



The bridge to possible

White paper  
Cisco public

# Cisco HyperFlex All-NVMe Systems for Oracle Database: Reference Architecture

July 2020

---

# Contents

Executive summary	3
Cisco HyperFlex HX Data Platform All-NVMe storage with Cascade Lake CPUs	3
Why use Cisco HyperFlex All-NVMe systems for Oracle Database deployments	4
Oracle Database 19c on Cisco HyperFlex systems	5
Solution overview	8
Solution components	9
Solution design	12
Engineering validation	16
Oracle Single Instance Database backup, restore and recovery: Oracle Recovery Manager	30
Duplicate a database using Oracle RMAN	33
Reliability and disaster recovery	35
Disaster recovery	35
Oracle Data Guard	36
Conclusion	40
For more information	41

---

## Executive summary

Organizations of all sizes require an infrastructure for their Oracle Database deployments that provides high performance and enterprise-level reliability for transactional Online Transaction Processing (OLTP) workloads. Oracle Database is a leading Relational Database Management System (RDBMS). It has a large installed base that is now running on outdated hardware architecture that often no longer provides the levels of performance and scale that enterprises require. With the Cisco HyperFlex™ solution for Oracle Database, organizations can implement Oracle databases using a highly integrated solution that scales as business demand increases. This reference architecture provides a configuration that is fully validated to help ensure that the entire hardware and software stack is suitable for a high-performance OLTP workload, enabling rapid deployment of Oracle databases. This configuration meets the industry best practices for Oracle Database in a VMware virtualized environment.

## Cisco HyperFlex HX Data Platform All-NVMe storage with Cascade Lake CPUs

Cisco HyperFlex systems are designed with an end-to-end software-defined infrastructure that eliminates the compromises found in first-generation products. With All-NVMe (Non-Volatile Memory Express) memory storage configurations and a choice of management tools, Cisco HyperFlex systems deliver a tightly integrated cluster that is up and running in less than an hour and that scales resources independently to closely match your Oracle Database requirements. For an in-depth look at the Cisco HyperFlex architecture, see the whitepaper [Deliver Hyperconvergence with a Next-Generation Platform](#).

An All-NVMe storage solution delivers more of what you need to propel mission-critical workloads. For a simulated Oracle OLTP workload, it provides 71 percent more I/O operations Per Second (IOPS) and 37 percent lower latency than our previous-generation All-Flash node. The behavior discussed here was tested on a Cisco HyperFlex system with NVMe configurations, and the results are provided in the “[Engineering validation](#)” section of this document. A holistic system approach is used to integrate Cisco HyperFlex HX Data Platform software with Cisco HyperFlex HX220c M5 All-NVMe nodes. The result is the first fully engineered hyperconverged appliance based on NVMe storage.

- **Capacity storage:** The data platform’s capacity layer is supported by Intel® 3D NAND NVMe Solid-State Disks (SSDs). These drives currently provide up to 64TB of raw capacity per node. Integrated directly into the CPU through the PCI Express (PCIe) bus, they eliminate the latency of disk controllers and the CPU cycles needed to process SAS and SATA protocols. Without a disk controller to insulate the CPU from the drives, we have implemented Reliability, Availability, and Serviceability (RAS) features by integrating the Intel Volume Management Device (VMD) into the data platform software. This engineered solution handles surprise drive removal, hot pluggability, locator LEDs, and status lights.
- **Cache:** A cache must be even faster than the capacity storage. For the cache and the write log, we use Intel Optane™ DC P4800X SSDs for greater IOPS and more consistency than standard NAND SSDs, even in the event of high-write bursts.
- **Compression:** The optional Cisco HyperFlex Acceleration Engine offloads compression operations from the Intel Xeon® Scalable CPUs, freeing more cores to improve virtual machine density, lowering latency, and reducing storage needs. This helps you get even more value from your investment in an All-NVMe platform.

- 
- **High-performance networking:** Most hyperconverged solutions consider networking as an afterthought. We consider it essential for achieving consistent workload performance. That's why we fully integrate a 40-Gbps unified fabric into each cluster using Cisco Unified Computing System™ (Cisco UCS®) fabric interconnects for high-bandwidth, low-latency, and consistent-latency connectivity between nodes.
  - **Automated deployment and management:** Automation is provided through Cisco Intersight™, a Software-as-a-Service (SaaS) management platform that can support all your clusters from the cloud to wherever they reside in the data center to the edge. If you prefer local management, you can host the Cisco Intersight Virtual Appliance, or you can use Cisco HyperFlex Connect management software.

All-NVMe solutions support most latency-sensitive applications with the simplicity of hyperconvergence. Our solutions provide the first fully integrated platform designed to support NVMe technology with increased performance and RAS. This document uses an 8-node Cascade Lake-based Cisco HyperFlex cluster.

## Why use Cisco HyperFlex All-NVMe systems for Oracle Database deployments

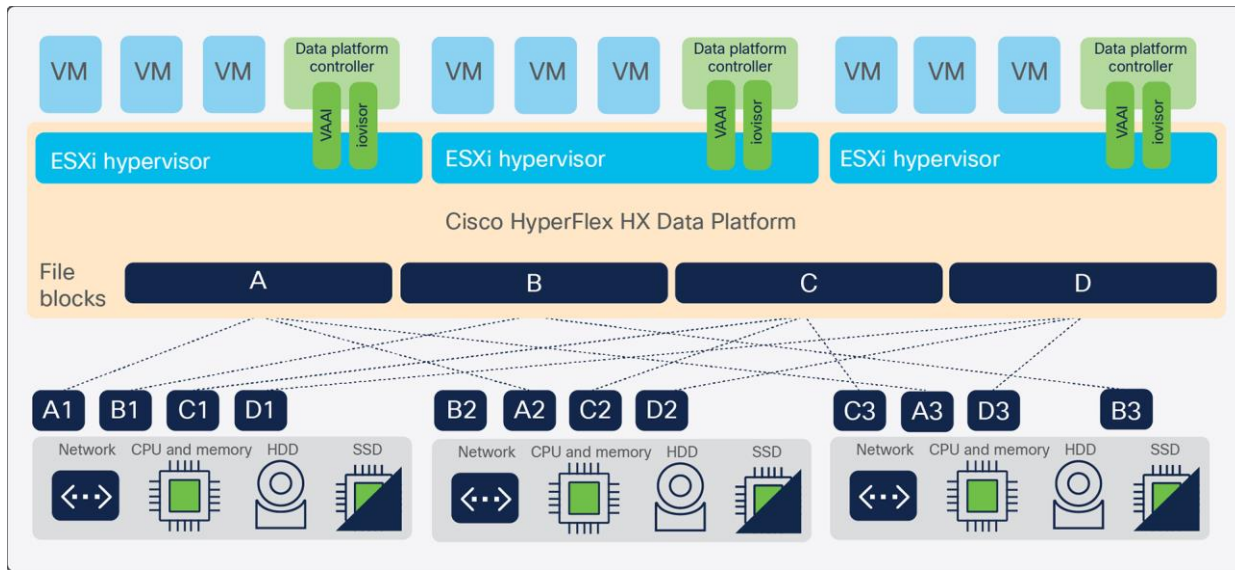
Oracle Databases act as the backend for many critical and performance-intensive applications. Organizations must be sure that they deliver consistent performance with predictable latency throughout the system. Cisco HyperFlex All-NVMe hyperconverged systems offer the following advantages, making them well suited for Oracle Database implementations:

- **High performance:** NVMe nodes deliver the highest performance for mission-critical data center workloads. They provide architectural performance to the edge with NVMe drives connected directly to the CPU rather than through a latency inducing PCIe switch.
- **Ultra-low latency with consistent performance:** Cisco HyperFlex All-NVMe systems, when used to host the virtual database instances, deliver extremely low latency and consistent database performance.
- **Data protection (fast clones, snapshots, and replication factor):** Cisco HyperFlex systems are engineered with robust data-protection techniques that enable quick backup and recovery of applications to protect against failures.
- **Storage optimization (always-active inline deduplication and compression):** All data that comes into Cisco HyperFlex systems is by default optimized using inline deduplication and data-compression techniques.
- **Dynamic online scaling of performance and capacity:** The flexible and independent scalability of the capacity and computing tiers of Cisco HyperFlex systems allow you to adapt to growing performance demands without any application disruption.
- **No performance hotspots:** The distributed architecture of the Cisco HyperFlex HX Data Platform helps ensure that every virtual machine can achieve the storage IOPS capability and make use of the capacity of the entire cluster, regardless of the physical node on which it resides. This feature is especially important for Oracle Database virtual machines because they frequently need higher performance to handle bursts of application and user activity.
- **Nondisruptive system maintenance:** Cisco HyperFlex systems support a distributed computing and storage environment that helps enable you to perform system maintenance tasks without disruption.

## Oracle Database 19c on Cisco HyperFlex systems

This reference architecture guide shows how Cisco HyperFlex systems can provide intelligent end-to-end automation with network-integrated hyperconvergence for an Oracle Database deployment. The Cisco HyperFlex system provides a high-performance, integrated solution for Oracle Database environments.

The Cisco HyperFlex data distribution architecture allows concurrent access to data by reading and writing to all nodes at the same time. This approach provides data reliability and fast database performance. Figure 1 shows the data distribution architecture.



**Figure 1.**  
Data distribution architecture

This reference architecture uses a cluster of eight Cisco HyperFlex HX220cM5 All-NVMe Nodes to provide fast data access. Use this document to design an Oracle Database 19c solution that meets your organization's requirements and budget.

This hyperconverged solution integrates servers, storage systems, network resources, and storage software to provide an enterprise-scale environment for an Oracle Database deployment. This highly integrated environment provides reliability, high availability, scalability, and performance to handle large-scale transactional workloads. This is an Oracle on VMware solution utilizing Oracle Enterprise Linux 7 as the OS across two VM profiles to validate performance, scalability, and reliability and achieving the best interoperability for Oracle databases.

Cisco HyperFlex systems also support other enterprise Linux platforms such as SUSE and Red Hat Enterprise Linux (RHEL). For a complete list of virtual machine guest operating systems supported for VMware virtualized environments, see the [VMware Compatibility Guide](#).

This reference architecture document is written for the following audience:

- Database administrators
- Storage administrators
- IT professionals with the responsibility of planning and deploying an Oracle Database solution

To benefit from this reference architecture guide, familiarity with the following is required:

- Hyperconvergence technology
- Virtualized environments
- SSD and All-NVMe storage
- Oracle Database 19c
- Oracle Automatic Storage Management (ASM)
- Oracle Enterprise Linux

### Oracle Database scalable architecture overview

This reference architecture uses two different Oracle virtual server configuration profiles to validate the scale-up and scale-out architecture of this environment. [Table 1](#) summarizes the two profiles. The use of two different configuration profiles allows additional results and performance characteristics to be observed. The Oracle Medium (MD) and Large (LG) profiles are base configurations commonly used in many production environments. However, this architecture can support different database configurations and sizes to fit your deployment requirements.

**Table 1.** Oracle server configuration profiles

Specification	Oracle Medium (MD)	Oracle Large (LG)
Virtual CPU (vCPU)	4	8
Virtual RAM (vRAM)	64 GB	96 GB
Database size	512 GB	512 GB
Schema	32	64

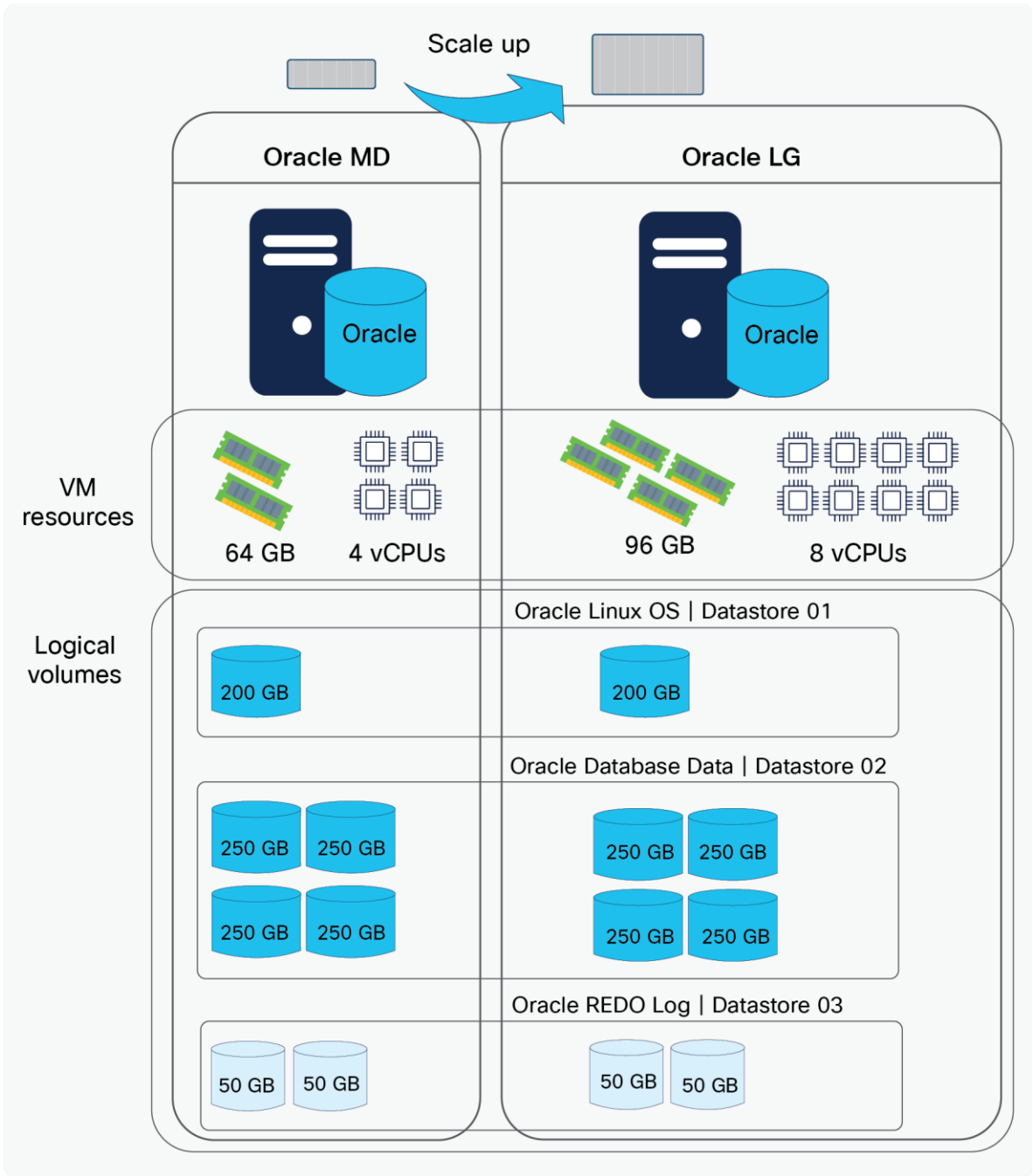
Customers have two options for scaling their Oracle databases: scale up and scale out. Either of these is well suited for deployment on Cisco HyperFlex systems.

#### Oracle scale-up architecture

The scale-up architecture shows the elasticity of the environment, increasing virtual resources to virtual machines on demand as the application workload increases. This model enables the Oracle database to scale up or down for better resource management. In this model, customers can grow the Oracle MD virtual machine into the Oracle LG configuration with no negative impact on Oracle Database data and performance. The scale-up architecture is shown in [Figure 2](#).

The distributed architecture of the Cisco HyperFlex system allows a single Oracle database running on a single virtual machine to consume and properly use resources across all cluster nodes, thus allowing a single database to achieve the highest cluster performance at peak operation. These characteristics are critical for any multitenant database environment in which resource allocation may fluctuate.

At the time of publication of this paper: A Cisco HyperFlex All-NVMe cluster supports up to 16 nodes, with the capability to extend to 48 nodes by adding 32 external computing-only nodes. For more recent information, please refer to the release notes of the corresponding HyperFlex release. This approach allows any deployment to start with a smaller environment and grow as needed, for a "pay as you grow" model.

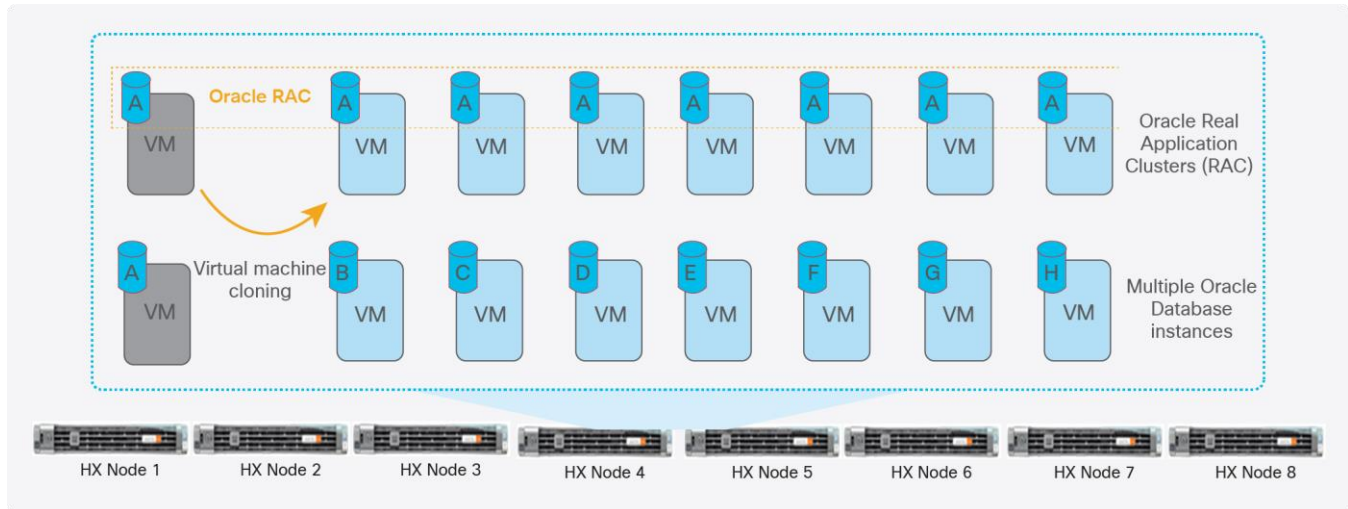


**Figure 2.**  
Oracle scale-up architecture

## Oracle scale-out architecture

The scale-out architecture validates the capability to add Oracle database virtual machines within the same cluster to support scale-out needs such as database consolidation to increase the ROI for infrastructure investment. This model also enables rapid deployment of Oracle Real Application Clusters (RAC) nodes for a clustered Oracle Database environment. [Figure 3](#) shows the scale-out architecture.

Oracle RAC is beyond the scope for this reference architecture. This document focuses on single-instance Oracle Database deployments and validates different sizes of Oracle VMs and the use of several virtual machines running simultaneously.



**Figure 3.**  
Oracle scale-out architecture

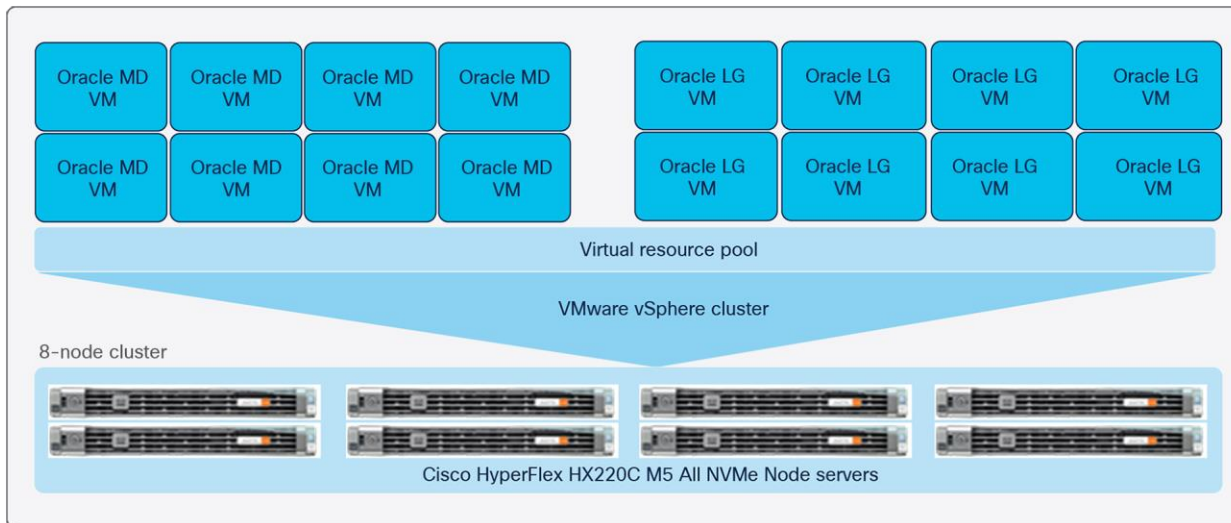
## Solution overview

This section describes how to implement Oracle Database 19c on a Cisco HyperFlex system using an 8-node cluster with a replication factor of 3 (RF=3), which provides failure protection from two simultaneous, uncorrelated failures. Important best-practice recommendations:

- Always use a replication factor of 3 (RF=3) for production deployments for the best resiliency.
- HyperFlex has an ability to automatically heal from component failures (drives, HX nodes, etc.) by regenerating the lost copies of data on the remaining surviving HX nodes. However, this requires that there is sufficient storage capacity in the remaining cluster for the regenerated copies. This should be considered when sizing the cluster.
- Large clusters (8 HX nodes or higher) should use Logical Availability Zones (LAZ) for better failure resiliency.

This reference configuration helps ensure proper sizing and configuration when you deploy Oracle Database 19c on a Cisco HyperFlex system. This solution enables customers to rapidly deploy Oracle databases by eliminating engineering and validation processes that are usually associated with deployment of enterprise solutions. During the validation and testing phases, only virtual machines that are under test are powered on. [Figure 4](#) presents a high-level view of the environment.





**Figure 4.**  
High-level view of solution design

## Solution components

This section describes the components of this solution. [Table 2](#) summarizes the main components of the solution. [Table 3](#) summarizes the Cisco HyperFlex HXAF220c All-NVMe M5 Node configuration for the cluster.

### Hardware components

This section describes the hardware components used for this solution.

#### Cisco HyperFlex systems

The Cisco HyperFlex systems provide next generation hyperconvergence with intelligent end-to-end automation and network integration by unifying computing, storage, and networking resources. The Cisco HyperFlex HX Data Platform is a high-performance, flash-optimized distributed file system that delivers a wide range of enterprise-class data management and optimization services. HX Data Platform is optimized for flash memory, reducing SSD wear while delivering high performance and low latency without compromising data management or storage efficiency. The main features of Cisco HyperFlex systems include:

- Simplified data management
- Continuous data optimization
- Optimization for flash memory
- Independent scaling
- Dynamic data distribution

Visit [Cisco's website](#) for more details about the Cisco HyperFlex HX-Series.

## Cisco HyperFlex HX220cM5 All-NVMe Nodes

Nodes with All-NVMe storage are integrated into a single system by a pair of Cisco UCS 6400 or 6300 Series Fabric Interconnects. Each node includes an M.2 boot drive 240G, an NVMe drive (500G) for data-logging drive, a single Optane NVMe SSD serving as write-log drive, and up to eight 1-TB NVMe SSD drives, for a contribution of up to 8 TB of raw storage capacity. The nodes use Intel Xeon Gold 6248 processor family with Cascade Lake-based CPUs and next-generation DDR4 memory and offer 12-Gbps SAS throughput. They deliver significant performance and efficiency gains and outstanding levels of adaptability in a 1-Rack-Unit (1RU) form factor.

This solution uses eight Cisco HyperFlex HX220c M5 All-NVMe Nodes for an eight-node server cluster to provide two-node failure reliability as the Replication Factor (RF) is set to 3.

See the [Cisco HX220c M5 All NVMe Node data sheet](#) for more information.

## Cisco UCS 6400 Series Fabric Interconnects

The Cisco UCS 6400 Series Fabric Interconnects are a core part of the Cisco Unified Computing System, providing both network connectivity and management capabilities for the system. The Cisco UCS 6400 Series offer line-rate, low-latency, lossless 10/25/40/100 Gigabit Ethernet, Fibre Channel over Ethernet (FCoE), and Fibre Channel functions.

The Cisco UCS 6400 Series Fabric Interconnects provide the management and communication backbone for the Cisco UCS B-Series Blade Servers, UCS 5108 B-Series Server Chassis, UCS Managed C-Series Rack Servers, and UCS S-Series Storage Servers. All servers attached to a Cisco UCS 6400 Series Fabric Interconnect become part of a single, highly available management domain. In addition, by supporting a unified fabric, Cisco UCS 6400 Series Fabric Interconnects provide both LAN and SAN connectivity for all servers within their domain.

From a networking perspective, the Cisco UCS 6400 Series Fabric Interconnects use a cut-through architecture, supporting deterministic, low-latency, line-rate 10/25/40/100 Gigabit Ethernet ports, switching capacity of 3.82 Tbps for the 6454, 7.42 Tbps for the 64108, and 200 Gbps bandwidth between the Fabric Interconnect 6400 series and the IOM 2408 per 5108 blade chassis, independent of packet size and enabled services. The product family supports Cisco low-latency, lossless 10/25/40/100 Gigabit Ethernet unified network fabric capabilities, which increase the reliability, efficiency, and scalability of Ethernet networks. The fabric interconnect supports multiple traffic classes over a lossless Ethernet fabric from the server through the fabric interconnect. Significant TCO savings come from an FCoE-optimized server design in which Network Interface Cards (NICs), Host Bus Adapters (HBAs), cables, and switches can be consolidated.

**Table 2.** Reference architecture components

Hardware	Description	Quantity
<b>Cisco HyperFlex HX220c M5 All NVMe Node servers</b>	Cisco 1RU hyperconverged nodes that allow cluster scaling in a small footprint	8
<b>Cisco UCS 6400 Series Fabric Interconnects</b>	Fabric interconnects	2

**Table 3.** Cisco HyperFlex HX220c All NVMe M5 Node configuration

Description	Specifications	Notes
CPU	2 Intel Xeon Gold 6248 CPU @ 2.50GHz	
Memory	24 × 32-GB DIMMs	
SSD	500-GB NVMe	Configured for housekeeping tasks
	240-GB M.2	Boot drive
	375-GB Optane SSD	Configured as cache
	8x1-TB NVMe SSD	Capacity disks for each node
Hypervisor	VMware vSphere, 6.5.0	Virtual platform for Cisco HyperFlex HX Data Platform software
Cisco HyperFlex HX Data Platform software	Cisco HyperFlex HX Data Platform Release 4.0(2a)	
Replication Factor	3	Failure redundancy from two simultaneous, uncorrelated failures

## Software components

This section describes the software components used for this solution. [Table 4](#) lists the software and versions for the solution.

### VMware vSphere

VMware vSphere helps you get the best performance, availability, and efficiency from your infrastructure while dramatically reducing the hardware footprint and your capital expenses through server consolidation. Using VMware products and features such as VMware ESX, vCenter Server, High Availability (HA), Distributed Resource Scheduler (DRS), and Fault Tolerance (FT), vSphere provides a robust environment with centralized management and gives administrators control over critical capabilities.

VMware provides product features that can help manage the entire infrastructure:

- vMotion: vMotion allows nondisruptive migration of both virtual machines and storage. Its performance graphs allow you to monitor resources, virtual machines, resource pools, and server utilization.
- Distributed Resource Scheduler (DRS): DRS monitors resource utilization and intelligently allocates system resources as needed.
- High Availability (HA): HA monitors hardware and OS failures and automatically restarts the virtual machine, providing cost-effective failover.
- Fault Tolerance (FT): FT provides continuous availability for applications by creating a live shadow instance of the virtual machine that stays synchronized with the primary instance. If a hardware failure occurs, the shadow instance instantly takes over and eliminates even the smallest data loss.

For more information, visit the [VMware website](#).

## Oracle Database 19c

Oracle Database 19c now provides customers with a high-performance, reliable, and secure platform to easily and cost-effectively modernize their transactional and analytical workloads on-premises. It offers the same familiar database software running on premises that enables customers to use the Oracle applications they have developed in-house. Customers can therefore continue to use all their existing IT skills and resources and get the same support for their Oracle databases on their premises.

For more information, visit the [Oracle website](#).

**Note:** The validated solution discussed here uses Oracle Database 19c Release 3. Limited testing shows no issues with Oracle Database 19c Release 3 or 12c Release 2 for this solution.

**Table 4.** Reference architecture software components

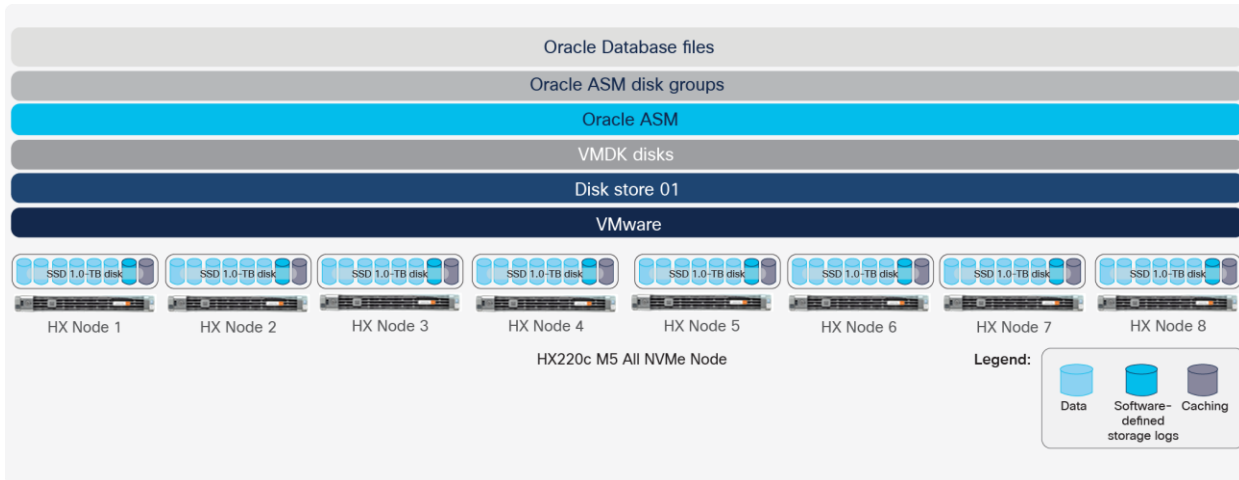
Software	Version	Function
Cisco HyperFlex HX Data Platform	Release 4.0(2a)	Data platform
Oracle Enterprise Linux	Version 7.6	OS for Oracle RAC
Oracle UEK Kernel	4.14.35-1902.10.8.el7uek.x86_64	Kernel version in Oracle Linux
Oracle Grid and Automatic Storage Management (ASM)	Version 19c Release 3	Automatic storage management
Oracle Database	Version 19c Release 3	Oracle Database system
Oracle Silly Little Oracle Benchmark (SLOB)	Version 2.4	Workload suite
Swingbench Suite	Version 2.5	Workload suite

## Solution design

This section describes the architecture for this reference environment. It describes how to implement scale-out and scale-up Oracle Database environments using Cisco HyperFlex HXAF220c M5 All NVMe Nodes on All-NVMe SSDs for Oracle transactional workloads.

### Storage architecture

This reference architecture uses an All-NVMe configuration. The Cisco HyperFlex HXAF220c M5 All NVMe Nodes allow eight NVMe SSDs. However, two per node are reserved for cluster use. NVMe SSDs from all eight nodes in the cluster are striped to form a single physical disk pool. (For an in-depth look at the Cisco HyperFlex architecture, see the Cisco white paper [Deliver Hyperconvergence with a Next-Generation Platform](#)). A logical data store is then created for placement of Virtual Machine Disk (VMDK) disks. The storage architecture for this environment is shown in [Figure 5](#). This reference architecture uses 1-TB NVMe SSDs.



**Figure 5.**  
Storage architecture

**Storage configuration**

This solution uses VMDK disks to create Linux disk devices. The disk devices are used as ASM disks to create the ASM disk groups ([Table 5](#)).

**Table 5.** Oracle ASM disk groups

Oracle ASM disk group	Purpose	Stripe size	Capacity
DATA-DG	Oracle Database disk group	4 MB	1000 GB
REDO-DG	Oracle Database redo group	4 MB	100 GB

**Virtual machine configuration**

This section describes the Oracle Medium (MD) and Oracle Large (LG) virtual machine profiles. These profiles are properly sized for a 512-GB Oracle database ([Table 6](#)).

**Table 6.** Oracle VM configuration

Resource	Details for Oracle MD	Details for Oracle LG
<b>VM specifications</b>	4 vCPUs 64-GB vRAM	8 vCPUs 96 GB vRAM
<b>VM controllers</b>	4 × Paravirtual Small Computer System Interface (PVSCSI) controller	4 × Paravirtual Small Computer System Interface (PVSCSI) controller
<b>VM disks</b>	1 × 200-GB VMDK for VM OS 4 × 250-GB VMDK for Oracle data 2 × 50-GB VMDK for Oracle redo log	1 × 200-GB VMDK for VM OS 4 × 250-GB VMDK for Oracle data 2 × 50-GB VMDK for Oracle redo log

## Oracle Database configuration

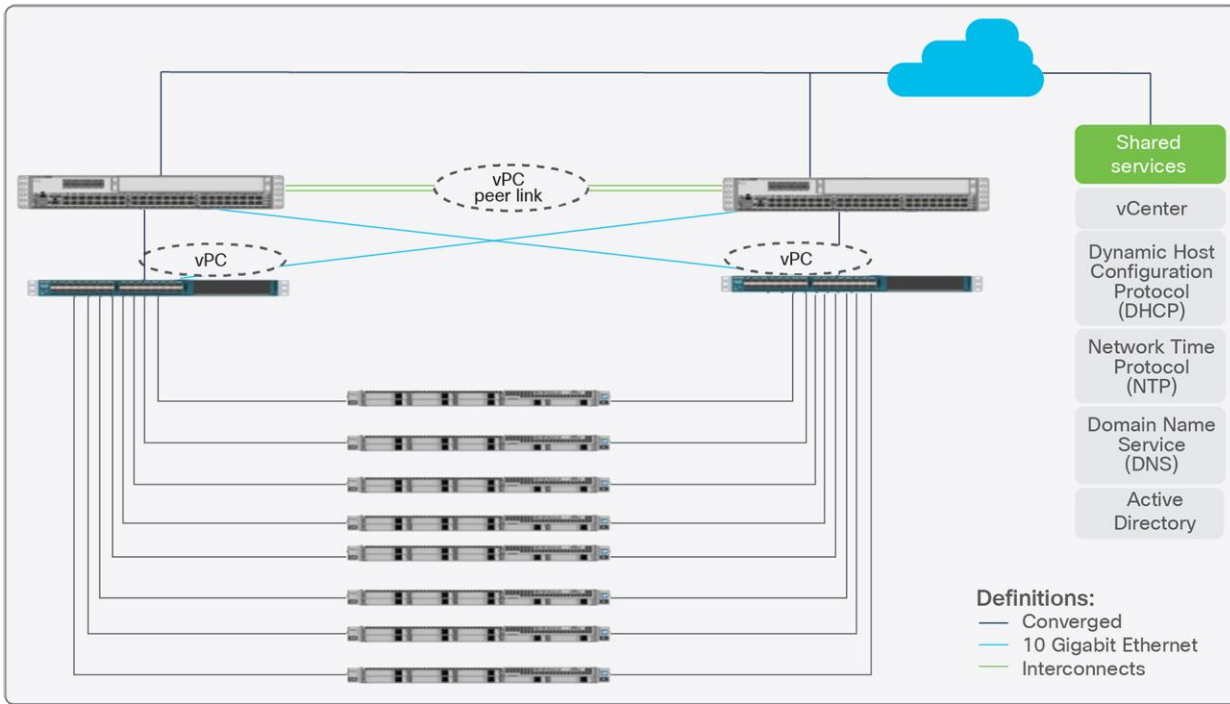
This section describes the Oracle Database configuration for this solution. [Table 7](#) summarizes the configuration details.

**Table 7.** Oracle Database configuration

	Oracle MD	Oracle LG
SGA_TARGET	30 GB	50 GB
PGA_AGGREGATE_TARGET	10 GB	10 GB
Data files placement	ASM and DATA DG	
Log files placement	ASM and REDO DG	
Redo log size	30GB	30GB
Redo log block size	4KB	4KB
Database block	8KB	8 KB

## Network configuration

The Cisco HyperFlex network topology consists of redundant Ethernet links for all components to provide the highly available network infrastructure that is required for an Oracle Database environment. No single point of failure exists at the network layer. The converged network interfaces provide high data throughput while reducing the number of network switch ports. [Figure 6](#) shows the network topology for this environment.



**Figure 6.** Network topology

---

## Storage configuration

For most deployments, a single datastore for the cluster is enough, resulting in fewer objects that need to be managed. The Cisco HyperFlex HX Data Platform is a distributed file system that is not vulnerable to many of the problems that face traditional systems that require data locality. A VMDK disk does not have to fit within the available storage of the physical node that hosts it. If the cluster has enough space to hold the configured number of copies of the data, the VMDK disk will fit because the HX Data Platform presents a single pool of capacity that spans all the hyperconverged nodes in the cluster. Similarly, moving a virtual machine to a different node in the cluster is a host migration; the data itself is not moved.

In some cases, however, additional datastores may be beneficial. For example, an administrator may want to create an additional HX Data Platform datastore for logical separation. Because performance metrics can be filtered to the data-store level, isolation of workloads or virtual machines may be desired. The datastore is thinly provisioned on the cluster. However, the maximum datastore size is set during datastore creation and can be used to keep a workload, a set of virtual machines, or end users from running out of disk space on the entire cluster and thus affecting other virtual machines. In such scenarios, the recommended approach is to provision the entire virtual machine, including all its virtual disks, in the same datastore and to use multiple datastores to separate virtual machines instead of provisioning virtual machines with virtual disks spanning multiple datastores.

Another good use for additional datastores is to assist in throughput and latency in high-performance Oracle deployments. If the cumulative IOPS of all the virtual machines on an ESX host surpasses 10,000 IOPS, the system may begin to reach that queue depth. In ESXTOP, you should monitor the Active Commands and Commands counters, under Physical Disk NFS Volume. Dividing the virtual machines into multiple datastores or increasing the ESX queue limit (the default value is 256) to up to 1024 can relieve the bottleneck.

Another place at which insufficient queue depth may result in higher latency is the SCSI controller. Often the queue depth settings of virtual disks are overlooked, resulting in performance degradation, particularly in high-I/O workloads. Applications such as Oracle Database tend to perform a lot of simultaneous I/O operations, resulting in virtual machine driver queue depth settings insufficient to sustain the heavy I/O processing (the default setting is 64 for PVSCSI). Hence, the recommended approach is to change the default queue depth setting to a higher value (up to 254) as suggested in this [VMware knowledgebase article](#).

For large-scale and high-I/O databases, you always should use multiple virtual disks and distribute those virtual disks across multiple SCSI controller adapters rather than assigning all of them to a single SCSI controller. This approach helps ensure that the guest virtual machine accesses multiple virtual SCSI controllers (four SCSI controllers maximum per guest virtual machine), thus enabling greater concurrency using the multiple queues available for the SCSI controllers.

Another operation that helps reduce application-level latency is to change the amount of time that the virtual machine NIC (vmnic) spends trying to coalesce interrupts. (More details about the parameters can be found [here](#)).

```
esxcli network nic coalesce set -n vmnic3 -r 15 -t 15
```

```
esxcli network nic coalesce set -n vmnic2 -r 15 -t 15
```

---

## Engineering validation

This section describes the methodology used to test and validate this solution. The performance, functions, and reliability of this solution were validated while running Oracle Database in a Cisco HyperFlex environment. The SLOB and Swingbench test kit were used to create and test an OLTP equivalent database workload.

The test includes:

- Test profile of 70-percent read and 30-percent update operations
- Test profile of 100-percent read operations
- Test profile of 50-percent read and 50-percent update operations
- Testing of scale-up options using Medium virtual machine sizes and multiple virtual machines running simultaneously using the SLOB test suite
- Testing of scale-up options using Large virtual machine sizes and multiple virtual machines running simultaneously using the Swingbench test suite

These results are presented to provide some data points for the performance observed during the testing. They are not meant to provide comprehensive sizing guidance. For proper sizing of Oracle or other workloads, please use the Cisco HyperFlex Sizer available at <https://hyperflexsizer.cloudapps.cisco.com/>.

### Test methodology

The test methodology validates the computing, storage, and database performance advantages of Cisco HyperFlex systems for Oracle Database. These scenarios also provide data to help you understand the overall capabilities when scaling Oracle databases.

This test methodology uses the SLOB and Swingbench test suite to simulate an OLTP-like workload. It consists of various read/write workload configuration I/O distribution to mimic an online transactional application.

### Test results

To better understand the performance of each area and component of this architecture, each component was evaluated separately to help ensure that optimal performance was achieved when the solution was under stress.

#### SLOB performance on 8-node HyperFlex All-NVMe servers

The Silly Little Oracle Benchmark (SLOB) is a toolkit for generating and testing I/O through an Oracle database. SLOB is very effective in testing the I/O subsystem with genuine Oracle SGA-buffered physical I/O. SLOB supports testing physical random single-block reads (Oracle db file sequential read) and random single-block writes (DBWR [DataBase WRiter] flushing capability).

SLOB issues single-block reads for the read workload that are generally 8K (because the database block size was 8K). The following tests were performed and various metrics like IOPS and latency were captured along with Oracle AWR reports for each test.

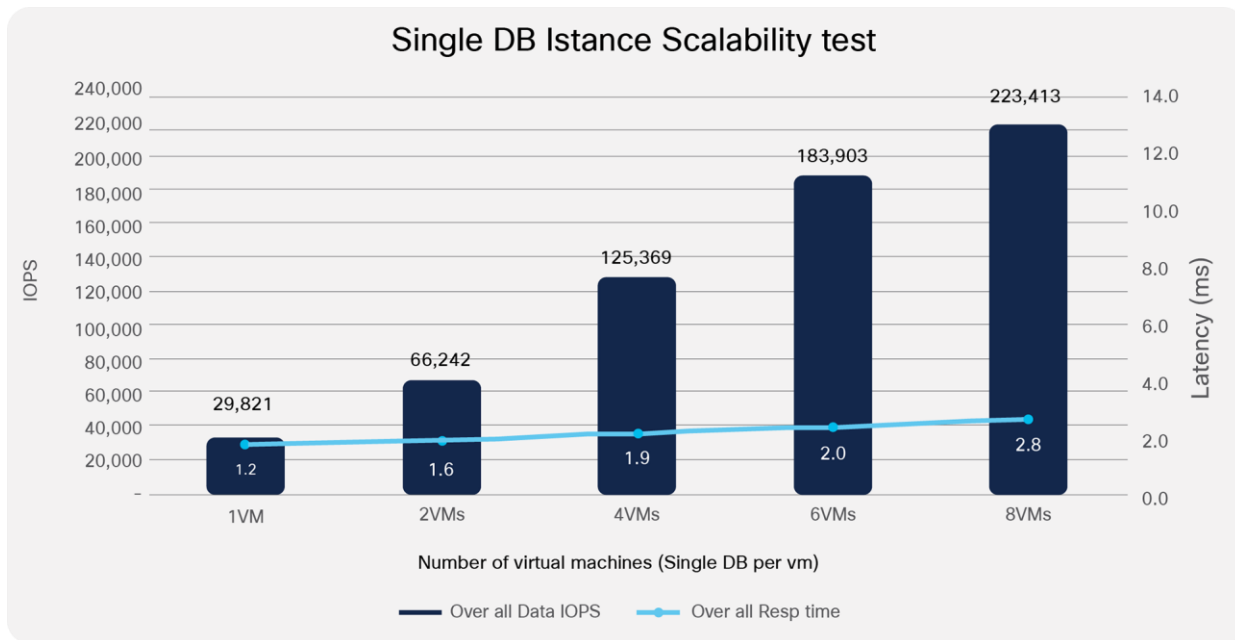


### Single DB Instance-scalability performance using Oracle medium virtual machines

This test uses eight Oracle medium virtual machines to show the capability of the environment to scale as additional virtual machines were used to run the SLOB workload. Scaling is very close to linear. Figure 7 shows the scale testing results. Think time is enabled and set to five in the SLOB configuration file for the below Single DB Instance scalability test.

To validate the Single DB Instance scalability, we ran the SLOB test for the following scenario:

- Single DB Instance scalability testing with one, two, four, six, and eight VMs.
- 70-percent read, 30-percent update workloads.
- 32 users per VM.



**Figure 7.** Single DB Instance scale test results-performance as seen by the application

This feature gives you the flexibility to partition the workload according to the needs of the environment. The performance observed from the above image includes the latency observed at the virtual machine and includes the overhead of the VM device layer, ESXi, and HyperFlex.

---

## Concurrent-workload performance tests using Medium virtual machines

The concurrent-workload test uses eight Oracle Medium virtual machines to test the solution's capability to run many Oracle Single DB Instances concurrently. The test ran for one hour and revealed that this solution stack has no bottleneck and that the underlying storage system can sustain high performance throughout the test. The following figure shows the performance results for the test with different read and write combinations. The Cisco HyperFlex maximum IOPS performance for this specific All-NVMe configuration was achieved while maintaining ultra-low latency. Note that the additional IOPS come at the cost of a moderate increase in latency.

SLOB was configured to run against all eight of the Oracle virtual machines, and the concurrent users were equally spread across all the VMs.

- 32 users per VM
- Varying workloads
  - 100-percent read
  - 70-percent read, 30-percent update
  - 50-percent read, 50-percent update

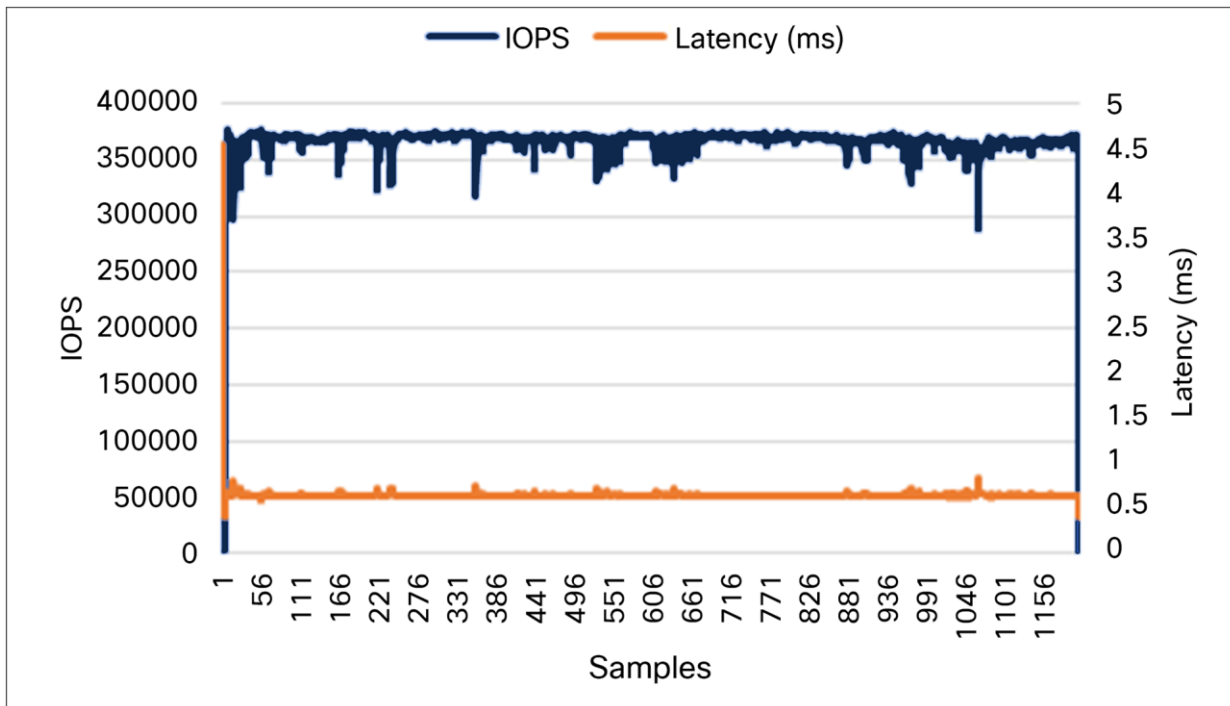
**Note:** The term “Samples” referenced in the figures throughout this paper means the horizontal axis that usually defines a time scale. A line graph is used to show the change in information over a time period and therefore, the horizontal axis in the upcoming graphs define the intervals based on the data points for the entire test duration.

### Case 1: SLOB test at moderate load

[Figures 8, 10, and 12](#) and [Tables 8, 10, and 12](#) show performance of a 100 percent read workload, 70 percent read and 30 percent update, and 50 percent read and 50 percent update, respectively, at a moderate workload and the corresponding latency as observed by the application, that is, measured from the virtual machine. This includes the overhead introduced by the virtual machine device layer and the hypervisor I/O stack over and above the HyperFlex level storage latency.

[Figures 9, 11, and 13](#) and [Tables 9, 11, and 13](#) show the performance of the same workloads (100 percent read, 70 percent read / 30 percent update, and 50 percent read / 50 percent update, moderate workload) at the HX storage layer. The difference in the latency between the two is the additional overhead of the layers about the HyperFlex storage.

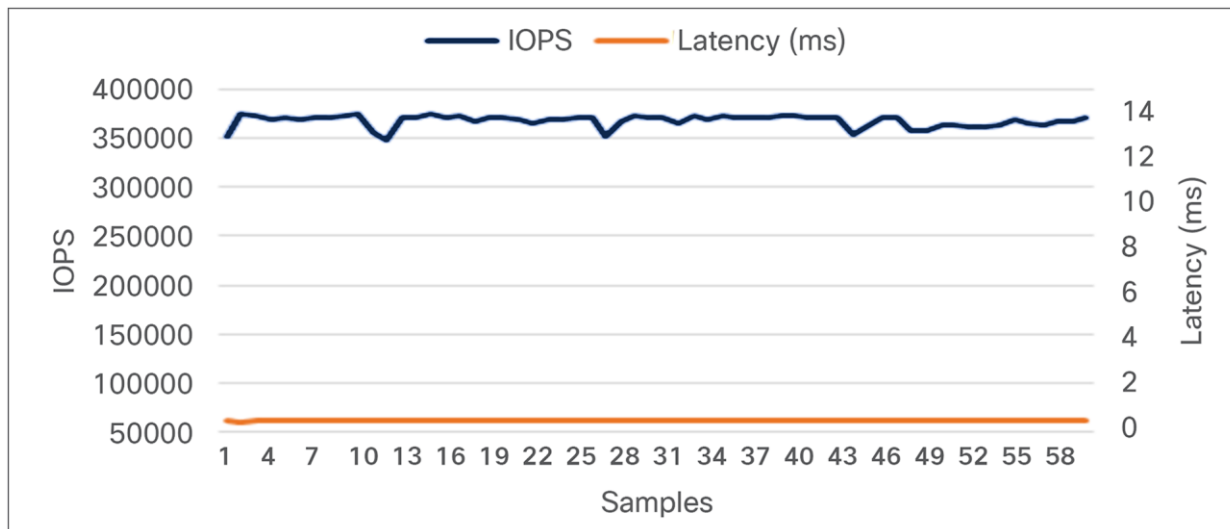
Concurrent workload test with 100-percent read configuration:



**Figure 8.** Concurrent workload test for 100-percent read (performance as seen by the application)

**Table 8.** Concurrent workload application-level performance for 100-percent read

Average IOPS	Average latency (ms)
365,819.4	0.63

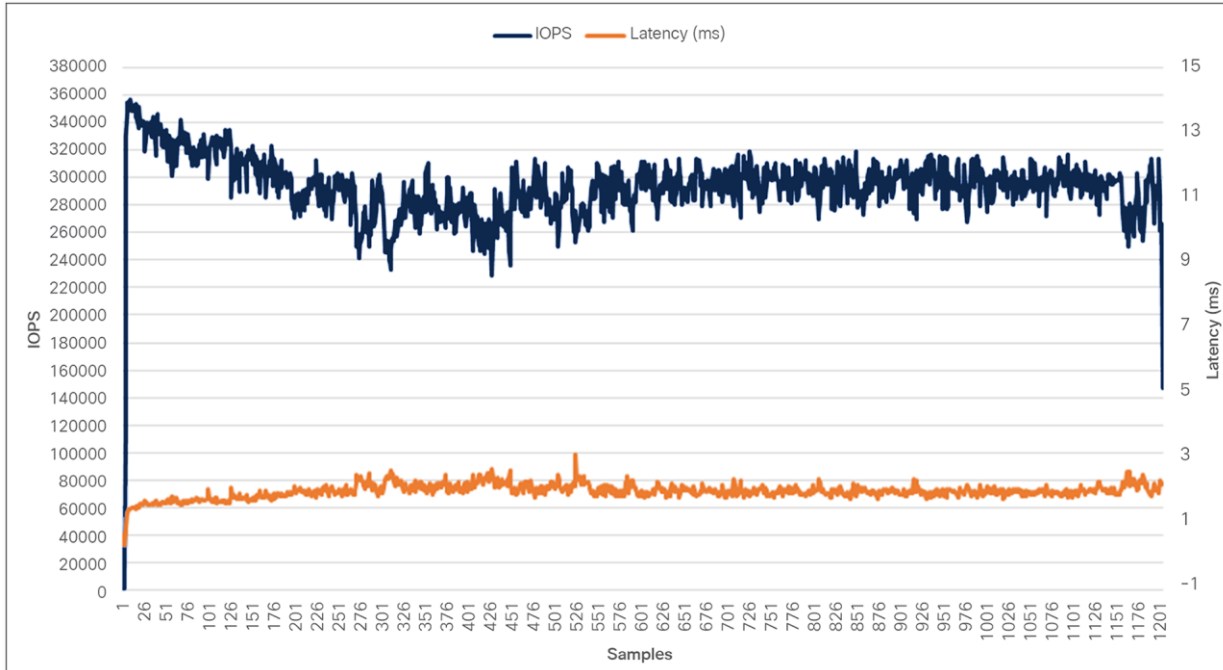


**Figure 9.** HyperFlex Connect cluster performance dashboard for 100-percent read (performance at storage level)

**Table 9.** Concurrent workload cluster-level performance for 100-percent read

Average IOPS	Average latency (ms)
367,474.6	0.48

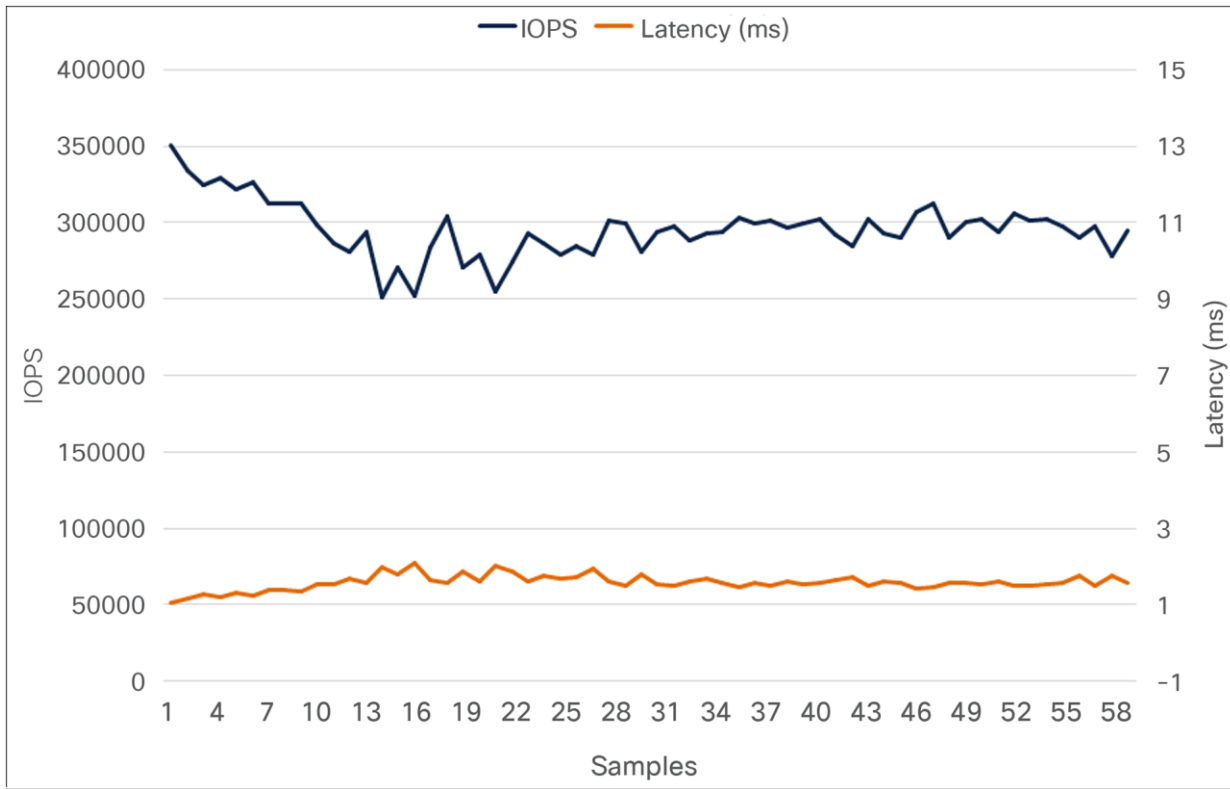
**Concurrent workload test with 70-percent read, 30-percent update configuration:**



**Figure 10.** Concurrent workload test for 70-percent read, 30-percent update (performance as seen by application)

**Table 10.** Concurrent workload application-level performance for 70-percent read, 30-percent update

Average IOPS	Average latency (ms)
293,271.6	2.0



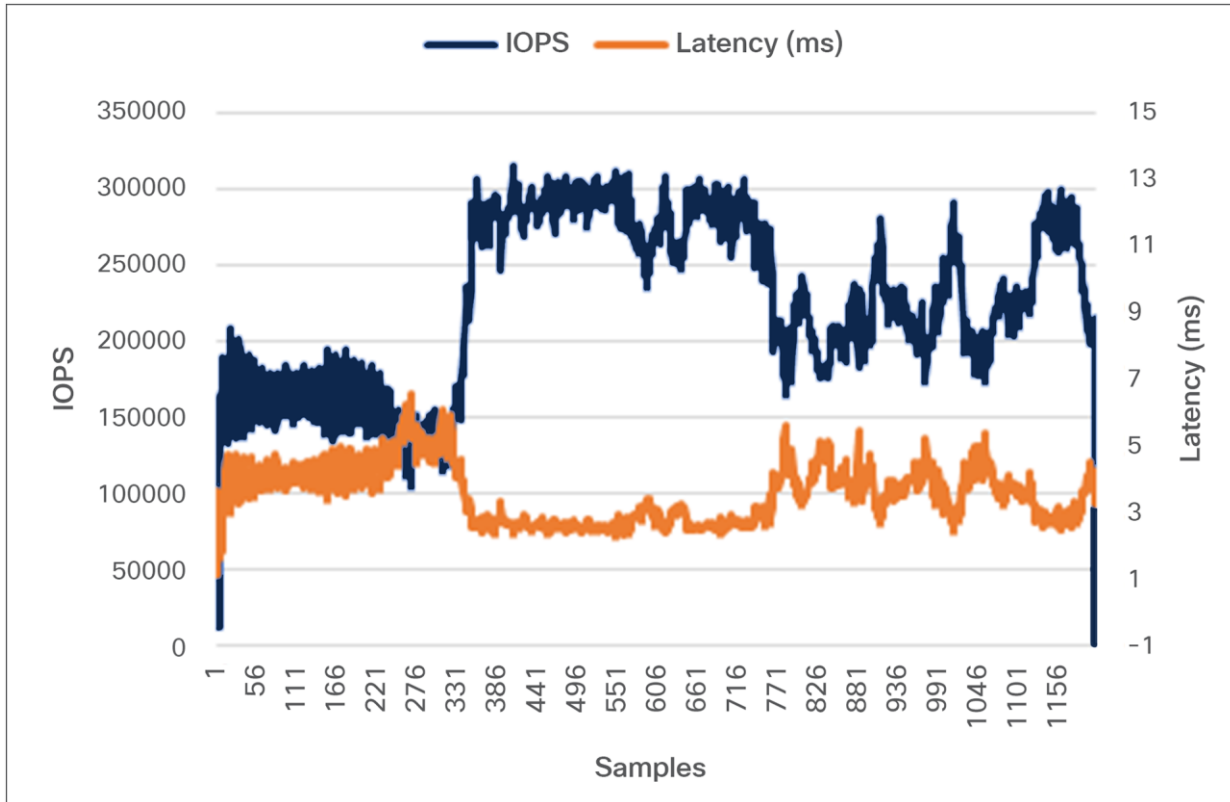
**Figure 11.** HyperFlex Connect cluster performance dashboard for 70-percent read, 30-percent update (performance at storage level)

**Table 11.** Concurrent workload cluster-level performance for 70-percent read, 30-percent update

Average IOPS	Average latency (ms)
295,741.6	1.5

**Concurrent workload test with 50-percent read, 50-percent update configuration:**

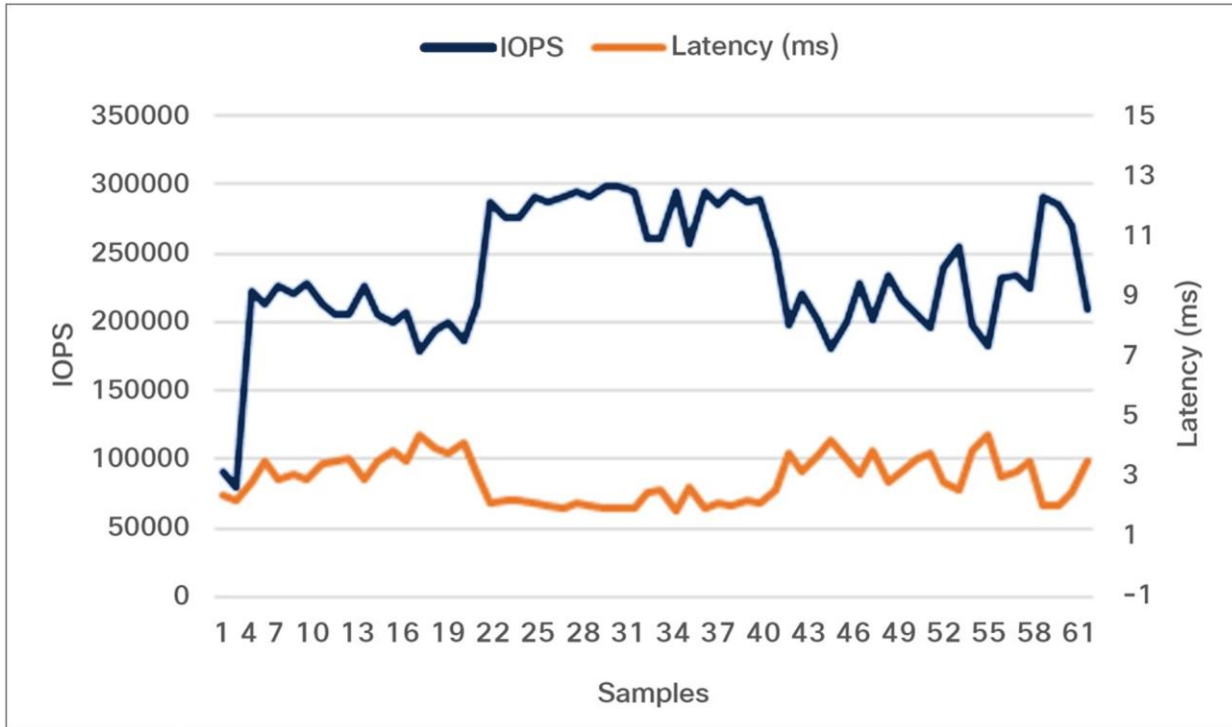
The following graph illustrates the IOPS and latency exhibited at the application level across a 50 percent read and 50 percent update workload. As it's a write-intensive workload, the sudden spike observed at the sample 331 in the figure is due to the completion of write log flushing just before 331 samples (cross-checked using iostat reports by looking at the read/write breakdown during this interval). A similar behavior is also observed at the cluster level performance, as shown in Figure 13.



**Figure 12.** Concurrent workload test for 50-percent read, 50-percent update (performance as seen by application)

**Table 12.** Concurrent workload application-level performance for 50-percent read, 50-percent update

Average IOPS	Average latency (ms)
224,645.2	3.5



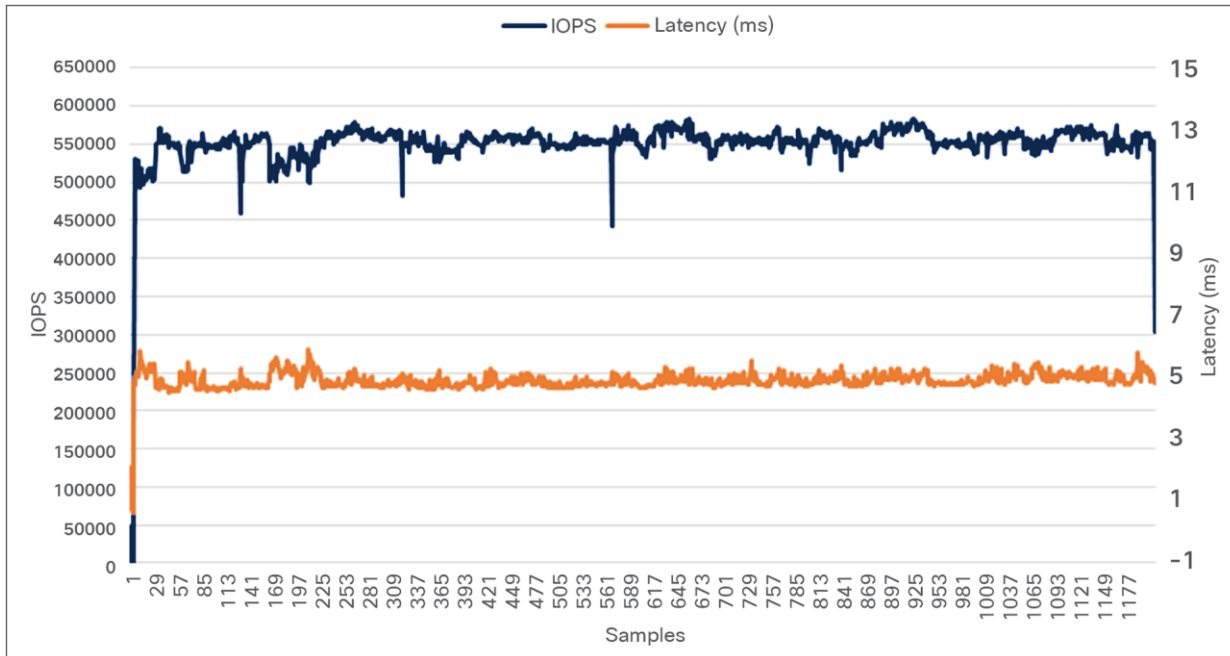
**Figure 13.** HyperFlex Connect cluster performance dashboard for 50-percent read, 50-percent update (performance at storage level)

**Table 13.** Concurrent workload cluster-level performance for 50-percent read, 50-percent update

Average IOPS	Average latency (ms)
235,565.2	2.9

**Case 2: SLOB test at higher load**

Concurrent workload test with 100-percent read configuration:



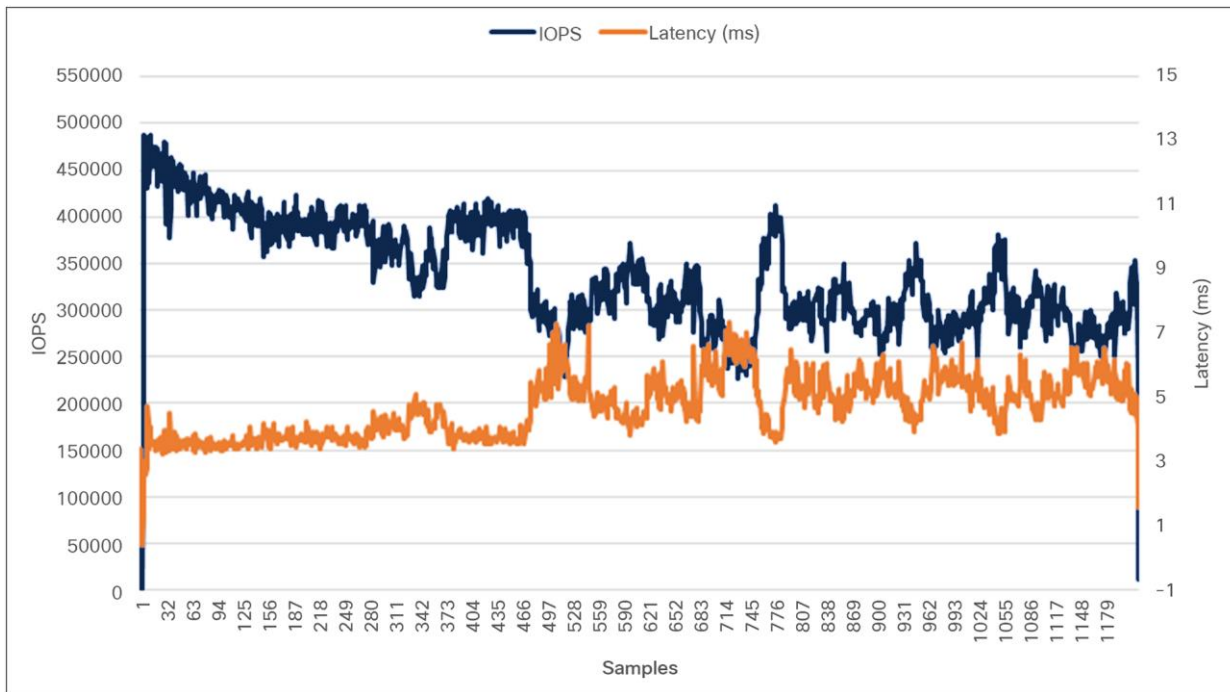
**Figure 14.** Concurrent workload test for 100-percent read (performance as seen by application)

**Table 14.** Concurrent workload application-level performance for 100-percent read

Average IOPS	Average latency (ms)
551,205.3	4.9



**Concurrent workload test with 70-percent read, 30-percent update configuration:**

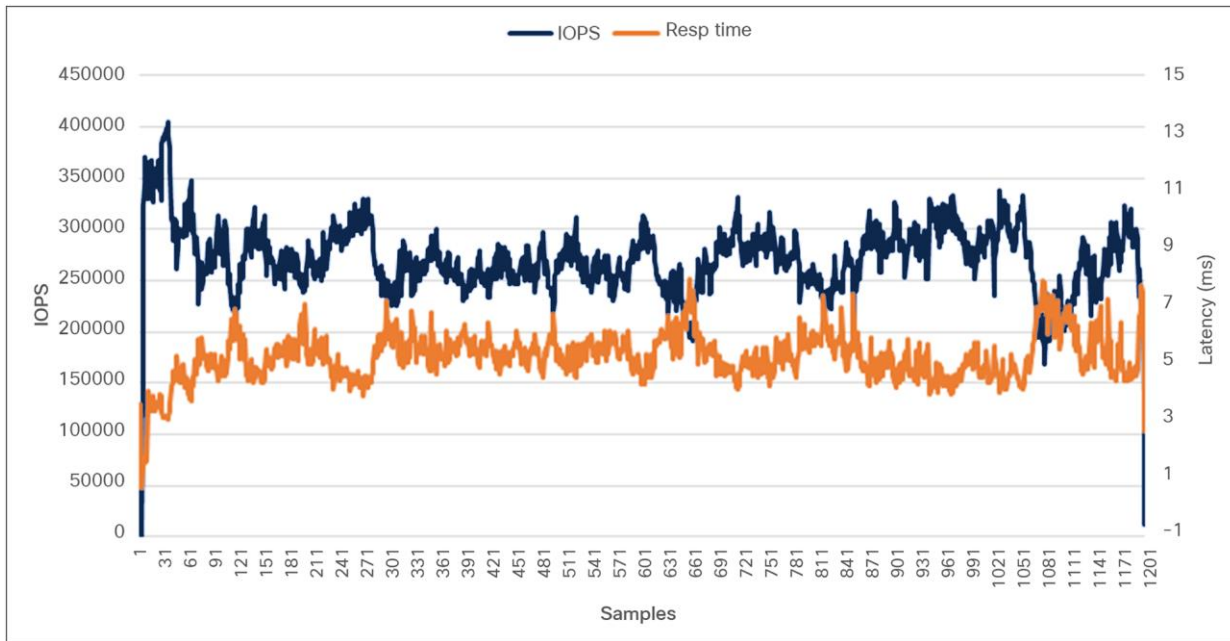


**Figure 15.** Concurrent workload test for 70-percent read, 30-percent update (performance as seen by application)

**Table 15.** Concurrent workload application-level performance for 70-percent read, 30-percent update

Average IOPS	Average latency (ms)
<b>336,181.0</b>	4.5

**Concurrent workload test with 50-percent read, 50-percent update configuration:**



**Figure 16.** Concurrent workload test for 50-percent read, 50-percent update (performance as seen by application)

**Table 16.** Concurrent workload application-level performance for 50-percent read, 50-percent write

Average IOPS	Average latency (ms)
271,692.9	5.1

This test demonstrates the unique capability of the Cisco HyperFlex system to provide a single pool of storage capacity and storage performance resources across the entire cluster. You can scale your Oracle databases using the following models:

- Scale the computing resources dedicated to the virtual machine running Oracle Database, thus removing a computing bottleneck that may be limiting the total overall performance
- Scale by adding virtual machines running Oracle Database or use the additional performance headroom for other applications

---

## Swingbench performance on 8-node HyperFlex All-NVMe servers

Swingbench is a simple-to-use, free, Java-based tool to generate database workloads and perform stress testing using different benchmarks in Oracle database environments. Swingbench can be used to demonstrate and test technologies such as Real Application Clusters, Online table rebuilds, Standby databases, online backup and recovery, etc.

Swingbench provides four separate benchmarks, namely, Order Entry, Sales History, Calling Circle, and Stress Test. For the tests described in this solution, Swingbench Order Entry benchmark was used for OLTP workload testing and the Sales History benchmark was used for the DSS workload testing.

The Order Entry benchmark is based on SOE schema and is TPC-C-like by types of transactions. The workload uses a very balanced read/write ratio around 60/40 and can be designed to run continuously and test the performance of a typical Order Entry workload against a small set of tables, producing contention for database resources.

For this solution, we created one OLTP (Order Entry) database to demonstrate database consolidation, performance, and sustainability. The OLTP database is approximately 512 GB in size.

Typically encountered in the real-world deployments, we tested a combination of scalability and stress-related scenarios that ran on current 8 Single DB Instances.

### Scalability performance

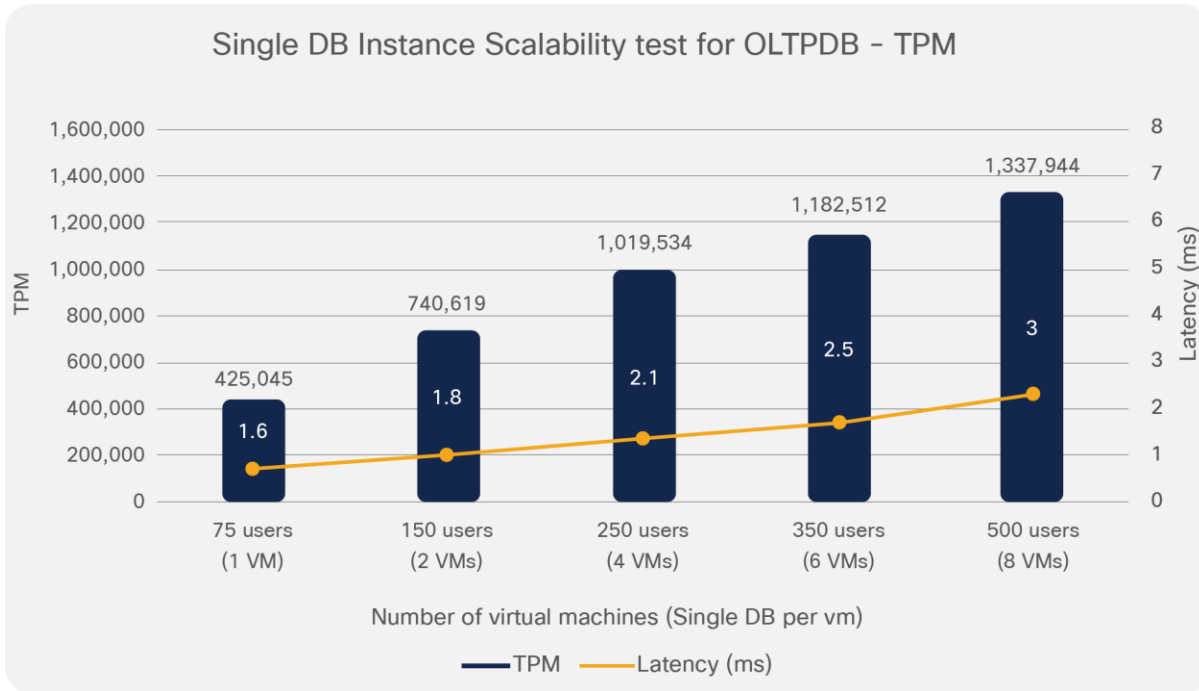
The first step after the databases' creation is calibration, regarding the number of concurrent users, DB Instances, OS and database optimization.

For OLTP database workloads featuring Order Entry schema, we used one OLTPDB database. For the OLTPDB database (512 GB), we used 50GB size of System Global Area (SGA). We also ensured that Huge Pages were in use. The OLTP Database scalability test was run for at least twelve hours and ensured that results were consistent for the duration of the full run.

Run the Swingbench scripts on each Oracle virtual machine to start OLTP1DB database, and generate AWR reports for each scenario, as shown below.

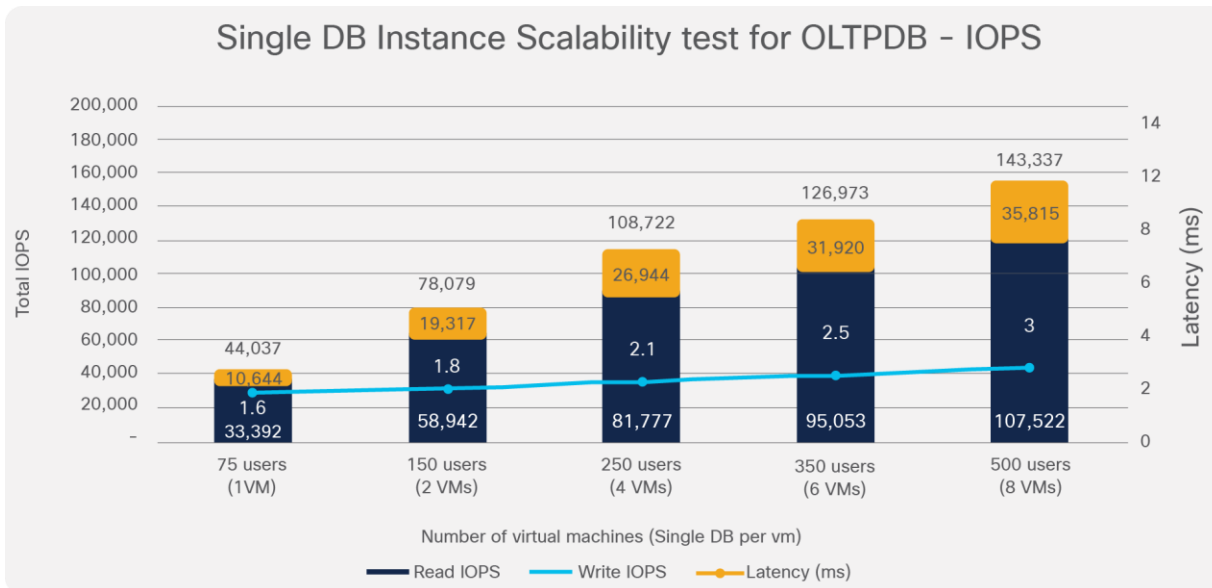
### Single DB Instance-scalability performance using Oracle large virtual machines

The graph in [Figure 17](#) illustrates the TPM for the OLTPDB database Single DB Instance scale. TPM for 1,2,4,6, and 8 virtual machines (single DB per VM) are around 425,000, 740,000, 1,019,000, 1,182,000, and 1,337,000, with latency under 3.1 milliseconds all the time.



**Figure 17.** Single DB Instance scalability TPM for OLTPDB (performance as seen by application)

The graph in [Figure 18](#) illustrates the total IOPS for OLTPDB database Single DB Instance scale. Total IOPS for 1,2,4,6, and 8 virtual machines (single DB per VM) are around 44,000, 78,000, 108,000, 126,000, and 143,000, with latency under 3.1 milliseconds all the time.



**Figure 18.** Single DB instance scalability IOPS for OLTPDB (performance as seen by application)

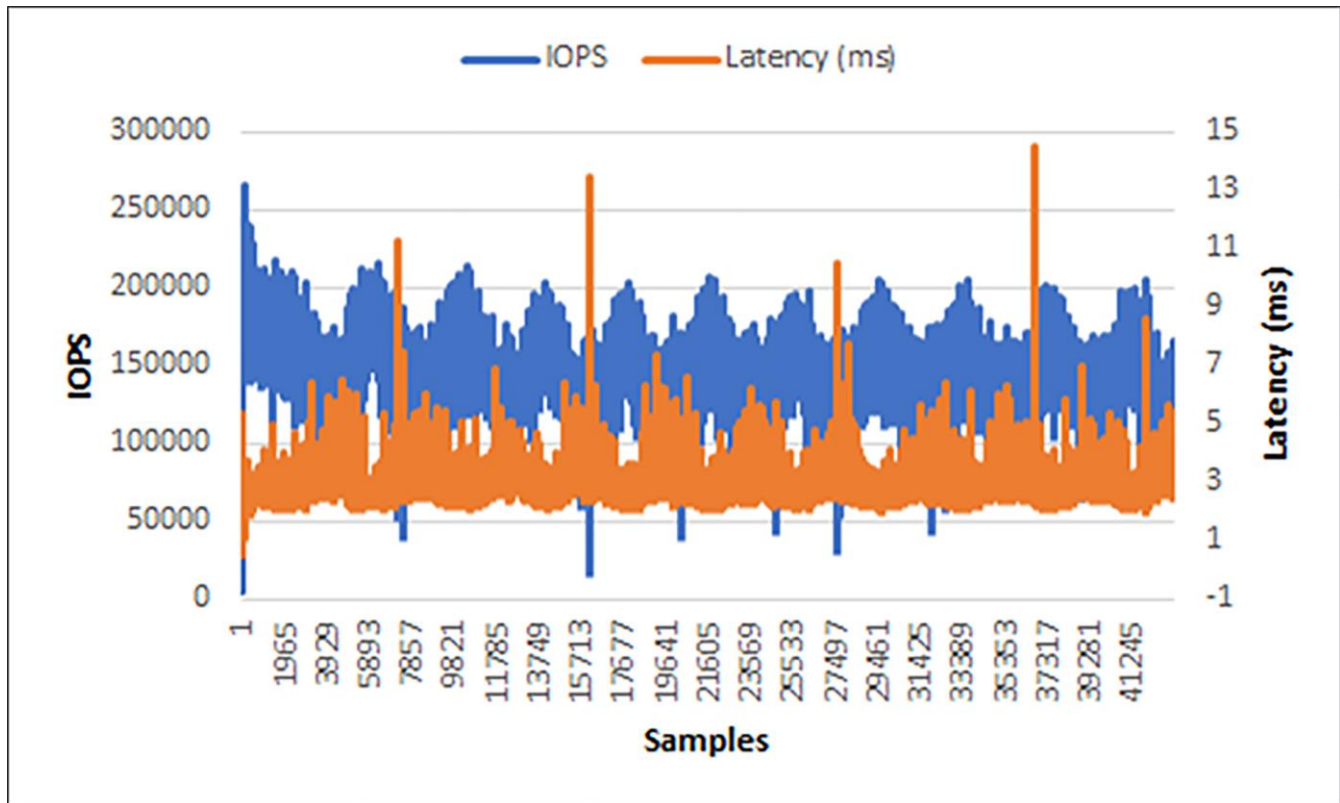
**Concurrent-workload performance tests using Large virtual machines**

[Table 17](#) illustrates the TPM for an OLTPDB database user scale on 8 virtual machines (Single DB per VM). TPM for 64 users (per vm) are around 1,533,000 with latency under 3 milliseconds all the time.

**Table 17.** Concurrent workload application-level performance

TPM	Latency (ms)
1,533,964	2.8

The graph in [Figure 19](#) illustrates the Total IOPS for an OLTPDB database user scale on 8 virtual machines (Single DB per vm). Total IOPS for 64 users (per vm) are around 149,000 IOPS with a latency of 2.8 ms all the time. The latency spikes observed in Figure 19 are due to the long-duration stress tests that were run using a Swingbench workload suite designed to stress-test an Oracle database using an order entry schema (60/40 workload ratio).



**Figure 19.** Concurrent workload test for 60-percent read, 40-percent update (performance as seen by application)

**Table 18.** Concurrent workload application-level performance for 60-percent read, 40-percent update\*\*\*\*

Average IOPS	Avg latency (ms)
149,095.3	2.8

## Oracle Single Instance Database backup, restore and recovery: Oracle Recovery Manager

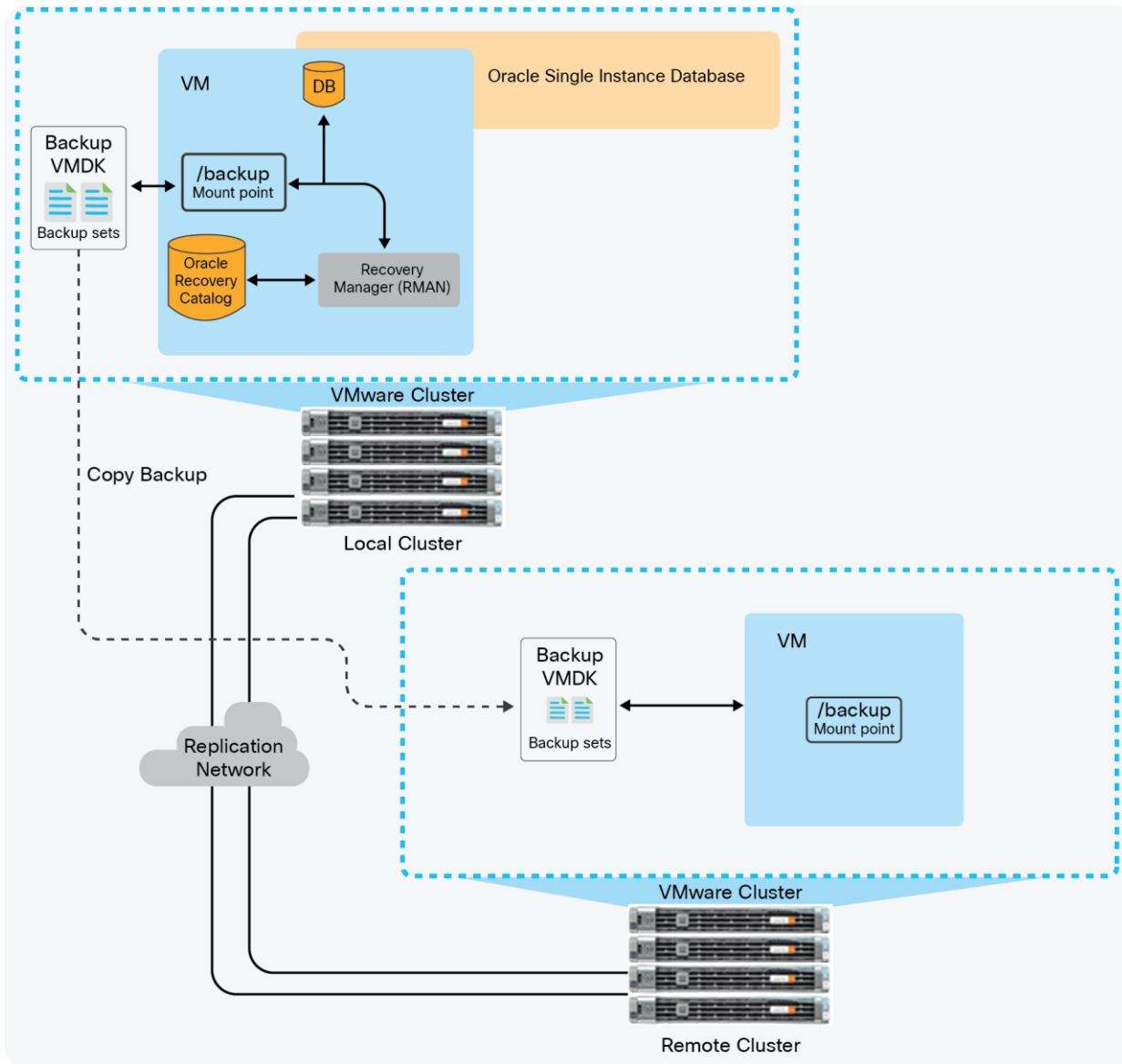
Oracle Recovery Manager (RMAN) is an application-level backup and recovery feature that can be used with Oracle Single Instance Database. It is a built-in utility of Oracle Database. It automates the backup and recovery processes. Database Administrators (DBAs) can use RMAN to protect data in Oracle databases. RMAN includes block-level corruption detection during both backup and restore operations to help ensure data integrity.

### Database backup

In the lab environment, an Oracle Single Instance with a database that is approximately 360 GB in size has been used for testing and RMAN was configured on the single instance. A dedicated 1500-GB VMDK disk was configured as the backup repository under the /backup mount point. By default, RMAN creates backups on disk and generates backup sets rather than image copies. Backup sets can be written to disk or tape. Using RMAN, initiate the full backup creation and provide the mountpoint location to store the backup sets that are generated. Once the backup creation is complete, transfer or copy the backup sets from the local cluster to the remote cluster using an operating system utility such as FTP or a cp command. Afterward, connect RMAN to the source database as TARGET and use the CATALOG command to update the RMAN repository with the location of the manually transferred backups.

The RMAN test was performed running a 32-user Swingbench test workload on the Oracle Single Instance Database. The backup operation did not affect the TPM count in the test. However, CPU use on the Single Instance increased as a result of the RMAN processes.

Figure 20 shows the RMAN environment for the backup testing. The figure shows a typical setup. In this case, the backup target is the mount point /backup residing on the remote cluster. A virtual machine with a backup mount point has been created on a remote host to store the backup sets created on the local cluster. However, if the plan is to duplicate the database in the remote cluster and not just store the backup copies, it needs the exact replica of the test setup that the local cluster consists of. The procedure is explained in detail in the further section under Duplicate Database using RMAN.



**Figure 20.**  
RMAN test environment

Two separate RMAN backup tests were run: one while the database was idle with no user activity, and one while the database was under active testing with 200 user sessions. Table 19 shows the test results.

**Table 19.** RMAN backup results

	During idle database	During active Swingbench testing
<b>Backup type</b>	Full backup	Full backup
<b>Backup size</b>	360 GB	360 GB
<b>Backup elapsed time</b>	00:17:21	00:22:11
<b>Throughput</b>	689 MBps	701.45 MBps

Table 20 shows the average TPM for two separate Swingbench tests: one when RMAN was used to perform a hot backup of the database, and one for a Swingbench test with no RMAN activity. Test results show little impact on the Oracle Single Instance Database during the backup operation.

**Table 20.** Impact of RMAN backup operation

	Average TPM	Average TPS
<b>RMAN hot backup during Swingbench test</b>	145,344	2176
<b>Swingbench only</b>	192,940	3241

**Note:** These tests and several others were performed to validate the functions. The results presented here are not intended to provide detailed sizing guidance or guidance for performance headroom calculations, which are beyond the scope of this document.

### Restore and recover database

To restore and recover an Oracle database from backup, use the Oracle RMAN RESTORE command to restore, validate, or preview RMAN backups. Typically, the restoration is performed when a media failure has damaged a current data file, control file, archived redo log, or before performing a point-in-time recovery.

In order to restore data files to their current location, the database must be started, mounted, or open with the table spaces or data files to be restored offline. However, if you use RMAN in a Data Guard environment, then connect RMAN to a recovery catalog. The RESTORE command restores full backups, incremental backups, or image copies and the files can be restored to their default or a different location.

The purpose for using the RECOVER command is to perform complete recovery of the whole database or one or more restored data files. Or you can use it to apply incremental backups to a data file image copy (not a restored data file) to roll it forward in time, or to recover a corrupt data block or set of data blocks within a data file.

On a high level, the following steps are performed in restoring and recovering the database from RMAN backup:

- Verify the current RMAN configuration on the server where the restore operation is performed and make sure you provide the right backup location
- Restore control file from backup
- Restore the database
- Recover the database



---

## Duplicate a database using Oracle RMAN

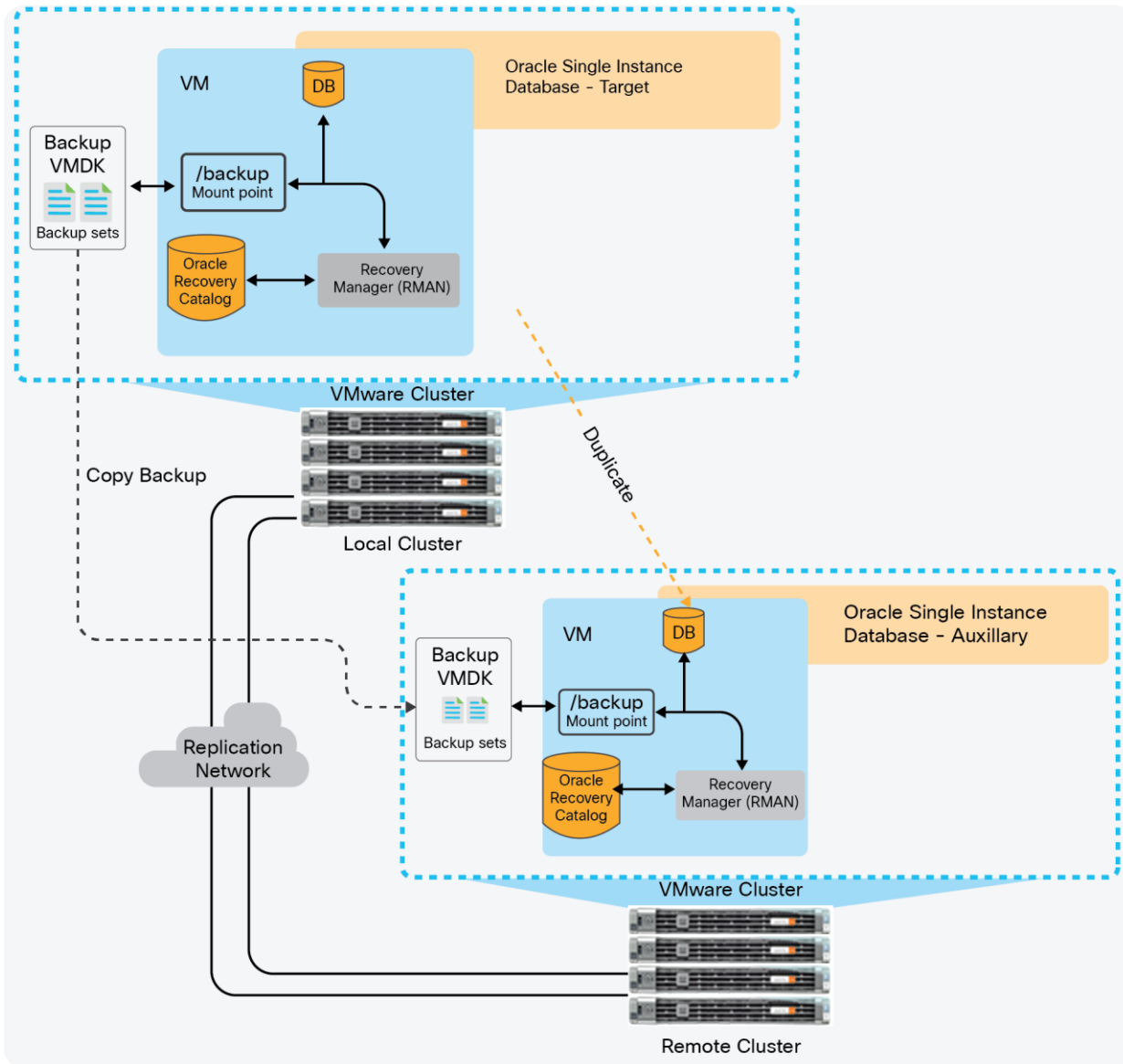
Database duplication is the use of the oracle RMAN DUPLICATE command to copy all or a subset of the data in a source database from a backup. The duplicate command functions entirely independently from the primary database. The tasks that can be performed and useful in a duplicate database environment, most of which involve testing, are: 1) test backup and recovery procedures; 2) test an upgrade to a new release of oracle database; 3) test the effect of applications on database performance; 4) create a standby database; and 5) generate reports. For example, you can duplicate the production database to a remote host and then use the duplicate environment to perform the tasks listed above while the production database on the local host operates as usual.

As part of the duplicating operation, RMAN manages the following:

- Restores the target data files to the duplicate database and performs incomplete recovery by using all available backups and archived logs
- Shuts down and starts the auxiliary database
- Opens the duplicate database with the RESETLOGS option after incomplete recovery to create the online redo logs
- Generates a new, unique DBID for the duplicate database

In the test environment, an Oracle Single Instance with a database approximately 680 GB in size has been used. The backup set created in the local cluster has been copied to the virtual machine hosted on the remote server. In order to prepare for database duplication, the first step would be to create an auxiliary instance, which should be a replica of the production environment. For the duplication to work, connect RMAN to both the target (primary) database and an auxiliary instance started in NOMOUNT mode. If RMAN can connect to both instances, the RMAN client can run on any server. However, all backups, copies of data files, and archived logs for duplicating a database must be accessible by the server on the remote host.

Figure 21 shows the RMAN environment for the database duplication testing. The figure shows a typical setup. In this case, a remote server with the same file structure has been created and the backup target is the mount point /backup residing on the remote cluster. The auxiliary instance (Single Instance Database) on the remote cluster consists of the backup target and RMAN in order to initiate the duplicate process.



**Figure 21.**  
Test environment for database duplication

**Table 21.** RMAN duplicate database results

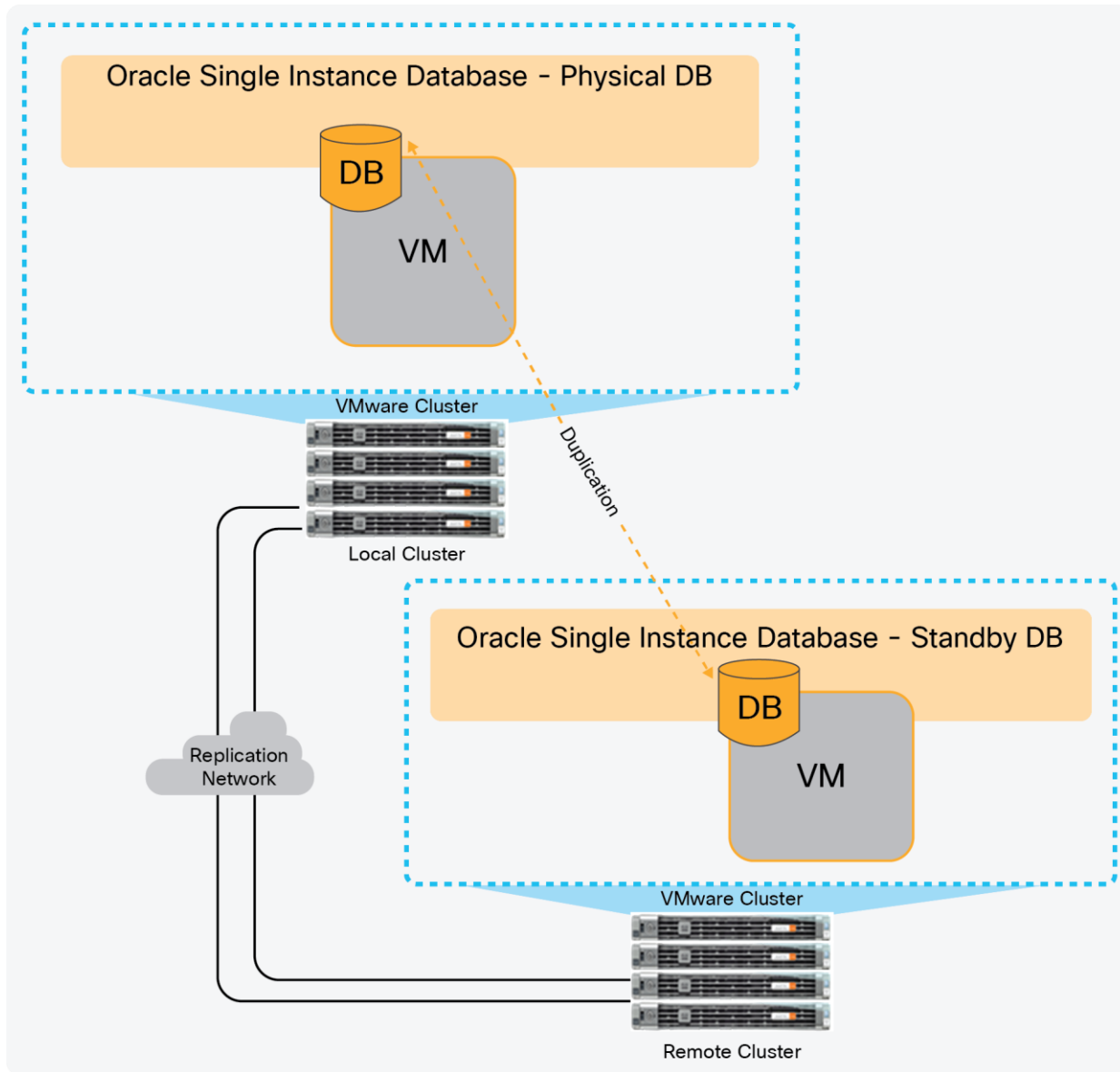
	Duplicate database
<b>Backup type</b>	Full backup
<b>Backup size</b>	673 GB
<b>Duplication elapsed time</b>	00:40:36

## Reliability and disaster recovery

This section describes some additional reliability and disaster-recovery options available for use with Oracle Single Instance databases.

### Disaster recovery

Oracle databases are usually used to host mission-critical solutions that require continuous data availability to prevent planned or unplanned outages in the data center. Oracle Data Guard is an application-level disaster-recovery feature that can be used with Oracle database.



**Figure 22.**  
Database duplication for disaster and reliability recovery

---

## Oracle Data Guard

Oracle Data Guard helps ensure high availability, data protection, and disaster recovery for enterprise data. It provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions. Data Guard maintains these standby databases as copies of the production database. Then, if the production database becomes unavailable because of a planned or an unplanned outage, Data Guard can switch any standby database to the production role, reducing the downtime associated with the outage. Data Guard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability. Data Guard transport services are also used by other Oracle features, such as Oracle Streams and

Oracle Golden Gate, for efficient and reliable transmission of redo logs from a source database to one or more remote destinations

A Data Guard configuration consists of one primary database and up to nine standby databases. The databases in a Data Guard configuration are connected by Oracle Net and may be dispersed geographically. There are no restrictions on where the databases are located if they can communicate with each other. For example, you can have a standby database on the same system as the primary database, along with two standby databases on another system.

The Data Guard broker is the utility used to manage data guard operations. The Data Guard broker logically groups the primary and standby databases into a broker configuration, which allows the broker to manage and monitor them together as an integrated unit. Broker configurations are used from the Data Guard command-line interface. The Oracle Data Guard broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Data Guard configurations. The following list describes some of the operations the broker automates and simplifies:

- Creating Data Guard configurations that incorporate a primary database or a new or existing (physical, logical, or snapshot) standby database, as well as redo transport services and log-apply services, where any of the databases could be Oracle Real Application Clusters (RAC) databases.
- Adding new or existing (physical, snapshot, logical, RAC, or non-RAC) standby databases to an existing Data Guard configuration, for a total of one primary database, and from 1 to 9 standby databases in the same configuration.
- Managing an entire Data Guard configuration, including all databases, redo transport services, and log apply services, through a client connection to any database in the configuration.
- Managing the protection mode for the broker configuration.
- Invoking switchover or failover with a single command to initiate and control complex role changes across all databases in the configuration.
- Configuring failover to occur automatically upon loss of the primary database, increasing availability without manual intervention.
- Monitoring the status of the entire configuration, capturing diagnostic information, reporting statistics such as the redo apply rate and the redo generation rate, and detecting problems quickly with centralized monitoring, testing, and performance tools.

---

The test environment consists of a Single Instance primary database created on the local cluster that is approximately 350 GB in size and a standby database on the remote cluster. Both the primary and standby databases have Oracle installed on them (Oracle Linux 7.6). The primary database server has a running instance whereas the standby server has just only the software installed in it. However, both primary and standby database servers should have proper network configurations set up in the listener config file, and entries in the listener config can be configured using Network Configuration Utility or manually. In the test setup used, network entries are configured manually. Once network and listener configurations are complete, duplicate the active primary database to create a standby database. Create online redo log files on the standby server to match the configurations created on the primary server. In addition to the online redo logs, create standby redo logs on both standby and primary databases (in case of switchovers). The standby redo logs should be at least as big as the largest online redo log and there should be one extra group per thread compared the online redo logs. Once the duplicate database is successful, bring up the standby database instance. Configure the Data Guard broker in both the primary and standby servers to access and manage databases.

**Duplicate the active database from primary database to standby database:**

using target database control file instead of recovery catalog

allocated channel: p1

channel p1: SID=380 device type=DISK

allocated channel: s1

channel s1: SID=660 device type=DISK

Starting Duplicate Db at 01-JUN-20

Starting backup at 01-JUN-20

Finished backup at 01-JUN-20

Starting restore at 01-JUN-20

channel s1: starting datafile backup set restore

channel s1: using network backup set from service dborcl

channel s1: restoring SPFILE

output file name=/u01/app/oracle/product/19.2/db\_home1/dbs/spfileorcl.ora

channel s1: restore complete, elapsed time: 00:00:01

Finished restore at 01-JUN-20

channel s1: using network backup set from service dborcl

channel s1: specifying datafile(s) to restore from backup set

channel s1: restoring datafile 00007 to +DATA

channel s1: restore complete, elapsed time: 00:00:01

Finished restore at 01-JUN-20

Finished Duplicate Db at 01-JUN-20

---

released channel: p1

released channel: s1

RMAN>

**Set up an Oracle Data Guard broker configuration on both the primary and standby databases:**

DGMGRL> show database dborcl

Database - dborcl

Role: PRIMARY

Intended State: TRANSPORT-ON

Instance(s):

orcl

Database Status:

SUCCESS

DGMGRL> show database stbbydb

Database - stbbydb

Role: PHYSICAL STANDBY

Intended State: APPLY-ON

Transport Lag:0 seconds (computed 0 seconds ago)

Apply Lag:0 seconds (computed 0 seconds ago)

Average Apply Rate: 80.00 KByte/s

Real Time Query:OFF

Instance(s):

orcl

Database Status:

SUCCESS

DGMGRL> show configuration

Configuration - dg\_config

Protection Mode: MaxPerformance

Members:

dborcl- Primary database

stbbydb - Physical standby database

Fast-Start Failover:Disabled

Configuration Status:

SUCCESS (status updated 39 seconds ago)

DGMGRL>

**Switch over from the primary database to the standby database:**

DGMGRL> switchover to stdbydb

Performing switchover NOW, please wait...

Operation requires a connection to database "stdbydb"

Connecting ...

Connected to "stdbydb"

Connected as SYSDBA.

New primary database "stdbydb" is opening...

Oracle Clusterware is restarting database "dborcl" ...

Connected to an idle instance.

Connected to an idle instance.

Connected to an idle instance.

Connected to an idle instance.

Connected to an idle instance.

Connected to "dborcl"

Connected to "dborcl"

Switchover succeeded, new primary is "stdbydb"

DGMGRL>

SQL> select name, db\_unique\_name, database\_role, open\_mode from v\$database;

NAME	DB_UNIQUE_NAME	DATABASE_ROLE	OPEN_MODE
ORCL	stdbydb	PRIMARY	READ WRITE

**Note:** These tests and several others were performed to validate the functions. The results presented here are not intended to provide detailed sizing guidance or guidance for performance headroom calculations, which are beyond the scope of this document.

---

## Conclusion

Cisco's extensive experience with enterprise solutions and datacenter technologies enables us to design and build Oracle Database reference architecture on hyperconverged solutions that is fully tested to protect our customers' investments and offer high ROI. Cisco HyperFlex systems enable databases to perform at optimal performance with very low latency: capabilities that are critical for enterprise-scale applications.

Cisco HyperFlex systems provide the design flexibility needed to engineer a highly integrated Oracle Database system to run enterprise applications that meet industry best practices for a virtualized database environment. The balance and distributed data access architecture of Cisco HyperFlex systems support Oracle scale-out and scale-up environments that reduce the hardware footprint while increasing datacenter efficiency.

By deploying Cisco HyperFlex systems with All-NVMe configurations, you can run your database deployments on an agile platform that delivers insight in less time and at lower cost.

The graphs provided in the "[Engineering validation](#)" section of this document show that the All-NVMe storage configuration has higher performance capabilities at ultra-low latency. Thus, to handle Oracle OLTP workloads at very low latency, the All-NVMe solution is a preferred choice over previous-generation All-Flash nodes.

This solution delivers many business benefits, including the following:

- Increased scalability
- High availability
- Reduced deployment time with a validated reference configuration
- Cost-based optimization
- Data optimization
- Cloud-scale operation

Cisco HyperFlex systems provide the following benefits in this reference architecture:

- Optimized performance for transactional applications with very low latency
- Balanced and distributed architecture that increases performance and enhances IT efficiency with automated resource management
- Capability to start with a smaller investment and grow as business demands increase
- Enterprise-application-ready solution
- Efficient data storage infrastructure
- Scalability



---

## For more information

For additional information, consult the following resources:

- Cisco HyperFlex white paper: Deliver Hyperconvergence with a Next-Generation Data Platform:  
<https://www.cisco.com/c/dam/en/us/products/collateral/hyperconverged-infrastructure/hyperflex-hx-series/white-paper-c11-736814.pdf>.
- Cisco HyperFlex systems solution overview:  
<https://www.cisco.com/c/dam/en/us/products/collateral/hyperconverged-infrastructure/hyperflex-hx-series/solution-overview-c22-736815.pdf>.
- Oracle Databases on VMware Best Practices Guide, Version 1.0, May 2016:  
<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/vmware-oracle-databases-on-vmware-best-practices-guide.pdf>.
- HyperFlex All NVMe at-a-glance:  
<https://www.cisco.com/c/dam/en/us/products/collateral/hyperconverged-infrastructure/hyperflex-hx-series/le-69503-aag-all-nvme.pdf>.
- Hyperconvergence for Oracle: Oracle Database and Real Application Clusters:  
<https://www.cisco.com/c/dam/en/us/products/collateral/hyperconverged-infrastructure/hyperflex-hx-series/le-60303-hxsql-aag.pdf>.

**Americas Headquarters**  
Cisco Systems, Inc.  
San Jose, CA

**Asia Pacific Headquarters**  
Cisco Systems (USA) Pte. Ltd.  
Singapore

**Europe Headquarters**  
Cisco Systems International BV Amsterdam,  
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)