VMware vCenter Server Performance and Best Practices

VMware vSphere 6.0

vmware®

Table of Contents

Introduction

VMware vCenter Server™ 6.0 substantially improves performance over previous vCenter Server versions. This paper demonstrates the improved performance in vCenter Server 6.0 compared to vCenter Server 5.5, and shows that vCenter Server with the embedded vPostgres database now performs as well as vCenter Server with an external database, even at vCenter Server's scale limits. This paper also discusses factors that affect vCenter Server performance and provides best practices for vCenter Server performance.

What's New in vCenter Server 6.0

vCenter Server 6.0 brings extensive improvements in performance and scalability over vCenter Server 5.5:

- Operational throughput is over 100% higher, and certain operations are over 80% faster.
- VMware vCenter Server™ Appliance™ now has the same scale limits as vCenter Server on Windows with an external database: 1,000 ESXi hosts, 10,000 powered-on virtual machines, and 15,000 registered virtual machines [1].
- VMware vSphere® Web Client performance has improved, with certain pages over 90% faster.

In addition, vCenter Server 6.0 provides new deployment options:

- Both vCenter Server on Windows and VMware vCenter Server Appliance provide an embedded vPostgres database as an alternative to an external database. (vPostgres replaces the SQL Server Express option that was available in previous vCenter versions.)
- The embedded vPostgres database supports vCenter's full scale limits when used with the vCenter Server Appliance[1].

Performance Comparison with vCenter Server 5.5

In order to demonstrate and quantify performance improvements in vCenter Server 6.0, this section compares 6.0 and 5.5 performance at several inventory and workload sizes. In addition, this section compares vCenter Server 6.0 on Windows to the vCenter Server Appliance at different inventory sizes, to highlight the larger scale limits in the Appliance in vCenter 6.0. Finally, this section illustrates the performance gained by provisioning vCenter with additional resources.

The workload for this comparison uses vSphere Web Services API clients to simulate a self-service cloud environment with a large amount of virtual machine "churn" (that is, frequently creating, deleting, and reconfiguring virtual machines). Each client repeatedly issues a series of inventory management and provisioning operations to vCenter Server. Table 1 lists the operations performed in this workload. The operations listed here were chosen from a sampling of representative customer data. Also, the inventories in this experiment used vCenter features including DRS, High Availability, and vSphere Distributed Switch. (See Appendix A for precise details on inventory configuration.)

---

[1] vCenter Server for Windows with an embedded vPostgres database still supports only a small deployment of 20 ESXi hosts and 200 virtual machines.

Operation	Description
Add Port Group	Create a new port group on a virtual switch.
Remove Port Group	Remove a port group from a virtual switch.
Clone VM	Create a clone of a virtual machine.[2]
Create Folder	Create a new host folder in the vCenter inventory hierarchy.
Delete Folder	Delete a host 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 VMs	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.[3]
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 and datastore.
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 1. Operations performed in performance comparison workload

The size of the workload was defined by the number of vSphere Web Services API clients that were issuing operations to vCenter Server. Table 2 lists the workload sizes used; these were chosen to represent typical provisioning workloads from very light to very heavy.

---

[2] The performance of a clone operation depends on the disk size of the VM being cloned and on the underlying hosts' disk performance

[3] In particular, this operation edits the VM's Memory Shares setting.

Size	Workloads (No. of API Clients)
Small	1, 3, 6, 12, 24
Medium	1, 13, 26, 52, 104
Large	1, 32, 64, 128, 256

Table 2. Workload sizes.

Testbed

This experiment was performed using vCenter Server 5.5 Update 2 Windows, vCenter Server 6.0 on Windows, and vCenter Server Appliance 6.0.

Figure 1 illustrates the testbed used for vCenter Server 5.5 and vCenter Server 6.0 on Windows. vCenter Server was configured to use an external Microsoft SQL Server 2012 database. vCenter Server and the database were each installed in a virtual machine running Microsoft Windows Server 2012 R2; these two virtual machines ran on a single physical host.
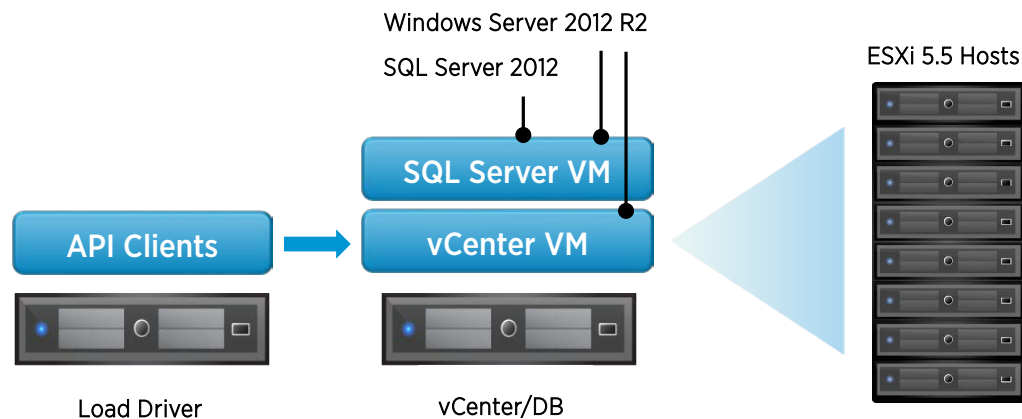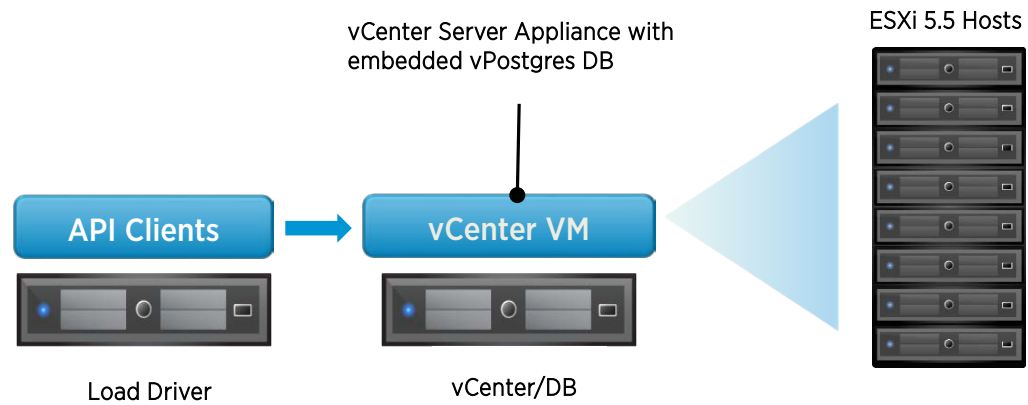

Figure 1. Testbed for vCenter Windows. One physical host runs the load driver. A second physical host runs two Windows Server 2012 R2 VMs, one running vCenter Server and the other running SQL Server 2012. Additional hosts run ESXi and VMs managed by vCenter.

Figure 2 shows the testbed used for vCenter Server Appliance 6.0. The appliance was installed on the same physical machine that hosted vCenter on Windows, but the appliance was configured to use an embedded vPostgres database instead of an external database.

Figure 2. Testbed for vCenter Server Appliance. One physical host runs the vCenter Server Appliance with its embedded vPostgres database. A second physical host runs the load driver; additional hosts run ESXi and VMs managed by vCenter.

Table 3 shows the configuration for the vCenter Server and database virtual machines. These experiments used the supported minimum configurations; see the vSphere 6.0 installation and setup guide [2] and the vCenter 6.0 deployment guide [3] for full details on vCenter Server installation and provisioning.

Size	Hosts	Virtual Machines	vCenter CPU	vCenter RAM	DB CPU	DB RAM
Small	100	1,000	4 vCPUs	16GB	2 vCPUs	8GB
Medium	400	4,000	8 vCPUs	24GB	4 vCPUs	12GB
Large	1,000	10,000	16 vCPUs	32GB	8 vCPUs	16GB

Table 3. Sizing for vCenter Server and database machines for performance comparison.

vCenter and its external database (where applicable) were hosted on a physical server with these specifications:

Intel Xeon E7-4870 v2 @ 2.3 GHz (4 processors, 15 cores each), HT enabled; 6 SSDs in RAID-10

The vCenter Server's physical machine was equipped with solid-state disks to prevent or minimize any bottlenecks in the storage system. The "Performance Considerations" section further discusses storage requirements.

Results

Figure 3 shows vCenter Server operation throughput (in operations per minute) for the heaviest workload for each inventory size. Performance has improved considerably at all sizes. For example, for the large inventory setup (Figure 3, right), operational throughput has increased from just over 600 operations per minute in vCenter Server 5.5 to over 1,200 operations per minute in vCenter Server 6.0 for Windows: an improvement of over 100%. The other inventory sizes show similar gains in operational throughput.
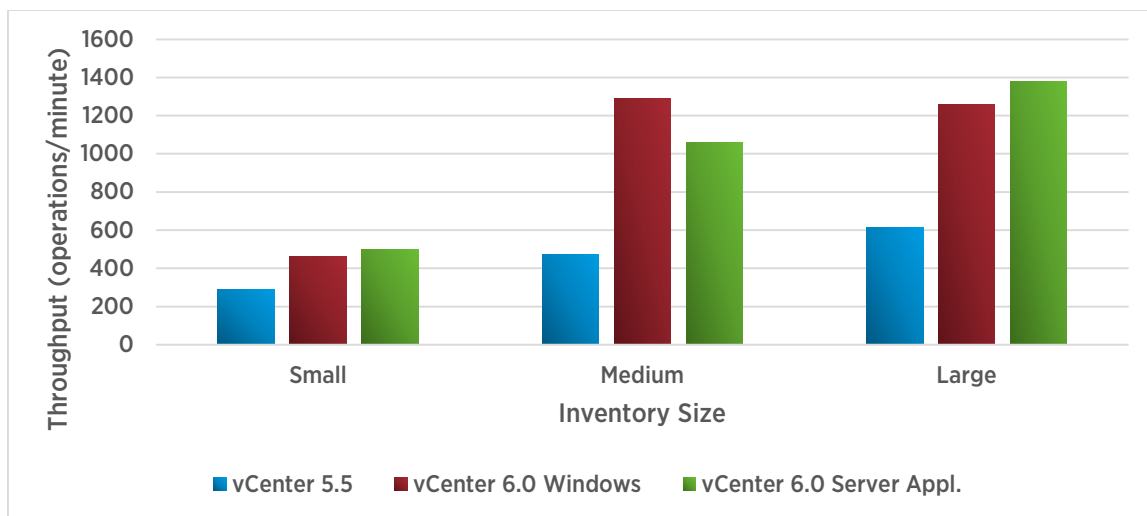
Figure 3. vCenter throughput at several inventory sizes, with heavy workload (higher is better). Throughput has increased at all inventory sizes in vCenter Server 6.0.

Figure 4 shows median latency across all operations in the heaviest workload for each inventory size. Just as with operational throughput in Figure 3, latency has improved at all inventory sizes. For example, for the large inventory setup (Figure 4, right), median operational latency has decreased from 19.4 seconds in vCenter Server 5.5 to 4.0 seconds in vCenter Server Appliance 6.0: a decrease of about 80%. The other inventory sizes also show large decreases in operational latency.
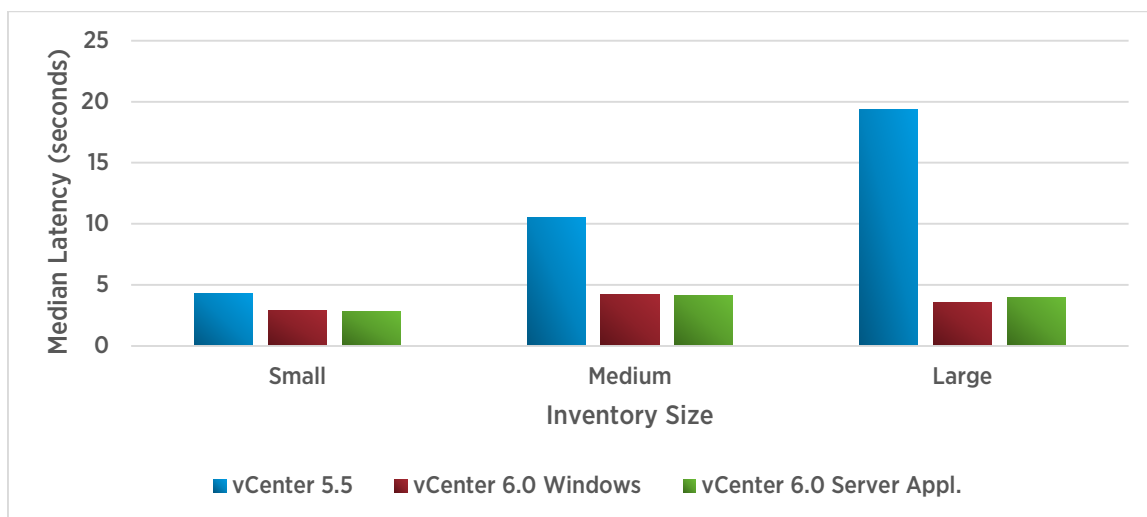

Figure 4. vCenter Server median latency at several inventory sizes, with heavy workload (lower is better). Latency has decreased at all inventory sizes in vCenter 6.0.

Additional Resource Allocation

The charts in Figures 3 and 4 show vCenter performance with the minimum required hardware configuration. Frequently, performance can be improved by allocating additional resources. (See the "Performance Best Practices" section for more details.) Figure 5 shows the improvement in throughput achieved by providing extra resources for vCenter. This additional test used the large inventory size, with vCenter Server given an additional 8 CPUs and 16GB of RAM above the minimum configuration. The external database, where applicable, was not given additional CPU or RAM. For vCenter Server Appliance, this resulted in a 64% gain in operational throughput.

vCenter Server 5.5 and vCenter Server 6.0 on Windows exhibited similar gains in throughput when given additional resources. In general, providing vCenter Server with additional resources will improve performance.

Note: These experiments were performed on physical machines with SSD storage. Increasing vCenter Server performance by allocating additional resources may require adding more disk spindles or using faster disks, such as SSDs.
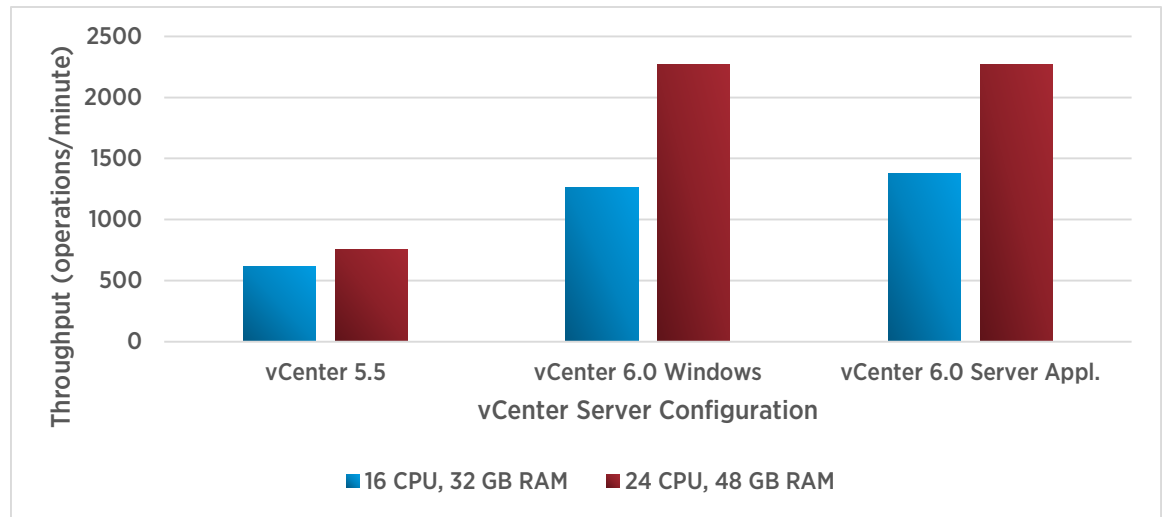
Figure 5. vCenter throughput with additional hardware resources (higher is better). The left bar in each pair shows throughput with the minimum resource configuration; the right bar shows throughput with eight additional CPUs and 16 additional GB of RAM provisioned for vCenter. As the figure shows, vCenter can scale and perform better with more hardware resources.

Web Client Performance Comparison

vSphere Web Client performance has also improved in vCenter Server 6.0 relative to vCenter Server 5.5. Table 4 lists representative Web Client views with the improvement in each view's average latency.

Operation	Improvement in Latency
Login	39%
Action Menu	88%
VM List	77%
Host Summary	65%
Create VM Wizard	73%
DRS Cluster Settings	91%

Table 4. Latency improvements for representative Web Client views.

Performance Considerations

As shown in the "Performance Comparison" section, vCenter Server performance depends on proper provisioning for vCenter Server, its database, and ESXi hosts. In general, these factors affect vCenter Server resource usage:

- **Inventory size:** A larger vCenter Server inventory uses more memory and database storage space for holding inventory information, and uses more CPU, database bandwidth, and network bandwidth for handling larger amounts of performance data and maintaining host and virtual machine configuration and runtime state.
- **Inventory churn (vCenter Server operations):** Performing more operations requires more CPU and more database and network bandwidth for persisting inventory changes.
- **Clients (Web Client or Web Services API):** Each active vCenter client requires additional memory for per-session state and additional CPU for issuing and tracking operations.
- **Extensions and plugins:** Each installed extension or plugin requires memory and CPU for the vSphere Web Client server, in addition to the extension's resource requirements.

The following sections illustrate the effects of each of these factors on vCenter Server's resource usage.

CPU

CPU usage is determined by inventory size, inventory churn, attached clients, and extensions and plugins.

Figure 6 shows how CPU usage varies with the size of the vCenter Server inventory, with no operations being executed. In this example, the inventory size is taken as the number of ESXi hosts managed by vCenter Server, with a constant number of VMs on each host (15 in this case). CPU usage scales linearly with inventory size, so doubling the number of hosts and virtual machines will approximately double the CPU usage. (This assumes that the numbers of hosts, virtual machines, datastores, and so on are all varied together; the exact effect of adding inventory objects on CPU usage depends on the number and types of objects.)
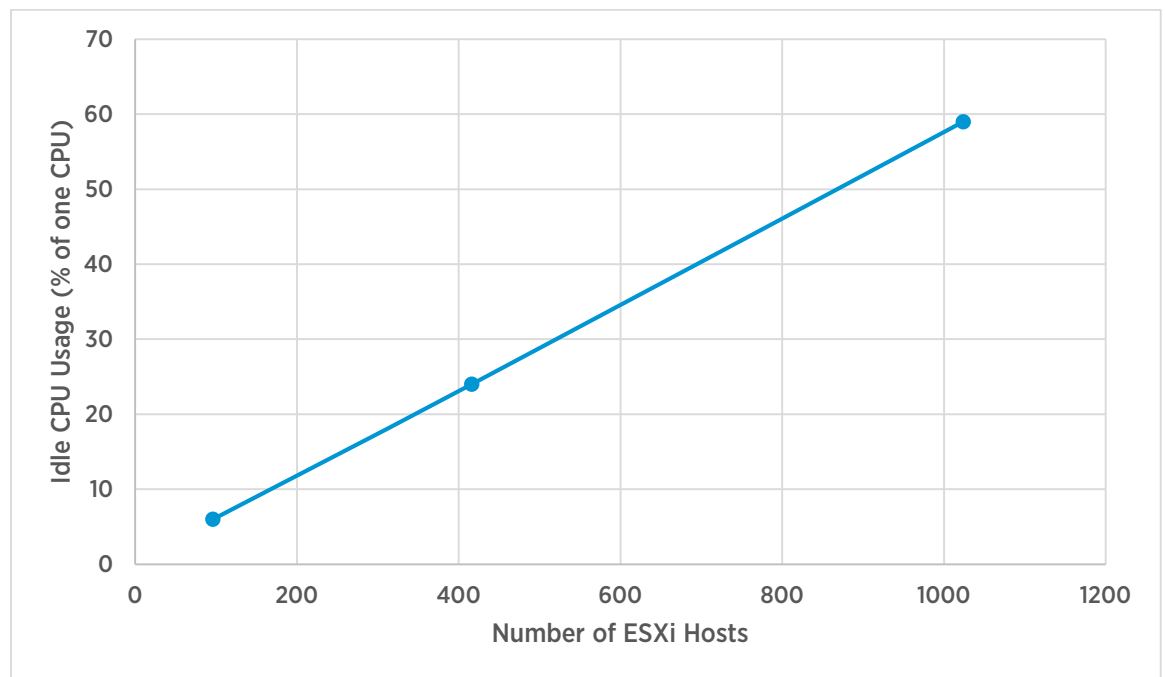

Figure 6. Idle CPU usage for varying vCenter Server inventory sizes.

Figure 7 shows how vCenter Server CPU varies with its operational throughput for a 1,000-host environment. Here, as in Figure 6, CPU usage varies linearly with throughput, so doubling vCenter Server's operational throughput will approximately double its CPU usage. As an example from this chart, eight CPU cores will provide around 650 operations per minute of throughput. (Actual vCenter Server performance depends on more than just the number of CPUs provided, so results will differ in practice.)
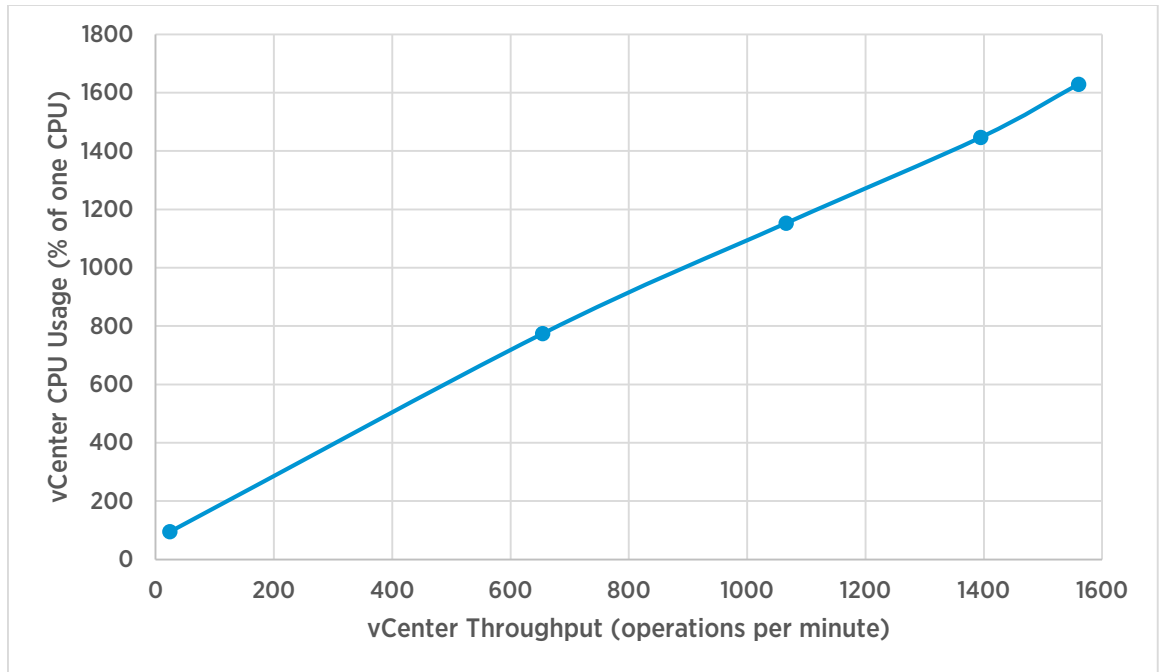

Figure 7. vCenter CPU usage at varying operational throughput, for a 1,000-host environment. The y-axis shows CPU usage as a fraction of one CPU; for example, 200% represents two CPUs being fully utilized.

Memory

Memory usage is primarily determined by inventory size and attached clients, as Figures 8 and 9 show. Figure 8 shows the variation in vCenter Server memory usage with varying inventory sizes. The blue line in Figure 9 shows memory usage during the first experiment performed on a vCenter Server. Its memory usage starts around 18GB before any workload is applied, and increases to 25GB as progressively heavier loads are applied. The red line shows memory usage for a second experiment performed on the same vCenter Server, and shows that vCenter Server's memory usage remains essentially constant, even as operational throughput varies, once the peak workload has been applied.
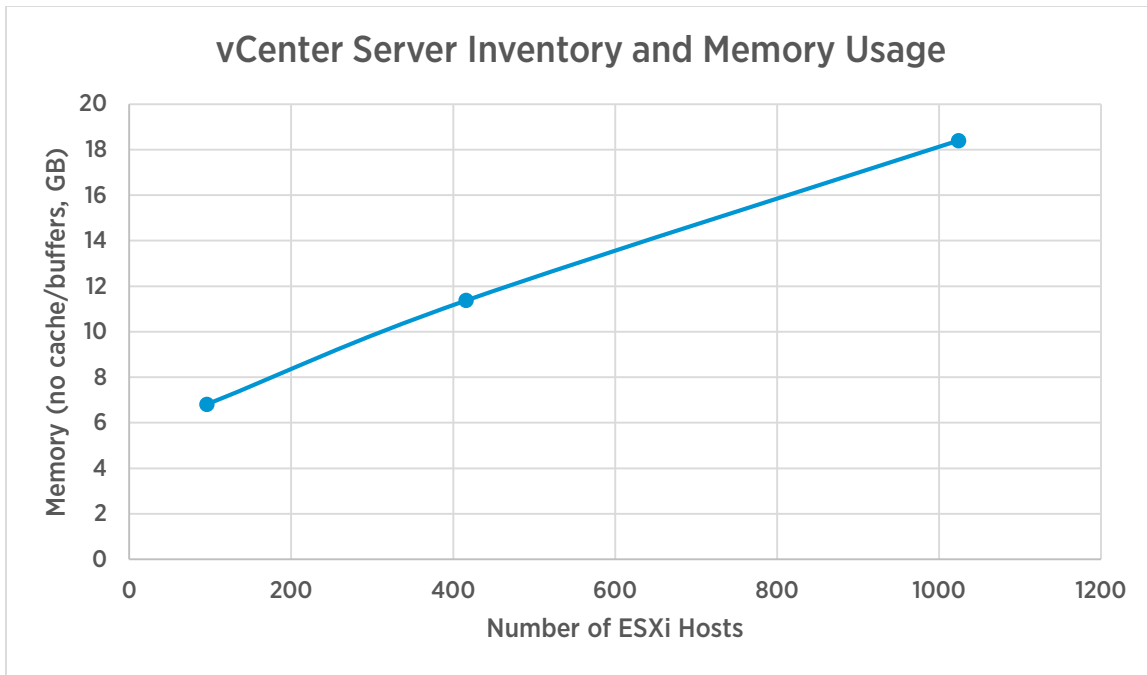
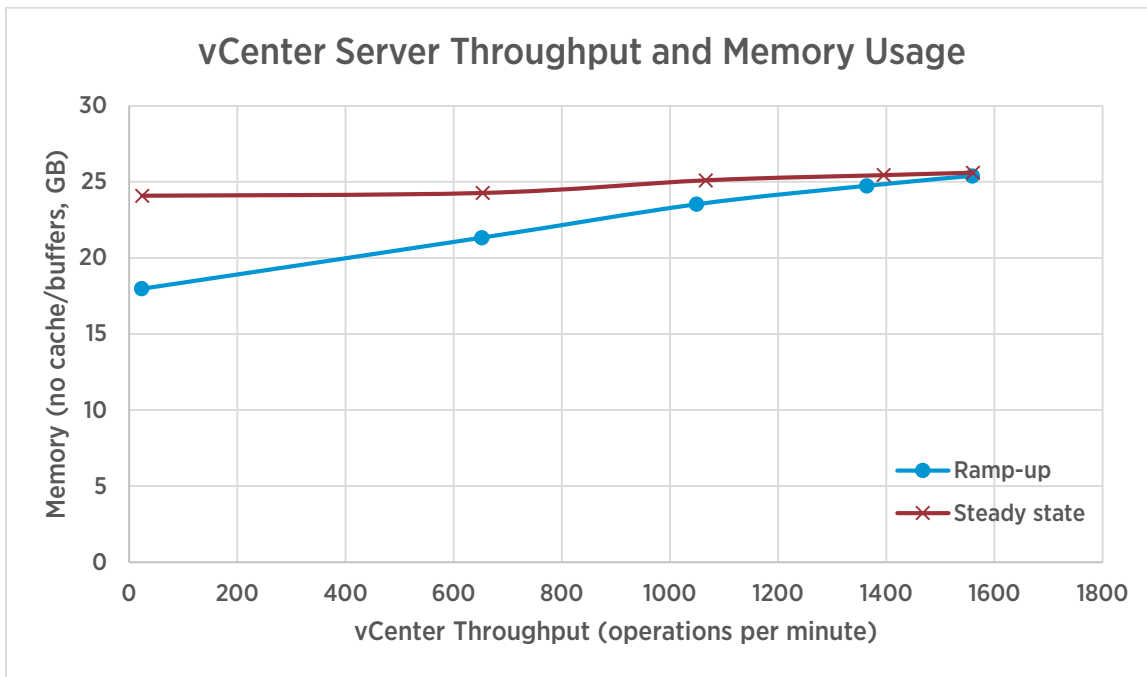Figure 8. Memory usage at varying vCenter inventory sizes.

Figure 9. vCenter memory usage at varying operational throughput, for a 1,000-host environment. "Ramp-up" indicates

that no workload has been applied to vCenter previously; "Steady state" indicates that a workload had previously been applied. Memory usage varies from 18GB, with no workload having been applied, to 26GB, after a 256-client workload has been applied.

Disk I/O

Figure 10 shows vCenter Server disk usage varying with the size of the inventory, with no operations being executed. vCenter Server's idle disk usage is primarily database traffic due to performance data collection; a small amount of disk traffic is for vCenter Server diagnostic logging.
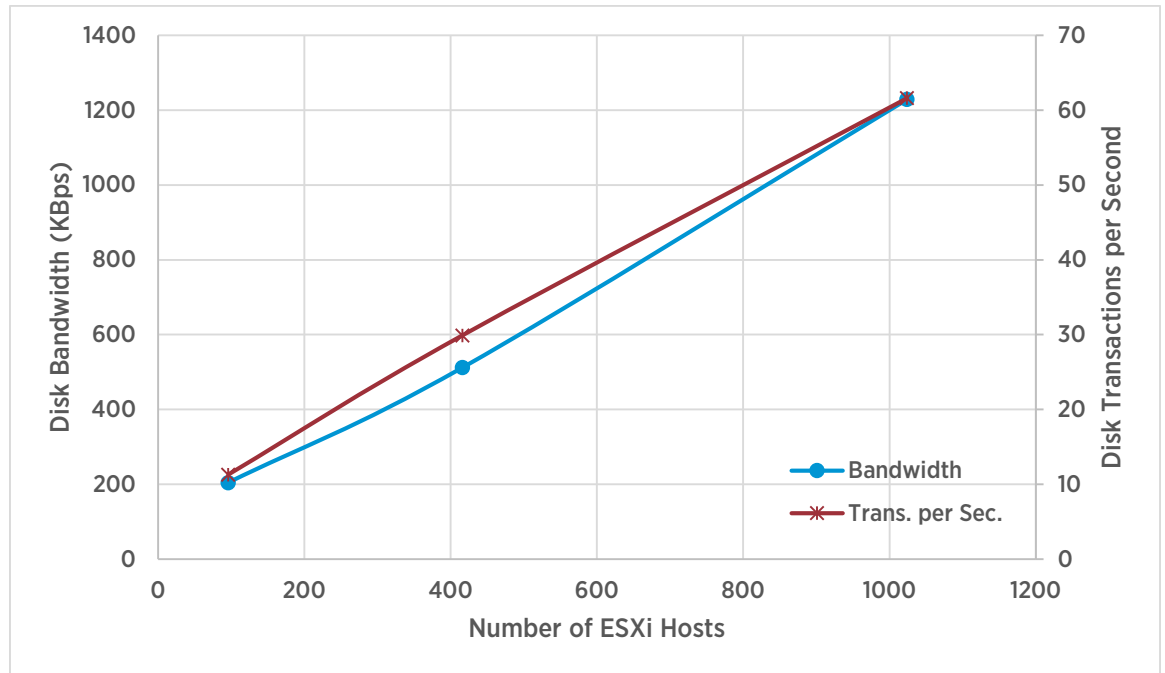

Figure 10. vCenter Server disk usage at varying inventory sizes.

Figure 11 shows how vCenter Server disk usage varies with operational throughput. As in Figure 10, some of this disk traffic is for performance data, but most of it is for persisting the results of vCenter operations to the vCenter database and the vCenter Inventory Service.

Note: In figures 10 and 11, vCenter Server is configured with the default (level 1) setting for performance statistics.

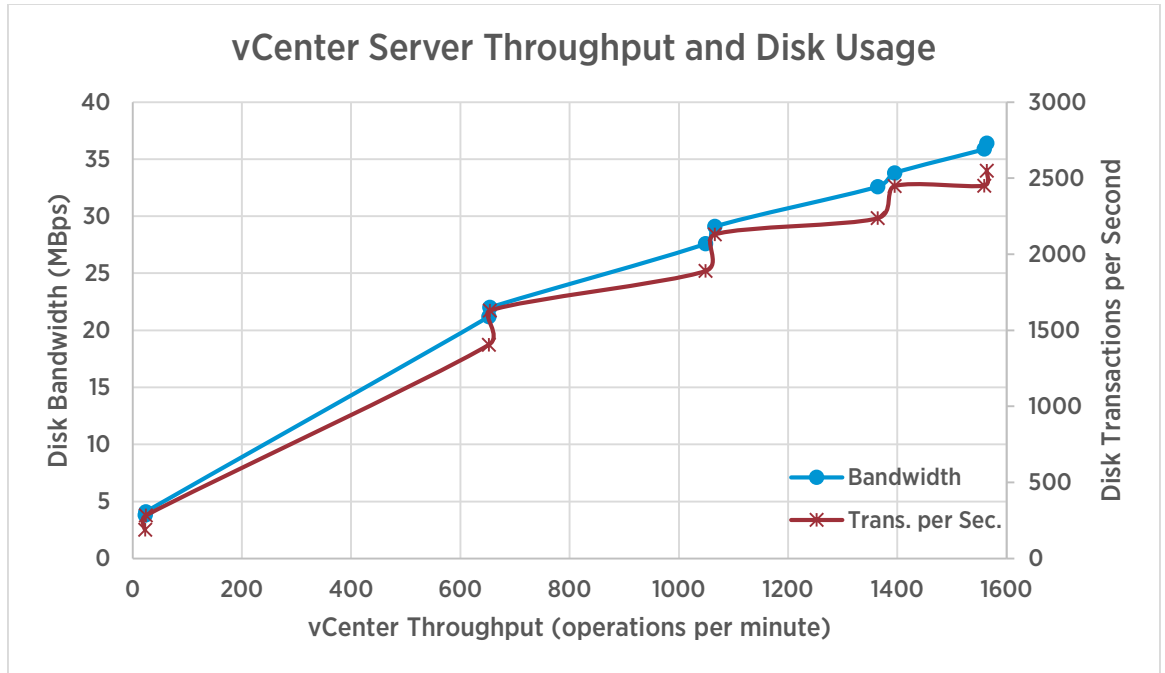**Figure 11. vCenter Server disk usage plotted against operational throughput.**
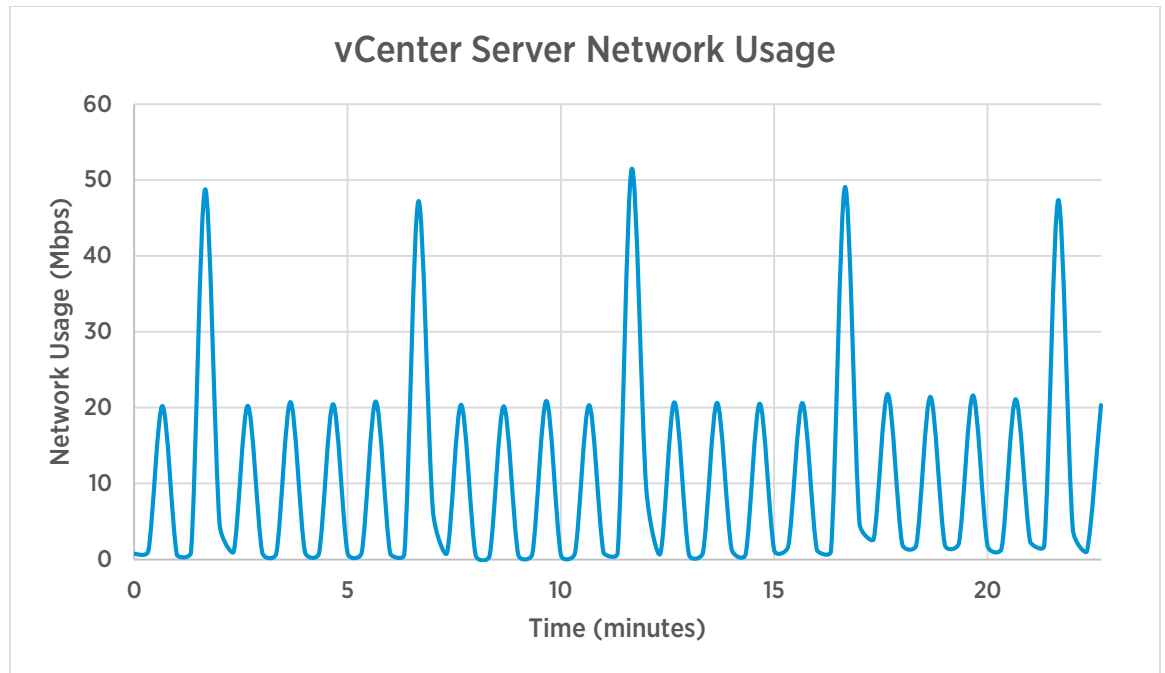
Network

Network usage depends strongly on operational throughput, as Figure 12 shows. Network usage scales approximately linearly with throughput, so doubling throughput will result roughly in doubling network usage. Figure 12 includes traffic both between vCenter Server and hosts, and between vCenter Server and its external database.
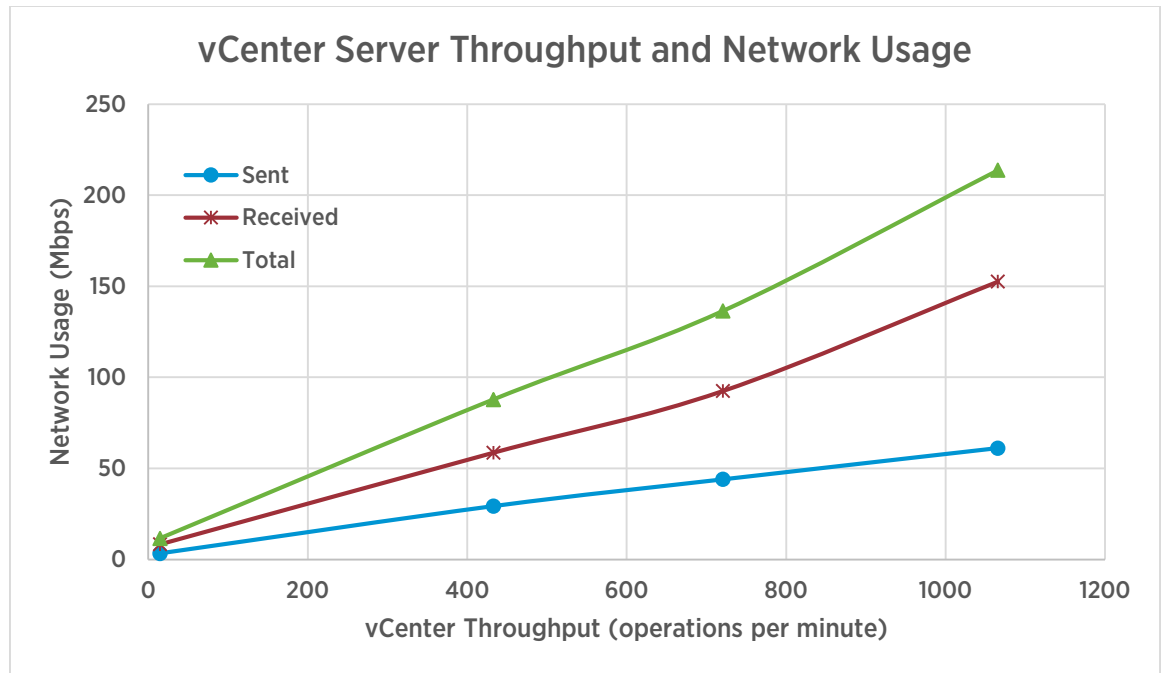
Figure 12. vCenter Server network usage at varying operational throughput, for a 1,000-host environment.

Figure 13 shows vCenter Server's network usage for a large inventory with no operations being executed. Note the spike in network usage that occurs every five minutes. These spikes are caused by vCenter Server collecting performance data from ESXi hosts and persisting that data to the vCenter Server database. (Note that Figure 13 shows network traffic both between hosts and vCenter Server and between vCenter Server and the external database. Also, in this example, vCenter Server is configured with the default setting (level 1) for performance statistics. The peak network usage will vary with the chosen statistics level.) Ensure that your network infrastructure can handle these periodic spikes, not just the average network traffic from vCenter Server.

Figure 13. vCenter Server network usage with no operations being executed. The periodic spikes are caused by vCenter Server collecting performance information from ESXi hosts and persisting it to the vCenter database.

Concurrency

In order to maintain good performance, vCenter Server limits the number of operations that may be performed concurrently. A single vCenter Server supports up to 2,000 concurrent sessions, where a session is a logged-in user, API client, or remote console session. After that limit is reached, requests for new sessions will be denied.

Also, vCenter Server will run up to approximately 600 operations concurrently. Additional requests for operations are queued, up to a maximum of around 5,000; after that, incoming requests for operations will be denied.

In addition to limits in vCenter Server itself, individual ESXi hosts limit concurrent operations as well, to avoid saturating the host and affecting performance of the virtual machines running on that host. A host can perform up to eight concurrent provisioning operations (that is, VM clone, vMotion, and VM relocate). A datastore can be involved in up to 128 concurrent vMotions and up to 8 Storage vMotions. A 1 Gbps host NIC can be involved in up to 4 concurrent vMotions; a 10 Gbps host NIC can be involved in up to 8 concurrent vMotions.

Provisioning operation performance will depend on the resources available to ESXi, especially storage performance. Please ensure that your ESXi storage has adequate capacity, especially for concurrent provisioning operations.

Performance Best Practices

Resource Provisioning

VMware's specified minimum configurations might not be adequate for every workload or environment. Complex environments (that is, those with an extensive usage of vSphere Distributed Switches, tags, annotations, storage profiles, and other features), those with large inventories, or those with additional solutions or extensions may require additional resources for vCenter Server. Also, workloads with high operational throughput will benefit from additional resources, as shown in Figure 5.

Many vCenter services are Java-based Web servers that depend on the configuration of their Java Virtual Machine (JVM), particularly the JVM's heap memory settings. The heap memory allocated to each JVM is a function of the size of the vCenter Server Appliance. Environments that make extensive use of one or more Java services (that is, an environment using many storage profiles, or with many Web Client plugins) may need to use a larger vCenter Server appliance in order to allocate more heap memory to the services of interest. Starting with vCenter Server 6.0, increasing the total memory provisioned for the vCenter Server appliance will automatically allocate more memory to each Java service after the vCenter Server virtual machine is rebooted.

Running vCenter on upgraded hardware will also provide better performance. In an experiment similar to the one described in the previous section "Performance Comparison," a large inventory test achieved 60% better performance on a server with Intel Xeon Processors E5-2670 v3 (code-named Haswell-EP), compared to a server with Intel Xeon Processors E7-8837 (code-named Westmere-EX).

Proper resource provisioning for the vCenter Server database (either embedded or external) is crucial for overall vCenter Server performance. Placing the database tables and logs on separate disk spindles can improve database performance. Using SSDs for the database can also improve performance. Finally, it may be necessary to limit database growth by setting appropriate retention policies for event and task data. Reducing the retention time for events and tasks will reduce the amount of data stored in the database, and can improve database performance. The default retention time is 30 days; at that setting, no additional tuning should be required. Refer to the vSphere documentation for details on setting the retention time [4].

Network Configuration

The vCenter database must be connected to vCenter by a network path with adequate bandwidth. As shown in Figure 12, a vCenter Server running over 1,000 operations per minute generates more network traffic than a single 100 Mbit/s network link can handle.

When using an external database, a low-latency network link (less than 10ms, preferable less than 1ms round-trip time) between vCenter Server and its database is recommended for best performance. The effect of a longer latency network is more pronounced in high-churn environments.

ESXi hosts should be attached to vCenter Server by network paths with the lowest possible latency. Using a high-latency network link between vCenter and ESXi hosts is fully supported, but will result in higher latencies for vCenter Server operations [5].

vCenter Server Configuration and Inventory Design

Environments that use vSphere Distributed Switches (VDS) can attain better performance by using multiple switches, instead of a single VDS for the entire vCenter Server inventory. Generally, having fewer virtual machines on any one switch will increase overall vCenter Server throughput. An experiment with the Large configuration, using four switches instead of one, achieved a 14% increase in throughput.

vCenter Server 6.0 provides the option to deploy a separate Platform Services Controller (PSC), which provides common services shared between multiple vCenter Servers. The PSC can also be embedded into the vCenter Server in smaller configurations. (See Figure 14.) Also, vCenter Server 6.0 provides an embedded vPostgres database server as an option for any environment. Organizational or other concerns may dictate choosing one configuration over another, but an embedded PSC will provide the best performance for a single vCenter Server, as long as adequate resources are provisioned for it. For a full description of vCenter Server deployment options, see the vCenter Server 6.0 deployment guide [3].
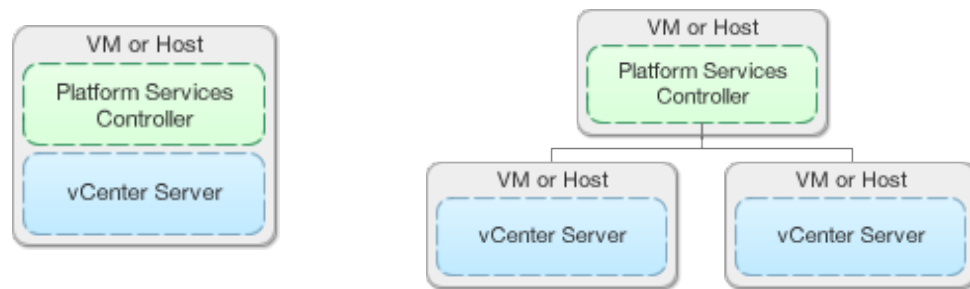
Figure 14. vCenter Server 6.0 deployment configurations. At left, embedded configuration with vCenter Server and Platform Services Controller (PSC) together. At right, PSC deployed separately and shared by several vCenter servers.

Finally, the choice of vCenter Server statistics level can affect performance. Increased statistics levels will result in more traffic to the vCenter Server database, causing increased storage traffic (and network traffic, if not using the embedded vPostgres database). VMware recommends keeping statistics at Level 1 unless a higher statistics level is needed, and using higher statistics levels only as long as needed. See "Data Collection Levels" [6] for more details on statistics levels.

Performance Best Practices for the Web Client

As with other vCenter components, Web Client performance depends on proper provisioning. Ensure that your vCenter Server has been provisioned with adequate CPU, memory, and storage. Refer to the vSphere 6.0 installation and setup guide [2] for full details on provisioning.

Solutions that provide plugins for the Web Client (for example, VMware NSX) may require additional memory for the Web Client. If you use several such solutions, you might need to provide additional memory to the Web Client server by increasing the amount of memory provisioned to vCenter Server.

Web Client performance also depends on adequate provisioning for the client. VMware recommends using the client on a machine with at least 2 CPUs and 4GB of RAM. Also, Web Client performance is affected by network latency between the Web Client server and Web browser clients; minimize client-server latency as much as possible for best performance. For example, minimize the number of router hops between the Web Client server and your Web browser. To access the Web Client server from a different geographical area, use RDP to connect to a machine in the same geographical area as the Web Client server, and then use a Web browser on that machine to access the Web Client.

Conclusion

The performance comparisons in this paper demonstrate several important points:

- vCenter Server 6.0 exhibits substantially improved performance over vCenter Server 5.5
- The vCenter Server Appliance achieves at-scale performance that is comparable to vCenter Server on Windows.
- CPU, memory, disk, and network resources affect vCenter Server's inventory size and throughput performance.
- vCenter Server users should follow best practices to achieve optimal vCenter Server performance, including proper resource provisioning, inventory layout, and configuration of the vCenter Server environment and infrastructure.

Appendix A. Inventory Configuration Details

Inventory sizes used in this performance study are shown in Table 5.

Size	Clusters	Hosts	Registered VMs	Powered-On VMs
Small	3	96	1440	960
Medium	13	416	6240	4160
Large	32	1024	15360	10240

Table 5. Inventory sizes for tests presented in this paper.

These sizes were chosen to approximate the supported vCenter Server configuration limits. "Powered-On VMs" is 2/3 of "Registered VMs." Inventories were configured with 1 vSphere Distributed Switch with portgroups of 14 ports each: small, 404 portgroups; medium, 1,747 portgroups; large, 4,300 portgroups.

Each virtual machine was configured with 1 vCPU, 1 GB RAM, and 2 vNICs.

In addition, the "Large" inventory was also tested with a vCenter with 24 vCPUs and 48GB RAM; the external database (where applicable) was configured as for "Large" above.

Appendix B. Monitoring vCenter Resource Usage

The vCenter Server Appliance provides vimtop, a tool for monitoring its resource usage. To access vimtop, type `vimtop` at the appliance console. Pay particular attention to the overall CPU and memory stats at the top of the screen. High values for user CPU, system CPU, and swap memory used, or a low value for memory free, suggest that allocating additional resources may increase vCenter performance.

For vCenter Server on Windows, the Windows Performance Monitor (perfmon) can provide similar stats. Refer to the Windows Performance Monitor documentation for details on its use.

Appendix C. Tiny Environment

In addition to the three inventory sizes listed in the previous section "Performance Comparison," vCenter Server supports a fourth configuration, "Tiny," which supports up to 10 hosts and 100 virtual machines, and is intended for home labs and pre-production environments. A Tiny vCenter Server requires 2 vCPUs and 8GB of RAM. This section describes an experiment performed on a Tiny vCenter Server in order to demonstrate its suitability in lab environments.

The vCenter Server Appliance 6.0 was tested in its Tiny configuration with an environment with these specifications: 10 hosts, 150 registered VMs, 100 powered-on VMs, and 1 vSphere Distributed Switch with 42 portgroups, each with 14 ports. The workload was run at loads of 1, 2, and 5 API clients.

The heaviest workload achieved a median latency of 2.41 seconds, and a throughput of 119 operations per minute. This throughput is higher than the throughput typically seen in a lab or pre-production environment, and this performance is as good as or better than the performance seen in the Small, Medium, and Large configurations when the difference in inventory size is taken into account.

As with the Small, Medium, and Large configurations, the Web Client shows performance improvements in the Tiny configuration, making it suitable for managing testing or small production environments. Initial login is around 15 seconds or less; most action menus display in 0.5 seconds or less; and most summary pages appear in up to 1.5 seconds, with many appearing in no more than 0.5 seconds.

References

[1] VMware, Inc. (2015, March) What's New in VMware vSphere 6?
https://www.vmware.com/files/pdf/vsphere/VMware-vSphere-Whats-New.pdf

[2] VMware, Inc. (2015, March) vSphere 6.0 Installation and Setup. http://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-installation-setup-guide.pdf

[3] VMware, Inc. (2015, March) VMware vCenter 6.0 Deployment Guide.
https://www.vmware.com/files/pdf/techpaper/vmware-vcenter-server6-deployment-guide.pdf

[4] VMware, Inc. (2015) Configure Database Retention Policy. https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.vcenterhost.doc%2FGUID-DD999601-2F41-498D-A577-DDC7C06648EE.html

[5] Fei Chen. (2012, June) Performance of VMware vCenter 5.0 in Remote Offices and Branch Offices.
http://www.vmware.com/files/pdf/techpaper/VMW-WP-Performance-vCenter.pdf

[6] 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

About the Authors

Mike Stunes is a senior member of technical staff in the vCenter performance group at VMware. He focuses primarily on internal tools and automation for improving vCenter performance and scalability. Mike holds SB and MEng degrees in electrical engineering and computer science from MIT.

Ravi Soundararajan is a principal engineer in the performance group at VMware. He focuses primarily on vCenter Server performance and scalability, and he has given numerous presentations at VMworld and Partner Exchange on vCenter and vSphere API performance. He holds an SB degree in electrical engineering from MIT, and an MS andPhD in electrical engineering from Stanford University. Find Ravi on Twitter as @vCenterPerfGuy.

Acknowledgements

vmware®