ORACLE DATABASE PERFORMANCE: VMWARE CLOUD ON AWS

vmware®

Table of Contents

Executive Summary

VMware Cloud™ on AWS lets customers easily run their applications in virtual machines on Amazon's cloud infrastructure.

Customers have successfully run their business-critical applications with high performance demands on VMware vSphere for many years. It is now possible to do the same thing on AWS infrastructure in the cloud. This allows customers to deploy a complete vSphere environment, ready in hours; including servers, storage, networking, monitoring, and administration. The vSphere environment on AWS is deployed on physical infrastructure, which means that performance is essentially the same as that of a similar environment setup in an onsite datacenter.

This paper examines the performance of Oracle databases on the VMware Cloud on AWS environment and finds that it has excellent performance to support a large number of high performing Oracle databases.

Introduction

A complete introduction to VMware Cloud on AWS can be found in many places. There are white papers, blogs, videos, and presentations available from VMware, Amazon, and others. A great place to find many of these resources is on the VMware Cloud on AWS Resources [1] page.

Summarized, VMware Cloud on AWS is a full VMware vSphere cloud that is deployed, ready, and running on Amazon Web Services cloud infrastructure. It runs directly on the servers with no additional layers between vSphere and the bare-metal system. It is automatically deployed and configured with the current version of vSphere, using vSAN for storage, and NSX for networking. It is a hybrid cloud solution that can be connected to your onsite datacenter. Existing VMs can be migrated via vMotion into the new VMware Cloud on AWS over this connection, which means you can easily add more capacity to your existing vSphere cluster.

Amazon Web Services (AWS) is a market leader in cloud computing and provides an incredibly easy and attractive way to quickly build out new capacity and new applications. The high availability and easy ability to scale resources at AWS provides many customers with great options for running their applications that were not possible in the past.

Oracle databases have long been a key part of many enterprises. These databases can be used to scale to meet almost any performance requirements in the market. Oracle's longtime position as a leader of database software and partnerships with many other companies in the industry have resulted in an exceptional track record for supporting business-critical applications. As a result, many customers use Oracle Database software to support crucial business functions that require the highest levels of availability and performance.

The combination of VMware vSphere with AWS provides a robust cloud environment for running business-critical applications like Oracle databases, and this provides you with new options for supporting your business needs.

Test Environment

For our tests, we deployed a four server software-defined data center (SDDC) from our VMware Cloud on AWS portal.

The servers of the SDDC used Intel Xeon E5 2686 v4 processors running at 2.30 GHz with 512 GB of RAM. Each processor had 18 cores and 36 logical threads with hyperthreading enabled (Figure 1).
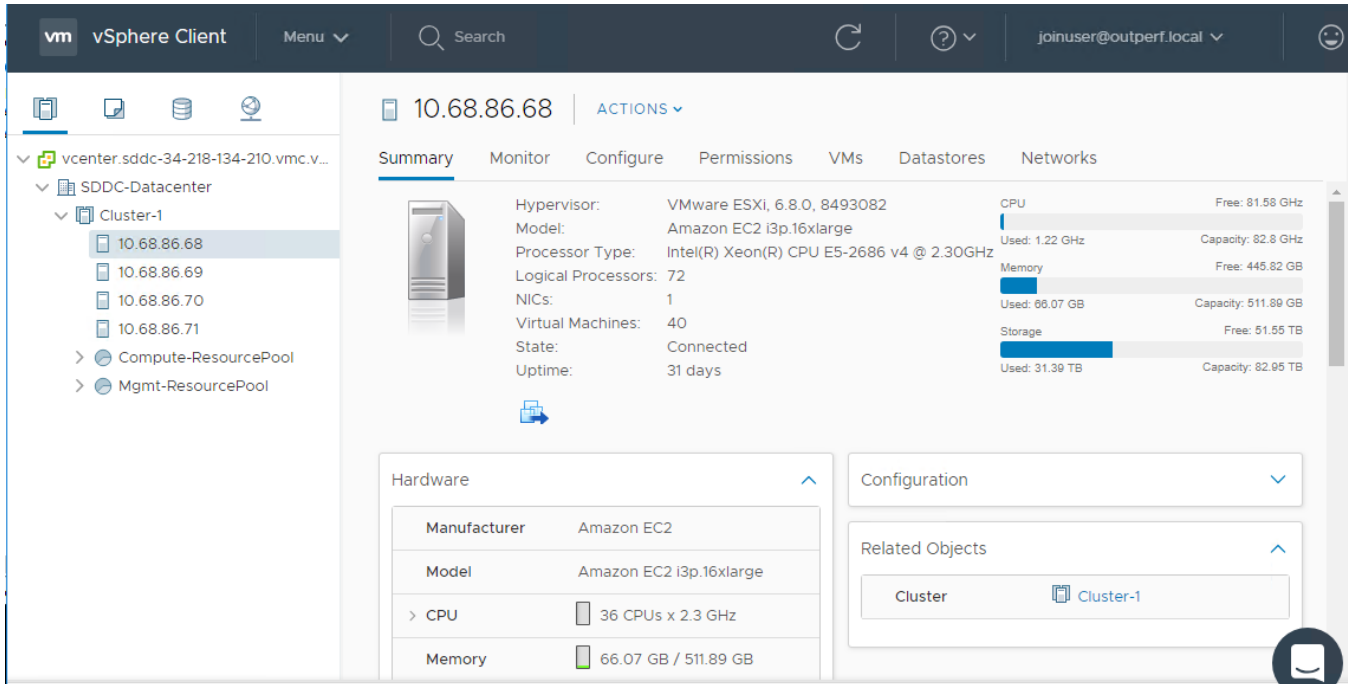

Figure 1. Screenshot of the test environment

On VMware Cloud for AWS, vSAN is used for the storage. It used the 8 NVMe devices that were in each of the four servers. Each NVMe had a capacity of 1.7 TB, so the entire cluster had 32 NVMe devices and a total raw capacity of approximately 40 TB (Figure 2). Deduplication and compression features of vSAN are enabled on VMware Cloud on AWS which will provide additional effective capacity. As part of the automatic deployment, management and workload datastores are created. The management datastore is for things like the vCenter and NSX-related VMs. The workload datastore is where all the VMs that are created or migrated into the environment are stored. This is where all the Oracle database VMs that were created for these tests were stored.
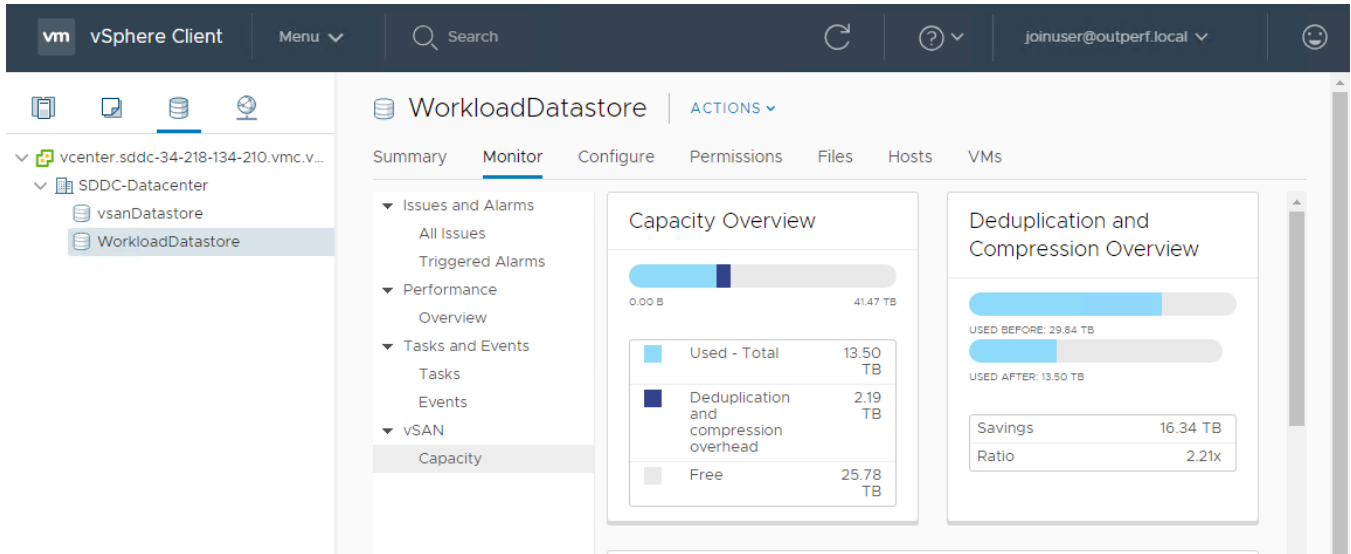
Figure 2. Screenshot of workload datastore configuration

Figure 3 shows a four node SDDC on AWS and its major components. We configured only the VMs that comprised the Oracle database and load driver. The rest of the components and configuration were done automatically as part of the deployment from the VMware Cloud on AWS portal.
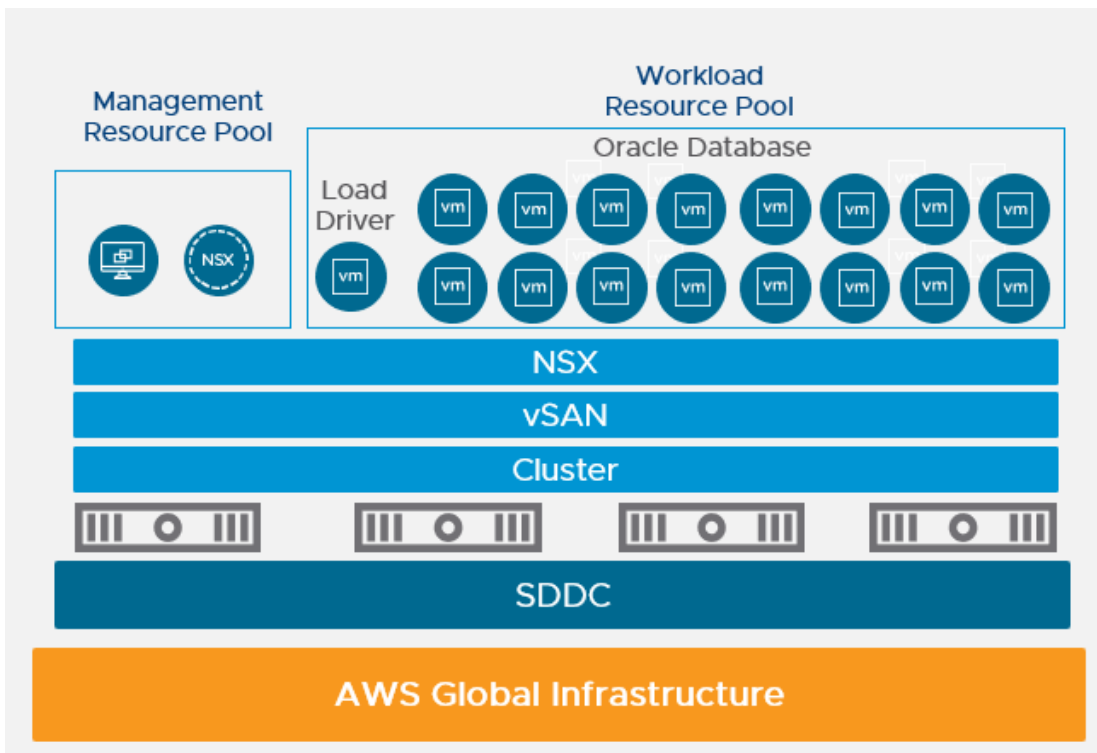

Figure 3. Cloud testbed configuration

Test Workload

We used the workload from the open-source benchmark DVD Store 3 [2]. DVD Store simulates an online store that allows customers to log on, search for DVDs, read reviews from other customers, rate the helpfulness of reviews, and purchase DVDs. DVD Store uses many common database features to support these operations, including tables, indexes, triggers, stored procedures, transactions, foreign keys, simple queries, and complex multi-join queries. The benchmark includes a client program that generates load against the database by simulating the actions of users on the site. Each simulated user thread does the full set of steps for a user from logon through purchase. The workload reports throughput in orders per minute (OPM).

For each test configuration, the number of user threads was increased with each iteration until the throughput reached its maximum level. We observed the maximum throughput corresponded to a host CPU utilization of between 90 and 99 percent.

DVD Store 3 allows for any size database to be created. For these tests, a database with approximately 140 GB of on-disk size was used.

Virtual Machine Configuration

We kept the configuration of the VMs constant except for the number of vCPUs, which we varied depending on the specific test. The VMs used the VMXNET3 virtual network adapter and paravirtual SCSI (PVSCSI) adapters. We assigned data and log virtual disks to separate PVSCSI adapters. We also configured the memory to be 128 GB.

We installed the VM with Red Hat Enterprise Linux (RHEL) 7.2 as the operating system and Oracle Database 12.0.1.2 using the Oracle-provided installer RPM. We followed Oracle best practices documentation for this installation. We configured huge pages for RHEL and enabled them in Oracle, and we set the Oracle sga_target parameter to 100 GB. This ensured that a large amount of the 140 GB database would be available in the buffer cache, but not all, so that some disk accesses would occur during the tests.

Scale-Up Performance

Scale-up performance testing entails testing to see how a single VM performs from small configurations up to larger and larger configurations. We performed scale-up tests with a VM configured with 8 vCPUs, up to 64 vCPUs. The DVD Store driver, which ran on a separate VM with 18 vCPUs, generated the load. The load-driver VM and the Oracle database VM both ran in the VMware Cloud on AWS, but were on different physical hosts in that cloud.

The results show good scaling from about 30,000 OPM with an 8 vCPU VM to just over 101,000 OPM with a 32 vCPU VM (Figure 4).

VMware Cloud on AWS Scale-Up Performance: Oracle on RHEL Running DVD Store 3
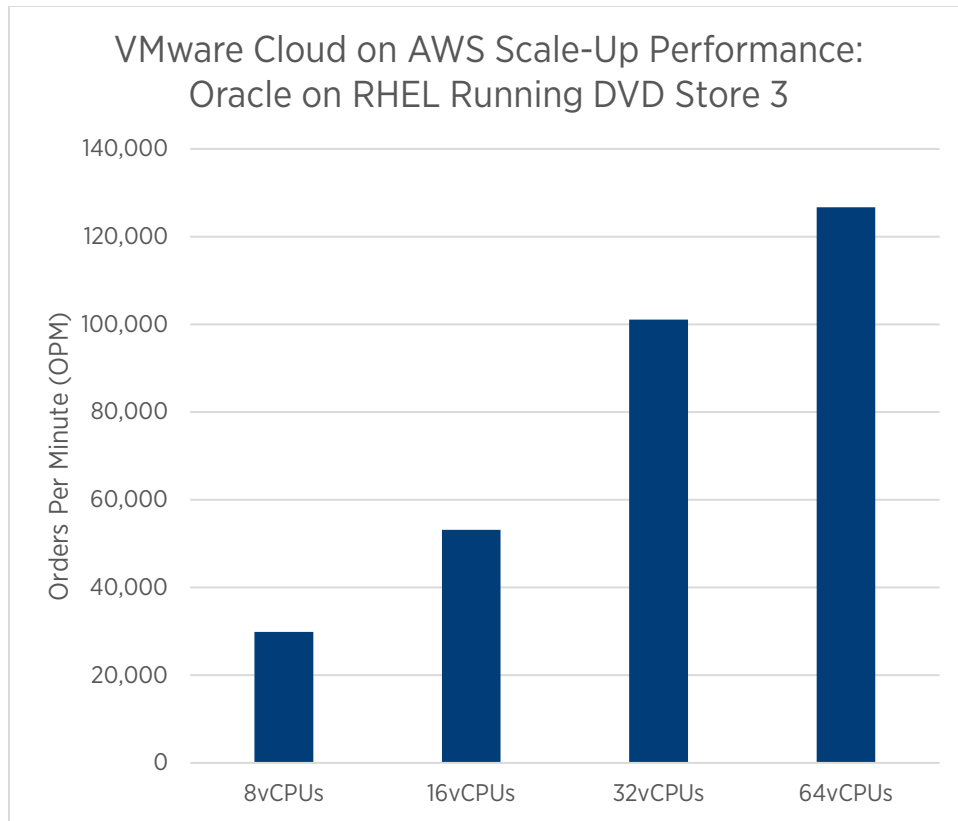
Figure 4. Oracle DB VM scale-up performance (adding processors)

The physical servers in this cloud were two-socket servers with 18 cores per socket, resulting in 36 cores total. Once a VM in this environment exceeded 36 vCPUs, it began to use hyperthreads for some of its vCPUs. Because hyperthreads are two logical threads running on a single physical core, the performance gain is usually about 15 to 30 percent. In this case, the performance scaling from 32 vCPUs to 64 vCPUs was in that range at about 25 percent. This is because almost all of the additional vCPUs were supported through the use of hyperthreads.

Scale-Out Performance

Another way to measure performance is with multiple VMs running at the same time, which is known as scale-out performance. The setup is the same as the previous single VM scale-up tests, but the workload driver system now spreads the worker threads across a number of target VMs simultaneously. The database VMs are spread out across the cluster as well, based on the best load balancing as determined by vSphere's Distributed Resource Scheduling (DRS). The total number of OPM achieved across all the Oracle Database VMs is then reported for each set of VMs.

Currently, VMware Cloud on AWS supports up to 30 hosts per cluster and up to 10 clusters per SDDC. In these tests, we used one cluster with only 4 hosts in our SDDC. The performance of the scale-out tests is largely due to the number of hosts. If more hosts were added to the SDDC, then more VMs could be run and the total amount of throughput achieved would be higher. An SDDC with more physical hosts would be capable of achieving higher scale-out performance than what we measured with the 4 host SDDC for these tests.

8 vCPU VMs

We performed the first set of tests with 8 vCPU VMs. We increased the number of VMs up to the maximum that successfully ran, which, in this case, was 14.

The scaling, as measured in throughput, from 1 to 14 VMs is good, going from 30,000 to 307,000 OPM (Figure 5).
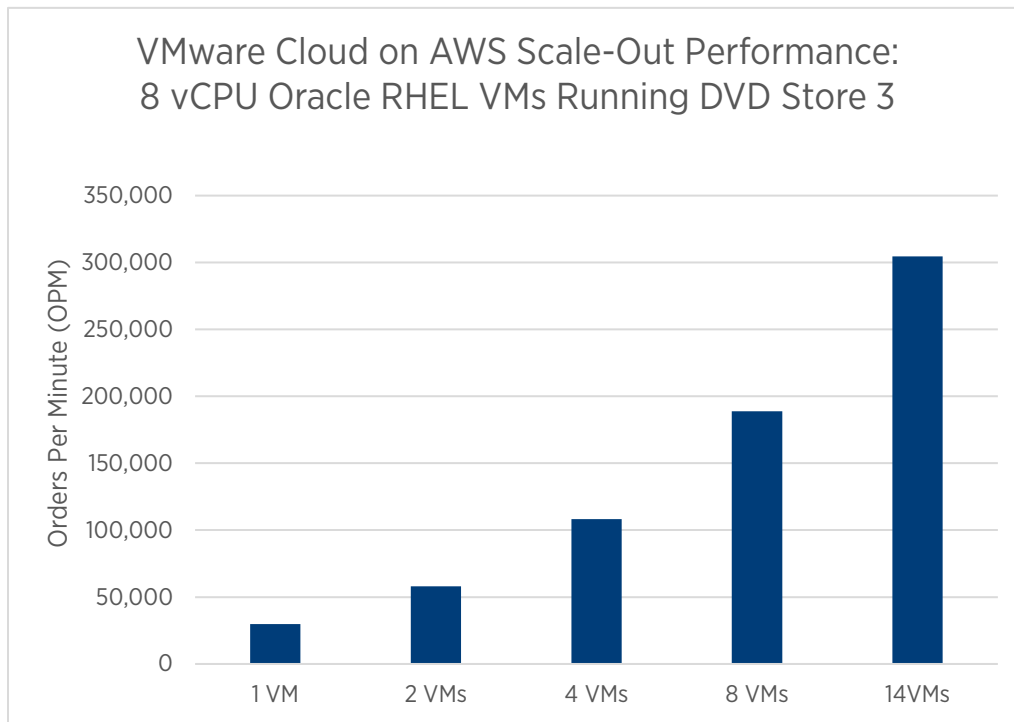

Figure 5. Scale-out performance with 8 vCPU VMs

As in all of the tests in this paper, each VM had 128 GB of memory, and the database had 140 GB on disk. This means that each VM actively used most of its assigned memory for database caching. Each host in the four-node SDDC had 512 GB of RAM for a total of 2 TB across the cloud. With 14 Oracle database VMs actively running, the memory assigned to them was 1,792 GB. While it was possible to get 16 VMs powered on for a test, they required all 2 TB of RAM on the cloud, which did not leave any room for the load-driver VM or any of the other cluster-related VMs like vCenter or NSX controllers. This is why, in the tests, we limited the 8 vCPU case to 14 VMs.

16 vCPU VMs

The next test case used 16 vCPU VMs in a scale-out test scenario. Each processor in the physical servers that were used for the SDDC had 18 cores. The 16 vCPU test case dealt with VMs that were just two cores smaller than the NUMA node size of the physical server.

In this round of tests, the scaling remains strong up through 12 VMs, but it begins to fall off with 13 (Figure 6). While the previous test with 8 vCPU VMs was limited by memory, this test was limited by CPU.

VMware Cloud on AWS Scale-Out Performance: 16 vCPU Oracle RHEL VMs Running DVD Store 3
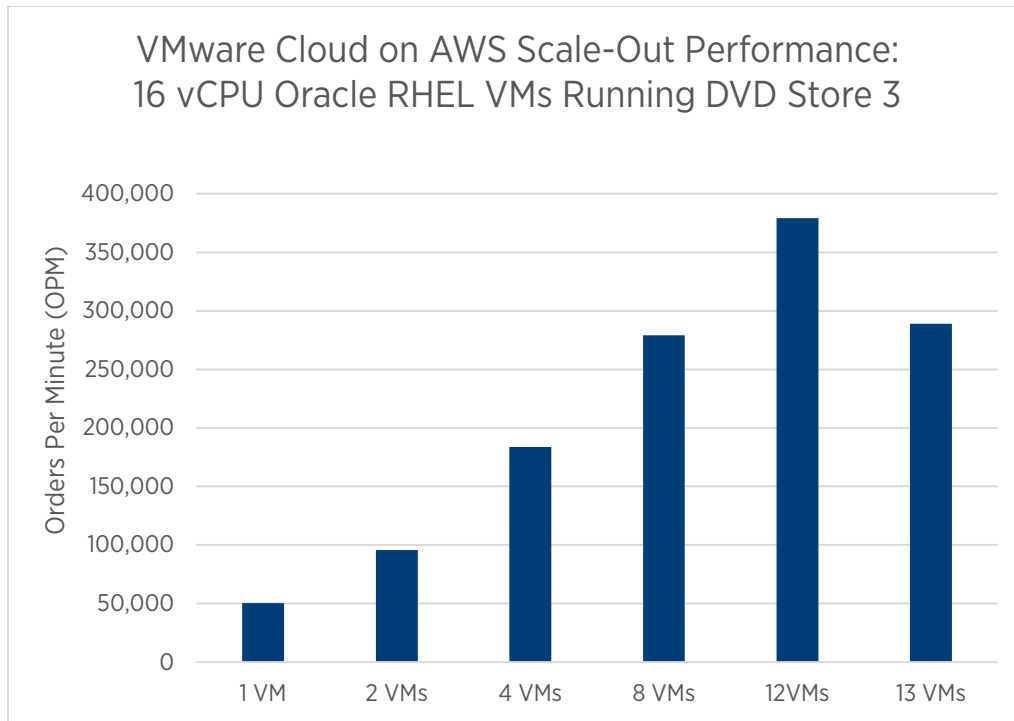
Figure 6. Scale-out performance with 16 vCPU Oracle/RHEL VMs running DVD Store 3

Each host had 36 cores, with the four-node SDDC having a total of 144 cores. With twelve 16-vCPU VMs, there were 192 vCPUs allocated, which means that the cloud had 48 more vCPUs allocated than it had physical cores.

Regardless, performance still scaled well because of hyperthreading, which gave the system a total of 288 threads. At 13 VMs, the total vCPUs reach 212 and performance degraded a bit, but all VMs continue to achieve significant throughput under this load. But at 14 VMs, the tests did not complete successfully. The load driver VM and all of the database VMs reach a total of 244 vCPUs assigned, and not all database VMs were able to get enough resources to complete queries in a timely manner.

18 vCPU VMs

The final test was with 18-vCPU VMs that we sized to exactly match the number of physical cores in each processor. The results are shown in Figure 7.
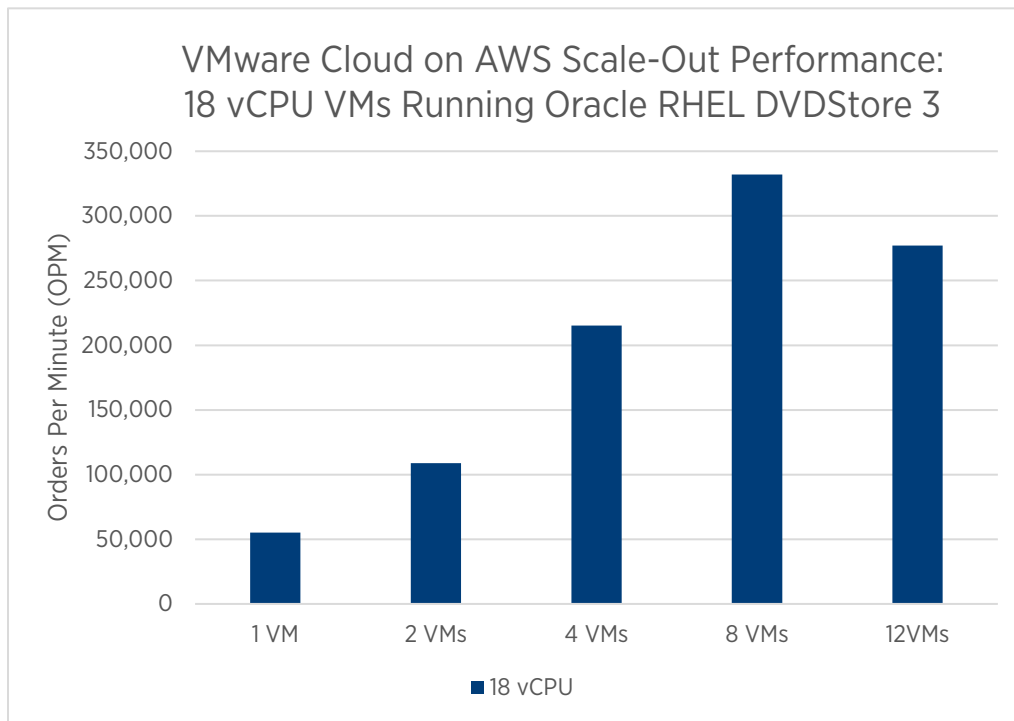

Figure 7. 18 vCPU RHEL VM scale-out performance running Oracle and DVD Store 3 workload

The 18-vCPU VMs are advantaged in the lower number of VMs tests because the system is not overcommitted, and it is sized to exactly use the full resources of single socket. Each VM has more resources than the 16-vCPU VMs, and it puts them to good use and achieves performance that is better than the 16 vCPU VMs at 1, 2, 4, and 8 VMs. But at 12 VMs, its performance is lower than the 16-vCPU VMs. That extra 2 vCPUs per socket are available to be used by the load driver VM and all of the vSphere infrastructure including vSAN, NSX, and vCenter.

Combined Scale-Out Results

Figure 8 is a combination of the results presented in the previous section to allow for easy side-by-side comparison.

The test cases with 8 VMs or less show that more throughput is achieved in proportion to the number of vCPUs. In other words, the bigger the VM, the more throughput that was achieved. There are enough physical cores in all these test cases to essentially provide each database VM with plenty of CPU resources. There are 8 database VMs and 8 physical processors with 18 cores, each, in the cloud. The 18-vCPU VMs provide the best performance in all of the test cases with up to 8 VMs.
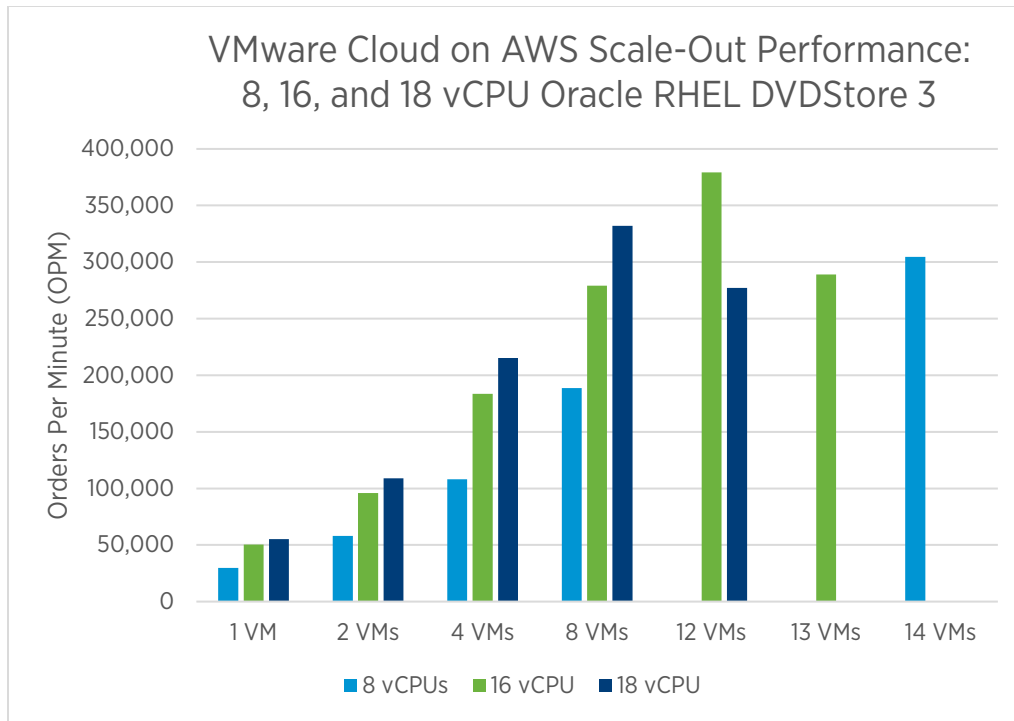
Figure 8. VMware Cloud on AWS scale-out performance with 8, 16, and 18 vCPU

The test cases with more than 8 VMs (the right side of the chart) are interesting. These are the cases where resources are more scarce and where either CPU or memory constraints come into play in the performance results. At the 12 VM count, the 16 vCPU VM test case achieves the highest performance of all the tests with about 380,000 OPM, while the 18 vCPU test case falls off to around 275,000 OPM. While the efficiency of having a full socket dedicated to a single 18 vCPU VM provided the best performance with 8 VMs, the lack of any additional room for other VMs was limiting when more were added. Having more vCPUs than physical cores results in some real resource contention that in turns leads to lower overall performance.

This same type of resource contention also ends up affecting the 16 vCPU VMs once the thirteenth VM is added as the total throughput for that test case drops to 289,000 OPM. This is a little bit higher than what the twelve 18-vCPU VMs achieved, but still represents a significant drop from 380,000 OPM.

Analysis

Oracle database workloads on VMware Cloud on AWS provide high performance, achieving hundreds of thousands of orders per minute. The scale-up and scale-out performance shows that Oracle running in this environment is capable of supporting big workloads as well as a large number of small- to medium-sized workloads as well. Most of the best practices that have been recommended for Oracle on vSphere still apply to this new environment. The VMware-published Oracle Databases on VMware Best Practices Guide [3] is the best source for complete details on best practices.

Distributed Resource Scheduling

Distributed Resource Scheduling (DRS) is a feature of vSphere that decides where to place VMs upon power-on, and when to move running VMs. This is the VM load balancing feature that is used to make sure that VMs are placed so that performance is optimal across a vSphere cluster. When using VMware Cloud on AWS, this feature is turned on and configured as part of the service. VMs are placed and moved around the cloud automatically.

Initial placement of the VMs was fairly easy for DRS in the test cases where there were enough resources—this was all of the test cases with 8 VMs or less. In cases where there were more, DRS would sometimes decide to move VMs around after load was applied to the VMs, meaning that, in some of the tests with more than 8 VMs, multiple vMotion migrations occurred during the first test run with that number of VMs. This sometimes caused the initial set of tests to have lower than expected throughput in the form of low OPMs. Once the vMotion migrations completed, throughput increased back to expected levels.

VM Sizing

It is important to size the VM with the amount of virtual CPUs and memory that are needed for that workload. Creating VMs that are larger than they need to be can result in wasted resources and lower overall performance across all VMs. Looking at the results of the 16 vCPU VMs compared with the 18 vCPU VMs, it is clear that better overall performance was possible with more slightly smaller VMs. At the same time, an 18 vCPU VM did outperform a 16 vCPU VM. Rightsizing the VMs so that they have the resources they need can lead to overall better utilization.

Performance

The performance scalability that we observed in these tests on the VMware Cloud on AWS environment were very similar to what is seen with Oracle database test workloads running on VMware vSphere onsite. Because the software stack is essentially the same and the key difference is location and deployment methodology, this was expected. High-performance Oracle databases can be run successfully with good performance on VMware Cloud on AWS.

Conclusion

The ability to quickly get a complete VMware vSphere environment up and running on AWS cloud infrastructure is a powerful solution. A best practices–based setup with compute, storage, and network resources can be quickly ready to run Oracle databases with high performance requirements.

References

[1] VMware, Inc. (2018) VMware Cloud on AWS Resources.
https://cloud.vmware.com/vmc-aws/resources

[2] Todd Muirhead. (2017, December) DVD Store version 3.
http://www.github.com/dvdstore/ds3

[3] VMware, Inc. (2016, May) Oracle Databases on VMware Best Practices.
https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/vmware-oracle-databases-on-vmware-best-practices-guide.pdf

Disclaimer: Results do not represent the performance of Oracle software, neither are they meant to measure Oracle performance nor compare the performance of Oracle to another DBMS. We simply use Oracle on VMware Cloud on AWS to observe and optimize the performance of virtual machines in this environment.

About the Author

Todd Muirhead is a performance engineer at VMware and has worked with many different databases, servers, and storage systems to develop best practices and performance guides. He is also the co-creator and maintainer of the DVD Store open source benchmark.

Acknowledgements

The author would like to thank David Morse for valuable conversations on database performance; Dave Jaffe, for his partnership on the DVD Store Benchmark; and many colleagues in the performance engineering organization for their support and feedback.