VMware vCloud® Director™ 1.5 Performance and Best Practices

Performance Study

TECHNICAL WHITE PAPER

Table of Contents

Introduction

VMware vCloud® Director™ 1.5 gives enterprise organizations the ability to build secure private clouds that dramatically increase datacenter efficiency and business agility. Coupled with VMware vSphere®, vCloud Director delivers cloud computing for existing datacenters by pooling virtual infrastructure resources and delivering them to users as catalog-based services. vCloud Director 1.5 helps you build agile infrastructure-as-a-service (IaaS) cloud environments that greatly accelerate the time-to-market for applications and responsiveness of IT organizations. vCloud Director 1.5 adds the following new features specific to accelerating application delivery in the cloud:

- Fast Provisioning
- vCloud Custom Guest Data
- Expanded vCloud API and SDK
- vCloud API Query Service
- vCloud Messages
- Cisco Nexus 1000v Integration
- vSphere 5.0 Support
- Microsoft SQL Server Support
- Globalization
- vShield Five Tuple Firewall Rules
- Static Routing
- IPSec Site-to-Site VPN

This white paper addresses three areas regarding vCloud Director performance:

- vCloud Director sizing guidelines and software requirements
- Best practices in performance and tuning
- Performance characterization for key vCloud Director operations

vCloud Director Architecture

Figure 1 shows the deployment architecture for vCloud Director. A customer accesses vCloud Director by using a Web browser or REST API. Multiple vCloud Director Server instances can be deployed with a shared database. In the vCloud Director 1.5 release, both Oracle and Microsoft SQL Server databases are supported. A vCloud Director Server instance connects to one or multiple VMware vCenter™ Servers. From now on, we use *vCloud Director Server instance* and *cell* interchangeably.

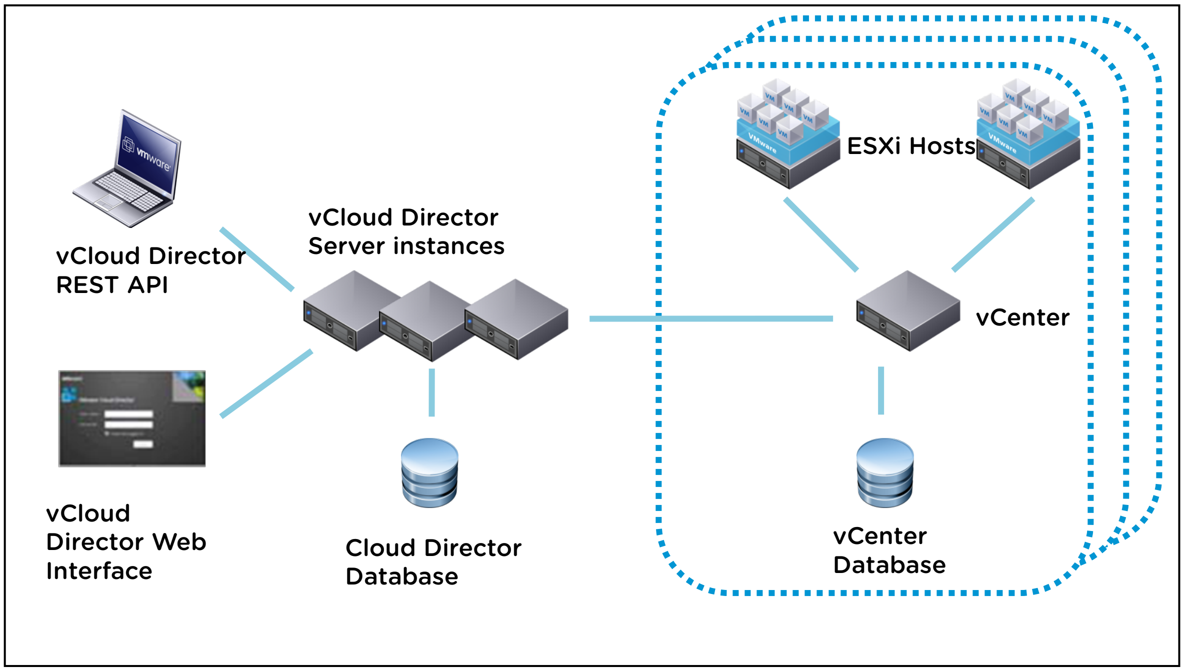

Figure 1. VMware vCloud Director high level architecture

Next we introduce the definitions for some key concepts in vCloud Director 1.5. These terms have been used extensively in this white paper. For more information, refer to the *vCloud API Programming Guide*[8].

vCloud Organization

A vCloud organization is a unit of administration for a collection of users, groups, and computing resources. Users authenticate at the organization level, supplying credentials established by an organization administrator when the user was created or imported.

vCloud Virtual Datacenters

A vCloud virtual datacenter (vDC) is an allocation mechanism for resources such as networks, storage, CPU, and memory. In a vDC, computing resources are fully virtualized and can be allocated based on demand, service level requirements, or a combination of the two.

There are two kinds of vDCs:

- **Provider vDCs**
  A provider virtual datacenter (vDC) combines the compute and memory resources of a single vCenter Server resource pool with the storage resources of one or more datastores available to that resource pool.

  Multiple provider vDCs can be created for users in different geographic locations or business units, or for users with different performance requirements.
- **Organization vDCs**
  An organization virtual datacenter (vDC) provides resources to an organization and is partitioned from a provider vDC. Organization vDCs provide an environment where virtual systems can be stored, deployed, and operated. They also provide storage for virtual media, such as floppy disks and CD-ROMs.

  A single organization can have multiple organization vDCs.

  An organization administrator specifies how resources from a provider vDC are distributed to the vDCs in an organization.

Catalogs

Organizations use catalogs to store vApp templates and media files. The members of an organization that have access to a catalog can use the catalog's vApp templates and media files to create their own vApps. A system administrator can allow an organization to publish a catalog to make it available to other organizations. Organization administrators can then choose which catalog items to provide to their users.

Catalogs contain references to virtual systems and media images. A catalog can be shared to make it visible to other members of an organization and can be published to make it visible to other organizations. A vCloud system administrator specifies which organizations can publish catalogs, and an organization administrator controls access to catalogs by organization members.

Test Environment

For the experiment results in this paper, we used the following test bed setup. Actual results may vary significantly and depend on many factors including hardware and software configuration.

Hardware Configuration

- **vCloud Director Cell:** 64-bit Red Hat Enterprise Linux 5, 4 vCPUs, 8GB RAM
- **vCloud Director Database:** 64-bit Windows Server 2003, 4 vCPUs, 8GB RAM
- **vCenter**: 64-bit Windows Server 2003, 4 vCPUs, 8GB RAM
- **vCenter Database:** 64-bit Windows Server 2003, 4 vCPUs, 8GB RAM
- All of these components are configured as virtual machines and are hosted on a Dell PowerEdge R610 box with 8 Intel Xeon CPUs@2.40GHz, and 16GB RAM.

Software Configuration

- **vCenter:** vCenter Server 5.0
- **vCenter Database:** Oracle DB 11g
- **vSphere Host:** vSphere ESXi 5.0
- **Storage:** Dell EqualLogic model 70-0115

Oracle Database

A database server configured with 16GB of memory, 100GB storage, and 4 CPUs should be adequate for most vCloud Director clusters.

The database must be configured to allow at least 75 connections per vCloud Director cell plus about 50 for Oracle's own use. Table 1 shows how to obtain values for other configuration parameters based on the number of connections, where *C* represents the number of cells in your vCloud Director cluster.

ORACLE CONFIGURATION PARAMETER	VALUE FOR *C* CELLS
CONNECTIONS	75*C+50
PROCESSES	= CONNECTIONS
SESSIONS	= CONNECTIONS*1.1+5
TRANSACTIONS	= SESSIONS*1.1
OPEN_CURSORS	= SESSIONS

Table 1. Oracle Database Configuration Parameters

For more information on best database practices, refer to *vCloud Director Installation and Configuration Guide*[10].

Latency Overview for Frequent Operations

In this section, we present the latency overview for some typical vCloud Director Operations. Figure 2 shows latency results for the following operations, which are performed by a group of eight users simultaneously. Note that these latency numbers are only for reference. Actual latency could vary significantly with different deployment setups.

- Clone vApp in workspace
- Capture vApp as template from workspace to catalog
- Instantiate vApp from template to workspace
- Delete vApp in workspace
- Delete vApp template in Catalog
- Edit vApp
- Create Users
- Deploy vApp in workspace, with or without a fence
- Undeploy vApp in workspace, with or without a fence

Clone vApp, capture vApp, instantiate vApp all involves VM clone operations. Clone vApp occurs as a workspace-to-workspace copy inside of vCloud. Capture vApp includes a copy operation from workspace to catalogue. Instantiate vApp includes the copy from catalogue to workspace. The vApp and vApp template we tested include a single VM with the same size (400MB).

Figure 2 shows that with a linked clone, performance improves for all these operations. (Note that in Figure 2, (F) means full clone and (L) means linked clone.) We also observed the instantiate is faster than clone and capture operations, either in the full clone or linked clone case. For more information on a performance comparison between a linked clone and a full clone, refer to "Linked Clone." Other operations, including delete vApp, delete vApp template, edit vApp, and create users take only a minimal amount of time.

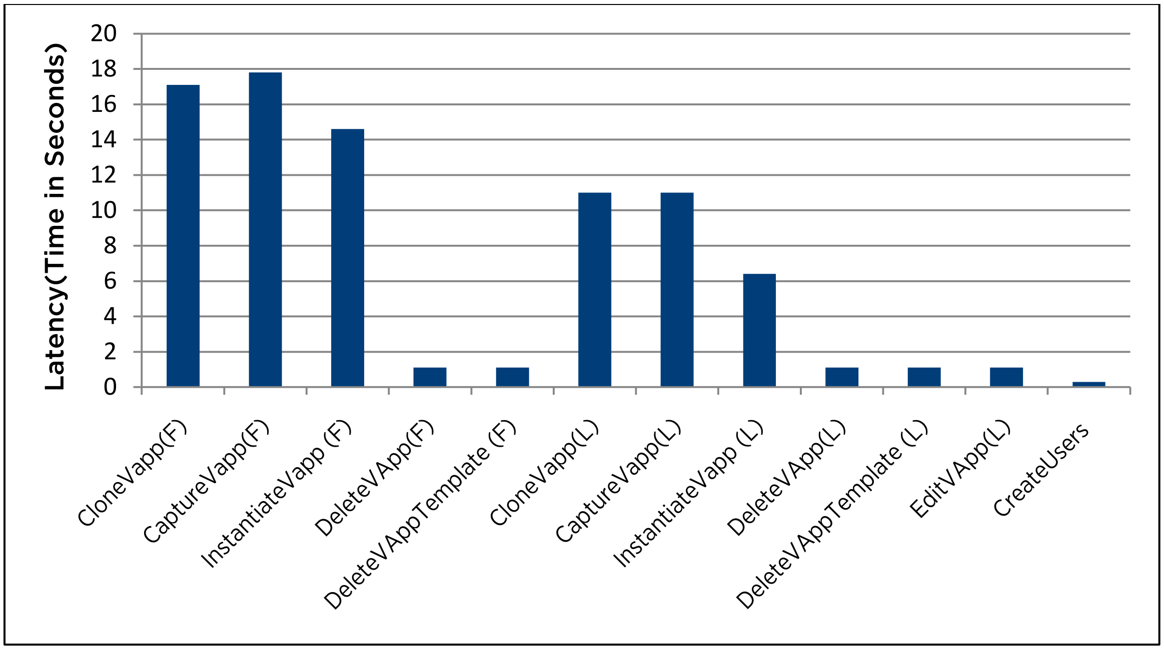

Figure 2. Latency overview for vCloud Director operations

Next, we looked into vApp deployment performance. vApp can be deployed with or without a fence as Figure 3 shows.

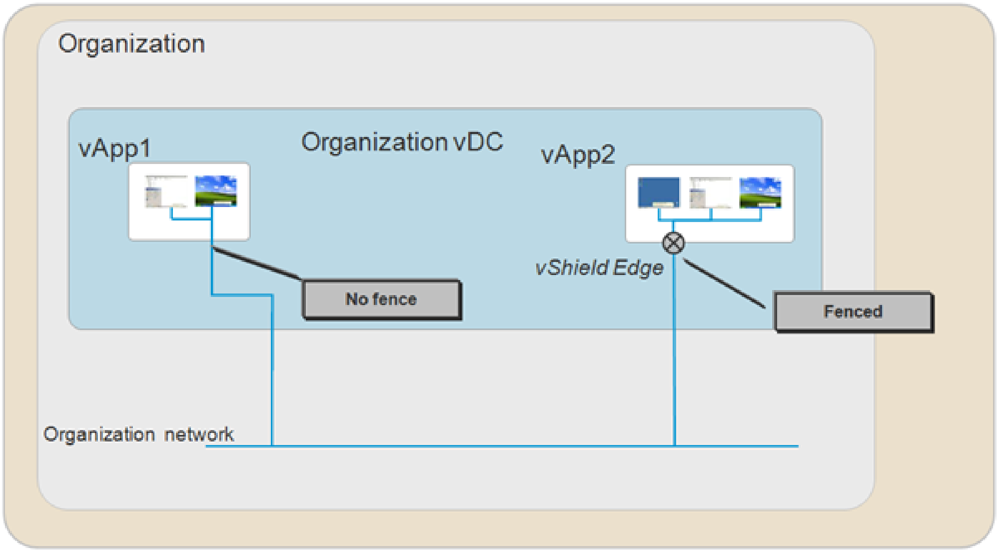

Figure 3. Deploy vApp with or without fence

Fence deploy and undeploy operations take extra time when compared to deploying and undeploying a vApp without a fence. This is because vCloud Director needs to perform extra configuration in order to deploy vApp with a fence. When a vApp is deployed without a fence, the vApp directly connects with the organization network. When a vApp is deployed with a fence, the connections between the vApp and the organization network traverse a vShield Edge virtual appliance, which provides protection for the vApp network and also enables extension of the organization network to run identical virtual machines in different vApps. By extending the organization network in this way, it is possible to run multiple identical vApps without conflict: the vShield Edge deployed on a per-vApp network basis isolates the overlapping Ethernet MAC and IP addresses. For more information on the vCloud Director network, including configuration details for various organization networks and vApp networks, please refer to vCloud Director Administrator's Guide 1.5[12] and vCloud Director User's Guide 1.5[13].

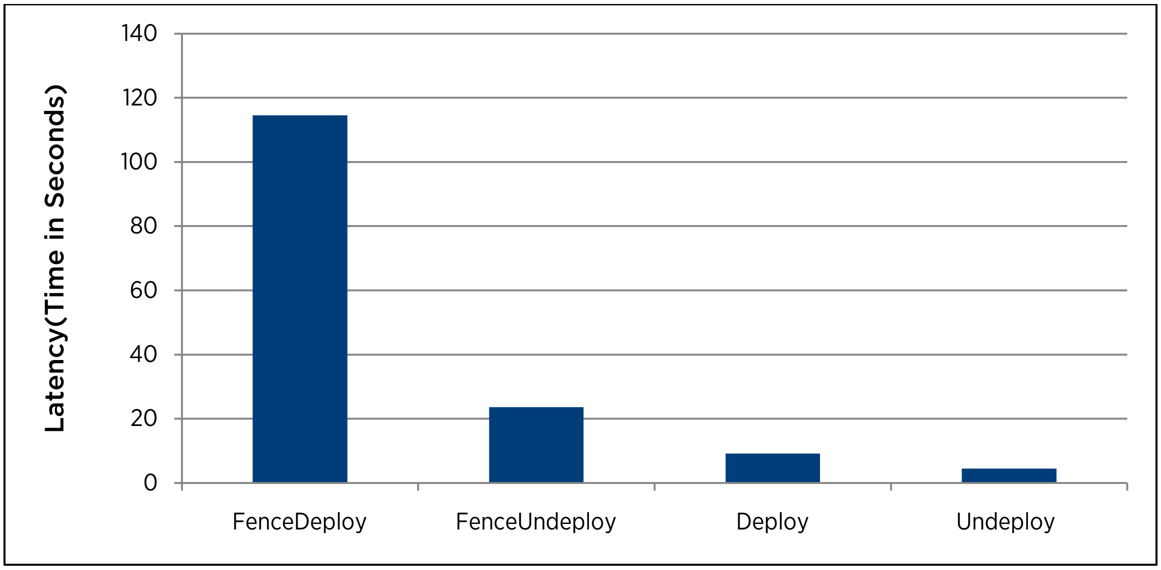

Figure 4. Fenced and unfenced deployment operations

Linked Clone

vCloud Director 1.5 provisions quickly using linked clones. Linked clones utilize the vSphere redo-log linked clone implementation to provide statistically fast VM provisioning within and across datastore and vCenter boundaries. Compared with full clone, linked clone improves agility in the cloud by reducing provisioning time, providing near-instant provisioning of virtual machines in a cloud environment.

NOTE: Fast Provisioning is supported only for vSphere 5.0. Mixed clusters of ESX/ESXi 4.x and ESXi 5.0 with vCenter 5.0 is not supported.

For the experiments in this section, the test bed is configured as described in "Test Environment." Other configurations include:

- Each test vApp has a single virtual machine.
- The virtual machine operating system is Linux 5.0.
- The test vCloud Director cell has one datacenter and two clusters.
- Two ISCSI datastores are connected.
- Each cluster has two vSphere hosts.

Comparison between Full Clone and Linked Clone

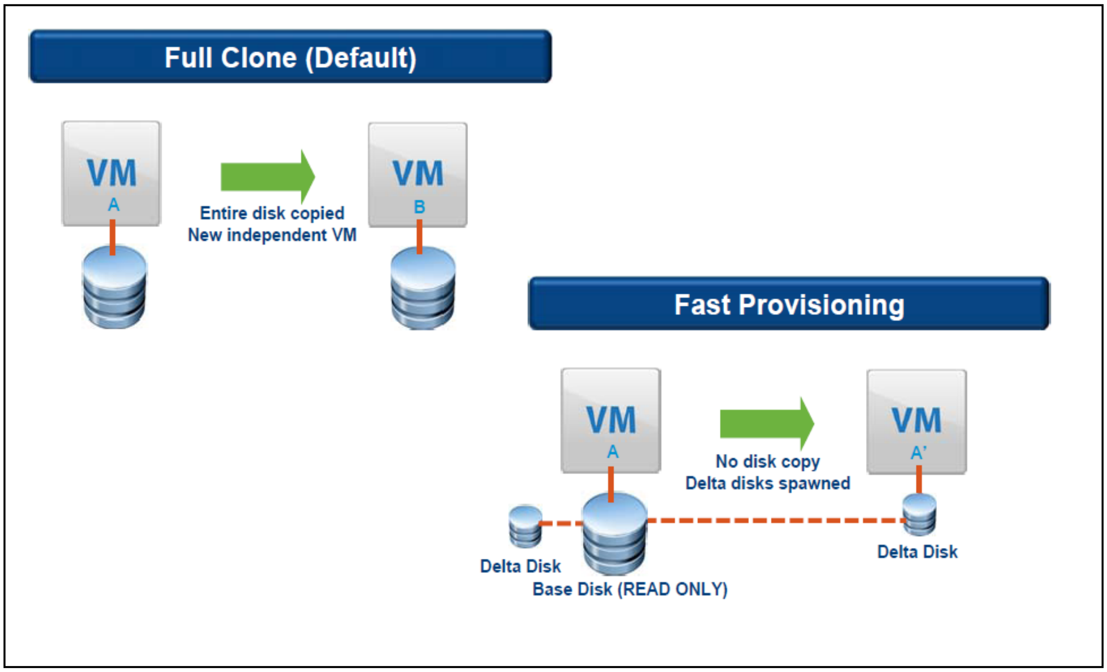

Figure 5. Linked clone and full clone

If provisioning a virtual machine using full clone, the entire virtual disk is replicated, then a new independent virtual machine is created. For linked clone, a delta disk will be created and a link with the base disk. Typically in the virtual machine in full clone, writes go to the VMDK and reads come from the same VMDK. In Figure 5, virtual machine A is a primary virtual machine in which reads and writes go to the same VMDK. When a linked clone virtual machine is provisioned, as shown by virtual machine A, a small 16MB VMDK is created. This VMDK is an empty delta disk that serves to capture disk writes for the newly created virtual machine. This takes very little time to create and consumes a very small amount of disk space. Writes to disk from virtual machine A then go to this delta disk, which grows to accommodate the writes. Disk read operations traverse up the linked clone chain until the desired block is found.

We conducted an experiment to compare the clone operation latency between full clone and linked clone. The virtual machine had one chain hop after the clone operation. We measured the linked clone latency by copying a vApp with a single virtual machine from a catalog to My Cloud work space. The virtual machine had one virtual disk. The virtual disk sizes were different for each run. Figure 6 shows the results of this test.

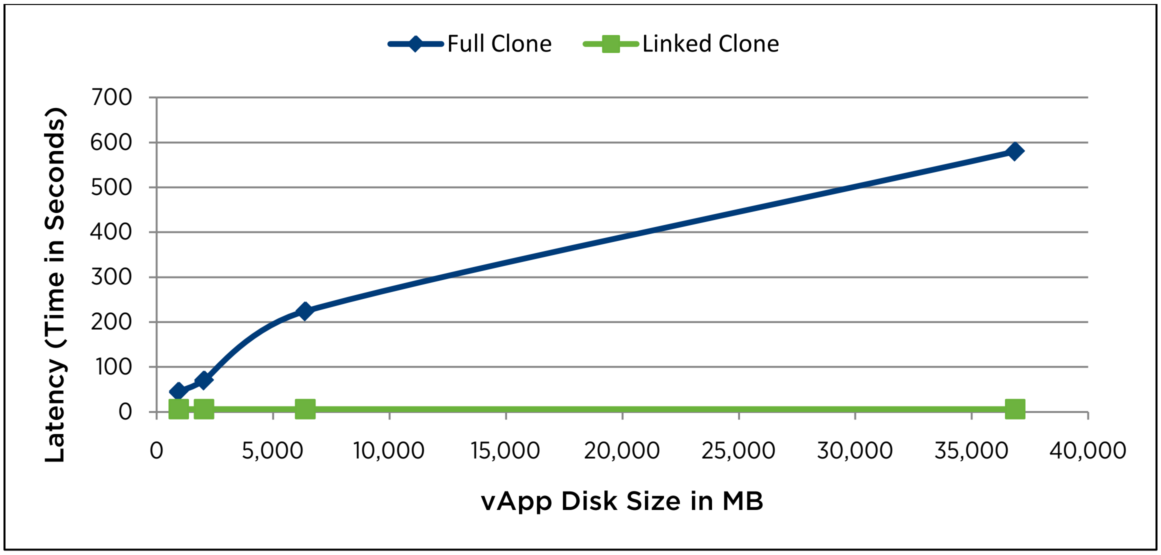

Figure 6. Linked clone and full clone latency in various disk sizes

Figure 6 shows that the latency of a vApp full clone increases as the vApp disk size grows. Linked clone latency remains the same short time regardless of the vApp disk size. During our tests, we measured linked clone latency as between 7-9 seconds. Because a linked clone is a copy of a delta disk file with a consistent small size, the operation latency is not increased by the primary vApp disk size.

NOTE: VMs with I/O-intensive workloads might not benefit from using linked clones. See "I/O Workflows for Linked Clone" for details.

Chain Length Limit

Every time a linked clone is created from a VM, a new delta disk is created on the primary, which increases the chain length by one. As more virtual machines are created, the linked clone chain increases in size and VM performance will begin to degrade. To ensure optimal linked clone performance, vCloud Director limits the chain length to 30. If the vApp is copied more than 29 times through a linked clone operation, the operation will change to a full clone. When this occurs, the clone response time increases as the underlying clone method is changed to a full clone. It is not possible to shorten the chain length by deleting the cloned VM because the delta file on the primary VM cannot be removed. Thus, the linked clone becomes a full clone after 29 linked clone copies, regardless of the deletions of the cloned vApp.

There are five types of operations involved in creating a linked clone:

- **Clone vApp**
  Occur as a workspace-to-workspace copy inside of vCloud Director.
- **Compose vApp**
  Build a vApp from existing virtual machines, including virtual machines contained by vApps and vApp templates
- **Capture vApp**
  Occur as a copy operation from workspace to catalog.
- **Instantiate vApp**
  Occur as a copy operation from catalog to workspace.
- **Clone template**
  Occur as a catalog-to-catalog copy inside of vCloud Director.

All five types of operations might have a chance to hit the linked clone limit by sequentially cloning, as shown in Figure 7.

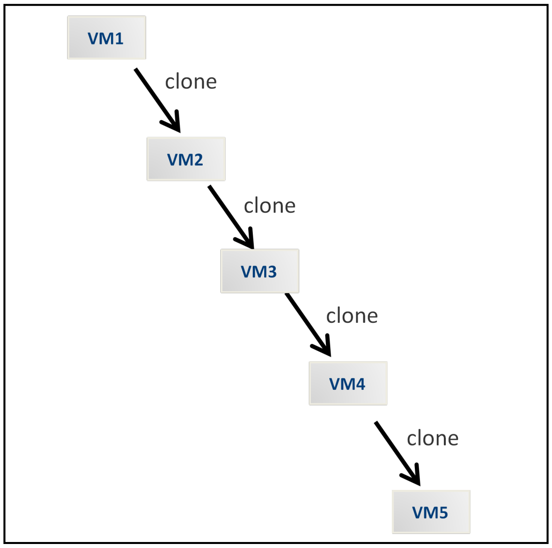

Figure 7. Linked clone chain length limit is 30

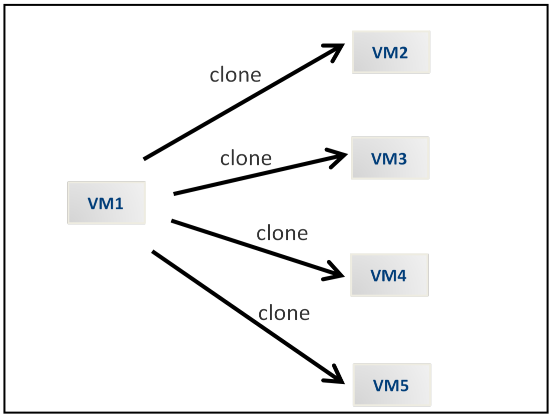

Figure 8. Linked clone from the same VM

Out of these five linked clone types, clone vApp, capture vApp, and compose vApp may encounter the chain length limit through the cloning pattern shown in Figure 8. The primary VM of clone vApp, capture vApp, and compose vApp from an existing vApp is located in the workspace. Because you can change the primary VM in the workspace once it is powered on, a running virtual machine needs to create a new snapshot point each time it is cloned. This creates a new delta disk on the primary, increasing the chain length by one. Conversely, the primary VM of the instantiate vApp, clone template, and compose-from-template vApp is in the catalog. This VM is not changeable since templates cannot be powered on. For these linked clone operations, the chain length will not increase after each clone operation. So we recommend you use instantiate vApp and clone template operations when possible to avoid the performance impact from the chain length limit.

Performance tuning tips:

- Since there is no chain length increase by using a template to clone a vApp, use the template for the cloning operation instead of vApp copy. For instance, if hundreds of new vApps need to be copied from an existing vApp, it would be better to capture a vApp to a template. After you create the template, copy it to the target organizations.
- For scenarios that need to generate many templates from a vApp, do not directly run capture vApp many times. Instead, capture this vApp to a template and copy the newly created template to catalogs in different organizations. In this way, the linked clone chain increase will be kept to a minimum.
- If the current clone has become very slow compared to the prior clone, the clone may have hit the snapshot chain length limit. This can be resolved by virtual machine consolation.

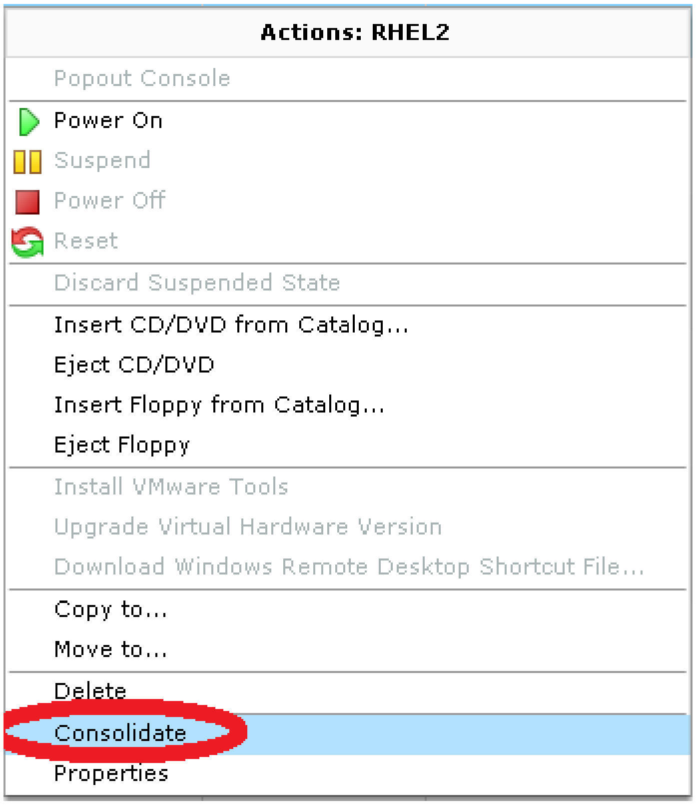

Figure 9. A snapshot of virtual machine actions menu

To consolidate a virtual machine:

1. Identify the primary VM in the vCloud Directory user interface.
2. Under My Cloud tab, in the left panel, click the VM and find the corresponding VM in the right panel.
3. Right click on the VM. The Actions menu appears, as shown in Figure 9.
4. Select Consolidate.

To check chain length number:

1. Select Properties in the Actions menu as shown Figure 9.
2. Find Chain Length in the Virtual Machine Properties window, which is shown in Figure 10.

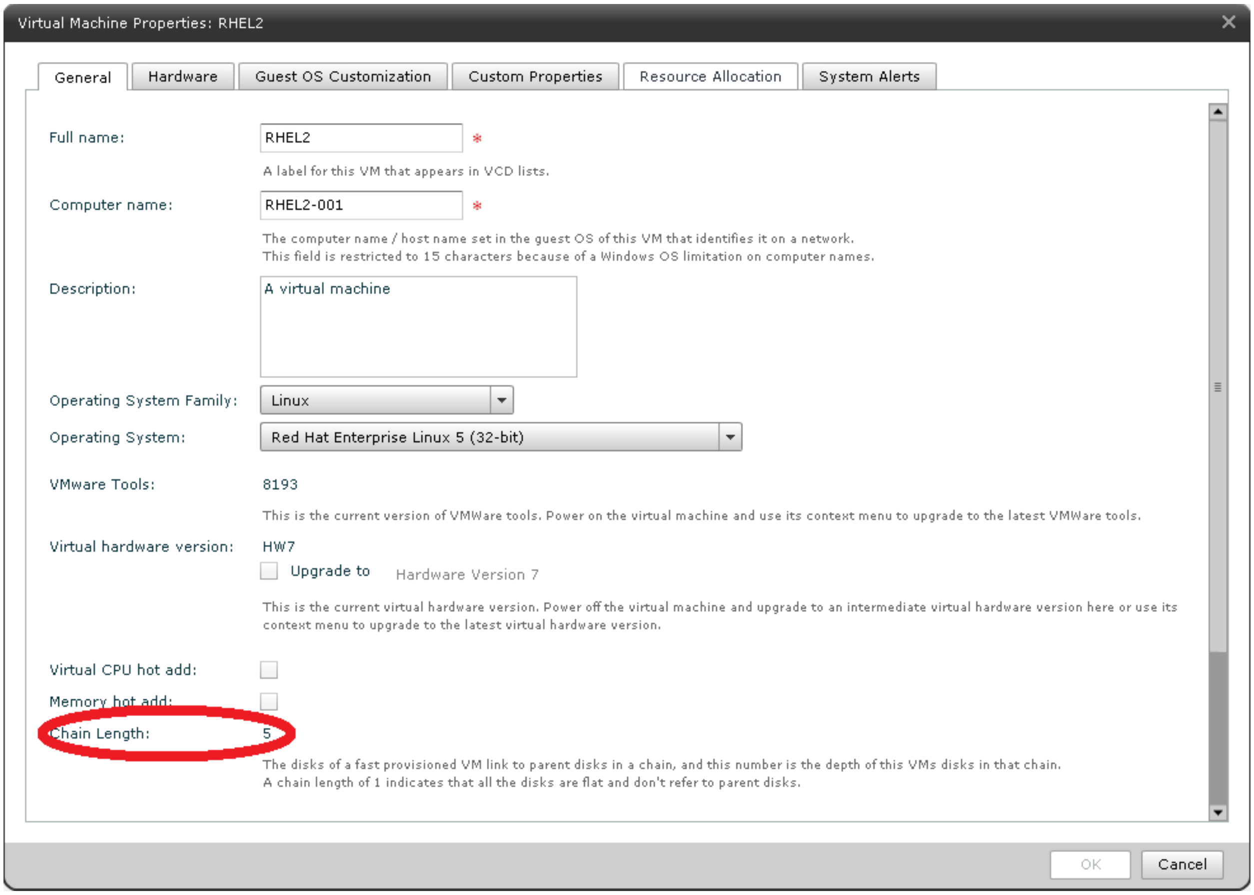

Figure 10. A snapshot of virtual machine properties

Scalability

The vCloud environment is architected to scale and support a large number of users. To ensure there is no impact for linked clone performance by concurrent users, we conducted an experiment of running a clone operation with one to 40 concurrent users. In the experiment, each vApp had a single virtual machine. Multiple users concurrently and consecutively cloned the respective vApp. We recorded the average latency regarding concurrent user number.

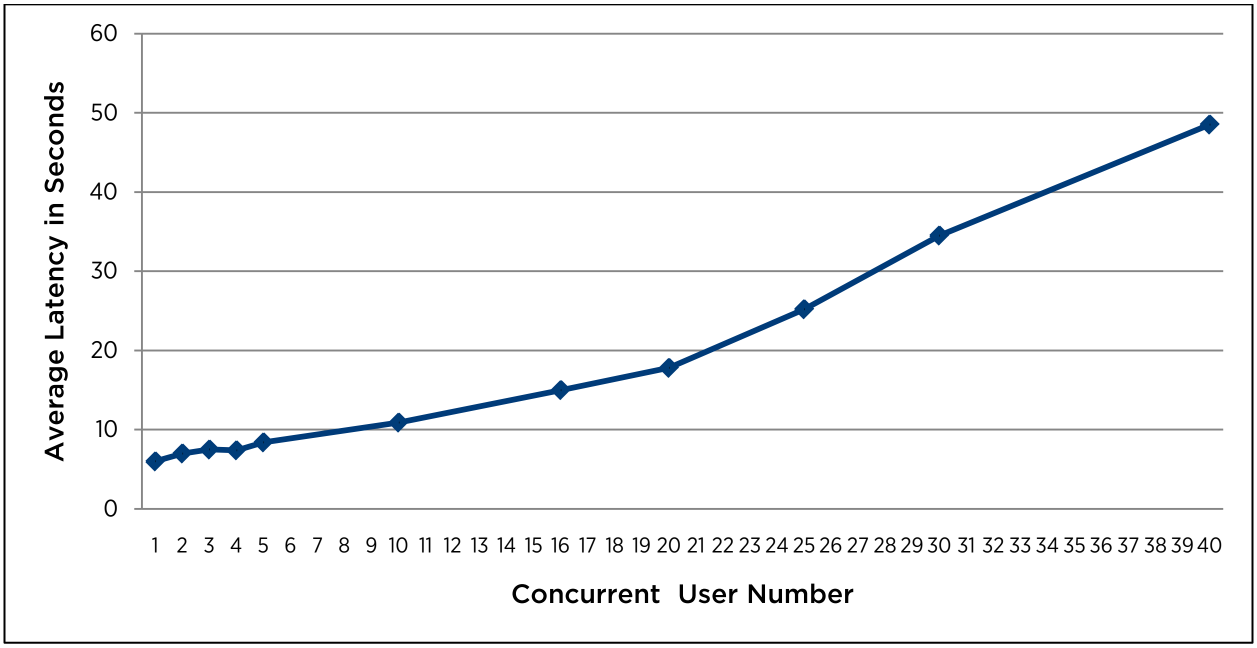

Figure 11. Linked clone latency regarding various concurrent user number

Figure 11 shows the result of our test. The average latency grows linearly as concurrent user number increases. This means that when vApp linked clone operations are performed concurrently, users will not expect any significant performance degradation.

Linked Clones across Datastore and vCenter

The direct creation of linked clones in vCloud Director is limited to a single datastore. To enable linked clones to be deployed across datastores in the cloud, vCloud Director uses a mechanism called shadow copying. When vCloud Director determines that it would be more advantageous to place a clone on a different datastore than that on which the source resides, a shadow copy is created. This shadow copy is a full clone on the destination datastore from which other linked clones can be built. Such a copy happens without user intervention, and substantially reduces any storage management overhead that might occur in using linked clones across datastores. In Figure 12, a shadow virtual machine (VM S) is first created when a linked clone must be placed on a different datastore than the source. This shadow copying is made regardless of whether the destination resides in the same vCenter server or in a different vCenter server. If the request is made to a different vCenter server, vCloud Director uses its image-transfer service to make a copy to the new vCenter server. Again, no special configuration is required from the vCloud administrator for this to happen. After the shadow virtual machine is created, subsequent linked clones (VM L in Figure 12) are as fast as linked clones from the original datastore.

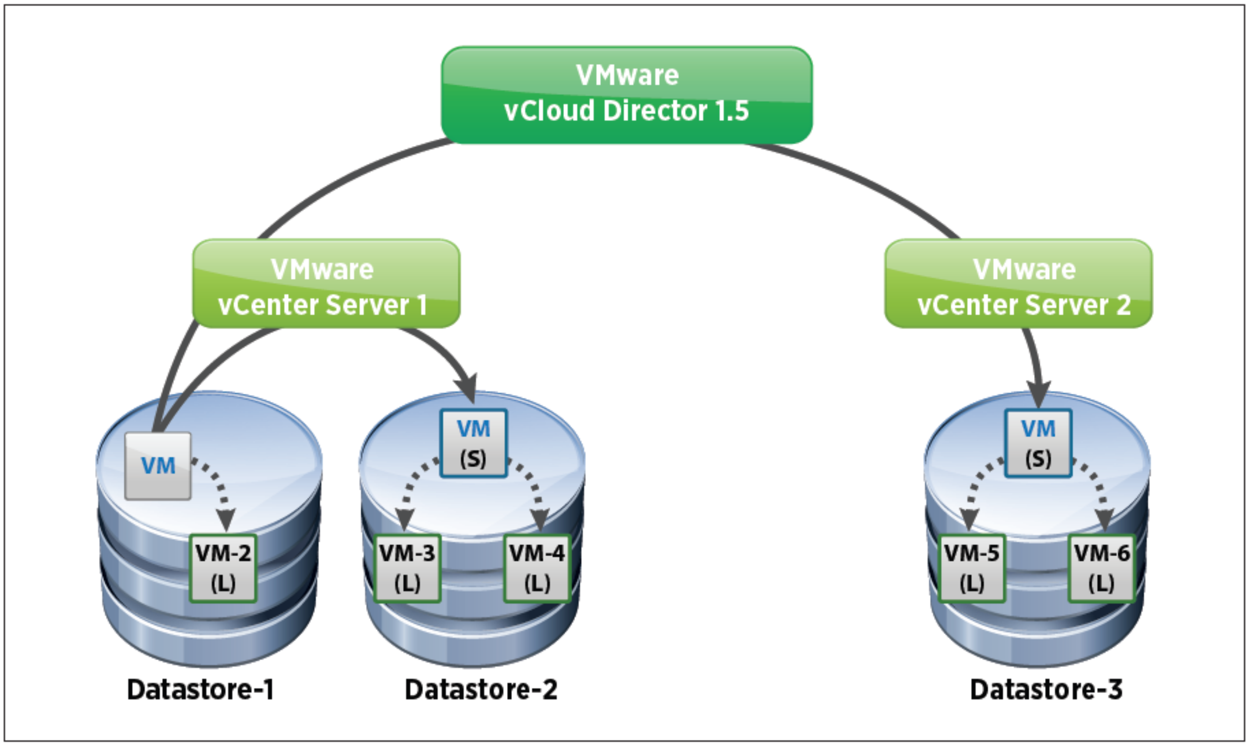

Figure 12. Shadow virtual machines deployed across datastores in the same vCenter Server and across vCenter Servers

For instance, a template is copied across vCenter servers to a different datastore. Since there is no support for cross-vCenter linked clones, a shadow VM is created in the destination datastore and registered with the destination vCenter. A linked clone is created out of the shadow VM. This is illustrated in Figure 12, where the primary VM, registered with "VC-1" and stored in "Datastore-1" is shadowed to "Datastore-2" and registered with "VC-2," and finally a linked clone is created out of it.

Shadow VM Copy

This section compares the latency of deploying linked clones and full clones in the same vCenter and with different target datastores. The vApp tested had a single 4GB virtual machine.

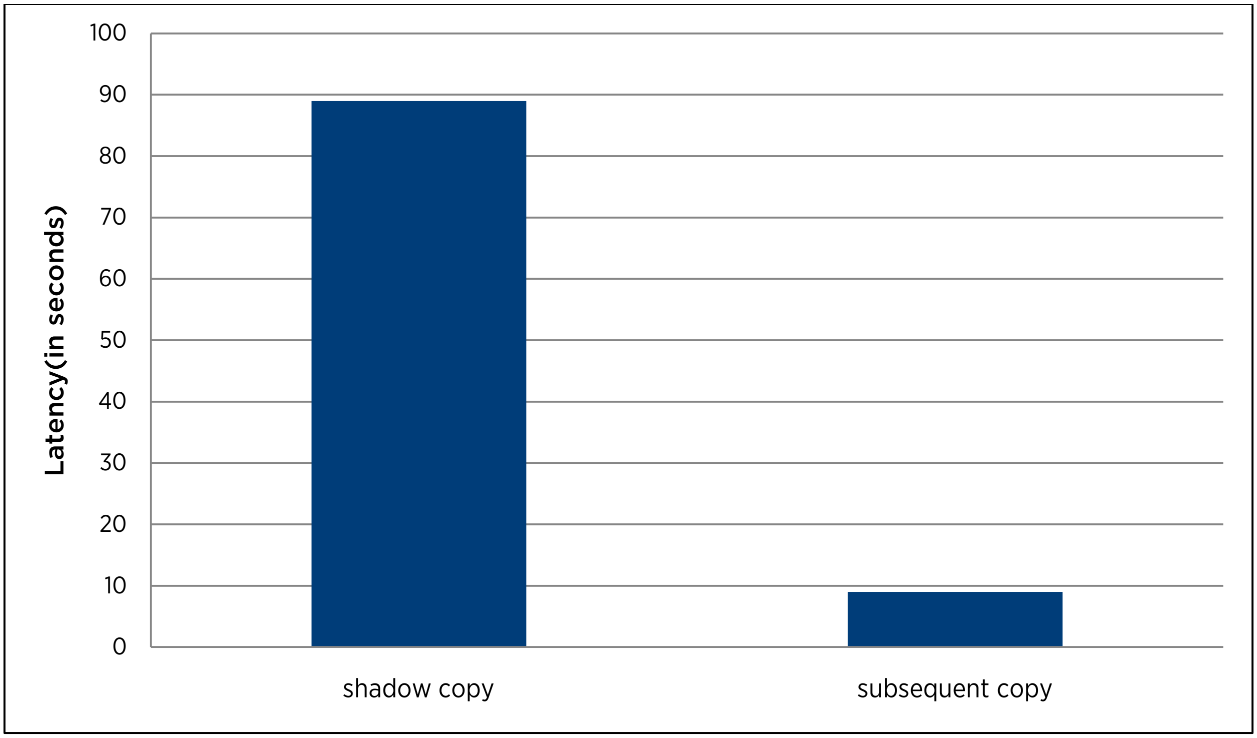

Figure 13. Latency comparison of first clone and subsequent clone for clone across datastore within same vCenter

As shown in Figure 12, a shadow VM is created during the initial clone. In the destination datastore, it actually performs as a full clone, which takes 89 seconds as shown in Figure 13. The subsequent clone is created as fast as a linked clone from the original source.

Performance tuning tips:

- Before starting critical operations, if the primary vApp and target vApp are on different datastores, pre-allocate the vApp to the target datastore by deploying a linked-clone across the datastore or vCenter. This shortens the subsequent copy time because the shadow VM has been already created in the desired datastore.

  If you notice that the datastore reaches a red threshold in vCenter, copy the primary vApp or template to a different datastore with sufficient capacity. This action will force a shadow copy to occur across datastores. Otherwise, the linked clone operation will fail.
- When removing a shadow VM, ensure all VMs cloned from this VM are also removed. After doing this, the removal of the primary VM in the source datastore will also remove the shadow VM in the target datastore.

Datastore Accessibility

In vCloud Director 1.5, when a VM clone operation occurs across a datastore, a shadow VM is created as the initial clone. Datastore accessibility plays a very important role in terms of the latency of shadow VM creation.

To demonstrate this, we created two provider vDCs based on two clusters from the same vCenter server. We derived the corresponding organization vDCs from the two provider vDCs. These two organization vDCs are presented in the same organization.

The way in which the datastores are connected to the ESX hosts will impact the first linked clone latency when cloning across datastores. In Figure 14, there are two separated datastores. The source vApp template resides on datastore 1. The instantiate operation will create a new vApp in Org vDC2. The new vApp will reside in datastore

2. Because ESX hosts in Org vDC2 do not have access to datastore 1, the shadow VM will be uploaded to the cell first and then a deployment OVF package API call will be issued to the vCenter server that is mapped to Org vDC2.

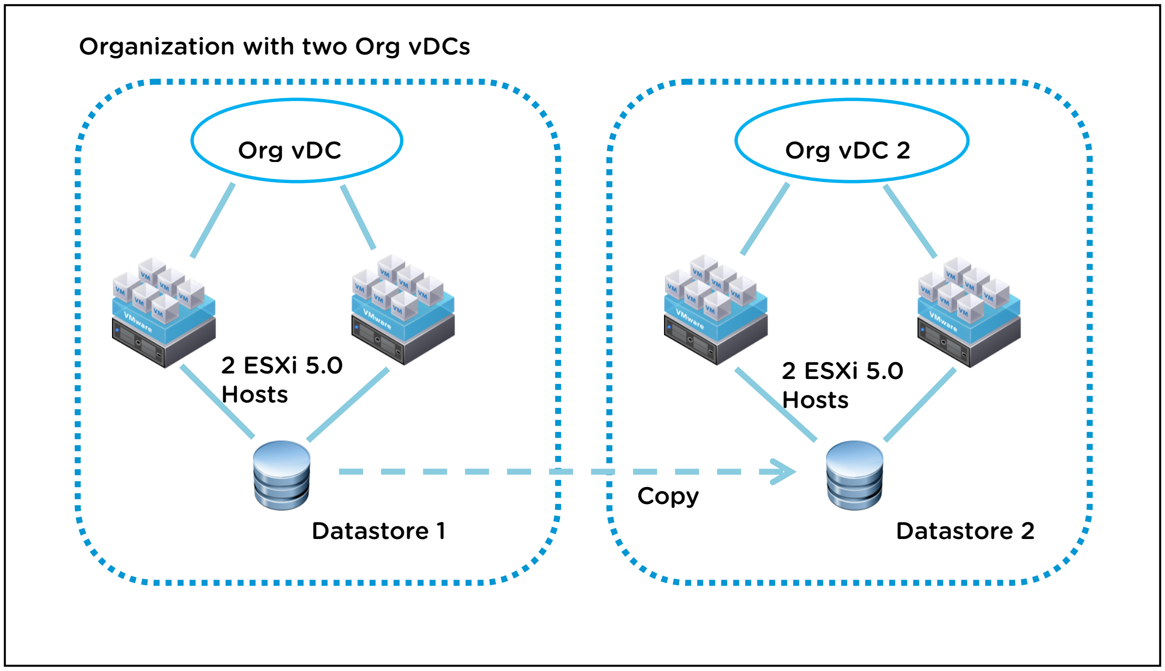

Figure 14. Two separated datastores

If ESX hosts in Org vDC 2 have access to the same datastore as shown in Figure 15, the process to upload the vApp template OVF package and then deploy it to vCenter server can be replaced by single file copy API calls to vCenter server. This design improves performance dramatically.

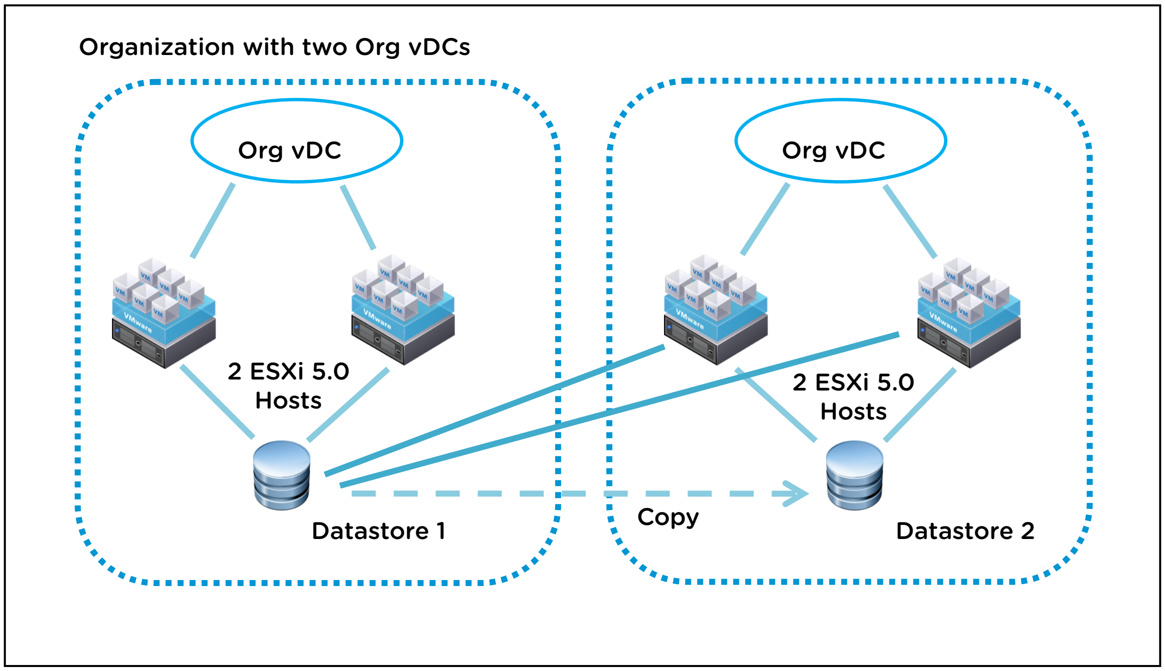

Figure 15. Shared and separated datastores among ESX hosts

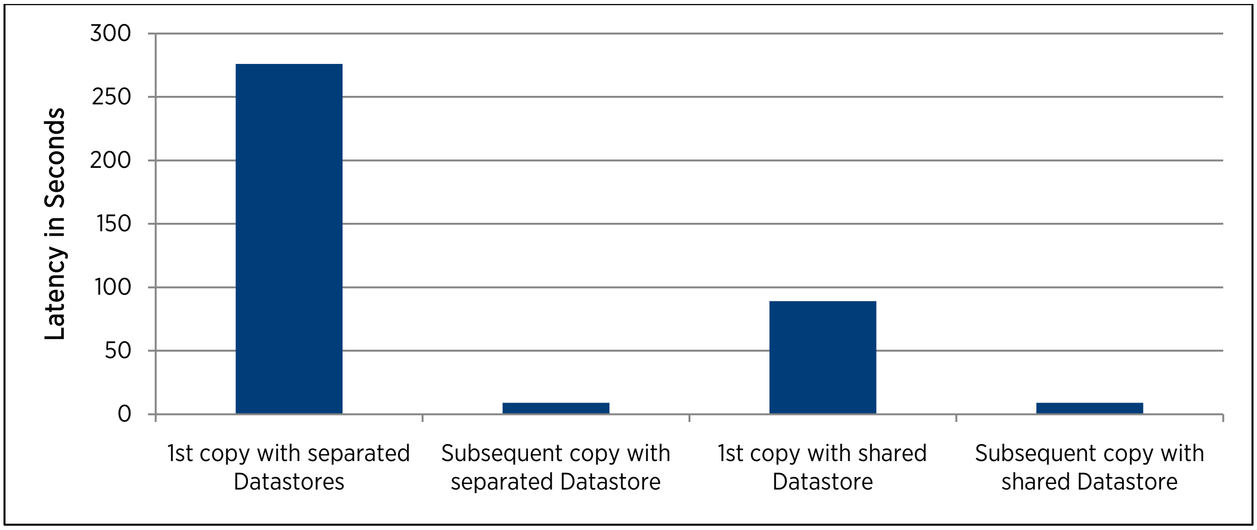

Figure 16. Latency comparison for datastore accessiblity

In the experiment, a VM with a 4GB disk is cloned from a template to a vApp across two datastores within the same vCenter. The first copy with separated datastores takes 4 minutes and 39 seconds. When the datastore is shared, the first linked clone takes 1 minute and 39 seconds—much less than separated detestores. Since the subsequent copy performs the regular linked clone operation, both operations with shared and separated datastores take the same small time.

Performance tips:

- To achieve better first linked clone performance across datastores in vCloud Director 1.5, we recommend having a shared datastore to hold the most popular vApp templates and media files and have this datastore mounted to at least one ESX host in a cluster. This way, the destination organization vDC has access to both the source and destination datastores. This removes the need to copy the OVF package files twice as discussed in "Datastore Accessibility."

For inter-vCenter clones, refer to "Clone vApps across vCenter Server Instances" in *VMware vCloud Director 1.0 Performance and Best Practices*.

I/O Workflows for Linked Clone

In an attempt to save space, linked clones use virtual disks that are sparsely allocated. These virtual disks are also called delta disks or redo logs, because they store the difference in contents between the linked clone and its parent. After the cloned virtual machines are created, powered on, and running, the delta disk grows in size. Appropriate workflows are recommended in order to handle this space over commitment over time. Sparse disks in vSphere are implemented using a 512 byte block size, and require additional metadata to maintain these blocks. The advantage of using a small block size is that it eliminates copy on write overheads and internal fragmentation. However, this design tends to add some overhead in processing I/O generated by the linked clone.

Performance tuning tips:

- For virtual machines not generating I/O-intensive workloads, linked clones offer the flexibility and agility of instant provisioning.
- For virtual machines generating I/O-intensive workloads, consider using full clones. Keep in mind that for linked clones there is additional I/O processing for delta disks. In our testing, we have seen that I/O loads in excess of 1500 IOPS will see a decreased throughput with linked clones.
- In order to mitigate this problem, we recommend that you shift the I/O load from the sparse disk to another thickly provisioned virtual disk within the virtual machine. This has the advantage of exploiting instant provisioning for the disks that contain the operating system, while taking advantage of improved performance of thickly allocated virtual disks for I/O-intensive applications

A thickly provisioned virtual disk can be created as shown in the following dialog box in vCenter server (Figure 17):

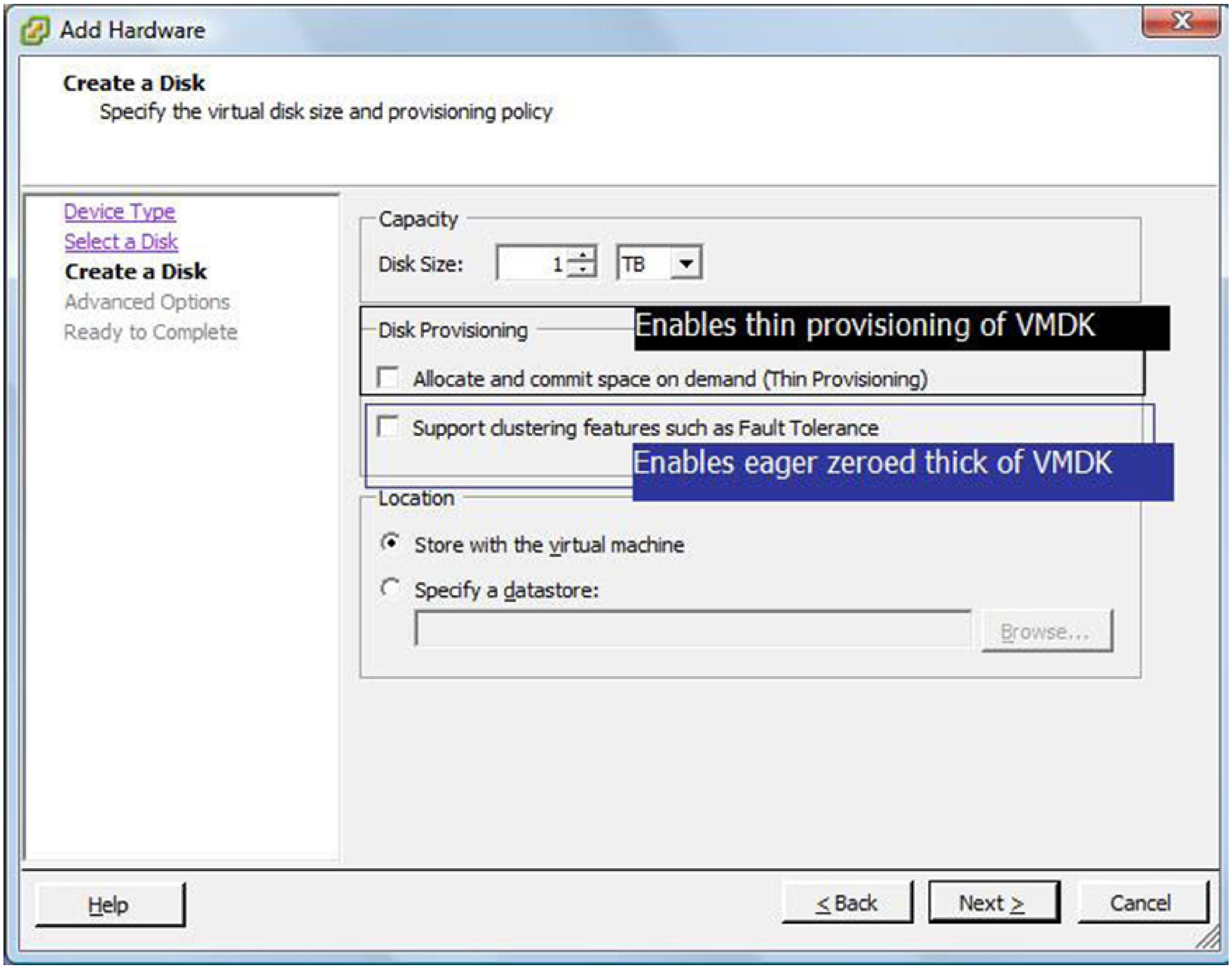

Figure 17. The Dialog window of provisioning type setup for a VM in the vSphere Client

The virtual disk file type can be found in Virtual Machine Properties, as shown in Figure 18.

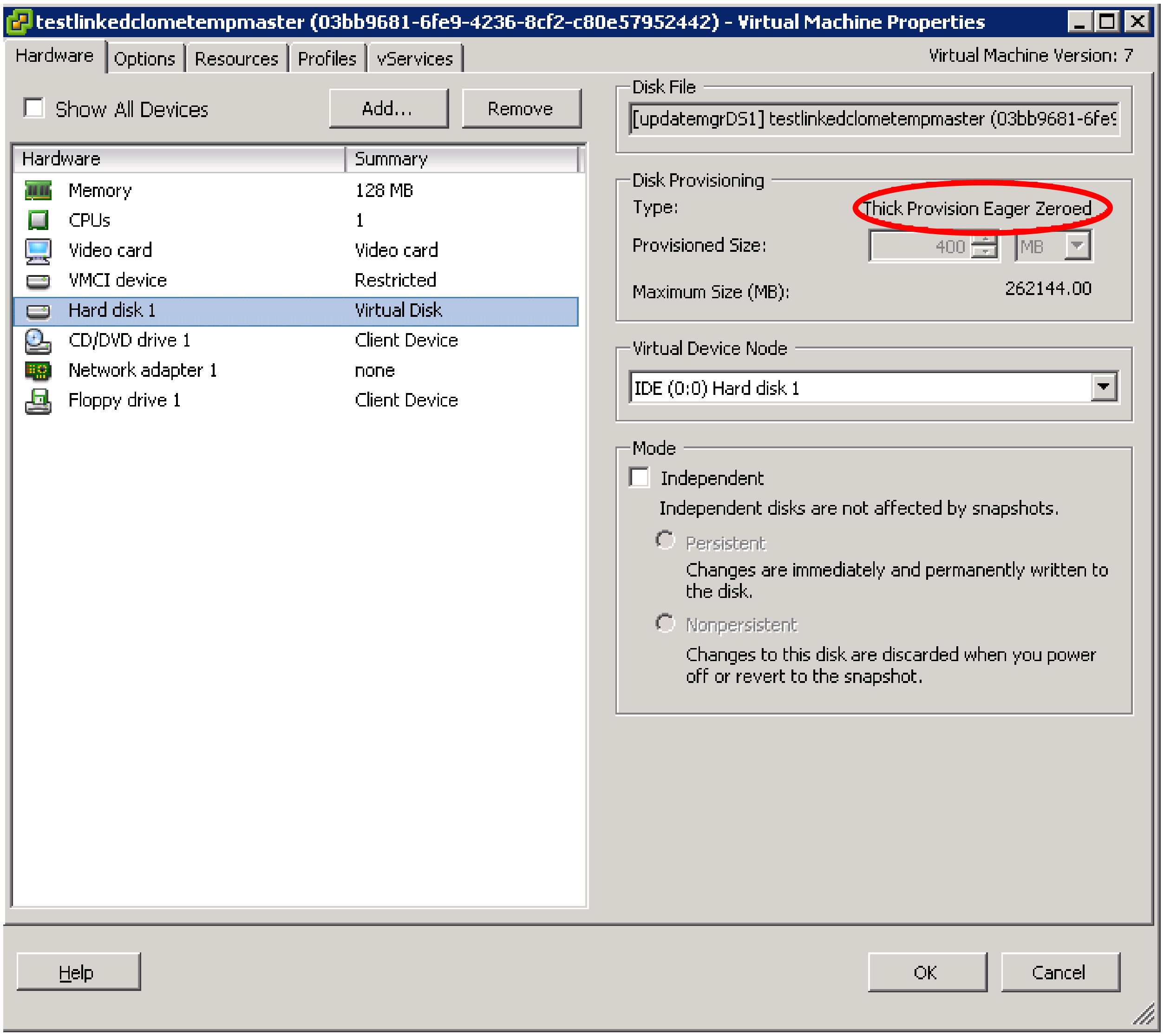

Figure 18. Dialog Window of a VM Property in the vSphere Client

Eight Host Limit

In vSphere 5.0, there's a VMFS limitation that only eight hosts may have a disk open at one time. So, while any number of virtual machines may share a common base disk, those virtual machines must reside on eight hosts or less. At the ESX level only powered-on virtual machines matter, but vCenter chooses to enforce this rule for powered-off virtual machines as well. So if using fast provisioning and a VMFS datastore, the virtual machine placed on the ninth host will fail to power on.

Performance tuning tips:

- When using fast provisioning (linked clones) and a VMFS datastore, do not exceed eight hosts in a cluster.
- For clusters larger than eight hosts that require fast provisioning (linked clones), use NFS datastores.

.

Sizing for Number of Cell Instances

vCloud Director scalability can be achieved by adding more cells to the system. Because there is only one database instance for all cells, the number of database connections can become the performance bottleneck as discussed in "Test Environment." By default, each cell is configured to have 75 database connections. The number of database connections per cell can become the bottleneck if there are not sufficient database connections to serve the requests. When vCloud Director operations become slower, increasing the database connection number per cell might improve the performance. Please check the database connection settings as described in section 3 to make sure it is configured for best performance.

For the purposes of this whitepaper, testing was performed with 12 cell instances and 10 fully loaded vCenter servers. The Oracle DB used by vCloud Director runs in a host with 12 cores and 16GB RAM. Each cell runs in a virtual machine with 2 vCPUs and 4GB RAM.

In general, we recommend the use of the following formula to determine the number of cell instances required:

number of cell instances = n+1 where n is the number of vCenter server instances

This formula is based on the considerations for VC Listener, cell failover, and cell maintenance. In "Configuration Limits," we recommended having a one-to-one mapping between the VC Listener and the vCloud Director cell. This ensures the resource consumptions for VC Listener are load balanced between cells. We also recommend having a spare cell to allow for cell failover. This allows for a level of high availability of the cell as a failure (or routine maintenance) of a vCloud Director cell will still keep the load of VC Listener balanced.

If the vCenters are lightly loaded (that is, they are managing less than 2,000 VMs), it is acceptable to have multiple vCenters managed by a single vCloud Director cell. In this case, the sizing formula can be converted to the following:

number of cell instances = n/3000 + 1 where n is the number of expected powered on VMs

For more information on the configuration limits in both VC 4.0, VC 4.1 and VC 5.0 , please refer to *VMware vCenter 4.0 Configuration Limits*[4], *VMware vCenter 4.1 Configuration Limits*[5], *VMware vCenter 4.1 Performance and Best Practice*[6], Configuration Maximums for VMware vSphere 5.0[7].

Configuration Limits

A vCloud Director installation has preconfigured limits for concurrent running tasks, various cache sizes, and other thread pools. These are configured with default values tested to work effectively within an environment of 10,000 VMs. Some of these are also user configurable, but will require a cell restart of the vCloud Director Cell.

THREAD POOL	DEFAULT SIZE (FOR EACH CELL)	USAGE/INFORMATION	ADJUSTMENT PROCEDURE
Tasks	128	Maximum number of concurrent tasks that can be executed per cell of a vCloud Director installation. This is a global task count and is not scoped per User or Organization. Different cells of the same vCloud Director installation can have different values.	org.quartz.threadPool.threadCount = N Where N is some number of concurrent tasks you want to run on a given cell.
VM Thumbnails	32	Maximum number of concurrent threads that can fetch VM thumbnail images from the vCloud Director Agent running on an ESX host. Only thumbnail images for running (powered on) VMs are collected. Thumbnails are also retrieved in batches so all VMs residing on the same datastore or host will be retrieved in batches. vCloud Director only fetches thumbnails if they are requested and once fetched, also caches them. Thumbnails are requested when a user navigates to various list pages or the dashboard that displays the VM image.	Not configurable.

Table 2. Thread pool limits

CACHE	DEFAULT SIZE (FOR EACH CELL)	USAGE/INFORMATION	ADJUSTMENT PROCEDURE
VM Thumbnail Cache	1000	Maximum number of VM thumbnails that can be cached per cell. Each cached item has a time to live (TTL) of 180 seconds.	cache.thumbnail.maxElementsInMemory = N cache.thumbnail.timeToLiveSeconds = T cache.thumbnail.timeToIdleSeconds = X
Security Context Cache	500	Holds information about the security context of logged in users. Each item has a TTL of 3600 seconds and idle time of 900 seconds.	cache.securitycontext.maxElementsInMemory = N cache.securitycontext.timeToLiveSeconds = T cache.securitycontext.timeToIdleSeconds = X
User Session Cache	500	Holds information about the user sessions for logged in users. Each item has a TTL of 3600 seconds and idle time of 900 seconds.	cache.usersessions.maxElementsInMemory = N cache.usersessions.timeToLiveSeconds = T cache.usersessions.timeToIdleSeconds = X
Inventory Cache	5000	Holds information about vCenter entities managed by vCloud Director. Each item has a LRU (least recently used) policy of 120 seconds.	inventory.cache.maxElementsInMemory = N

Table 3. Cache configuration limits in vCloud Director 1.5

To modify any of these pre-configured values:

1. Stop the cell.
2. Edit the `global.properties` file found in `<vCloud director install directory>/etc/`.
3. Add the desired configuration lines. For example, org.quartz.threadPool.threadCount = 256.
4. Save the file.
5. Start the cell.

vCenter configuration limits are very important because vCloud Director utilizes vCenter for many operations. For vCenter 4.0 configuration limits, refer to *VMware vCenter 4.0 Configuration Limits*[4]. For vCenter 4.1 configuration limits, refer to *VMware vCenter 4.1 Configuration Limits*[5], For vCenter 5.0 configuration limits, refer to Configuration Maximums for VMware vSphere 5.0[7].

Conclusion

In this paper, we discussed some of the features of the vCloud Director 1.5 release, performance characterizations including latency breakdown, latency trends, resource consumptions, sizing guidelines and hardware requirements, and performance tuning tips. Some highlights of vCloud Director performance and best practices include:

- Be aware that there is a chance to hit the snapshot chain length limit. If the current clone has become very slow compared to the prior clone, the clone may have hit the snapshot chain length limit 30. This can be resolved by virtual machine consolation.
- Since there is no chain length increase by using a template to clone a vApp, use the template for the cloning operation instead of vApp copy. For instance, if hundreds of new vApps need to be copied from an existing vApp, it is better to capture a vApp to a template. After the template is created, it is instantiated to the target organizations.
- For scenarios that need to generate many templates from a vApp, do not directly run capture vApp many times. Instead, capture this vApp to a template and copy the newly created template to catalogs in different organizations. In this way, the linked clone chain increase will be kept to a minimum.
- For cross-vCenter and cross-datastore linked clones, pre-allocating the vApp to the target datastore helps shorten the subsequent copy time.
- Before trying to remove a shadow VM, ensure all VMs cloned from this VM are removed. The shadow VM can be removed during the primary VM deletion.
- If you notice that the datastore reaches a red threshold, copy the primary vApp or template to a different datastore with sufficient capacity. This action forces a shadow copy to occur across datastores. Otherwise, the linked clone operation will fail.
- For virtual machines that are not generating I/O-intensive workloads, linked clones offer the flexibility and agility of instant provisioning.
- For virtual machines that generate I/O-intensive workloads, consider using full clones instead of linked clones. Keep in mind the additional I/O-processing for delta disks when using linked clones. In our testing, we have seen that I/O loads in excess of 1500 IOPS may see a decreased throughput with linked clones.
- In order to mitigate this problem, we recommend that you shift the I/O load from the sparse disk to another thickly provisioned virtual disk within the virtual machine. This has the advantage of exploiting instant provisioning for the disks that contain the operating system, while taking advantage of improved performance of thickly allocated virtual disks for I/O-intensive applications.
- Higher throughput to deploy vApps might be achieved with a higher concurrency level.
- Having a shared datastore to hold the most popular vApp templates and media files can get better performance than when these files are on separate datastores.
- When using fast provisioning (linked clones) and a VMFS datastore, do not exceed eight hosts in a cluster.
- For clusters larger than eight hosts that require fast provisioning (linked clones), use NFS datastores.

References

1. Open Virtualization Format Specification
http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

2. Open Virtualization Format White Paper
http://www.dmtf.org/standards/published_documents/DSP2017_1.0.0.pdf

3. WANem Simulator Web site
http://wanem.sourceforge.net/

4. VMware vCenter 4.0 Configuration Limits
http://www.vmware.com/pdf/vsphere4/r40/vsp_40_config_max.pdf

5. VMware vCenter 4.1 Configuration Limits
http://www.vmware.com/pdf/vsphere4/r41/vsp_41_config_max.pdf

6. VMware vCenter 4.1 Performance and Best Practice
http://www.vmware.com/files/pdf/techpaper/vsp_41_perf_VC_Best_Practices.pdf

7. Configuration Maximums for VMware vSphere 5.0

http://www.vmware.com/pdf/vsphere5/r50/vsphere-50-configuration-maximums.pdf

8. vCloud API Programming Guide
https://www.vmware.com/pdf/vcd_15_api_guide.pdf

9. vCloud Director 1.5 Release Note
https://www.vmware.com/support/vcd/doc/rel_notes_vcloud_director_15.html

10. Changing vCloud Director Java heap size to prevent java.lang.OutOfMemoryError messages
http://kb.vmware.com/kb/1026355

11. Cloud Director Installation and Configuration Guide
https://www.vmware.com/pdf/vcd_15_install.pdf

12. vCloud Director Administrator's Guide 1.5

http://www.vmware.com/pdf/vcd_15_admin_guide.pdf

13. vCloud Director User's Guide 1.5

http://www.vmware.com/pdf/vcd_15_users_guide.pdf

About the Authors

Joanna Guan is a Senior Member of Technical Staff in VMware where she works on several performance projects including VMware vCloud Director, VMware vCenter CapacityIQ™, and VMware vCenter Change Insight. Prior to VMware, Joanna was a senior software developer at Hewlett Packard and Agilent Technologies.

Xuwen Yu is a Member of Technical Staff in VMware where he works on several projects including VMware vCloud Director, VMware vCenter™ Update Manager and ESX host simulator. Xuwen received his Ph.D. degree in Computer Science and Engineering from the University of Notre Dame in 2009.

Ritesh Tijoriwala is a Staff Member of Technical Staff in VMware. Ritesh is a lead engineer for the Performance and Scalability for vCloud Director and a key engineer for Interoperability with vCenter, and Failover and Inventory related functionality for VMware vCloud Director. Ritesh has received his M.S. in Computer Science from California State University, Fresno.

John Liang is a Senior Manager at VMware. Since 2007, John has been working as a technical lead for performance projects on VMware products such as vCloud Director, Update Manager, VMware vCenter Site Recovery Manager™ (SRM), VMware vCenter Converter, and VMware vCenter Lab Manager™. Prior to VMware, John was a Principal Software Engineer at Openwave Systems Inc., where he specialized in large-scale directory development and performance improvements. John received a M.S. degree in Computer Science from Stony Brook University.

Acknowledgements

We want to thank Rajit Kambo and Jennifer Anderson for their valuable guidance and advice. Their willingness to motivate us contributed tremendously to this white paper. We would also like to thank Mahdi Ben Hamida, Sunil Satnur, Nikhil Jagtiani, Paul Herrera, Colin Zhang, Tom Stephens, Catherine Fan, Mornay Van Der Walt, and Clair Roberts. Without their help, this white paper would not have been possible. Finally, we would like to thank Julie Brodeur for help editing and improving this white paper.

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