Performance Characterization of Microsoft SQL Server Using i3en Instances on VMware Cloud on AWS

Performance Study - September 28, 2020

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Copyright © 2020 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark 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.

Table of Contents

Executive Summary

This paper analyzes the increase in performance and capacity of the i3en instance type of VMware Cloud™ on AWS for Microsoft SQL Server databases. The i3en instance type offers more compute cores, more memory, and other performance enhancements that allow it to host larger and more powerful VMs. Testing found that SQL Server databases running on i3en compared to the i3 instance provided about 18 percent more throughput per core and about double the throughput for similar-sized SDDCs. With this higher performance capability, VMware Cloud on AWS i3en instance types can run larger SQL Server databases than in the past.

Introduction

VMware Cloud on AWS is a jointly engineered service that brings VMware Cloud Foundation™ to Amazon Web Services™ (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 there are more than 300 certified or validated technology solutions available to VMware Cloud on AWS customers.

VMWare Cloud on AWS has historically run on i3 and r5 metal instances. In July of 2020, the new i3en instance, based on 2nd generation Intel® Xeon® Scalable Processors, became available. It's designed for storage-dense workloads with high-performance requirements and delivers superior economics at scale for datacenter migration and disaster recovery transformation projects. These new instances deliver 4x the raw storage capacity at roughly half the cost per gigabyte of storage per host of current offerings. In addition, it comes with low latency Non-Volatile Memory Express (NVMe™) SSD capacity for applications that require high random I/O access to large amounts of data such as relational databases.

Microsoft SQL Server is one of the most widely used databases in the enterprise. It provides core database functionality with high performance and scalability for a wide range of applications. Because of its widespread use with customers, it is one of the applications that is most often run on vSphere. Running SQL Server with high performance and high availability are key aspects of why customers choose VMware vSphere® as their platform.

vSphere has long been an excellent platform for SQL Server databases in customer datacenters around the world. With the introduction of VMware Cloud on AWS several years ago, the vSphere platform became available in the cloud to support SQL Server databases. With the new i3en instance types announced recently, additional and more powerful hardware is now available to support larger databases with higher performance requirements than the i3 instance types could handle.

To measure these performance gains specifically with SQL Server databases, a three-host software-defined datacenter (SDDC) was deployed on the VMware Cloud on AWS service using the new i3en instance type. A test environment was set up with SQL Server database VMs, and many tests were run to measure and compare the performance gains found on the new i3en instance type. Results showed significant performance gains for single VM and multiple VM–based tests.

Test Environment

For our tests, we deployed an SDDC using a three-host cluster from our VMware Cloud on AWS portal. The server instance type was i3en.metal which was introduced as a new instance type for VMware Cloud on AWS in July of 2020. This new instance type brings new servers with more cores and more memory. We compared the results from this test with results using the same tests on an i3.metal three-host cluster that were performed a few months earlier.

The servers of the SDDC used Intel Xeon Platinum 8259CL processors with 24 cores running at 2.5Ghz with 789GB of RAM. (Shown in **figure 1**).

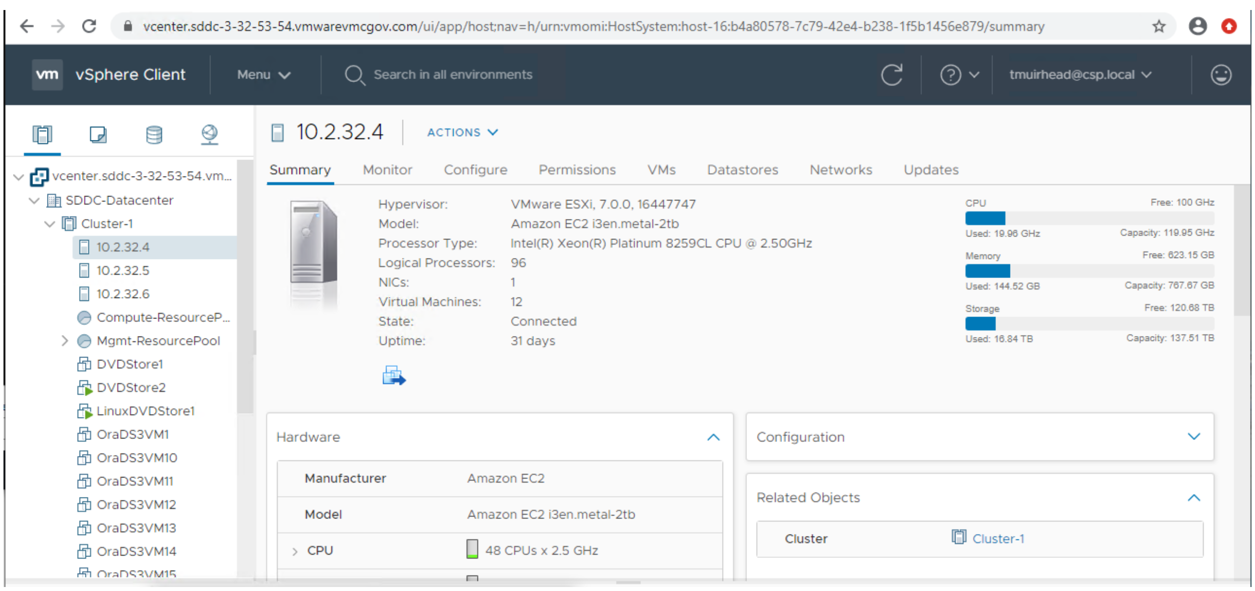

Figure 1. Screenshot of the test environment

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 amount of capacity available depends on the number of hosts in the SDDC. In our three-host SDDC environment used for testing, the capacity was 137.5TB as shown in **figure 2**.

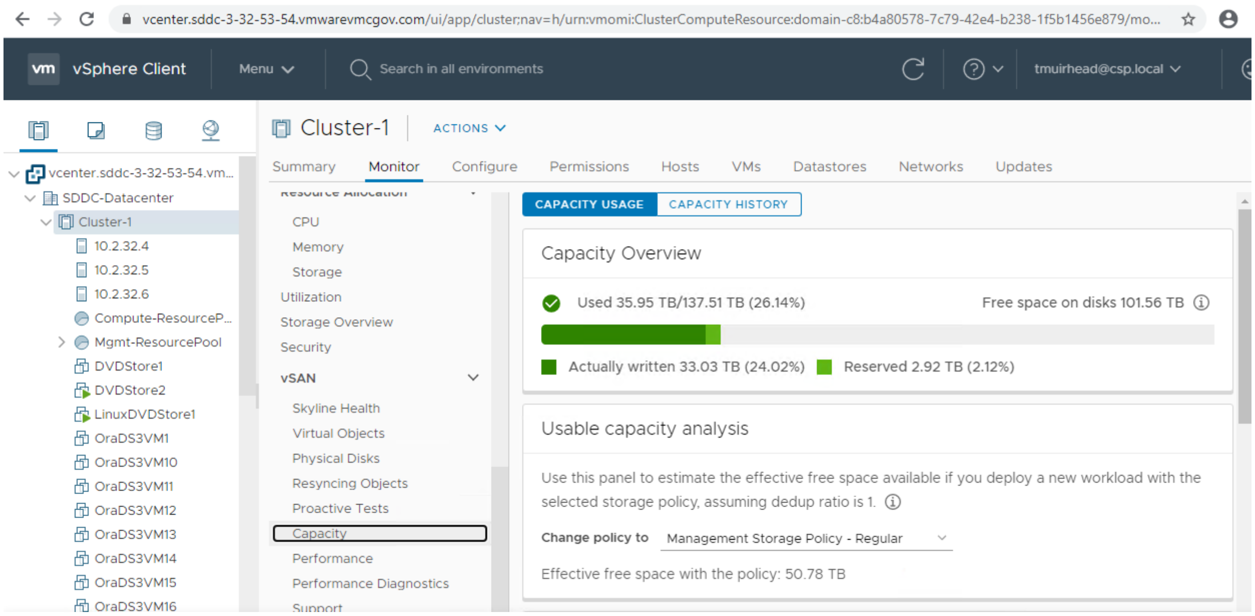

Figure 2. Screenshot of workload datastore configuration

Figure 3 shows a four-node SDDC on AWS and its major components. We configured only the VMs that composed the SQL Server 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.

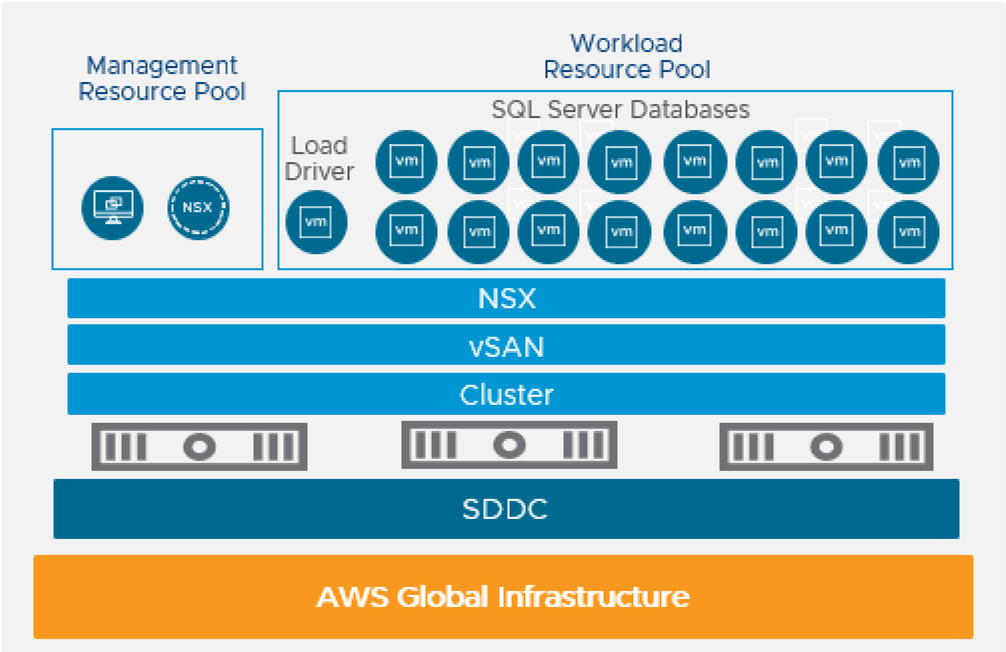

Figure 3. Cloud testbed configuration

Test Workload

We used the workload from the open-source benchmark DVD Store 3 [1]. 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 any size database to be created. For these tests, a database with approximately 250GB of on-disk size was used. This same size database was used in all tests in this paper,

Virtual Machine Configuration

Windows Server 2019 datacenter was installed as the guest operating system for all the VMs we tested (both the load-driving client VM and the database servers). SQL Server 2019 Enterprise Edition was the database engine used within all database server VMs. We adhered to Architecting Microsoft SQL Server on VMware vSphere: Best Practices Guide [2], and do not have any specific additional caveats for deploying SQL Server within VMware Cloud on AWS.

We configured the database servers with 128GB of RAM, while the number of vCPUs varied depending on the test. We configured the load driver VM with 24 virtual CPUs (vCPUs) and 16GB of virtual RAM. The VMs used the VMXNET3 virtual network adapter and paravirtual SCSI (PVSCSI) adapters. We assigned data and log disks to separate PVSCSI adapters.

Scale-Up Performance

A scale-up test, in the context of this study, is when a single VM is given increasing amounts of vCPUs for each iteration of the test to measure how well performance scales as these increases in vCPUs occur.

Scale-up tests on both i3 and i3en instance types were started at a baseline of 8 vCPUs and then increased to 32 vCPUs for i3 and 96 vCPUs for i3en. The much higher maximum vCPUs used for i3en is due to the additional cores available and hyperthreading being enabled. On i3 instances, hyperthreading was disabled due to the L1TF vulnerability that exists on this generation of processors. The processors used for i3en instances are newer and do not have the L1TF vulnerability. A separate VM running the DVD Store driver generated the load. The load-driver VM and the SQL Server database VM both ran in VMware Cloud on AWS, but they were on different physical hosts in that cloud.

The results are show in **figure 4** and explained in detail following the graph.

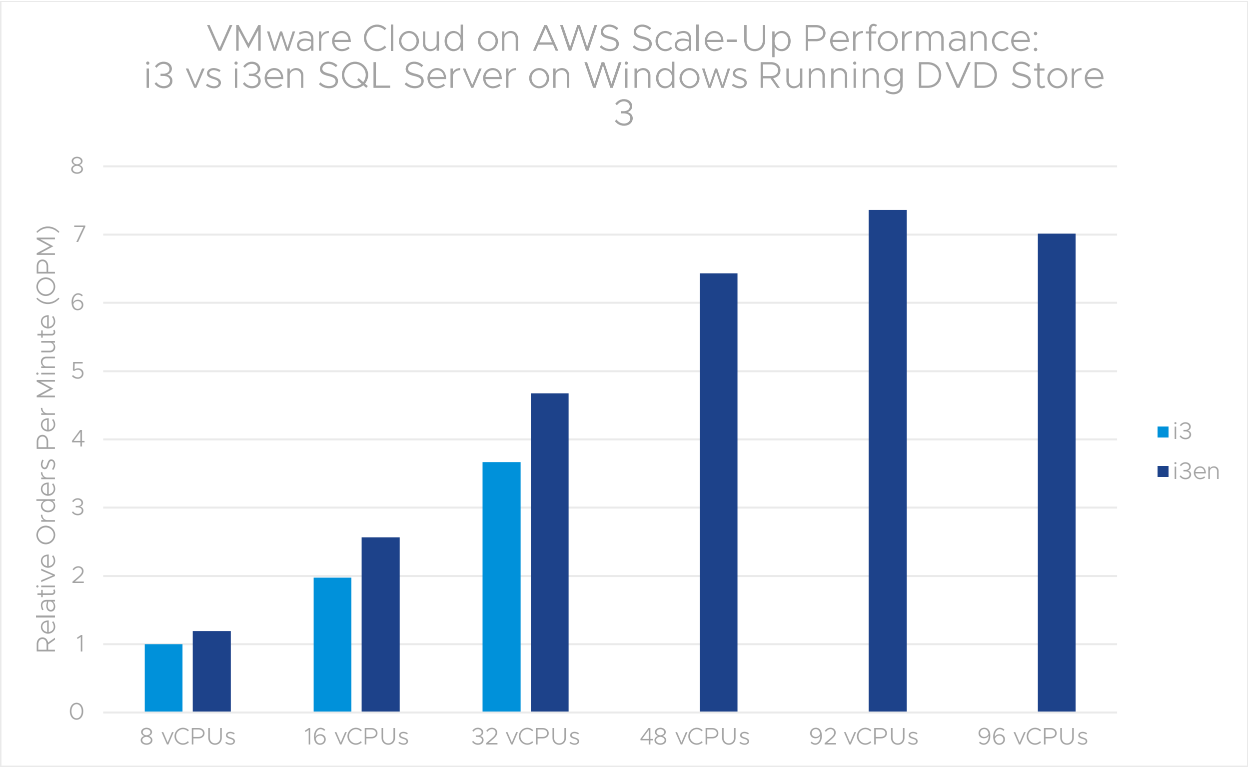

Figure 4. SQL Server scale-up performance (adding processors)

The scale-up test results highlight the increased capability of the hosts used for the i3en instance type that result in better performance for a SQL Server database. In the case of comparing an 8-vCPU VM running on i3 vs an 8-vCPU VM on i3en, the performance advantage is 19 percent. This means that with the same number of vCPUs, the VM on i3en was able to outperform the i3 VM by a significant margin.

A single VM on i3en (92 vCPUs) was able to achieve double the throughput of what a single VM on i3 (32 vCPUs) was able to achieve. There are two main factors that contribute to the big performance difference seen in this comparison. The i3en instance has more cores per socket than the i3. The i3 instance does not have hyperthreading enabled due to the L1TF vulnerability that existed in the older Intel processors, but this vulnerability does not exist in the newer Intel processors used in the i3en instance type. In addition to more cores and hyperthreading, the i3en instance type also has vSAN configured without compression or deduplication and better and faster NICs. Future versions of i3en instances will have compression and/or deduplication enabled, which will increase storage capacity while potentially decreasing performance.

An interesting final point on these results is that the best performing VM is not created at the absolute max size possible of 96 vCPUs, but instead was 92 vCPUs. This is often seen when running a VM that has all the CPU resources assigned to it, which in this case is 96 vCPUs, and that VM is run at very high utilization levels of over 95%. With this configuration and such a high utilization level, the system does not have resources to handle the hypervisor and virtual networking without taking cycles from the cores that the VM is using to perform its tasks. In this way, the VM and the system compete for resources, and the overall performance of the application suffers. By leaving a couple of threads per socket unassigned, the hypervisor and network functions have some resources to use without having to interrupt the VM, and this results in better overall performance. For this reason, we recommend you size VMs to be slightly smaller than the system to optimize performance.

These scale-up tests with a single VM show that the i3en instance type provides the capability to host much larger SQL Server database VMs with more performance than was possible with the i3 instance type.

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 SQL Server database VMs simultaneously. The database VMs are spread out across the cluster based on the best load balancing as determined by vSphere Distributed Resource Scheduling™ (DRS). The total number of operations per minute achieved across all the SQL Server database VMs is then reported for each set of VMs.

In these tests, we used one cluster with only three hosts in our SDDC, but more hosts could be added as needed, up to the maximum supported. 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 three-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 12 for i3 and 20 for i3en. The limit for the number of VMs in this case was due to the amount of RAM in the hosts. The i3 instances each have 512GB of RAM for a total of about 1.5TB with three hosts. Each of the SQL Server VMs has 128GB assigned, which means that with 12 of them powered on there is 1.5TB of RAM allocated for the database VMs. The i3en instances have 768GB per host, which means we have a total of 2.3TB allowing us to get 20 128GB SQL Server VMs with a total allocation of 2.5TB to run. This amount of memory oversubscription is a bit more than what we would recommend as a best practice, but the system was able to handle the test runs successfully. The results are shown in **figure 5**.

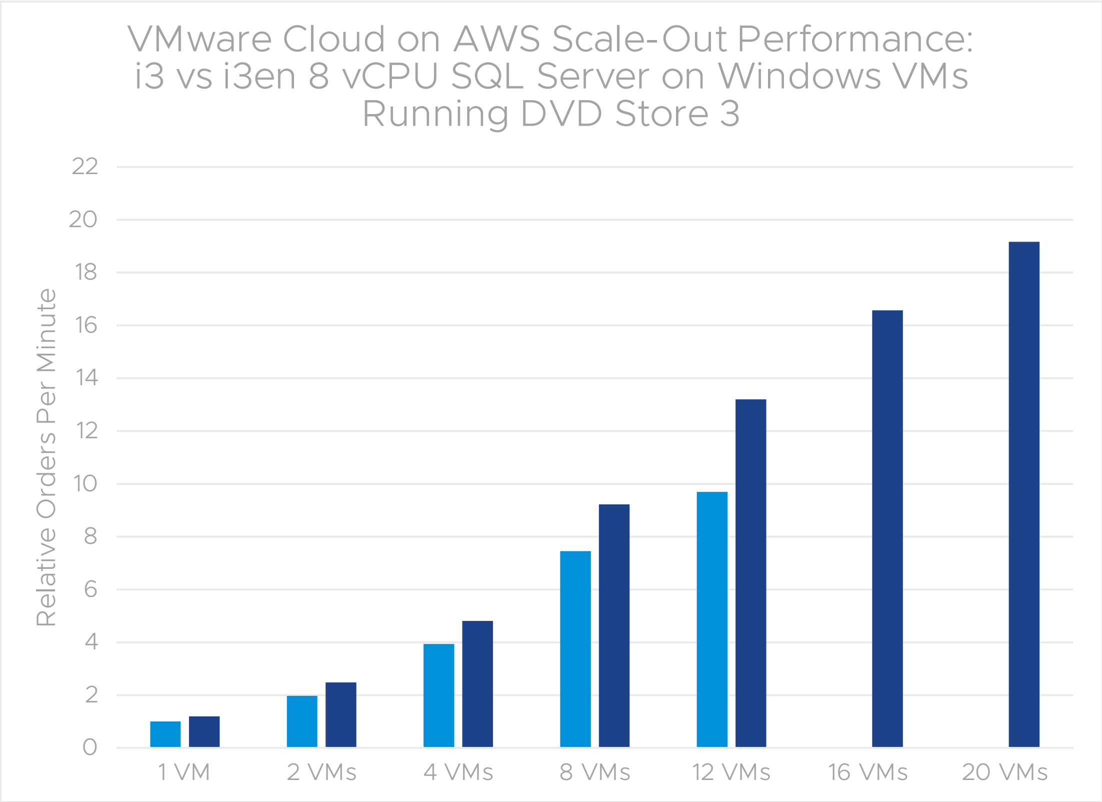

Figure 5. Scale-out performance with 8 vCPU VMs

The results for scale-out using 8 vCPUs show the same data points comparing 8 vCPU i3 vs i3en with a gain of 19 percent as shown in the initial point in the scale-up graph in the previous section. In this scale-out testing with 8-vCPU VMs, we see what happens to performance with multiple VMs instead of simply increasing the number of vCPUS in a single VM. The total throughput across all VMs is reported here to show the total performance across all VMs tested in each case. The performance advantage for i3en is between 22 and 25 percent for the 2, 4, and 8 VM cases, which is slightly better than what we saw in the 19 percent advantage with a single 8-vCPU VM. The performance advantage for i3en increases to 36 percent with 12 VMs as the i3 cluster becomes heavily loaded and performance scaling falls off a bit. The i3en cluster can support more 8-vCPU VMs and achieve additional performance levels due to the additional memory and cores with hyperthreading enabled. The maximum throughput achieved by i3en with 20 8-vCPU VMs was approximately double the maximum achieved by the 12 VMs of the i3 cluster.

16-vCPU VMs

The next test case used 16-vCPU VMs in a scale-out scenario to test scalability with larger VMs. In this test scenario, the main limiting factor is CPU instead of memory, as seen in the previous 8-vCPU VM tests. The hosts in the i3 cluster have 18 cores per socket for a total of 108 cores across all three hosts. In the final i3 test case here, CPU is oversubscribed with 128 vCPUs for 8 SQL Server VMs. With i3en there are 24 cores with hyperthreading enabled on each host in the cluster for a total of 144 cores and 288 threads. With 16 SQL Server VMs with 16 vCPUs each, there are a total of 256 vCPUs allocated. The results are shown in **figure 6**.

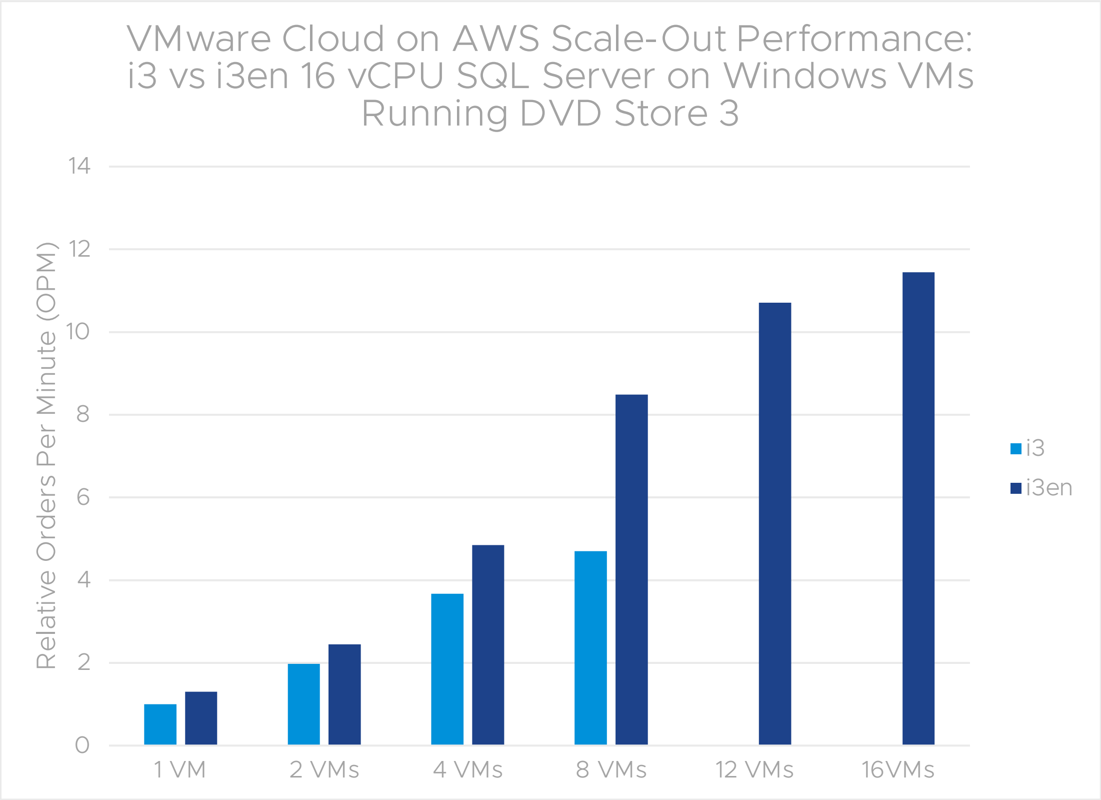

Figure 6. Scale-out performance with 16 vCPU VMs

On both the i3 and i3en clusters, once the number of vCPUs exceeded the number of physical cores, performance does not increase at the same rate as new VMs are added. With the i3 cluster when running with eight 16-vCPU VMs, there were 128 vCPUs allocated. This means there were more vCPUs allocated than the cloud had physical cores. With the i3en-based cluster, it was possible to run more VMs due to more cores that have hyperthreading enabled. While hyperthreading does double the number of logical threads, it does not double the performance. It can boost performance of the overall system by 15 to 30 percent depending on the workload. These advantages with the i3en instance type are seen clearly in these tests with the system able to run twice as many 16-vCPU VMs as i3, while also achieving 2.4x the total throughput as i3.

Analysis

i3en instances provide significant performance and capacity advantages over i3 instances. The major advantages of the i3en instances are more cores, hyperthreading, and additional RAM. These advantages are combined with many other smaller advantages found with the newer generation of server hardware, which provides a great platform for running SQL Server databases. The VMware-published Architecting Microsoft SQL Server on VMware vSphere: Best Practices Guide [2] 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 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. 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 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.

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 SQL Server 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 SQL Server databases can be run successfully with good performance on VMware Cloud on AWS.

Comparing i3 vs i3en Instances

In some ways, the comparison of i3 vs i3en is a bit of apple vs oranges due to some of the underlying configuration differences. For example, hyperthreading is disabled on i3 due to the L1TF vulnerability, but it is enabled on i3en because L1TF is not an issue on that generation of CPU. Another example is that vSAN on i3 has compression and deduplication enabled, but on i3en they are both disabled. These settings are locked down and cannot be changed by users. Further, VMware Cloud on AWS is a service that does allow these options to be changed. We compared theses services with these differences while making the things that we can control the same.

Conclusion

The results from this performance study show that customers running SQL Server database workloads within VMware Cloud on AWS can expect to attain higher performance with i3en instances than with i3. With the i3en instances, there are more cores, each core is more powerful, and there is more memory. As shown by the results in this study, this additional performance capacity means that larger database VMs can be run that can achieve up to 2x the throughput that was possible with i3. It also means that many more database VMs can be hosted, with the total throughput of those VMs being double what was possible on i3.

References

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

[2] VMware, Inc. (2019, April) Architecting Microsoft SQL Server on VMware vSphere: Best Practices Guide.
https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-best-practices-guide.pdf

[3] Oleg Ulyanov. (2019) Best Practices Guide for Microsoft SQL Server.
https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-best-practices-guide.pdf

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

[5] VMware, Inc. (2012, September) Virtualizing Business-Critical Applications on vSphere.
http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/solutions/vmware-virtualizing-business-critical-apps-on-vmware_en-white-paper.pdf

[6] VMware. (2019, May) Add Hosts. https://docs.vmware.com/en/VMware-Cloud-on-AWS/services/com.vmware.vmc-aws-operations/GUID-53393BD8-4CA2-4608-A700-E2E2B9691D41.html

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.