triggers. The DVD Store package includes data creation programs, database loader scripts, database creation scripts, Web application layer, and client driver programs. The scripts allow for any size database to be specified and, for these tests, a 25GB database was used for each instance (a total of 200GB across all 8 instances).

In order to drive utilization and throughput up to the maximum levels on the largest VM tested, it was necessary to create eight Oracle database instances of DVD Store, which is a reflection of the DVD Store application limits and not Oracle database limits. The tests were run with all eight instances for all VM sizes to keep the tests consistent. Each instance represents a separate online store and has a complete set of database tables and a separate memory allocation for database caching. This separation allows for higher throughput overall on the system with more parallel operations occurring across all the stores at the same time.

Database Configuration

The database configuration was exactly the same across all eight instances. The memory used by each instance, called the System Global Area (SGA), was set to 25GB to match the size of the DVD Store database in these tests. This ensured that most of the queries would be satisfied by cache and trips to disk would be minimized. In order to support the high throughput of orders, four redo logs of 3GB each were configured for each instance.

A 25GB DVD Store database was created on each instance. Each instance had its own set of data and log files that were spread across the configured virtual disks and LUNs evenly. With four LUNs configured for data and four LUNs configured for logs, this meant that two instances shared each LUN.

A variety of different NUMA-related settings were tried to find the optimal performance of this workload. The Oracle 12c installation RPM automatically disables NUMA at the operating system level by adding a kernel parameter that is applied at system startup. Additionally the Oracle configuration parameter for NUMA support is set to off by default. Finally, the default setting in vSphere for virtual machines that are created with a vCPU size that exceeds the number of cores in a single NUMA node is set to enable virtual NUMA or virtual sockets. This means that the architecture reported to the VM includes NUMA information that appears to the guest operating system. All of these features were enabled and disabled to see which combination performed the best in combination with several vCPU pinning scenarios. It turned out that the default settings were best for this workload: NUMA disabled at the OS level, NUMA disabled in Oracle, and NUMA enabled in the server BIOS with virtual NUMA enabled for the virtual machine in vSphere.

Performance Metrics

The DVD Store reports performance in terms of orders per minute (OPM). Each order represents a user logging into the store, browsing for products several times, and then purchasing a number of products. The OPM performance metric is a measure of the amount of throughput that can be achieved.

Tests were run to achieve the maximum throughput possible for each VM vCPU configuration. The number of user threads running against the database was increased until the performance began to decline. This would occur when CPU usage was extremely high, at 98% utilization or higher. While these tests were focused on saturating the CPUs on the server, other aspects of performance were also monitored. Storage performance is always critical for databases and this was an area that was watched by ensuring that disk latency did not dramatically increase as the workload increased. It was observed that as disk I/Os per second (IOPS) increased from approximately 8000 on average for the baseline to an average of almost 19,000 for the highest measured test run. While the IOPS scaled up, the disk latency remained fairly constant and averaged about 0.6ms per operation.

Scale-Up Testing

The server used for testing had 15 cores per socket, for a total of 60 physical cores in the system. With Hyper-Threading enabled, the system has 2 execution threads per core, for a total of 120 threads for the system. In all tests Hyper-Threading was enabled on the server, but in configurations where 60 vCPUs or less are assigned to the VM, Hyper-Threads are not used by the VM. This is a result of the default scheduling policy where vCPUs are preferred to be scheduled on one thread per core before using the second thread of any cores. This first set of results, shown in Figure 2, is focused on the tests that scale up to 60 vCPUs to see the scaling for the virtual machine without the Hyper-Threads assigned to the VM.

The aggregate OPM achieved by the VM at each vCPU configuration shows that the amount of throughput increased at essentially the same rate that vCPUs were added. This indicated the workload was processor-bound, and that the VM was able to efficiently use the additional processing power, as the performance results increased almost linearly.

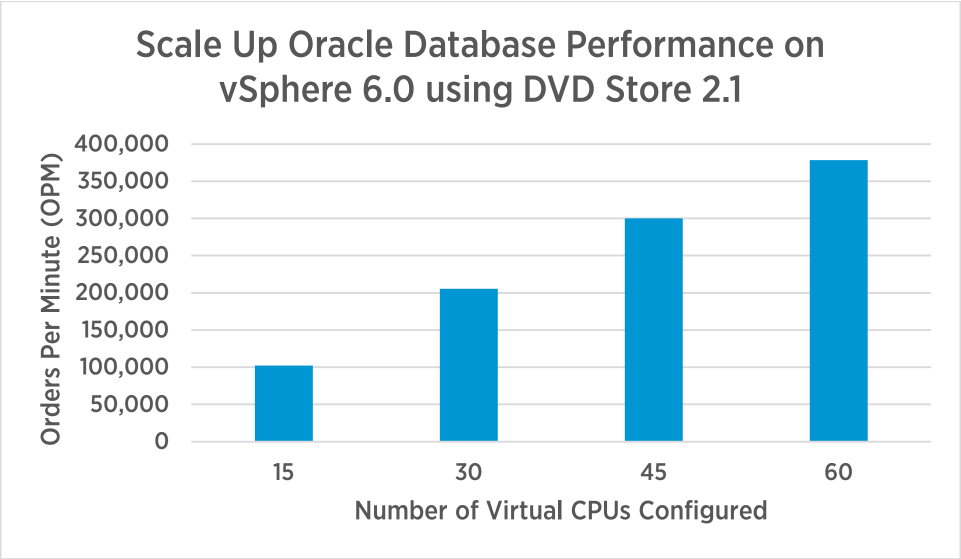

Figure 2. Scale-up test results up to 60 vCPUs

Scale-Up Testing with Hyper-Threads

While vSphere 6 supports up to 128 vCPUs per VM, these tests were limited to 120 vCPUs due to the number of threads available on the server. Additionally, the large VM created using 120 vCPUs must use both hardware execution threads (Hyper-Threads) on all the processor cores in order to be reach 120 vCPUs. In this case, there is one vCPU per execution thread.

Hyper-Threading doubles the number of execution threads on a core, but it does not double that core's performance. In order to measure the scale-up performance of the 120-vCPU VM, a 60-vCPU VM was configured with CPU affinity so that it was limited to only two of the server's four sockets. In this configuration, the 60-vCPU VM is the same as the 120-vCPU VM, as one vCPU was available per execution thread. By configuring a 60-vCPU VM in this way, it makes it easy to see the scale up performance at 120 vCPUs on this server where hyper-threads are enabled.

The results of the scale-up testing using the 60-vCPU VM configured with CPU affinity to only 2 sockets and the 120-vCPU VM using all four sockets showed near linear scaling, as shown in Figure 3.

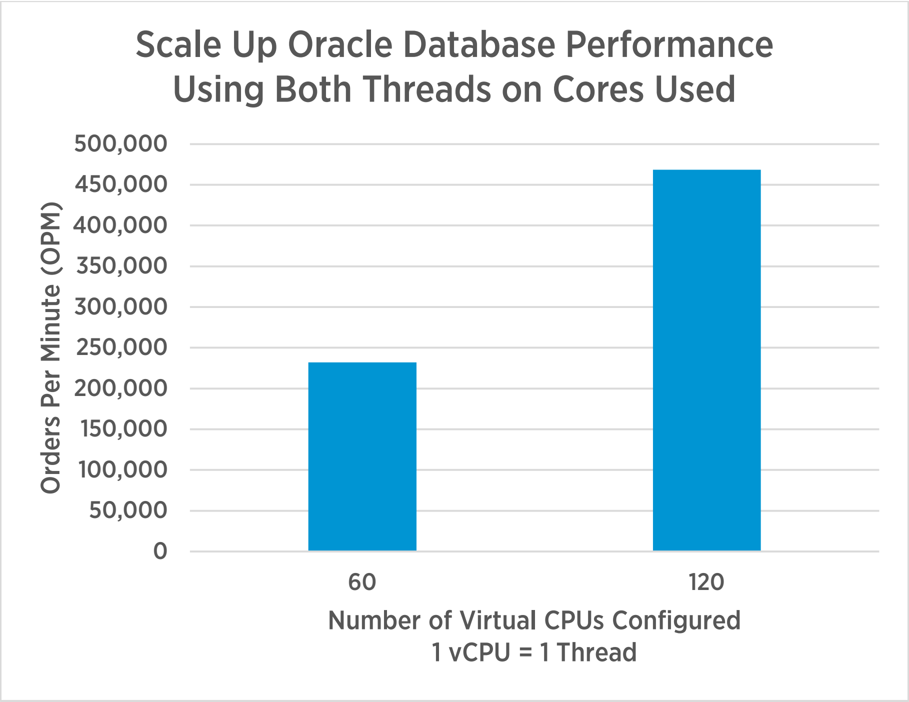

Figure 3. Scale up test results up to 120 vCPUs using one Hyper-Thread per vCPU

Hyper-Threading Performance Tests

As mentioned earlier, while Hyper-Threading does double the number of execution threads per core, it does not double performance. Using the results from these tests it is possible to measure the benefit obtained from Hyper-Threading. The 60-vCPU result from the first test, which is using all 60 cores without Hyper-Threading, is compared with the 120-vCPU result from the second test which uses all 60 cores with the Hyper-Threads in Figure 4. While it is estimated that Hyper-Threading usually provides between a 10 to 30 percent performance increase, the actual amount will depend on the workload. In the case of these tests, with a transactional workload running on Oracle Database, it was found that the benefit from Hyper-Threading was 24 percent.

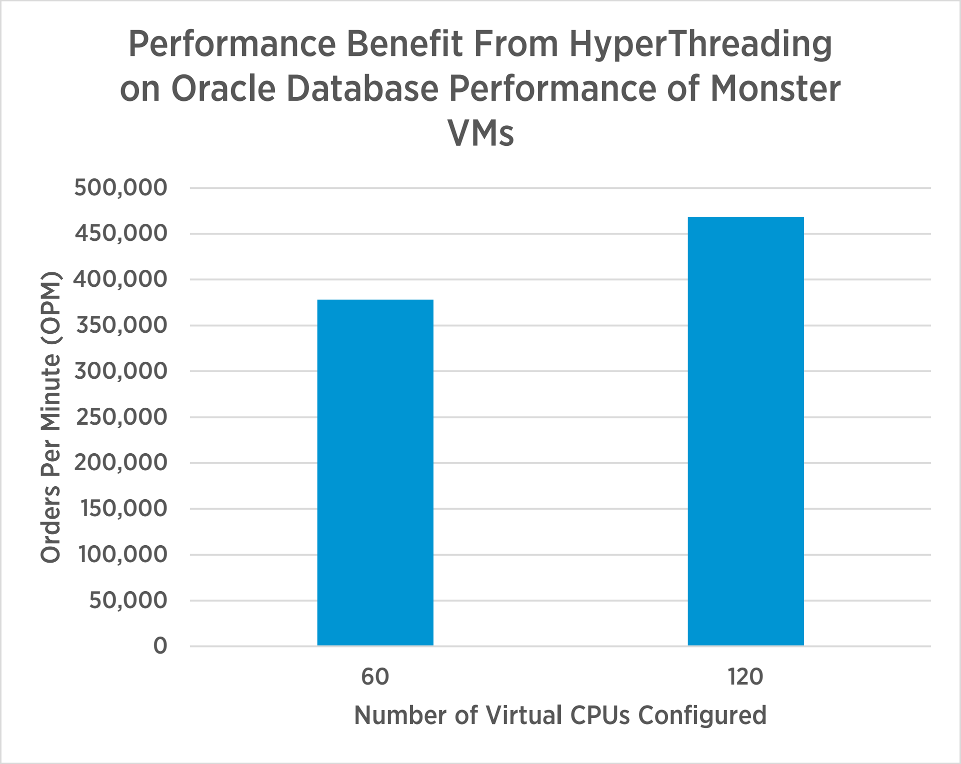

Figure 4. Performance benefit from Hyper-Threading

Complete Performance Test Results

Another way of looking at the results is to view them all from the same baseline. In this case, the baseline is the performance achieved at the smallest size of 15 vCPUs. The linear performance improvements as additional vCPUs are added up to 60 are clear with improvements of 2.0x, 2.9x, and 3.7x as vCPU counts were increased to 30, 45, and 60 vCPUs respectively. The additional performance from Hyper-Threads shows a gain from 60 to 120 vCPUs with an increase from 3.7x to 4.6x. Finally, the pinned 60 vCPUs on 30 physical cores result shows the linear scalability going up to 120 vCPUs, where the increase in performance goes from 2.3x to 4.6x. The complete test results are listed in Table 1.

NUMBER OF VCPUS	ORDERS PER MINUTE (OPM)	PERFORMANCE IMPROVEMENT FROM 15 VCPUS
15 vCPUs	102108	1x - Baseline
30 vCPUs	205642	2.0x
45 vCPUs	299933	2.9x
60 vCPUs	378269	3.7x
120 vCPUs	468348	4.6x
60 vCPUs (pinned to 30 cores on 2 processors)	231917	2.3x

Table 1. Complete test results

Conclusion

The new larger “Monster” VM support in vSphere 6 allows for virtual machines that can support larger workloads than ever before with excellent performance. These tests show that large virtual machines running on vSphere 6 can scale up as needed to meet extreme performance levels.

Appendix A

The following Table 2 describes the hardware and software used in the test bed.

VIRTUALIZATION PLATFORM	
Hypervisor Vendor, Product, Version	VMware ESXi 6.0
SERVER PLATFORM	
Server Manufacturer/Model	Dell PowerEdge R920
Processor Vendor/Model	Intel® Xeon® E7-4890 v2
Processor Speed	2.8 GHz (Intel Turbo Boost Technology up to 3.4 GHz)
Total Sockets/Total Cores/Total Threads	4 Sockets / 60 Cores / 120 Threads

L3 Cache	37.5MB
BIOS Version	1.2.2 05/05/2014
Memory Size (in GB, Number of DIMMs)	1024 GB, 32 x 32 GB
Memory Type and Speed	Quad rank x4 PC3-12800 (LRDIMM-1600)
Disk Subsystem Type	FC SAN
Number of Disk Controllers	1
Disk Controller Vendor and Model	Dell PowerEdge Raid Controller (PERC) H730P
Number of Host Bus Adapters	1 dual-port
Host Bus Adapter Vendors and Models	QLogic 2532 dual port 8Gb Fibre Channel HBA
STORAGE	
Array Vendor, Model, and Firmware Version	EMC VNX 5800 (version 05.33.0.5.038)
Array Cache Size	32GB per SP
Total Number of Physical Disks Used	100 (70 HDDs, 30 SSDs)
Total Number of Enclosures/Pods/Shelves Used	4
Disks	30 x SAS SSD 70 x SAS HDD
NETWORK	
Network Speed	1Gbps
Network Controller Vendors/Models	1 x Intel I350 1GbE Quad-Port
DATABASE VM INFORMATION	
Guest Operating System	CentOS 6.5
Database Application	Oracle Database 12c
Virtual Processors	15, 30, 45, 60, 120 vCPUs
Virtual RAM	256 GB
Virtual SCSI	4 x PVSCSI
Virtual NIC	1 x VMXNET3
Benchmarking Software	DVD Store 2.1

Table 2. Hardware and software

References

[1] Oracle Corporation. (2015, February) Oracle Database Installation Guide.
https://docs.oracle.com/database/121/LADBI/toc.htm

[2] GitHub, Inc. (2015, February) DVD Store Open Source Database Workload.
http://www.github.com/dvdstore

About the Author

Todd Muirhead is a staff engineer on the Performance Engineering team at VMware. He is also the co-creator of the open source DVD Store benchmark which has been used widely in the industry to measure and compare database performance.

Acknowledgements

The author would like to thank: David Morse and Bruce Herndon for valuable discussions around database performance and test scenarios and configurations; Sreekanth Setty, Jeff Buell, and Hal Rosenberg for feedback and advice on the testing and white paper review.

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