

VMWARE
vCENTER SERVER
HIGH AVAILABILITY
PERFORMANCE AND
BEST PRACTICES

VMware vCenter Server 6.5

Table of Contents

Introduction	3
Architecture Overview	3
Client Interaction and Failover	5
Test Environment	6
vCenter Server and Platform Services Controller Configurations.....	6
Test Bed Architecture.....	7
Performance Benchmark: VCbench	7
VCHA Performance Characterization	7
Performance of VCHA Failover/RTO	8
Service Ready Time for vpxd	8
Usability Metrics for vsphere-client.....	9
Performance of Enabling VCHA	10
VCHA Overhead	11
VCbench Throughput.....	11
VCbench Average Operation Latency.....	13
Comparison of Resource Utilization	14
Performance Impact of vCenter Server Statistics Level.....	18
Performance Impact of VCHA Private Network	19
External PSC vs Embedded PSC	21
VCHA Overhead and VCbench Throughput.....	21
Performance of VCHA Failover/RTO	22
Best Practices	23
Enabling VCHA.....	23
VCHA Configuration	23
Network Configuration	23
Conclusion	24
Appendix A - Performance Benchmark: VCbench	25
References	26

Executive Summary

Much work has gone into the high availability feature of VMware vCenter Server® 6.5 to ensure that this service and its operations minimally affect the performance of your vCenter Server and vSphere hosts. In this performance study, we examine comprehensive dimensions of vCenter Server performance when HA is enabled, including:

- **Performance of VCHA failover/recovery time objective (RTO):** In case of a failure, vCenter Server HA (VCHA) provides failover/RTO such that users can continue with their work in less than 2 minutes through API clients and less than 4 minutes through UI clients. While failover/RTO depends on the vCenter Server configuration and the inventory size, in our tests it is within the target limit, which is 5 minutes.
- **Performance of enabling VCHA:** We observed that enabling VCHA would take around 4 - 9 minutes depending on the vCenter Server configuration and the inventory size.
- **VCHA overhead:** When VCHA is enabled, there is no significant impact for vCenter Server under typical load conditions. We observed a noticeable but small impact of VCHA when the vCenter Server was under extreme load; however, it is unlikely for customers to generate that much load on the vCenter Server for extended time periods.
- **Performance impact of vCenter Server statistics level:** With an increasing statistics level, vCenter Server produces less throughput, as expected. When VCHA is enabled for various statistics levels, we observe a noticeable but small impact of 3% to 9% on throughput.
- **Performance impact of a private network:** VCHA is designed to support LAN networks with up to 10 ms latency between VCHA nodes. However, this comes with a performance penalty. We study the performance impact of the private network in detail and provide further guidelines about how to configure VCHA for the best performance.
- **External PSC vs Embedded PSC:** We study VCHA performance comparing these two deployment modes and observe a minimal difference between them.

Throughout the paper, our findings show that vCenter Server HA performs well under a variety of circumstances.

Introduction

VMware vSphere® is a virtualization platform that forms the foundation for building and managing an organization's virtual, public, and private cloud infrastructures. VMware vCenter® Server Appliance™ (vCSA) sits at the heart of vSphere and provides services to manage various components of a virtual infrastructure like ESXi hosts, virtual machines, and storage and networking resources. As large virtual infrastructures are built using vSphere, vCenter Server becomes an important element in ensuring the business continuity of an organization. vCenter Server must protect itself from a set of hardware and software failures in an environment and must recover transparently from such failures. vSphere 6.5 provides a high availability solution for vCenter Server, known as vCenter Server High Availability, or VCHA.

This paper shows:

- **VCHA performance characterization:** A preliminary performance evaluation of the VCHA feature.
- **VCHA best practices:** Some guidelines for VCHA deployment and configuration.

Architecture Overview

The vCenter High Availability architecture uses a three-node cluster to provide availability against multiple types of hardware and software failures. A vCenter HA cluster consists of one Active node that serves client requests, one Passive node to take the role of Active node in the event of failure, and one quorum node called the Witness node. Any Active and Passive node-based architecture that supports automatic failover relies on a quorum or a tie-breaking entity to solve the classic split-brain problem, which refers to data/availability inconsistencies due to network failures within distributed systems maintaining replicated data. Traditional architectures use some form of shared storage to solve the split-brain problem. However, in order to support a

VMWARE vCENTER SERVER HIGH AVAILABILITY PERFORMANCE AND BEST PRACTICES

vCenter HA cluster spanning multiple datacenters, our design does not assume a shared storage-based deployment. As a result, one node in the vCenter HA cluster is permanently designated as a quorum node, or a Witness node. The other two nodes in the cluster dynamically assume the roles of Active and Passive nodes. vCenter Server availability is assured as long as there are two nodes running inside a cluster. However, a cluster is considered to be running in a degraded state if there are only two nodes in it. A subsequent failure in a degraded cluster means vCenter services are no longer available.

A vCenter Server appliance is stateful and requires a strong, consistent state for it to work correctly. The appliance state (configuration state or runtime state) is mainly composed of:

- Database data (stored in the embedded PostgreSQL database)
- Flat files (for example, configuration files).

The appliance state must be backed up in order for VCHA failover to work properly. For the state to be stored inside the PostgreSQL database, we use the PostgreSQL native replication mechanism to keep the database data of the primary and secondary in sync. For flat files, a Linux native solution, *rsync*, is used for replication. Because the vCenter Server appliance requires strong consistency, it is a strong requirement to utilize a synchronous form of replication to replicate the appliance state from the Active node to the Passive node.

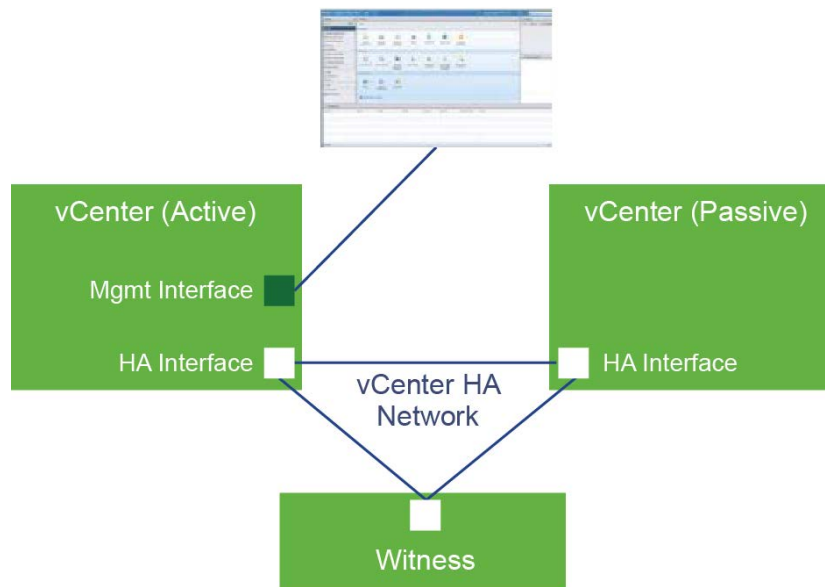


Figure 1. vCenter HA cluster overview.

A design assumption of good network (low latency and high bandwidth) connectivity between the Active and Passive node is made to guarantee zero recovery point objective (RPO).

A vCenter HA cluster requires a vCenter HA network that is separate from the management network for the vCenter Server appliance. As such, 3 FQDNs or static IP addresses are required to be assigned to each node that is used for VCHA cluster traffic on the isolated VCHA network. Clients can have access to the Active vCenter Server appliance via the management network interface, which is public.

Roles for each type of node in a vCenter HA cluster, as shown in Figure 1, are:

- **Active Node:**
 - Node that runs the active instance of vCenter Server.
 - Enables and uses the public IP address of the cluster.
- **Passive Node:**
 - Node that runs as the passive instance of vCenter Server.

- Constantly receives state updates from the Active node in synchronous mode.
- Equivalent to the Active node in terms of resources.
- Takes over the role of Active Node in the event of failover.
- **Witness Node:**
 - Serves as a quorum node.
 - Used to break a tie in the event of a network partition causing a situation where the Active and Passive nodes cannot communicate with each other.
 - A light-weight VM utilizing minimal hardware resources.
 - Does not take over role of Active/Passive nodes.

In the event of the Active vCenter Server appliance failing (due to hardware, software, or network failures), the Passive node appliance takes over the role of the Active node, assumes the public IP address for the cluster, and starts serving client requests. Meanwhile, it is expected that the clients need to re-log into the appliance for continued access. Because the HA solution utilizes synchronous database replication, there will never be any data loss during failover (RPO = 0).

Availability of the vCenter Server appliance works as follows under the following failure conditions:

1. Active node fails:

- As long as the Passive node and the Witness node can communicate with each other, the Passive node will promote itself to Active and start serving client requests.

2. Passive node fails:

- As long as the Active node and the Witness node can communicate with each other, the Active node will continue to operate as Active and continue to serve client requests.

3. Witness node fails:

- As long as the Active node and the Passive node can communicate with each other, the Active node will continue to operate as Active and continue to serve client requests. The Passive node will continue to watch the Active node for failover.

4. More than one node fails or is isolated:

- This means all three nodes—Active, Passive, and Witness—cannot communicate with each other. This is more than a single point of failure and when this happens, the cluster is assumed non-functional and availability is impacted because VCHA is not designed for multiple failures.

5. Isolated node behavior:

- When a single node gets isolated from the cluster, it is automatically taken out of the cluster and all services are stopped. For example, if an Active node is isolated, all services are stopped to ensure that the Passive node can take over as long as it is connected to the Witness node.
- Isolated node detection takes into consideration intermittent network glitches and resolves to an isolated state only after all retry attempts have been exhausted.

Client Interaction and Failover

A client (Web/UI client or any API client) connecting to the vCenter Server appliance will use the public IP address. In the event of failover, the Passive node will take over the exact personality of the failed Active node including the public IP address. The target recovery time objective (RTO) is in minutes, during which clients should be ready to receive failures.

Test Environment

The testbed consisted of ESXi 6.0 hosts managed by vCenter Server Appliance 6.5.

We performed experiments to study the various aspects that depend on the size of the inventory and the resource configuration. Table 1 shows a list of the configurations we used; these were chosen to approximate the supported vCenter Server configuration limits in the [VMware vCenter 6.0 Deployment Guide](#) [1].

- “Powered-On VMs” is 2/3 of “Registered VMs.”
- Each virtual machine was configured with 1 vCPU, 32MB memory, 2 vNICs, and 2 virtual disks.

Size	vCPUs	Memory(GB)	Clusters	Hosts	Registered VMs	Powered-On VMs
Small	4	16	3	96	1440	960
Medium	8	24	13	416	6240	4160
Large	16	32	32	1024	15,360	10,240
X-Large	32	64	64	2048	26,624	36,864

Table 1. Sizing for vCenter Server for performance comparison.

In order to study certain aspects of VCHA, we stressed the system within its supported limits. Unless it is indicated otherwise, we performed the testing and analysis by default with the largest inventory that vCenter Server supports:

- 64 clusters
- 2,000 hosts
- 25,000 powered-on and 35,000 registered VMs

vCenter Server and Platform Services Controller Configurations

With vSphere 6.0, the Platform Services Controller (PSC) component is introduced, which contains common infrastructure services such as vCenter Single Sign-On (SSO), VMware Certificate Authority (VMCA), Certificate Store (VECS), licensing, Lookup Service for Component Registrations and so on.

Basically, two PSC deployment models are supported:

- **External PSC:** vCenter Server and Platform Services Controller are deployed on different virtual machines.
- **Embedded PSC:** vCenter Server and Platform Services Controller are deployed on the same virtual machine.

Refer to the “vCenter Server Deployment Models” section in the [vSphere Upgrade Guide](#) for more information about when to use external or embedded deployment models. [2]

With these PSC deployment models, vSphere 6.0 can be configured for various topologies. Here is the list of [recommended topologies](#) for VMware vSphere 6.0. [3]

Embedded PSC is mostly suitable for small scale environments, while External PSC is for large scale environments. With the help of a load balancer distributing the load across multiple PSCs, the External PSC deployment model provides flexibility to support various topologies. That is why we chose to deploy the External PSC model as our default configuration. We also dedicated a separate section “[External PSC vs Embedded PSC](#)” to study the performance characterization of Embedded PSC mode comparing it with the Embedded PSC.

Test Bed Architecture

vCenter Server (and the VCHA secondary and witness nodes) were installed on Intel Haswell-based servers:

- Dell PowerEdge R730xd
- Intel® Xeon® Processor E5-2680 v3, 2.50 GHz
- 24 physical cores (48 cores with hyper-threading enabled)
- 256GB memory
- Storage: 2 x Dell Express Flash NVMe 400GB SSDs
- Network: 2 x Intel Ethernet Controller 10 Gigabit X540-AT2
 - One 10 Gigabit Ethernet was dedicated to the VCHA private network, which was used for PostgreSQL and file replication on an isolated network.
 - The other is used by vCenter Server to manage all the entities on the testbed.

Performance Benchmark: VCbench

We used a performance benchmark tool, VCbench, to generate background load for the vCenter Server to stress the performance and scalability of the system. We chose a list of management operations from a sampling of representative customer data for the vSphere Web Services SDK to deploy. Please refer to “Appendix A – Performance Benchmark: VCbench” for further details about this benchmark tool.

The number of concurrent operations that are issued to the server is also configurable. While we ran experiments with various levels of concurrency, in this report, we will be focusing mainly on:

- **64-client VCbench load:** This is **the common use case** for most customers.
- **256-client VCbench load:** This is **the worst case scenario**. While it is unlikely for customers to generate that much load on the vCenter Server for extended time periods, we studied the performance characterization of the vCenter Server generating heavy background load to stress the performance and scalability limits of the system.

We relied on the following metrics to measure the performance of vCenter Server:

- VCbench Throughput (operations per minute, or ops/min)
- VCbench Average Operation Latency (milliseconds per operation, or ms/op)

VCHA Performance Characterization

We studied the VCHA performance characterization in terms of:

- Performance of VCHA failover/RTO
- Performance of enabling VCHA
- VCHA overhead
- Performance impact of vCenter Server statistics level
- Performance impact of a private network
- External PSC vs Embedded PSC

Performance of VCHA Failover/RTO

There are mainly two ways for users to interact with vCenter Server:

- VMware vCenter Server (vpxd) through API clients
- vSphere Web Client (vsphere-client) through UI clients

We study the performance of VCHA failover/RTO in terms of:

- Service ready time for the vpxd service
- Usability metrics for the vsphere-client

Service Ready Time for vpxd

When the active VCHA node becomes non-responsive, VCHA detects the failure and initiates a failover process, where the Passive node becomes the Active node. During the failover process, the Passive node takes over and boots up all the dormant services.

Users rely on API clients to interact with the VMware vCenter Server (vpxd), which goes through some initialization phases during boot up. We measured how long it takes to reach a state where vpxd accepts incoming connection requests. We marked this as the service ready time of vpxd, as shown in Figure 2 and Table 2.

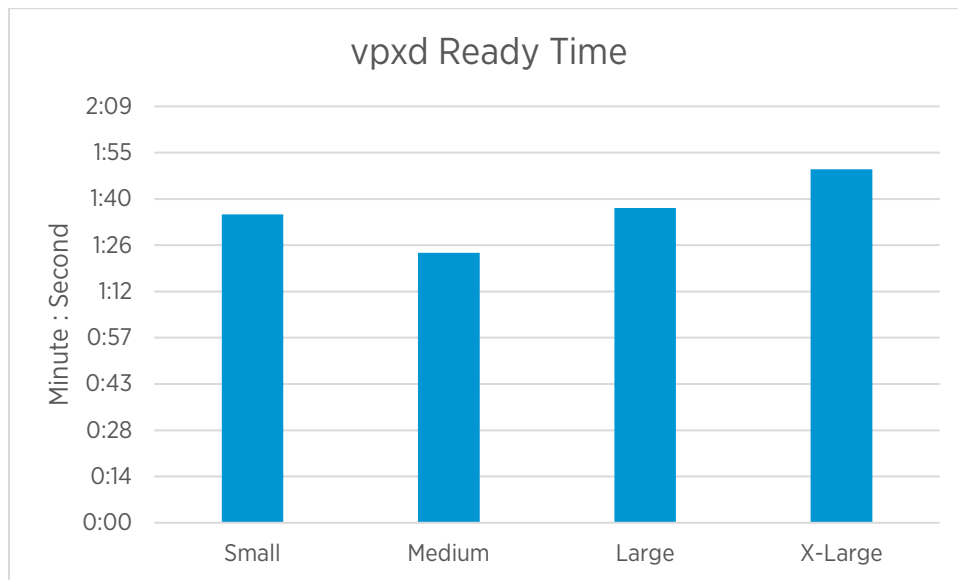


Figure 2. Service ready time of vpxd for various inventory sizes. (Lower is better.)

Size	vpxd Ready Time
Small	1:36
Medium	1:24
Large	1:38
X-Large	1:50

Table 2. Service ready time of vpxd for various inventory sizes. (minute:second)

VMWARE vCENTER SERVER HIGH AVAILABILITY PERFORMANCE AND BEST PRACTICES

The service ready time of vpxd depends on two competing factors:

- **vCenter Server resource configuration:** As the inventory sizes increase, vCenter Server is configured with higher CPU and memory resources. With more CPU resources, vpxd completes the initialization phases during the boot up faster and service ready time becomes faster.
- **vCenter Server inventory size:** As the inventory sizes increase, vpxd takes longer to complete the initialization phases during boot up, thus service ready time becomes slower.

Usability Metrics for vsphere-client

Most users rely on UI clients to interact with vCenter Server through the Web Client (vsphere-client) service. We set up a vSphere Web Client on the vCenter Server Summary page. After the Active vCenter Server was powered off, we manually refreshed the Web Client. After a VCHA failover is completed, the Web Client performs re-login, during which a user would see the following screens in order: blank screen, blue screen, inventory on the left-hand side, and, finally, the vCenter Server Summary page. We measured how long it takes for the Web Client to present these various UI screens after a VCHA failover, as shown in Figure 3 and Table 3.

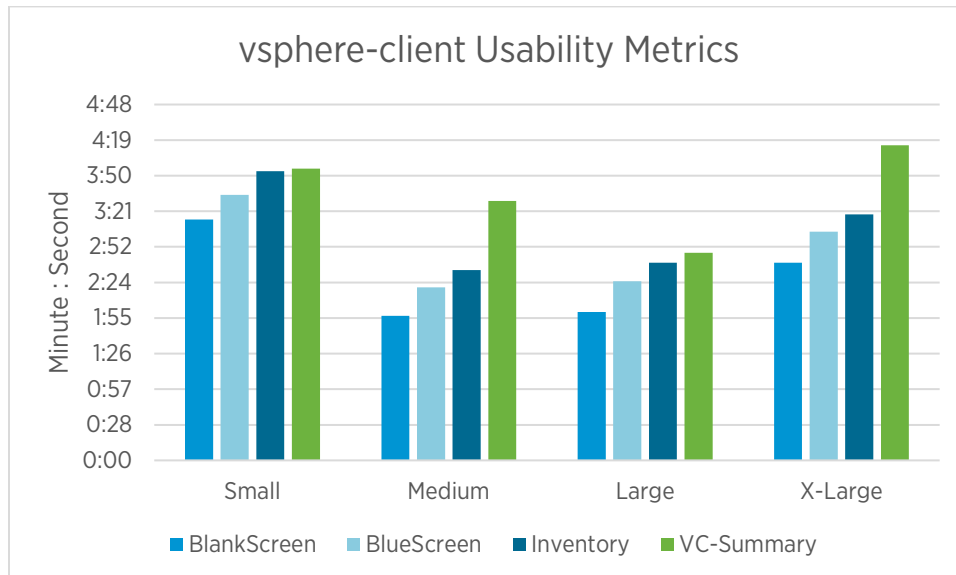


Figure 3. Usability metrics of vsphere-client for various inventory sizes. (Lower is better)

Size	BlankScreen	BlueScreen	Inventory	VC-Summary
Small	3:15	3:35	3:54	3:56
Medium	1:57	2:20	2:34	3:30
Large	2:00	2:25	2:40	2:48
X-Large	2:40	3:05	3:19	4:15

Table 3. Usability metrics of vsphere-client for various inventory sizes (minute:second).

Performance of Enabling VCHA

The first part of the lifecycle of the VCHA is to deploy it, which is what we measure in this section. Enabling VCHA initiates a method (FailoverClusterConfigurator.deploy) on the vpxd service, which performs a series of tasks. We can categorize these as:

- Cloning the Primary to the Secondary
- Cloning the Primary to the Witness
- Configuring the VCHA, including the PostgreSQL replication, file replication, and so on.

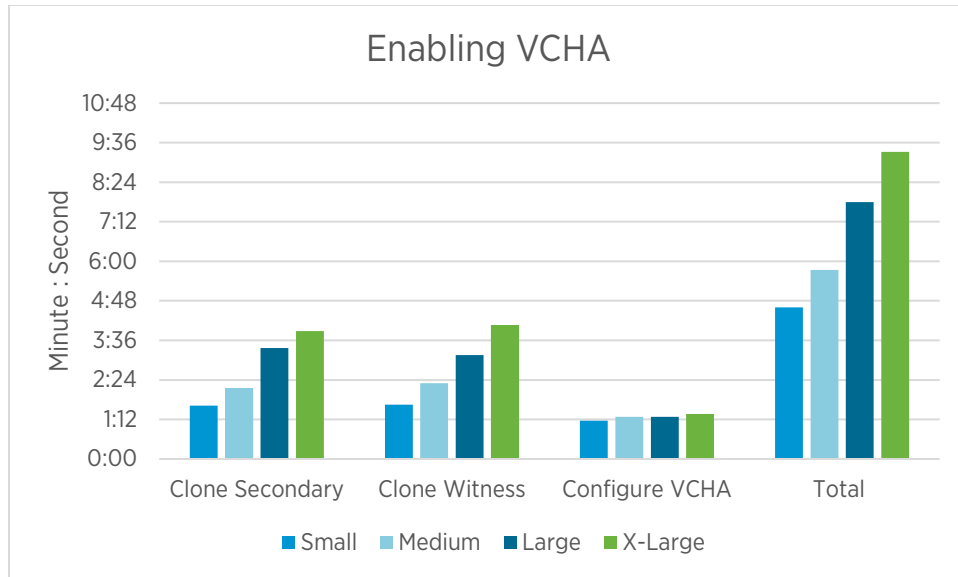


Figure 4. Enabling VCHA performance characterization. (Lower is better.)

Size	Clone Secondary	Clone Witness	Configure VCHA	Total
Small	1:37	1:39	1:10	4:36
Medium	2:09	2:18	1:17	5:44
Large	3:22	2:09	1:17	7:48
X-Large	3:53	4:04	1:22	9:19

Table 4. Enabling VCHA performance characterization (minute:second).

In Table 4 and Figure 4, we report the time it takes to perform these tasks during enabling VCHA for various configurations:

- It takes about the same time to clone the Primary and create a Secondary and a Witness because it is the same clone operation performed twice back to back.
- The clone operations take longer as we increase the inventory size. This is expected because the amount of the data stored on the virtual machine disks is increasing with each level. It is the amount of data that needs to be transferred through the network.
- We perform these tests on a setup where a 10 Gigabit network interface is used for the clone operations. The network is mostly saturated during these clone operations. Hence, the performance of enabling VCHA depends on the network bandwidth.
- There is a fixed overhead to configure the VCHA, including the PostgreSQL replication, file replication, and so on.

- The total performance of enabling VCHA depends on the clone operations:
 - Inventory size determines the amount of data that needs to be transferred during the clone operation.
 - The network bandwidth determines how fast the clone operation can transfer the data through the network.

VCHA Overhead

As a baseline, we set up our environment with VCHA disabled and ran experiments. Then we repeated these experiments with VCHA enabled to measure the overhead:

- **Baseline:** VCHA disabled. Single vCenter Server Appliance.
- **Control Runs:** VCHA enabled with a three-node cluster: Primary, Secondary, and Witness vCenter Server Appliances.

We ran two experiments, each generating different levels of background load:

- **64-client VCbench load:** Common use case for most of our customers.
- **256-client VCbench load:** Worst case scenario to stress the performance and scalability limits of the system.

VCbench Throughput

VCbench throughput (number of operations per minute) is used as the primary metric in our performance benchmarks.

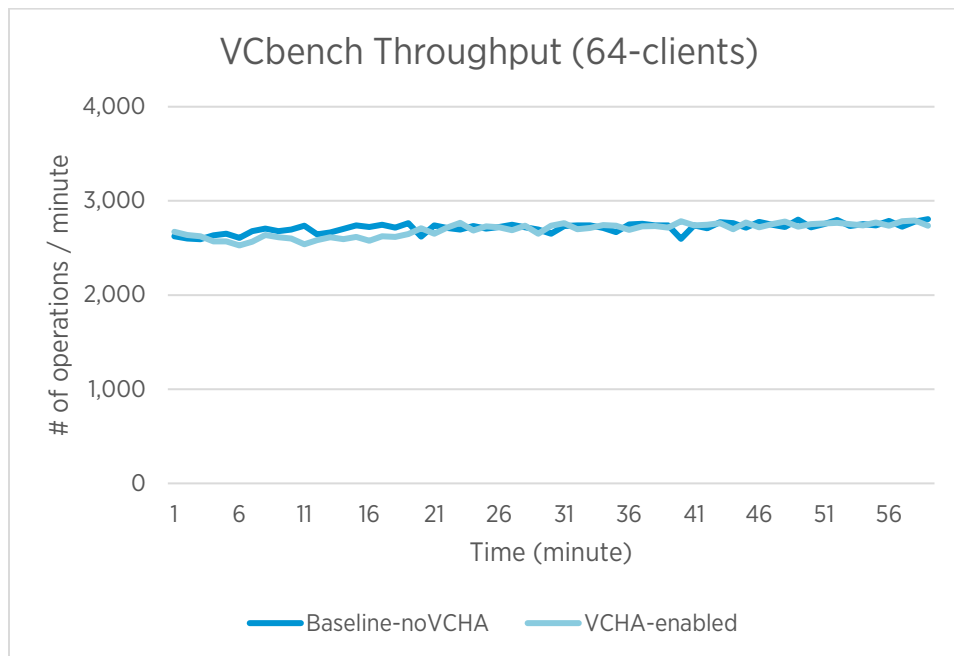


Figure 5. Periodic throughput during 64-client VCbench run. (Higher is better.)

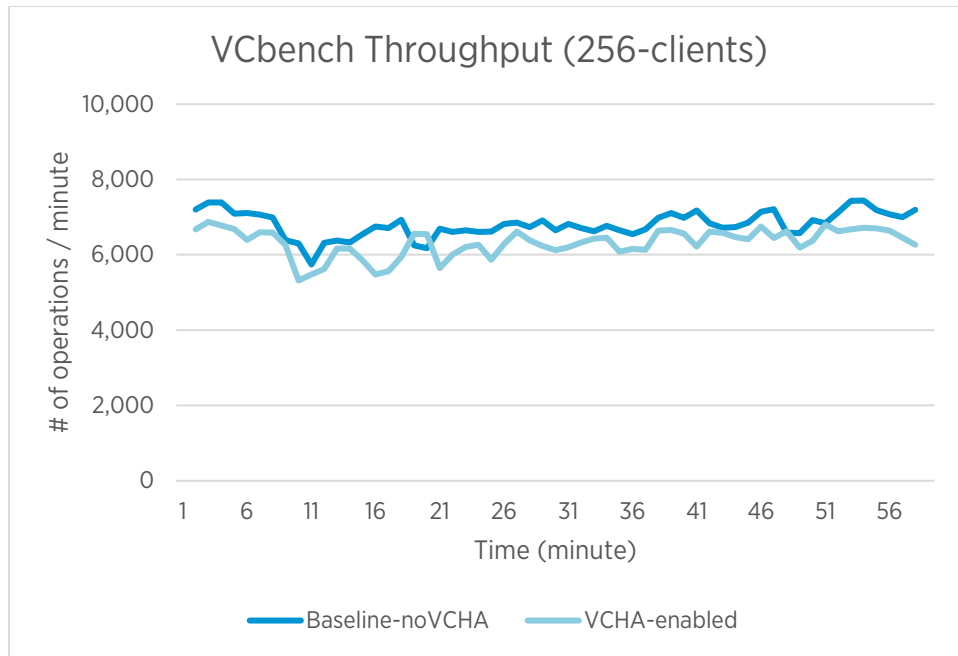


Figure 6. Periodic throughput during 256-client VCbench run. (Higher is better.)

In Figure 5 and Figure 6, the periodic throughput results during one hour experiments are shown for 64- and 256-client VCbench runs.

- In Figure 5 for the 64-client run, enabling VCHA has minimal impact on the performance of the vCenter Server. The results are almost the same.
- In Figure 6 for the 256-client run, enabling VCHA has a noticeable but small impact on the performance of the vCenter Server.

In order to quantify the impact of enabling VCHA, we computed the average throughput for each run, and the results are shown in Figure 7 and Table 5.

Observations:

- When VCHA is enabled, there is no significant impact for the 64-client VCbench throughput. This point represents the common use case.
- When VCHA is enabled, there is a noticeable but small impact of ~7% for the 256-client VCbench runs. With 256 clients generating background load, vCenter Server significantly uses the resources on the VM. This point represents the extreme use case, and it would be unlikely for customers to generate that much load on the vCenter Server for extended time periods.

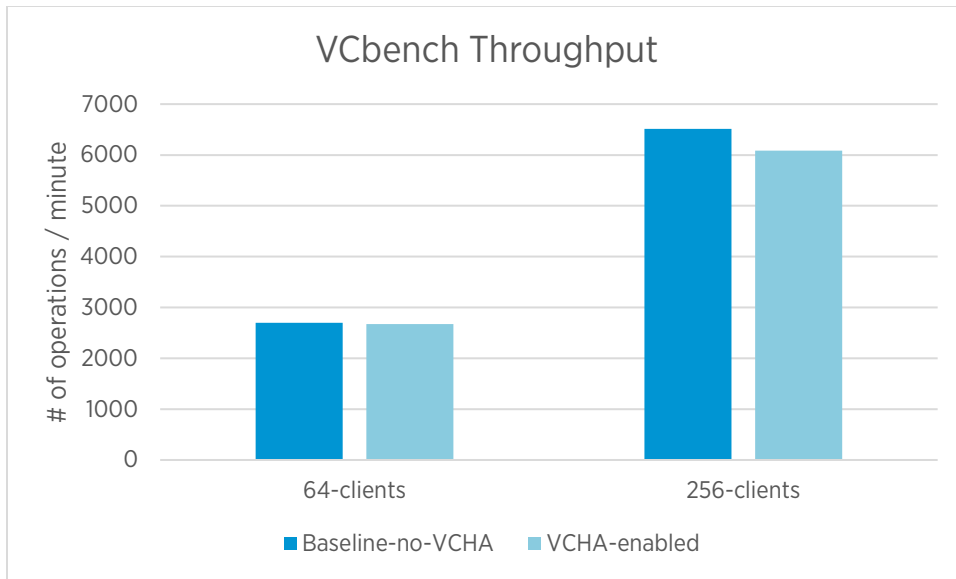


Figure 7. Average throughput results for entire 1-hour runs. (Higher is better)

# of operations / minute	64-client	256-client
Baseline-no-VCHA	2,696	6,513
VCHA-enabled	2,673	6,085
Difference	-0.9%	-6.6%

Table 5. Average throughput results for entire 1-hour runs.

VCbench Average Operation Latency

Another metric used for our performance benchmark is the VCbench average operation latency (measured in milliseconds per operation, or msec/operation), which is shown in Figure 8 and Table 6.

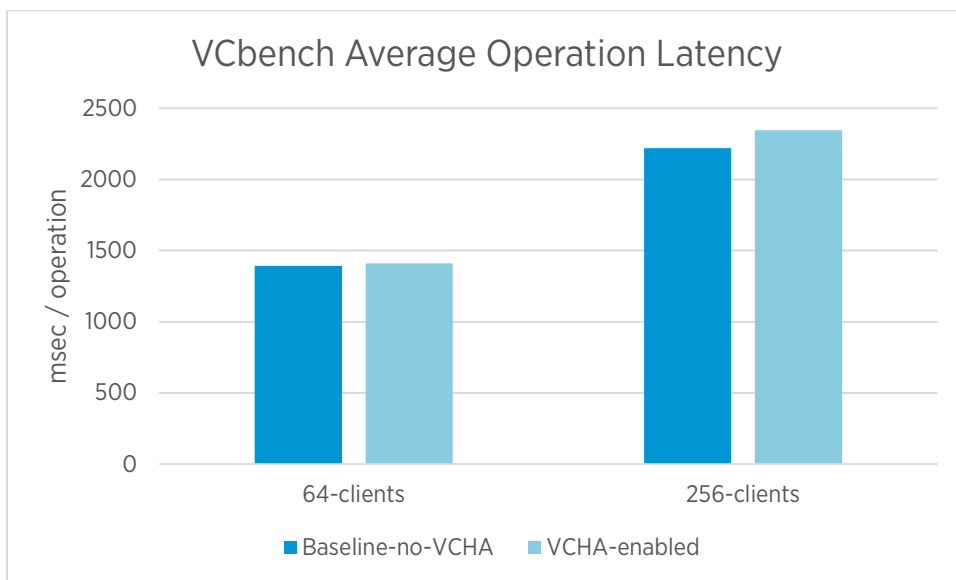


Figure 8. Average operation latency for entire 1-hour runs. (Lower is better.)

msec / operation	64-clients	256-clients
Baseline-no-VCHA	1,392	2,219
VCHA-enabled	1,411	2,345
Difference	1.4%	5.7%

Table 6. Average operation latency for entire 1-hour runs.

Observations:

- When VCHA is enabled, there is no significant impact for the 64-client VCbench average operation latency. This point represents the common use case.
- When VCHA is enabled, there is a noticeable increase of ~6% in the average operation latency with the 256-client VCbench. As we mentioned above, VCbench average operation latency is closely correlated to the VCbench throughput. Nonetheless, this point represents the extreme use case, and it would be unlikely for customers to generate that much load on the vCenter for extended time periods.
- VCbench average operation latency is closely correlated to the VCbench throughput, thus we focus on VCbench throughput from now on.

Comparison of Resource Utilization

We monitored the resource utilization on the vCenter Server Appliances. In this section, we present the resource utilization in terms of CPU, memory, disk, and network utilization.

CPU Utilization

In a vCenter HA cluster, both the Passive and Witness nodes use minimal CPU resource:

- **Witness:** VCHA is the only service running, which is minimal.
- **Passive:** PostgreSQL and VCHA are the only services running related to VCHA. Both use minimal CPU resource.

CPU utilization of the vCenter Server (vpxd) service is the most significant on vCenter Server Appliances with the VCbench load. All the other services use minimal CPU. Thus, we focus on the CPU utilization of vpxd and compare:

- **Baseline:** vCenter Server Appliance
- **VCHA:** Active vCenter Server Appliance of the VCHA cluster

In Figure 9 and Table 7, CPU Utilization of vpxd is presented, where 700% CPU utilization means the full 7 cores are being used on the virtual machine.

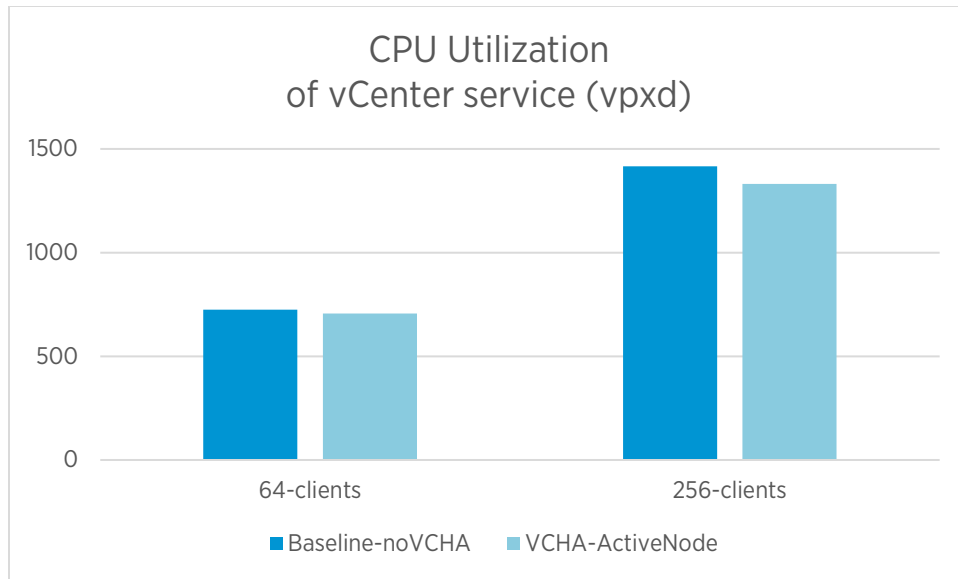


Figure 9. CPU Utilization of vCenter Server service (vpxd). 100% represents one core.

Load	Baseline-noVCHA	VCHA-ActiveNode
64-clients	725%	707%
256-clients	1,416%	1,332%

Table 7. CPU Utilization of vCenter Server service (vpxd). 100% represents one core.

Observations:

- For the 64-client VCbench load, when VCHA is enabled, the CPU utilization of vpxd does not change much because the VCbench throughputs are virtually the same, as shown above.
- For the 256-client VCbench load, when VCHA is enabled, the CPU utilization of vpxd is about 1 core less because the VCbench throughput is about 6% smaller.

Memory Utilization

In Table 8, we report the resident memory utilization, which is the actual memory occupied by the process, for the relevant services:

- VMware VirtualCenter service (vpxd)
- VMware PostgreSQL (postgres)
- VMware vCenter High Availability (vcha)

Load	Baseline-noVCHA	VCHA-ActiveNode	VCHA-PassiveNode		
64-clients	11,534	11,534	N/A	vpxd	VMware vCenter Server
	4,406	4,482	3,044	postgres	PostgreSQL Database
	N/A	56	52	vcha	vCenter High Availability
256-clients	14,680	14,680	N/A	vpxd	VMware vCenter Server
	4,222	4,218	3,045	postgres	PostgreSQL Database
	N/A	56	52	vcha	vCenter High Availability

Table 8. Peak resident memory (MB).

Observations:

- Enabling VCHA does not change the memory utilization on the vCenter Server Appliance.
- The VCHA service's memory utilization is minimal, around 50MB, and it stays constant.

Disk Utilization

The VCHA 6.5 virtual machine comes with 12 virtual disks. Each virtual disk is a separate vmdk. While all vmdks can be stored on the same datastore, within the vCenter Server Appliance 6.5 virtual machine, we can monitor the disk utilization for each of those virtual disks individually.

Each of these 12 virtual disks are used for different purposes, including storing logs, core dumps, netdump, auto deploy, Update Manager, and so on. In the interest of VCHA, we focus on the following virtual disks:

- **sdf**: PostgreSQL-DB. Database partition storing the core tables.
- **sdg**: PostgreSQL-Log. Database WAL files.
- **sdh**: PostgreSQL-SEAT. Database partition storing the SEAT tables including statistics, events, alarms, and tasks.

These three are the only disks that are generating I/O by the VCbench background load. All the other disks observe minimal and insignificant I/O. In Table 9 and Table 10, we report the I/O utilization for 64-client VCbench runs.

I/O STATS TPS (W/S)				
Baseline-noVCHA	VCHA-ActiveNode	VCHA-PassiveNode	HardDisk	Used for
2,29.5	244.3	288.6	sdf	PostgreSQL-DB
1,740.3	1,505.2	1,344.8	sdg	PostgreSQL-Log
60.1	61.5	59.7	sdh	PostgreSQL-SEAT
I/O STATS BANDWIDTH (MB/S)				
Baseline-noVCHA	VCHA-ActiveNode	VCHA-PassiveNode	HardDisk	Used for
4.2	4.1	4.0	sdf	PostgreSQL-DB
23	20.8	15.0	sdg	PostgreSQL-Log
1.8	1.6	1.6	sdh	PostgreSQL-SEAT

Table 9. I/O stats for 64-client VCbench runs.

I/O STATS TPS (W/S)				
Baseline-noVCHA	VCHA-ActiveNode	VCHA-PassiveNode	HardDisk	Used for
875.4	761.8	764.0	sdf	PostgreSQL-DB
1,944.3	1,710.3	1,223.7	sdg	PostgreSQL-Log
1,70.1	160.3	155.1	sdh	PostgreSQL-SEAT
I/O STATS BANDWIDTH (MB/S)				
Baseline-noVCHA	VCHA-ActiveNode	VCHA-PassiveNode	HardDisk	Used for
17.7	14.8	13.5	sdf	PostgreSQL-DB
31.3	28.6	20.4	sdg	PostgreSQL-Log
3.7	3.5	3.4	sdh	PostgreSQL-SEAT

Table 10. I/O statistics for 256-clients VCbench runs.

Network Utilization

The vCenter Server appliance has two network interfaces:

- **eth0**: The default interface is connected to the network managing all the ESXi hosts and virtual machines.
- **eth1**: If VCHA is enabled, then the second interface is connected to a private isolated network, which is dedicated to the VCHA traffic for PostgreSQL and file replication.

In Table 11, we report the network utilization of these network interfaces on the vCenter Server appliances running 64- and 256-client VCbench runs.

64-clients	Interface	Receive	Transmit
Baseline-noVCHA	eth0	10.5	5.2
VCHA-ActiveNode	eth0	10.2	5.0
	eth1	0.4	8.1
256-clients	Interface	Receive	Transmit
Baseline-noVCHA	eth0	23.5	9.7
VCHA-ActiveNode	eth0	21.6	9.1
	eth1	0.3	14.7

Table 11. Network utilization for 64- and 256- client VCbench runs. (Megabytes per second.)

Observations:

- On the eth0 interface, which vCenter Server uses to manage all the entities on the testbed, the network utilization is virtually the same for the 64-client run and is slightly lower with VCHA enabled. This matches the VCbench throughput.
- On the eth1 interface, which is used for VCHA data replication, the network utilization is significant and it is one-way traffic from the Active node to the Passive node.

Performance Impact of vCenter Server Statistics Level

vCenter Server collects statistics for all the entities it manages at different levels: [4] [5]

- Level 1 is the least-detailed statistics level and only includes the most critical statistics, such as aggregate CPU, memory, and network usage.
- Level 2 introduces a number of additional statistics.
- Level 3 incorporates per instance statistics; for example, CPU usage of a host on a per-CPU basis.
- Level 4 is the most detailed and is inclusive of all the other levels.

With an increasing stats level, vCenter Server collects more and more data. Thus, PostgreSQL stores more data in the database. As a result, VCHA needs to replicate more data to the Secondary while transferring more data through the network and writing more data to the disk. VCHA has a varying performance impact with the increasing statistics level. Here, we present the VCbench throughput results for the 256-client VCbench experiments in Figure 10 and Table 12.¹

¹ Large distributed systems inherently have some non-determinism, as VCbench throughput results for 256-client runs can vary quite a bit. Here, we report the average numbers of multiple runs.

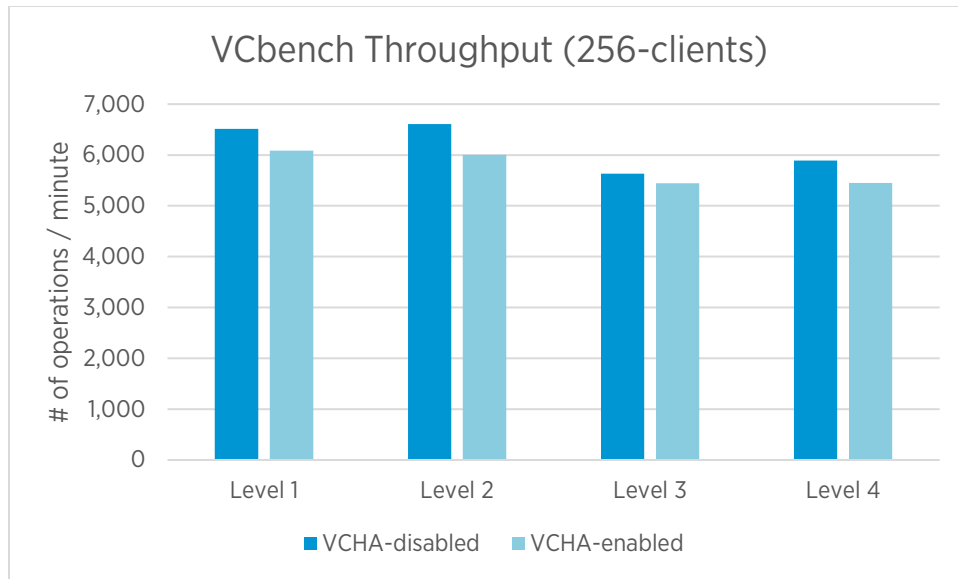


Figure 10. 256-client VCbench throughput with varying stats level. (Higher is better.)

VCbench Throughput (ops/min)	Level 1	Level 2	Level 3	Level 4
VCHA-disabled	6,513	6,609	5,631	5,889
VCHA-enabled	6,085	6,003	5,442	5,451
Enabled VCHA Overhead	-6.6%	-9.2%	-3.4%	-7.4%

Table 12. 256-client VCbench throughput with varying stats level.

Observations:

- With an increasing statistics level, VCbench produces less throughput, as expected.
- When VCHA is enabled for various statistics levels, there is a noticeable impact of 3% to 9% on VCbench throughput.
- The 256-client VCbench run represents the extreme use case, and it would be unlikely for customers to generate that much load on the vCenter Server for extended time periods. With fewer loads, expect the VCHA overhead to be minimal.

Performance Impact of VCHA Private Network

VCHA is designed for low latency networks. All the state kept at the Primary node, including the PostgreSQL database and all the files used by various services, are replicated to the Secondary node through the VCHA private network. By increasing the network latency, we measured the performance penalty on VCbench throughput, as shown in Figure 11 and Table 13.

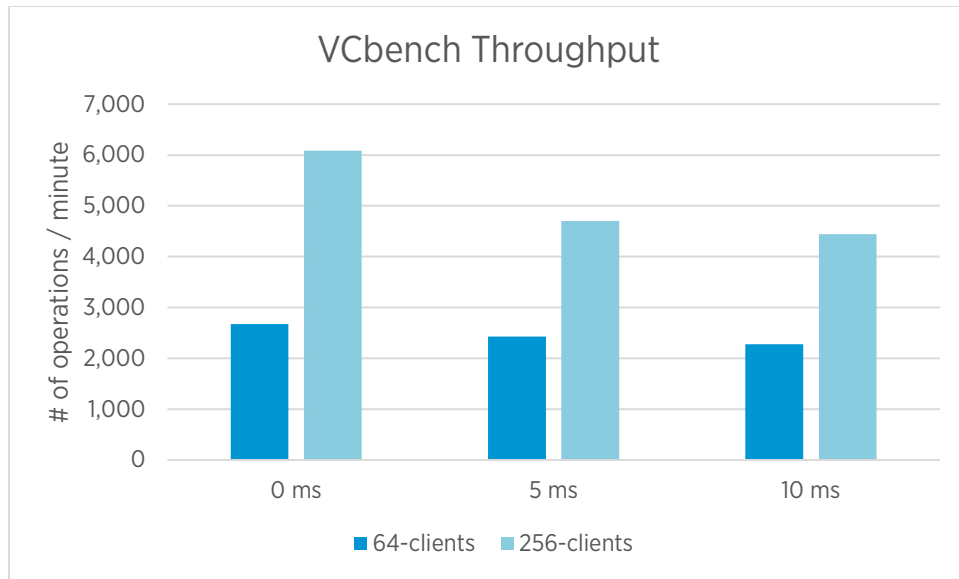


Figure 11. VCbench throughput with varying network latency on a private VCHA network. (Higher is better.)

Network Latency (ms)	VCbench Throughput (ops/min)	
	64-clients	256-clients
0 ms	2,673	6,085
5 ms	2,427	4,699
10 ms	2,280	4,445

Table 13. VCbench throughput with varying network latency on a private VCHA network.

Observations:

- VCHA is designed to support LAN networks with up to 10 ms latency between VCHA nodes. However, this comes with a performance penalty.
- VCHA private network latency has a measurable impact on vCenter Server performance if VCHA is enabled.
- With a 64-client VCbench load, which represents a realistic use case for customers, 5 ms latency shows a 9% drop and 10 ms latency shows a 15% drop in overall VCbench throughput, which are considerably modest.
- With a 256-client VCbench load, which is a more extreme and unlikely use case for most customers, 5 ms latency shows a 23% drop, and 10 ms latency shows a 27% drop in overall VCbench throughput.

External PSC vs Embedded PSC

Besides the external PSC configuration, VCHA also supports embedded mode: the PSC services are running on the vCenter Server appliance with all the management services. In this section, we study the VCHA performance comparing:

- Embedded PSC configuration
- External PSC configuration

Please refer to:

- The section “vCenter Server Deployment Models” in the [vSphere Upgrade Guide](#) for these deployment models. [2]
- The [recommended topologies](#) for VMware vSphere 6.0. [3]

VCHA Overhead and VCbench Throughput

We ran various levels of background load:

- VCbench loads: 64 and 256 clients.

Here are the VCbench throughput results for these experiments:

	64-cli	256-cli
VCHA-ExternalPSC	2,673	6,085
VCHA-EmbeddedPSC	2,616	6,044
Difference	-2.1%	-0.7%

Table 14. VCbench throughput results. (Operations per minute.)

Here are the average VCbench operation latencies for these experiments:

	64-cli	256-cli
VCHA-ExternalPSC	1,411	2,345
VCHA-EmbeddedPSC	1,430	2,376
Difference	1.3%	1.3%

Table 15. Average VCbench operation latencies. (Milliseconds per operation.)

Here are our key observations:

- We observe a minimal difference in the VCbench throughput, which is used as the primary metric for the performance benchmark.
- We observe a minimal difference in the VCbench average operation latency.
- The difference is well within the error margin of the experiment methodology.

Performance of VCHA Failover/RTO

We study the performance of VCHA failover/RTO comparing the Embedded PSC configuration with the External PSC configuration in terms of:

- Service ready time for vpxd
- Usability metrics for vsphere-client

Service Ready Time for vpxd

The vCenter service (vpxd) goes through some initialization phases during boot up. We measured how long it takes to reach a state where vpxd accepts incoming connection requests. We marked this as service ready time and compared the Embedded PSC configuration with the External PSC configuration:

	vpxd
VCHA-ExternalPSC	1:50
VCHA-EmbeddedPSC	2:42
Difference	47.3%

Table 16. Service ready time for vpxd (minute:second).

- The service ready time for vpxd is about 1 minute worse in the Embedded PSC mode compared to the External PSC mode. This is expected because, in the embedded PSC configuration, VCHA needs to start all the PSC services prior to the management services.
- These numbers depend on the vCenter Server configuration. The tests are performed with at-scale inventory and extra-large configuration (32 vCPUs, 64GB memory).

Usability Metrics for vsphere-client

We connected a vSphere Web Client and opened the Summary page of the vCenter Server. After the Active VCHA node was powered off, we manually refreshed the Web Client while the Passive VCHA node was becoming the Active. Here are the measured times for various screens to appear, comparing the Embedded PSC configuration with the External PSC configuration:

	InProgress	BlankScreen	BlueScreen	Inventory
VCHA-ExternalPSC	1:05	2:40	3:05	3:19
VCHA-EmbeddedPSC	1:40	3:20	3:35	4:09

Table 17. Usability metrics for vsphere-client (minute:second).

- These metrics are measured manually during a failover; hence, they have a larger error margin.
- Usability metrics for vsphere-client are about 30-40 seconds worse in the Embedded PSC mode compared to the External PSC mode. This is expected because, in the embedded PSC configuration, VCHA needs to start all the PSC services prior to the vsphere-client service.

Best Practices

Enabling VCHA

We support enabling VCHA while vCenter Server is running. As a best practice, it is highly recommended to pick a time period with a very light workload to enable VCHA. Otherwise, the newly cloned Passive node PostgreSQL DB could lag behind too much and not be able to catch up with the Active DB due to transaction logs on the Active DB getting purged on the Active node. We are working on adding archivelog support to address this issue in upcoming releases.

VCHA Configuration

VCHA is designed to provide high availability in case of a single failure. All the state data is replicated between the Active and Passive nodes, so that if something happens to the Active node, then the Passive node can take over and continue to run and provide availability. That's why it is very important to configure VCHA to avoid single points of failure affecting multiple nodes of VCHA (Active, Passive, and Witness).

- Configure the VCHA such that if a failure occurs, then it should not affect more than a single node of three VCHA nodes.
- Each node should be deployed on different servers as a precaution against hardware failures including CPU, memory, cable, or motherboard failures.
- Each node should be deployed on separated datastores as a precaution against disk failures.
- Redundant hardware, such as the power supply and cooling in the physical infrastructure, would increase availability. If a power failure disrupts only the Active node, as long as the other nodes are functional, then VCHA would initiate a failover promoting the Passive to Active.

On the other hand, VM configurations such as the resource settings of CPU, memory, and disk should be identical between nodes. This guarantees that if the Passive node becomes Active, then it would perform the same way.

VCHA supports all the configurations listed in Table 1 in the section "[Test Environment](#)." vCenter Server supports another configuration "**Tiny**" up to 10 hosts and 100 virtual machines, which is not supported by VCHA. The tiny configuration deploys 2 vCPUs and 10GB of RAM, which is not enough for VCHA to provide RTO guarantees. VCHA should be deployed on the vCenter Server appliance with a deployment size of small or higher, not tiny.

Network Configuration

VCHA is designed for a low latency, high bandwidth, isolated private network. There are various aspects of the network that impact the performance of VCHA overall.

- VCHA replicates the data between Active and Passive nodes. PostgreSQL handles this synchronously by default, and if it experiences any deviation, then it falls back to asynchronous mode. That is why it is important to have an isolated private network dedicated to VCHA, providing network resource that is free of glitches and interference of other applications.
- VCHA is supported officially for low latency networks up to 10 milliseconds. If the supported limits are exceeded, VCHA will continue providing high availability, but with a performance penalty; users will see significantly higher latency for operations and lower throughput. For the supported range, the performance impact of the VCHA network latency is demonstrated in the section "[Performance Impact of VCHA Private Network](#)."
- Once the VCHA feature is enabled, the network utilization for data replication, which is a continuous process, does not saturate the network resources as seen in the section "[VCHA Overhead](#)." However, it is important to have a high bandwidth network for enabling the VCHA feature, which performs two clone operations and would saturate the network. With low bandwidth networks, the performance numbers reported in the section "[Performance of Enabling VCHA](#)" would be much slower.

Conclusion

High availability services are important in any platform, and vCenter Server is no exception. As the main administrative and management tool of vSphere, it is a critical element that requires HA. VCHA delivers protection against software and hardware failures with excellent performance for common customer scenarios, as shown in this paper.

We thoroughly tested VCHA with a benchmark that simulates common vCenter Server activities in both regular and worst case scenarios. The result is solid data and a comprehensive performance characterization in terms of:

- Performance of VCHA failover/recovery time objective (RTO)
- Performance of enabling VCHA
- VCHA overhead
- Performance impact of vCenter Server statistics level
- Performance impact of a private network
- External PSC vs Embedded PSC

In addition to the performance study results, the paper describes the VCHA architecture and includes some useful performance best practices for getting the most from VCHA.

Appendix A - Performance Benchmark: VCbench

VCbench is the performance benchmark tool used to generate background load for the vCenter Server. A list of management operations is executed by the vSphere Web Services SDK clients. Table 18 shows the list of operations chosen from a sampling of representative customer data. [6]

The number of concurrent SDK clients is also configurable. With an increasing number of concurrent SDK clients, the vCenter Server can be stressed in terms of its performance and scalability limits.

OPERATION	DESCRIPTION
Add Port Group	Create a new port group on a vSphere Standard Virtual Switch.
Remove Port Group	Remove a port group from a vSphere Standard Virtual Switch.
Clone VM	Create a clone of a virtual machine.
Create Folder	Create a new folder in the vCenter inventory hierarchy.
Delete Folder	Delete a folder from the vCenter inventory hierarchy.
Create Snapshot	Create a snapshot of a virtual machine.
Delete Snapshot	Delete a snapshot of a virtual machine.
Group Power-On	Power on several virtual machines in a single operation.
vMotion VM	Move a powered-on virtual machine to a different host.
Power On VM	Power on a single virtual machine in a DRS cluster.
Power Off VM	Power off a single virtual machine.
Reconfigure VM	Edit a virtual machine's configuration settings.
Register VM	Add a .vmx file from a datastore to the vCenter inventory.
Unregister VM	Remove a virtual machine from the vCenter inventory without deleting its files from its datastore.
Relocate VM	Move a powered-off virtual machine to a different host.
Remove VM	Delete a virtual machine from the vCenter inventory, and delete its files from its datastore.
Reset VM	Reset a single virtual machine.
Suspend VM	Suspend a single virtual machine.
Resume VM	Resume a single virtual machine.

Table 18. List of management operations issued by VCbench performance benchmark.

References

- [1] VMware, Inc. (2015, March) VMware vCenter 6.0 Deployment Guide.
<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-vcenter-server6-deployment-guide-white-paper.pdf>
- [2] VMware, Inc. (2015) vSphere Upgrade, vSphere 6.0.
<http://pubs.vmware.com/vsphere-60/index.jsp?topic=%2Fcom.vmware.vsphere.upgrade.doc%2FGUID-ACCD2814-0F0A-4786-96C0-8C9BB57A4616.html>
- [3] VMware, Inc. (2016) List of recommended topologies for VMware vSphere 6.0.x (2108548).
<http://kb.vmware.com/kb/2108548>
- [4] Alper T. Mizrak and Raju Angani. (2012) VMware vCenter Server 5.1 Database Performance Improvements and Best Practices for Large-Scale Environments.
<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-vcenter-db-perf-best-practices-white-paper.pdf>
- [5] VMware, Inc. (2015) Data Collection Levels.
<http://pubs.vmware.com/vsphere-60/index.jsp?topic=%2Fcom.vmware.vsphere.vcenterhost.doc%2FGUID-25800DE4-68E5-41CC-82D9-8811E27924BC.html>
- [6] Mike Stunes and Ravi Soundararajan. (2015) VMware vCenter Server Performance and Best Practices - VMware vSphere 6.0.
<https://communities.vmware.com/docs/DOC-29203>

VMWARE vCENTER SERVER HIGH AVAILABILITY PERFORMANCE AND BEST PRACTICES

About the Authors

Alper T. Mizrak is a Staff Engineer with the Performance Engineering group at VMware. He joined the VMware Research & Development team in 2007 and he has been working on the performance and scalability of VMware products. Dr. Mizrak received his PhD in Computer Science in December 2007 from the University of California, San Diego. During his graduate studies, he conducted research in the area of network security and distributed systems. His graduate research resulted in the receipt of the William C. Carter Award—an award granted by the IEEE Technical Committee on Fault Tolerant Computing and the IFIP WG-10.4 on Dependable Computing and Fault Tolerance—to recognize an individual who has made a significant contribution to the field of dependable computing through his or her graduate dissertation research; a book, *Secure Networking: Detecting Malicious Routers*, published in 2008; 8 academic papers published in peer-reviewed journals, conferences and workshops; and various technical reports.

Pawan Saxena is a Senior Staff Engineer within the vCenter Core group at VMware. He joined the VMware Research and Development team in 2014 and has been working on architecting the availability solution for the vCenter Server appliance. Mr. Saxena received his MS in Computer Science from Indore University in India in 1990. Mr. Saxena holds several patents in the areas such as data storage, data protection, data deduplication, and replication solutions. Prior to joining VMware, Mr. Saxena worked extensively in the field of data storage and the design of next generation file systems to allow continuous data protection.

Acknowledgements

This white paper is the result of a collaborative effort among various teams within VMware, including the core vCenter High Availability team, the vCenter Server Platform Database team, as well as the vCenter Server Performance team. We would like to thank Ravi Soundararajan, Mike Stunes, Shanshan Song, and Feidhlim O'Leary for reviewing the document and providing insightful feedback to help improve the quality of the paper. We would also like to thank Priya Sethuraman, Hiep Ma, Dilpreet Bindra, Kostadis Roussos, and Sameh Zakhary for their support of this work. Finally, we would like to thank Julie Brodeur for editing the paper.



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

Copyright © 2016 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.

Comments on this document: <https://communities.vmware.com/docs/DOC-33044>