Oracle Database Performance: VMware Cloud on AWS i4i vs i3en vs i3

Performance Study - June 30, 2023

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

Table of Contents

Executive Summary

This paper analyzes the increase in performance and capacity of the i4i instance type of VMware Cloud™ on AWS for Oracle® Database. The i4i instance type offers more compute cores, memory, and other performance enhancements allowing it to host larger and more powerful VMs. You can learn more about instance types available with VMware Cloud on AWS in this TechZone article.

Testing found that Oracle databases running on i4i, compared to the i3en and i3 instances, provided about 25 and 50 percent more throughput per core and about 50 and 200 percent the throughput for similar-sized SDDCs. With this higher performance capability, VMware Cloud on AWS with i4i instance types can run larger Oracle databases than in the past.

Introduction

VMware Cloud on AWS delivers an integrated hybrid cloud solution that extends on-premises vSphere environments to a VMware software-defined data center (SDDC) running on Amazon Elastic Compute Cloud (Amazon EC2). This elastic, bare-metal infrastructure is fully integrated with AWS. The service offers ultra-fast cloud migration powered by VMware HCX® and vMotion combined with consistent hybrid-cloud infrastructure and operations from AWS. Once applications are migrated, customers can run, manage, and modernize these applications with the VMware Tanzu portfolio as well as integrate native AWS services. More than 500 channel partners have achieved a VMware Cloud on AWS service competency, and more than 300 certified or validated technology solutions are available to VMware Cloud on AWS customers.

The Oracle database management system has been a key component of many enterprises for decades, with proven performance and scalability to act as the backbone database for some of the largest applications. The systems administrators that work with the Oracle database administrators to put together database solutions for their customers know that performance is key. With both large and small deployments of Oracle, it is critical to maintain the highest levels of performance with quick responses to queries of information from the database.

VMware vSphere has long been an excellent platform for Oracle databases in customer data centers worldwide. With the introduction of VMware Cloud on AWS several years ago, the VMware vSphere platform became available in the cloud to support Oracle instances as well. With the new i4i instance types announced recently, additional, more powerful hardware is now available to support larger databases with higher performance requirements than that with the i3 and i3en instance types.

To measure these performance gains specifically with Oracle databases, we deployed a three-host SDDC on the VMware Cloud on AWS service using the new i4i instance type. We set up a test environment with Oracle database VMs and ran many tests to measure and compare the performance gains found on the new i4i instance type. Our results showed significant performance gains for single and multiple VM–based tests.

Test Environment

We deployed an SDDC using a three-host cluster from our VMware Cloud on AWS portal for our tests. The server instance type was i4i, which was introduced as a new instance type for VMware Cloud on AWS in July of 2022, bringing new servers with more cores and more memory. We compared the results from this test with those of the same tests on an i3en.metal and i3.metal three-host SDDCs performed previously.

The servers of the SDDC used dual Intel® Xeon® Platinum 8375C processors with 32 physical cores running at 2.9GHz with 1024GB of RAM. (See figure 1.)

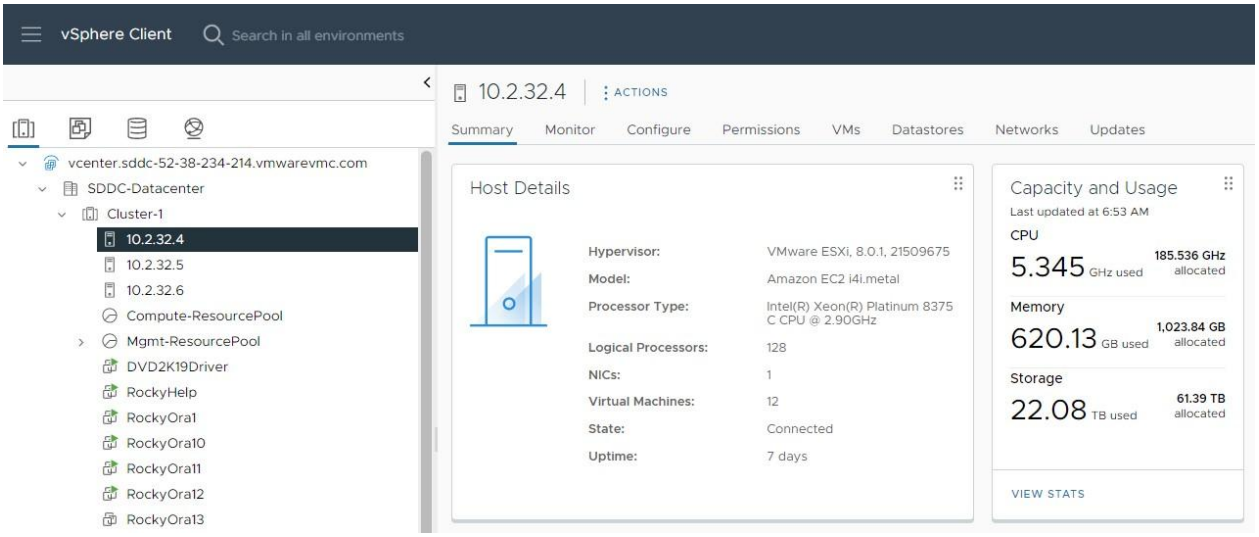

Figure 1. Screenshot of the test environment

Storage for the SDDC was provided by vSAN using the local NVMe devices in the hosts. vSAN is configured as part of the SDDC deployment, and the storage is ready when you log into vCenter for the first time. The capacity available depends on the number of hosts in the SDDC. In our three-host SDDC environment used for testing, the usable capacity was 61.39TB (RAID1, FTT=1), as shown in figure 2.

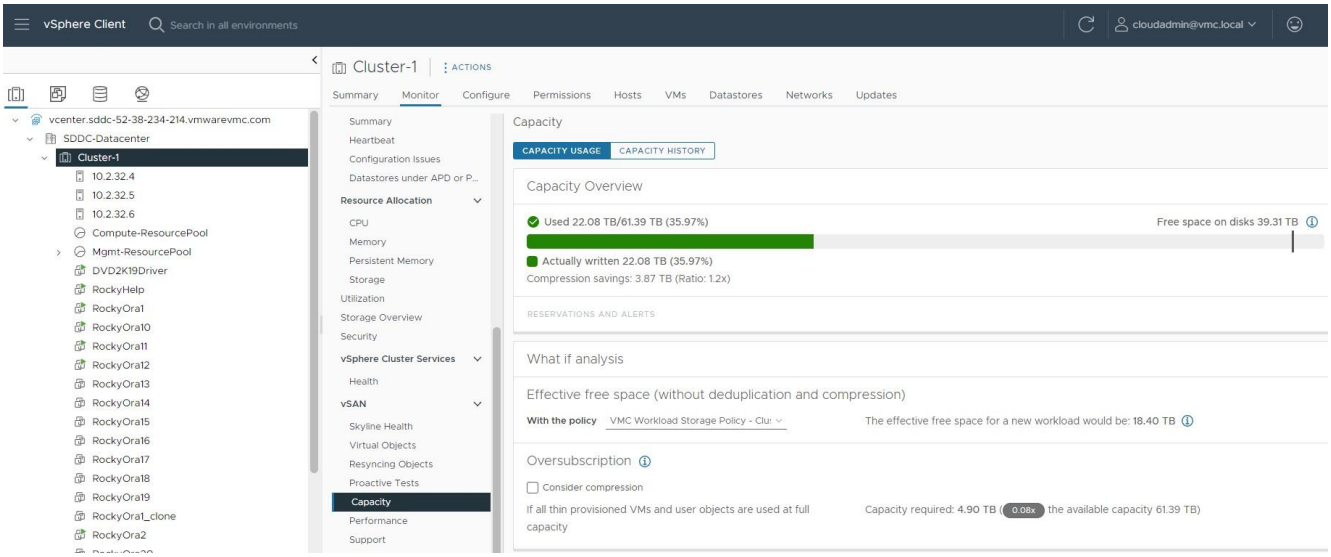

Figure 2. Screenshot of workload datastore configuration

Figure 3 shows a three-node SDDC on AWS and its significant components. We configured only the VMs that composed 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.

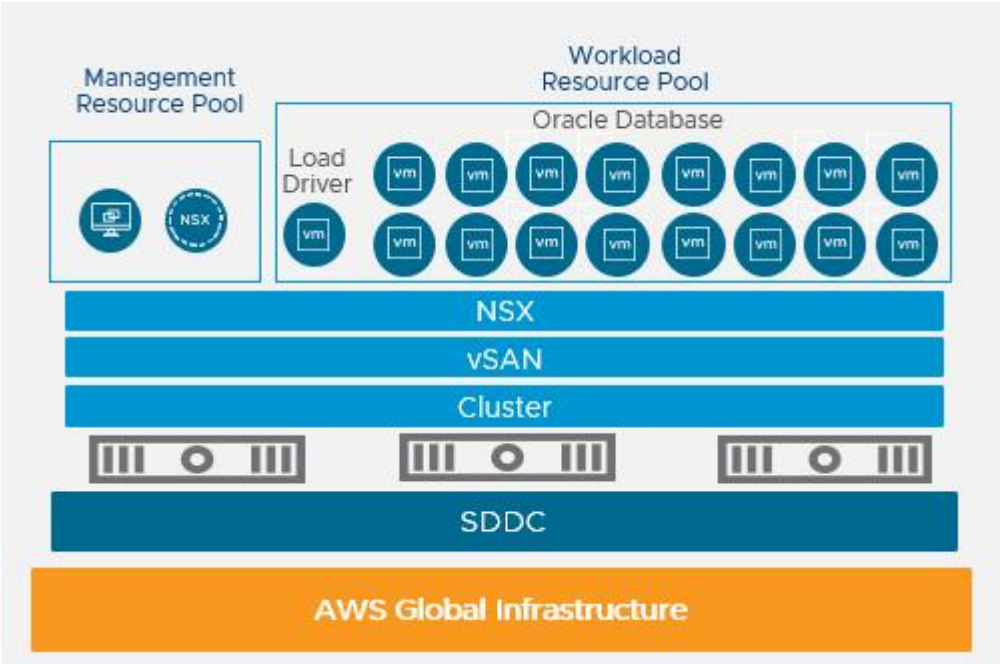

Figure 3. Cloud testbed configuration

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 login 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 that the maximum throughput corresponded to a host CPU utilization of between 90 and 99 percent.

DVD Store 3 lets you create any size database. We used a database with approximately 240GB of on-disk size for these tests.

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 128GB.

Using the Oracle-provided installer RPM, we installed the VM with Rocky Linux (RHEL-compatible) 8.4 as the operating system and Oracle Database 19c. We followed Oracle's best practices documentation for this installation. We configured huge pages for Linux and enabled them in Oracle, and we set the Oracle sga_target parameter to 100GB. This ensured that a large amount of the 240GB database would be available in the buffer cache, but not all, so 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 16 vCPUs up to 32 vCPUs for the i3 instance, 16 vCPUs up to 96 vCPUs for the i3en instance, and 16 vCPUs up to 128 vCPUs for i4i. These maximum sizes were based on the number of cores in a single host in each instance type.

The DVD Store driver, which ran on a separate VM, generated the load. The load-driver VM and the Oracle Database VM ran in VMware Cloud on AWS but were on different physical hosts in that cloud.

The results are shown in figure 4.

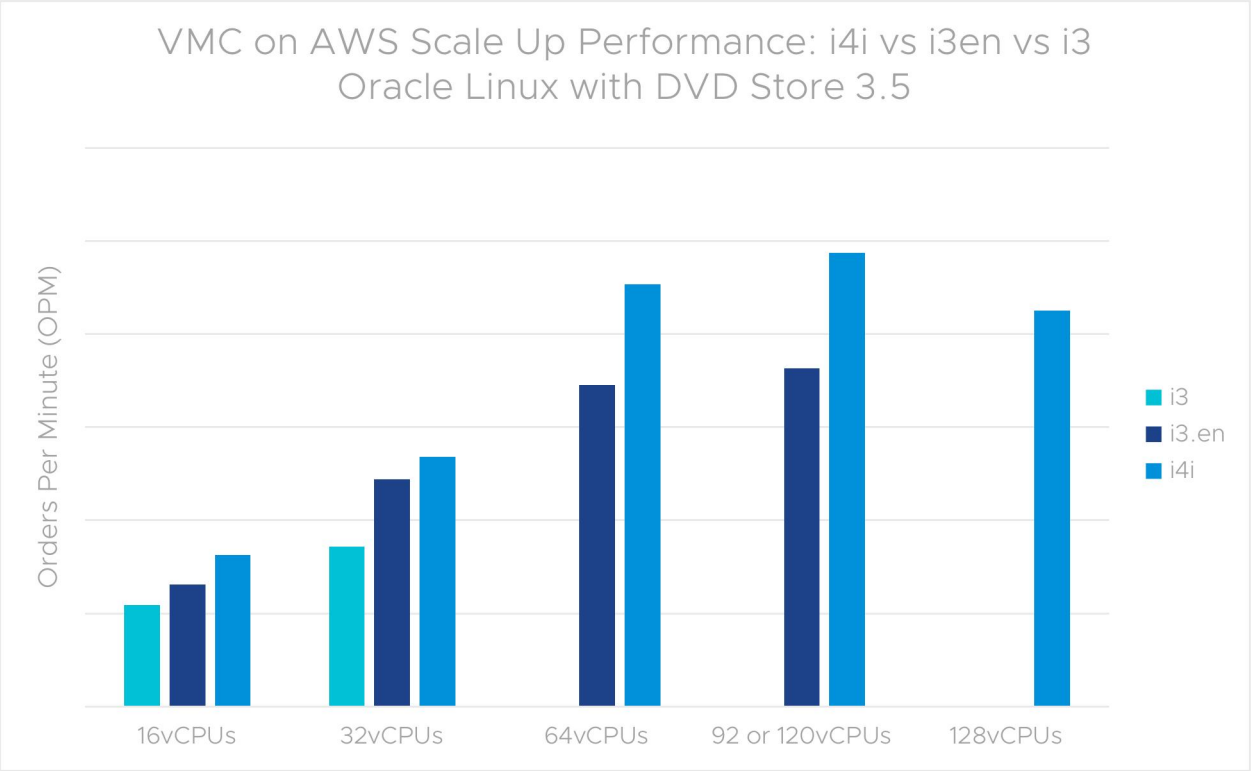

Figure 4. Oracle DB VM scale-up performance (adding processors)

The scale-up test results highlight the increased capability of the hosts used for the i4i instance type that improves Oracle Database performance. When comparing a 16-vCPU VM running on all instance types, the performance advantage of i4i is 24 percent better than i3en and 50 percent better than i3. This means that with the same number of vCPUs, the VM on i4i outperformed the i3en and i3 VMs by a significant margin.

A single VM on i41 (120 vCPUs) was able to achieve 35 percent more throughput than what a single VM on i3en (92 vCPUs) was able to achieve. The main factor contributing to this performance difference is that the i4i instance has more cores per socket than the i3en. This allows for a larger size VM to be created. In addition to having even fewer cores, the i3 instance does not have hyperthreading enabled due to the L1TF vulnerability that existed in the older Intel processors, but it does not exist in the newer Intel processors used in the i3en and i4i instance types.

An interesting final point on these results is that the best-performing VM is not created at the absolute max size possible of 128 vCPUs, but instead was 120 vCPUs. This is often seen when running a VM that has all the CPU resources assigned to it, which in this case is 128 vCPUs, and that VM is run at very high utilization levels of over 95 percent. With this configuration and such a high utilization level, the system does not have the resources to handle the hypervisor and virtual networking without taking cycles from the cores the VM uses to perform its tasks. In this way, the VM and the system compete for resources, and the application's overall performance suffers. By leaving a couple of

threads per socket unassigned, the hypervisor and network functions have some resources to use without interrupting the VM; this results in better overall performance. For this reason, we recommend that you size the VMs slightly smaller than the system to optimize performance.

These scale-up tests with a single VM show that the i4i instance type can host much larger Oracle Database VMs with more performance than was possible with the i3 or i3en instance types.

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 also spread out across the cluster, based on the best load balancing as determined by VMware vSphere® Distributed Resource Scheduling™ (DRS). The total number of OPMs achieved across all the Oracle Database VMs is then reported for each set of VMs.

Two VMs per Host Processor

We performed the first set of scale-out tests with 16-vCPUs for i4i, 12-vCPU VMs for i3en, and 8 vCPUs for i3. This was sizing the VMs in each case so that two would fit in each host processor. For example, the i4i hosts have 32 cores per socket, which means that two 16 vCPU VMs fit in each socket without over-provisioning. We increased the number of VMs up to 2 VMs per socket per host, which, in this case, was 12.

The results are shown in figure 5.

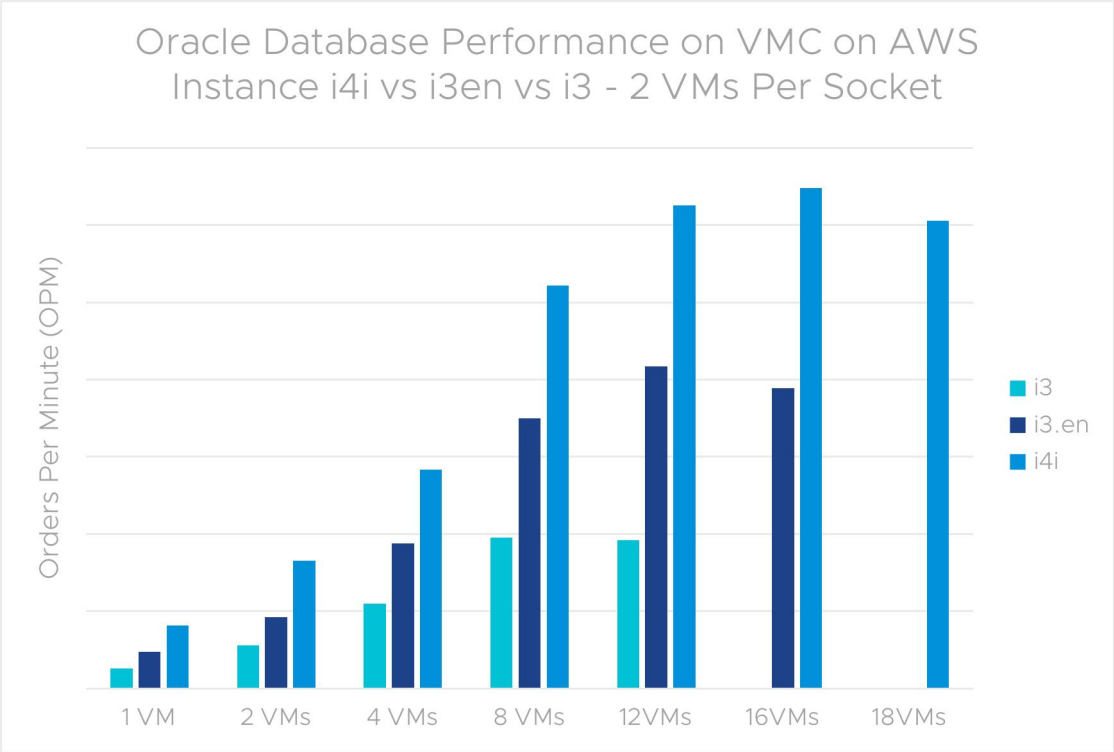

Figure 5. Scale-out performance with 8 vCPU VMs

By scaling the VM to take advantage of the additional cores in the i4i instance type, we can compare the total system performance as the number of VMs is scaled out. In the last case with four Oracle database VMs per host, the total throughput of the i4i SDDC is about 50 percent more than that of the i3en SDDC and nearly 200% more than i3. This is for the same reasons that a single i4i outperformed a single i3en VM: mainly more cores, better cores, and in the case of i3, hyperthreading.

Combined Results

Figure 6 combines the results presented in the previous sections to allow for easy side-by-side comparison. Up and to the right indicates more cores and more performance. The combination graph shows some interesting comparisons. The baseline here is the performance of an 8-vCPU VM running on i3. The i4i scale-up maximum is almost 10x as much throughput, about 25% higher than a single VM scale-up maximum throughput for i3en. Finally, the scale-out performance of i4i at 25x is significantly higher than i3en at 18x. These numbers represent the potential database performance across the 3-node clusters tested and the relative difference between the solutions.

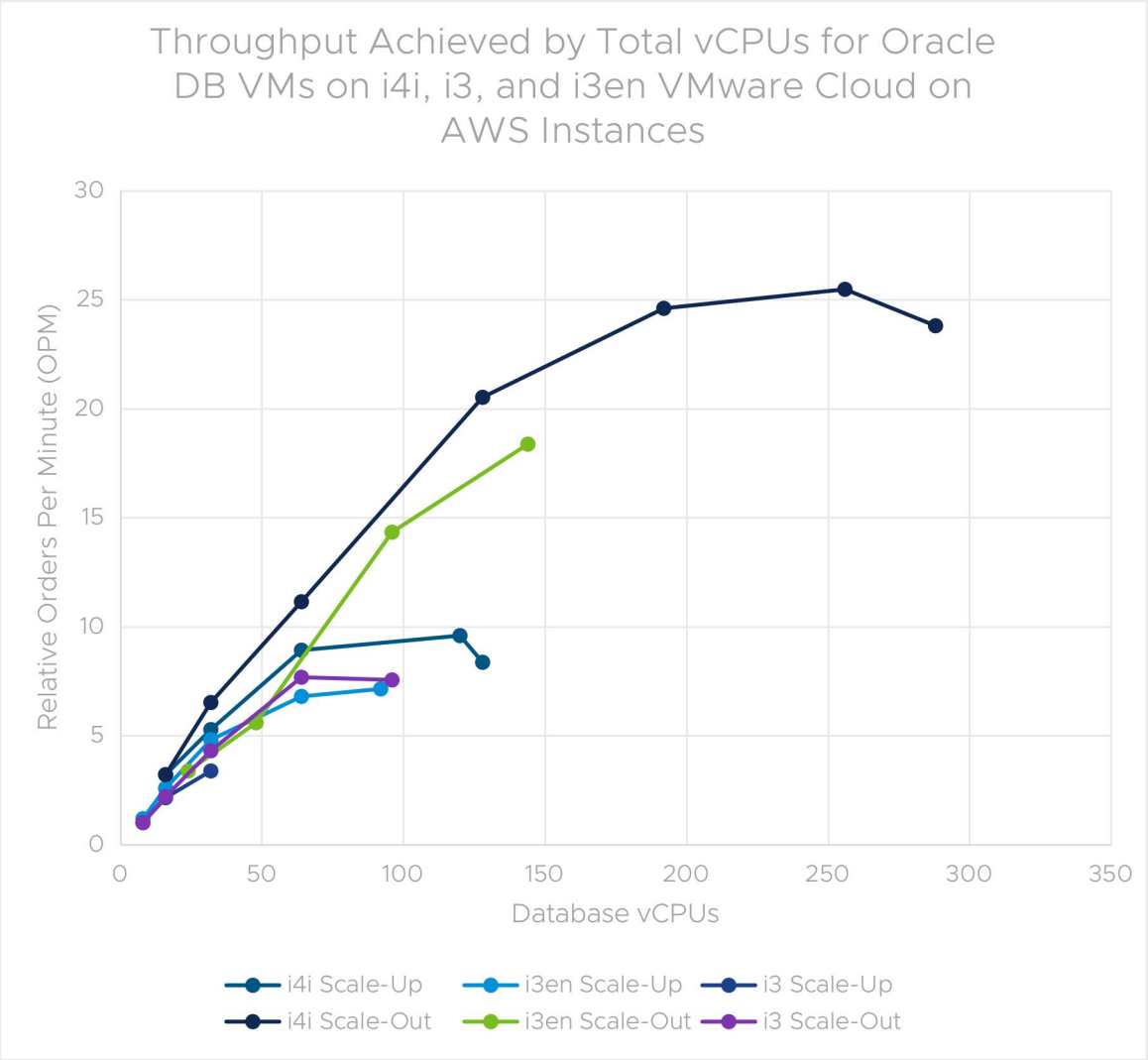

Figure 6. VMware Cloud on AWS scale-up and scale-out performance relative combination plotted graph

Analysis

I4i instances provide significant performance and capacity advantages over i3 and i3en instances and are well-designed to serve Oracle Database workloads. The VMware-published Oracle Databases on VMware Best Practices Guide [3] is the best source for complete details on best practices.

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 VM load balancing feature is used to ensure that VMs are placed to achieve optimal performance 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 with enough resources—this was all of the test cases with 8 VMs or less. In cases where there were more VMs, DRS would sometimes decide to move VMs around after load was applied to them; this means 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 were completed, throughput increased back to expected levels.

VM Sizing

It is essential to size the VM with the amount of virtual CPUs and memory 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. The scale-out results show that performance scales well with VMs that are right-sized based on a CPU core count that leads to efficient utilization.

Performance

The performance scalability we observed in these tests on the VMware Cloud on AWS environment was very similar to what is seen with Oracle Database test workloads running on VMware vSphere onsite. This was expected because the software stack is essentially the same, and the key difference is the location and deployment methodology. High-performance Oracle Databases run successfully with good performance on VMware Cloud on AWS.

Comparing i3 vs i3en and i4i Instances

In some ways, the comparison of i3 vs i3en or i4i is a bit of apples vs oranges due to some underlying configuration differences. For example, hyperthreading is disabled on i3 due to the L1TF vulnerability, but it is enabled on i3en and i4i because L1TF is not an issue on those CPU generations.

Conclusion

The ability to quickly deploy a complete VMware SDDC environment 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. For larger database deployments with higher performance requirements, the i4i instance type provides clear advantages over the i3 and i3en instance types due to its higher number of cores, additional RAM, and better overall performance.

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.

Author Info

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.