PERFORMANCE STUDY – JULY 2018

PERFORMANCE CHARACTERIZATION OF MICROSOFT SQL SERVER USING VMWARE CLOUD ON AWS

Table of Contents

Executive Summary

VMware Cloud™ on AWS brings VMware's enterprise software-defined data center (SDDC) to the AWS Cloud. It enables customers to easily migrate their existing virtual infrastructures to seamlessly run their business-critical applications like SQL Server, in the cloud, using the same VMware vSphere® environment that they have traditionally used on-premises.

The purpose of this paper is to answer a question that many customers have today: how do workloads that have traditionally been run in an on-premises infrastructure perform when transitioned to the cloud? The benchmark results show the answer is SQL Server continues to perform great within a VMware Cloud environment, for both small and large OLTP database VMs/workloads.

Introduction

VMware Cloud on AWS, powered by VMware Cloud Foundation™, integrates VMware flagship compute, storage, and network virtualization products—VMware vSphere, VMware vSAN™, and VMware NSX®—along with VMware vCenter Server® management. It optimizes them to run on elastic, bare-metal AWS infrastructure. With the same architecture and operational experience on premises and in the cloud, IT teams can now get instant business value via the AWS and VMware hybrid cloud experience.

The VMware Cloud on AWS solution enables customers to have the flexibility to treat their private cloud and public cloud as equal partners and to easily transfer workloads between them—for example, to move applications from DevTest to production or burst capacity. Users can leverage the global AWS footprint while getting the benefits of elastically scalable SDDC clusters, a single bill from VMware for its tightly integrated software plus AWS infrastructure, and on-demand or subscription services. For more information, visit our VMware Cloud on AWS Resources [1] page.

VMware vSphere, regardless of whether it is on-premises or in the VMware Cloud, provides an ideal platform for business-critical applications, including databases, ERP systems, email servers, and emerging technologies such as Hadoop. A full discussion of the benefits is included in the whitepaper "Virtualizing Business-Critical Applications on vSphere [2]."

A business-critical application that is often run on vSphere today is Microsoft SQL Server, which is "one of the most widely deployed database platforms in the world, with many organizations having dozens or even hundreds of instances deployed in their environments [2]." Consolidating these deployments onto modern multi-socket, multi-core, multi-threaded server hardware as virtual machines on vSphere is an effective solution.

VMware Cloud on AWS is ideal for customers looking to migrate their SQL databases to the public cloud or extend the capacity of their data centers for existing applications that leverage SQL Server.

Test Environment

Infrastructure (On-Premises)

For the on-premises environment, we leveraged an existing lab testbed in our datacenter, which consisted of 4 x HPE ProLiant DL380 Gen9 servers.

The physical hosts within this 4-node cluster each contained two Intel Xeon E5-2683 v4 processors running at 2.10 GHz with 512 GB of RAM. Each processor had 16 cores and 32 logical threads with hyperthreading enabled (Figure 1).

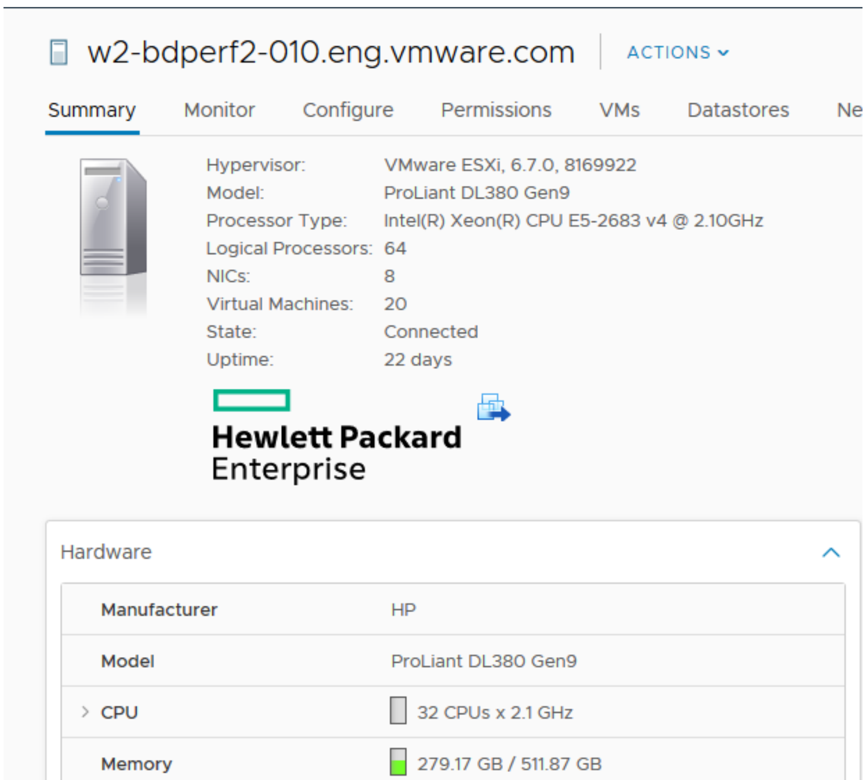

Figure 1. Screenshot of an on-premises host server

Much like VMware Cloud on AWS, these 4 hosts contained local NVMe storage and were configured as a vSAN cluster, with DRS enabled. While the two environments do not represent a 100% “apples-to-apples” comparison, it gives us a reasonably close approximation.

Infrastructure (VMware Cloud on AWS)

For our database benchmarks in the cloud, we deployed a four-node software-defined data center (SDDC) from the VMware Cloud on AWS portal. We used the latest SDDC available at the time of testing, which was version 1.4.

The physical hosts within the SDDC each contained two 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 2).

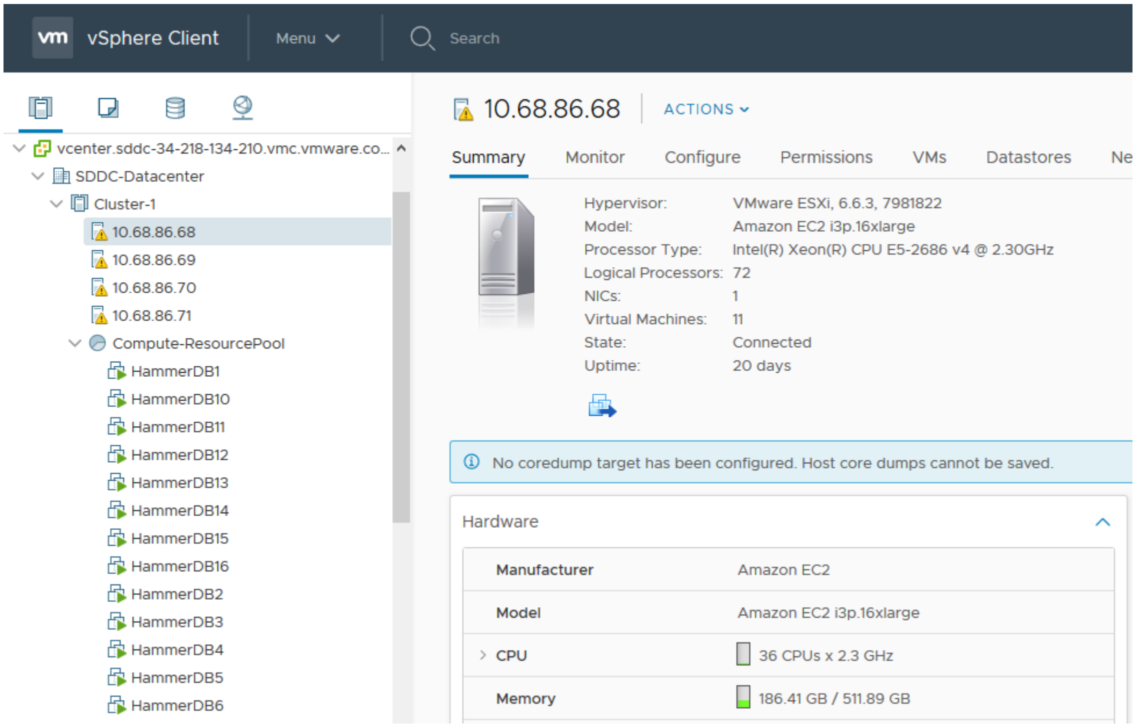

Figure 2. Screenshot of the VMware Cloud on AWS SDDC

On VMware Cloud for AWS, vSAN is used for the storage. It used the 8 NVMe devices that were in each of the 4 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 3). 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 of the SQL Server VMs that are created or migrated into the environment are stored.

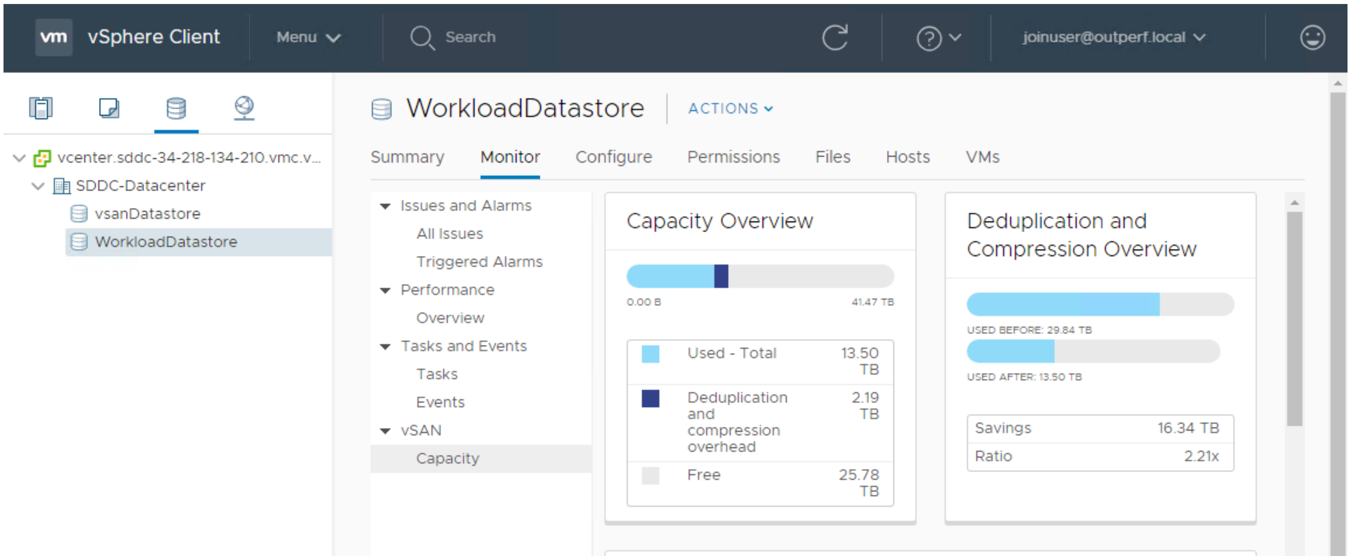

Figure 3. Screenshot of workload datastore configuration

Figure 4 shows a four-node SDDC on AWS and its major components. We only had to deploy and configure the SQL workload load drivers and database VMs; the rest of the components and configuration was done automatically as part of the deployment from the VMware Cloud on AWS portal.

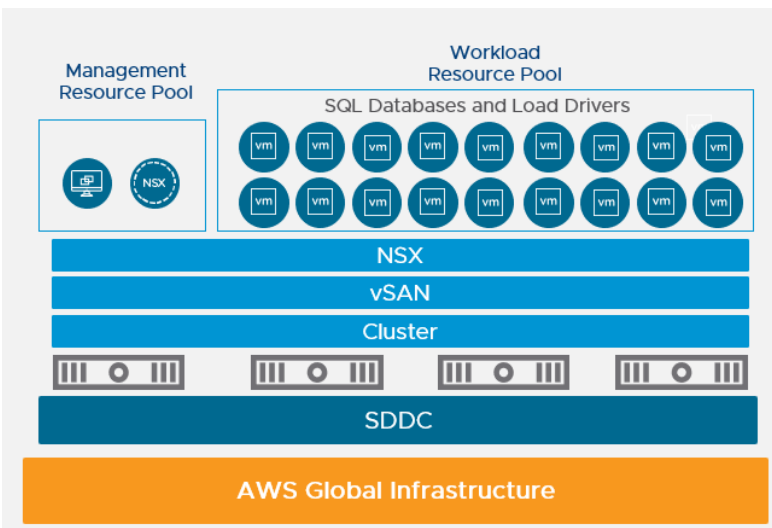

Figure 4. Cloud testbed configuration

While beyond the scope of this paper, it should be noted that our VMware Cloud on AWS solution consists of tools to ease the transition to the cloud, namely Hybrid Linked Mode, which allows us to vMotion VMs from our existing on-premises datacenter to VMware Cloud on AWS with ease. For more information, see the VMware Cloud on AWS Getting Started documentation [3].

Database Benchmarks

We used two online transaction processing (OLTP) benchmark workloads to verify SQL Server performance: HammerDB (for small VMs) and CDB (for large VMs).

HammerDB [4] is an open-source database load testing and benchmarking tool. It supports SQL Server, Oracle, and many other databases. It implements a workload like TPC-C, and reports throughput in **transactions per minute (TPM).** The TPC-C specification has been around since 1992, and thus is considered the "gold standard" of OLTP workloads.

CDB (Cloud Database Benchmark) is a database schema and workload mix designed by Microsoft. The benchmark is designed with cloud computing in mind, but the database schema, data population, and transactions have been designed to be broadly representative of the basic elements most commonly used in OLTP workloads. The benchmark driver reports throughput in terms of **transactions per second (TPS).** Since the benchmark is relatively new, the workload was designed to simulate heavier loads on the database with each simulated user/thread. We were first introduced to this workload during our project to measure performance of SQL Server 2017 on Linux using vSphere 6.5 [5].

It is important to note that since these two benchmarks have different workloads and database schemas, the results are not directly comparable.

Virtual Machine Configurations

We installed Windows Server 2016 as the guest operating system (OS) for our workload VMs (both load-driving clients and database servers). SQL Server 2017 Enterprise Edition was the database engine used within all database server VMs. The SQL Server databases were built to 100 GB in size for both HammerDB and CDB. We adhered to the Best Practices Guide for Microsoft SQL Server [6], and do not have any specific additional caveats for deploying SQL Server within VMware Cloud on AWS.

We configured all load driver VMs with 4 virtual CPUs (vCPUs) and 4 GB of virtual RAM; the HammerDB database servers with 8 vCPUs and 32 GB of RAM; and the CDB database servers with vCPUs equal to the number of cores of the physical host.

The VMs used the VMXNET3 virtual network adapter and paravirtual SCSI (PVSCSI) adapters. We assigned data and log disks to separate PVSCSI adapters.

Both benchmarks were run with an increasing number of simulated users until all of the database server VMs' vCPUs were fully saturated. This represented the maximum throughput that the test environment could achieve.

Database Performance Results (HammerDB)

For HammerDB, we ran the 8 vCPU/32 GB database VMs in a scale-out fashion. We started with 1 database VM, then ran 2 database VMs simultaneously, again for 4, and so on. Note that for every database VM, we had a 4 vCPU/4 GB load driver VM. vSphere's Distributed Resource Scheduling (DRS) was especially useful, as it intelligently migrated the VMs via vMotion to lesser-utilized hosts within the cluster.

The results show good scaling within VMware Cloud on AWS. With 1 database VM, we achieved 864,479 TPM, scaling out to over **6.7 million TPM** with 16 VMs (Figure 5).

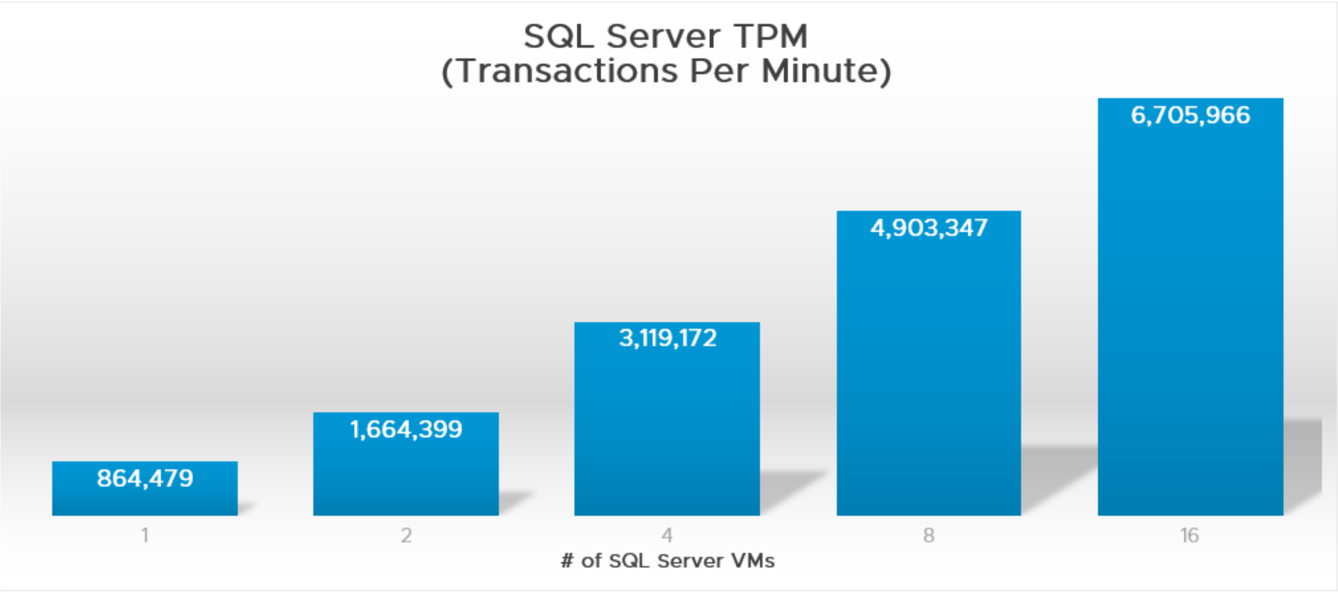

Figure 5. SQL Server VM scale-out performance with HammerDB

We then wanted to see how these results in the cloud would look compared to an on-premises vSphere environment. Unfortunately, due to storage capacity issues, we were not able to scale the on-premises testbed to the same degree, but the 1, 2, and 4-VM results are shown in Figure 6.

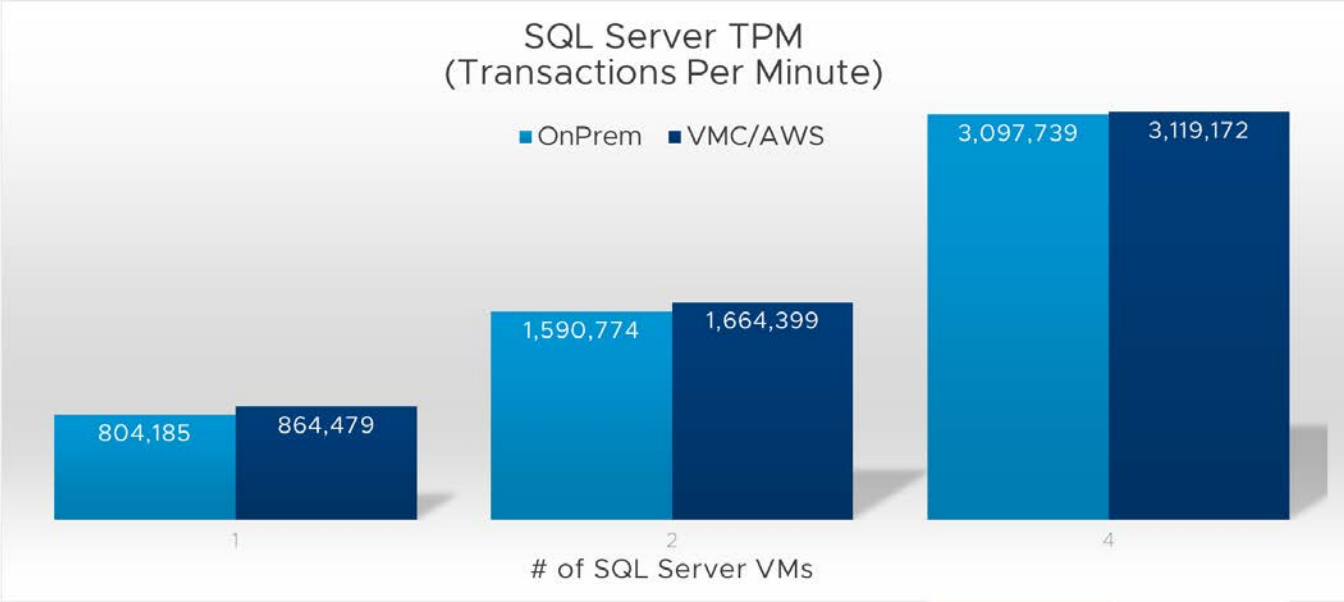

Figure 6. SQL Server VM scale-out performance with HammerDB

We can see from these HammerDB results that the performance is virtually identical between the on-premises and the VMware Cloud on AWS environments. The slight performance edge of the latter is due to the hosts' faster processors in the cloud environment; as mentioned earlier, we did not have the luxury of configuring the two environments identically.

Database Performance Results (CDB)

Microsoft's CDB (Cloud Database Benchmark) represents a heavier, more up-to-date OLTP workload that is specifically tailored to benchmarking private and public clouds. This allowed us to configure the database VMs to be much larger; namely, we were able to increase the number of vCPUs from 8 to the number of physical cores in the host, and the virtual RAM from 32 GB to the amount of virtual RAM in the host (512 GB). We often refer to these as "wide" or "monster" VMs, because they are larger than average VMs and consume most, if not all, of the host's resources. For reference, we did try running HammerDB with the larger VMs, but did not see the benchmark/database scale as expected. Figure 7 shows a side-by-side comparison of the CDB results we saw when we compared the on-premises environment to VMware Cloud on AWS.

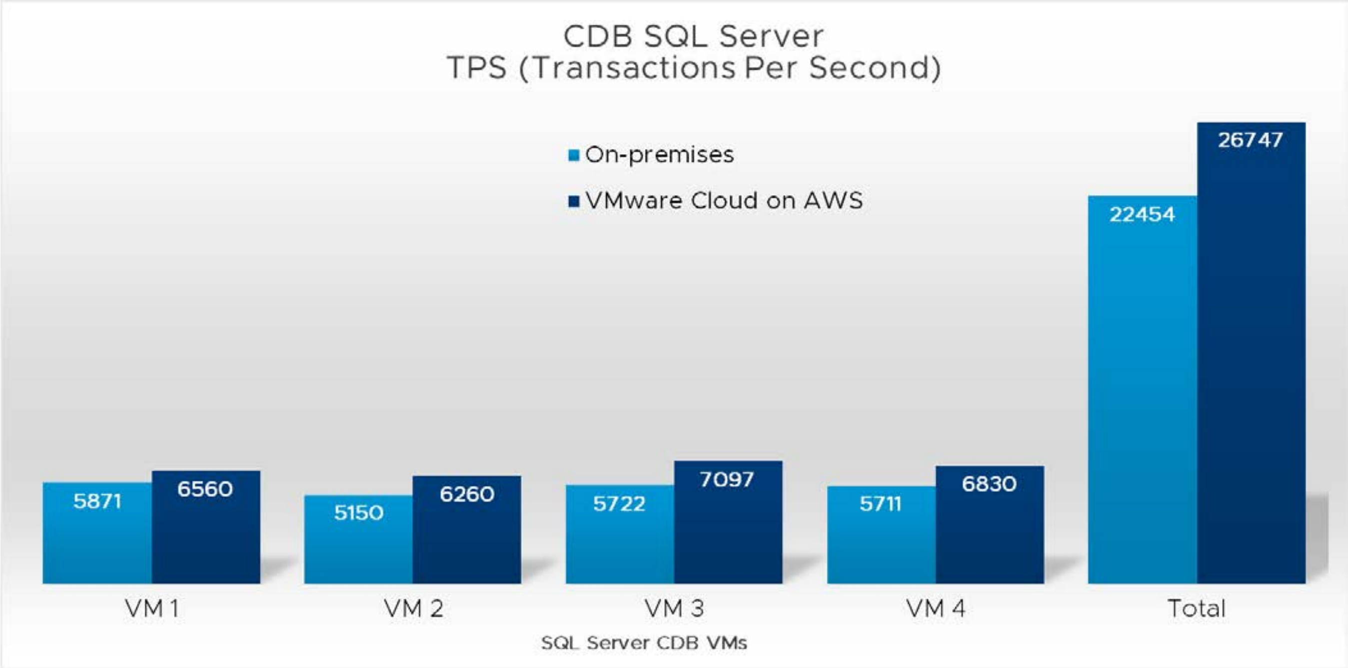

Figure 7. SQL Server "monster VM" performance with CDB

The CDB benchmark results showed similar performance across both on-premises and cloud environments, though the VMs were consistently higher-performing in the VMware Cloud on AWS case. Had the two environments been identical from a hardware perspective, we would have expected the results to be indistinguishable. Nonetheless, the key takeaway is that customers should not expect to lose any performance by transitioning their SQL database workloads to the public cloud.

Conclusion

The results from this performance study show that customers running SQL Server database workloads within vSphere virtual machines (regardless of size) can expect to attain the high performance that they have come to expect from their on-premises datacenters.

References

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

[2] 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

[3] VMware, Inc. (2018) VMware Cloud on AWS Getting Started.
https://docs.vmware.com/en/VMware-Cloud-on-AWS/services/com.vmware.vmc-aws.getting-started/GUID-3D741363-F66A-4CF9-80EA-AA2866D1834E.html

[4] HammerDB. (2018) HammerDB Benchmark.
http://www.hammerdb.com/

[5] Todd Muirhead (2017, October) Performance of SQL Server 2017 for Linux VMs on vSphere 6.5.
https://blogs.vmware.com/performance/2017/10/performance-sql-server-2017-linux-vms-vsphere-6-5.html

[6] Niran Even-Chen. (2017, March) Microsoft SQL Server on VMware Best Practices Guide.
http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-best-practices-guide.pdf

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