VMware vCloud Director® 5.1 Performance and Best Practices

Performance Study

vmware®

Table of Contents

Introduction

VMware vCloud Director® 5.1 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 5.1 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.

This white paper addresses three areas regarding vCloud Director performance:

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

Figure 1 shows the deployment architecture for vCloud Director. A user accesses vCloud Director through a Web browser or REST API. Multiple vCloud Director Server instances can be deployed with a shared database, and 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.
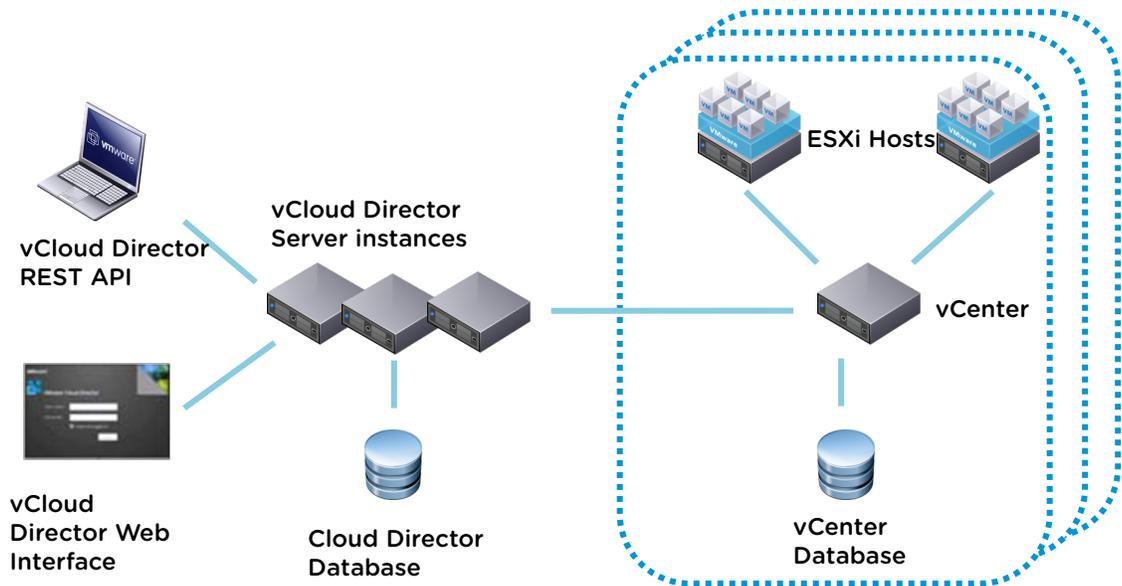
Figure 1. VMware vCloud Director high level architecture

Next we introduce the definitions for some key concepts in vCloud Director 5.1. 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 one or more vCenter Server resource pools 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 vApps can be stored, deployed, and operated. vDCs can also provide storage for virtual media, such as floppy disks and CD-ROMs.

  A single organization can have multiple organization vDCs.

  A system administrator specifies how resources from a provider vDC are distributed to the organization 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.

Throughput Improvements for Frequent Operations

Significant performance improvements have been made in vCloud Director 5.1 compared to previous releases. In this section, we present the test results and performance improvements for typical vCloud Director operations.

Test Environment

We used the following test-bed setup. Actual results may vary 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, 16GB RAM

vCenter: 64-bit Windows Server 2003, 4 vCPUs, 16GB RAM

vCenter Database: 64-bit Windows Server 2003, 4 vCPUs, 16GB RAM

All of these components are configured as virtual machines and are hosted on Dell PowerEdge R610 machines with 8 Intel Xeon CPUs@2.40GHz, and 48GB RAM.

Software Configuration

vCenter: vCenter Server 5.0

vCenter Database: Oracle Database 11g

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 database configuration, refer to the *vCloud Director Installation and Upgrade Guide* [7] or KB 2034540, "Installing and configuring a vCloud Director 5.1 database [10]."

Methodology

In our experiment, the vCD operations listed as follows are performed by a group of users against a vCloud Director cell simultaneously. The number of users varies from 8, 16, 32, and 64 to 128, and the operations performed include:

- Clone vApp

- Capture vApp as a template in a catalog

- Instantiate vApp from a template

- Delete vApp

- Delete vApp template in a catalog

- Edit vApp

- Create users

- Deploy vApp with or without a fence

- Undeploy vApp with or without a fence

Note that clone vApp, capture vApp, and instantiate vApp all involve virtual machine clone operations. The vApp and vApp template we tested include a single virtual machine with the same size (400MB).

Results

Figure 2 shows the throughput results for varying users in both vCD 5.1 and the previous release vCD 1.5. Compared with the previous release (vCD 1.5), operation throughput has been significantly improved. Also, when the concurrent user number increases from 8 to 128, we observed that the throughput keeps growing in a stable manner.
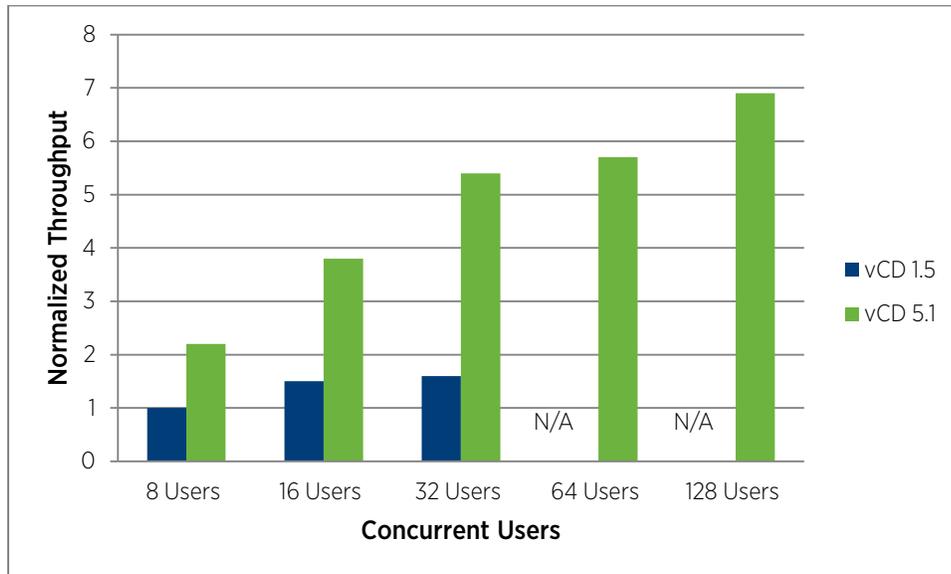
Figure 2. Throughput improvement for frequent operations

In order to make the figure more readable, we have normalized it to the throughput result of eight concurrent users in vCD 1.5; this result is used as one unit.

All of these tests are performed with a single cell and a single vCenter Server. More throughput can be achieved by adding more resources (cell, vCenter Server, and so on). Related information can be found in this paper in the section "Sizing for Number of Cell Instances."

In our experiments, we also noticed that rebuilding cell database indexes after intensive object create/delete operations helps to improve vCD performance. For more details on rebuilding cell database indexes, please refer to "Appendix."

Inventory Sync

In this section, we investigate two types of inventory sync:

- **Restart-cell-sync**
  The vCloud Director Server may be shut down and restarted. When it is restarted, it retrieves all the current vCenter Server inventory information. If there is anything different from the current state in the vCloud Director database, this change will be stored in the database. We call this process restart-cell-sync.

- **Reconnect-vCenter-sync**
  The vCenter Server may also be shut down and restarted. In this case, vCloud Director Server tries to reconnect to vCenter Server and re-sync the inventory information. We call this process reconnect-vCenter-sync.

Test Environment

The system used is the same as that described in the previous section "Throughput Improvements for Frequent Operations."

Results

Inventory Sync Time

Because vCloud Director Server has an inventory cache which stores the inventory information in memory, it is more efficient to re-sync inventory when vCenter Server is reconnected instead of when vCenter Server is restarted.
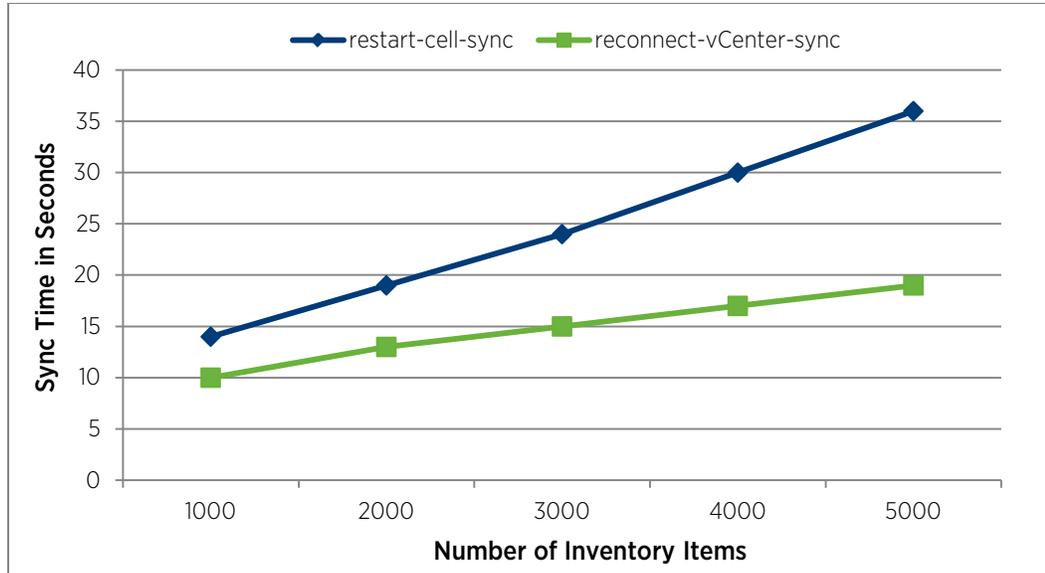

Figure 3. Inventory sync time

Figure 3 shows that both restart-vCenter-sync and restart-cell-sync latency proportionally grow as the number of inventory items in the system increases. For reconnect-vCenter-sync, because the in-memory inventory cache could potentially cache all or most of the inventory objects, the time to fetch these objects from the cell database is saved. This is why reconnect-vCenter-sync gives better performance than restart-cell syncs.

Overall, it is recommended to perform vCloud Director operations after inventory sync finishes if the cell or vCenter restarts. This ensures operations can be executed smoothly. The sync progress can be tracked in the vCD user interface as shown in Figure 4 (System→Manager & Monitor→vCenters→Status).
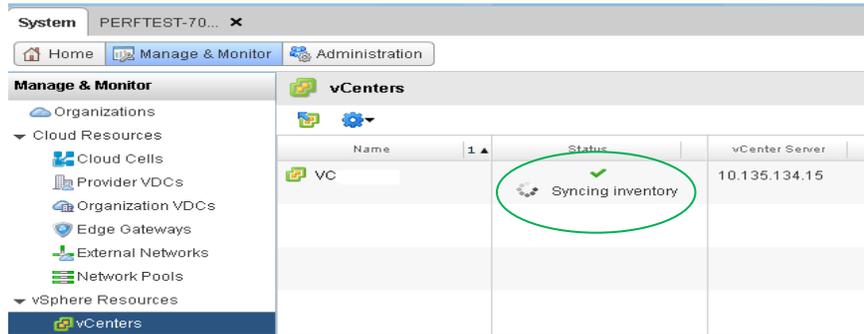
Figure 4. Tracking sync progress in the vCloud Director user interface

Tuning the Inventory Cache Size

An in-memory inventory cache is implemented in vCloud Director. The cache saves the cost to fetch inventory information from the database and the cost to de-serialize the database record. Figure 5 demonstrates the effectiveness of the inventory cache for reconnect-vCenter-sync usage. When the cache size is set to 10,000 inventory items, the cache hit ratio is much higher. The latency to sync 8000 inventory items is also much faster when the cache hit ratio is higher.
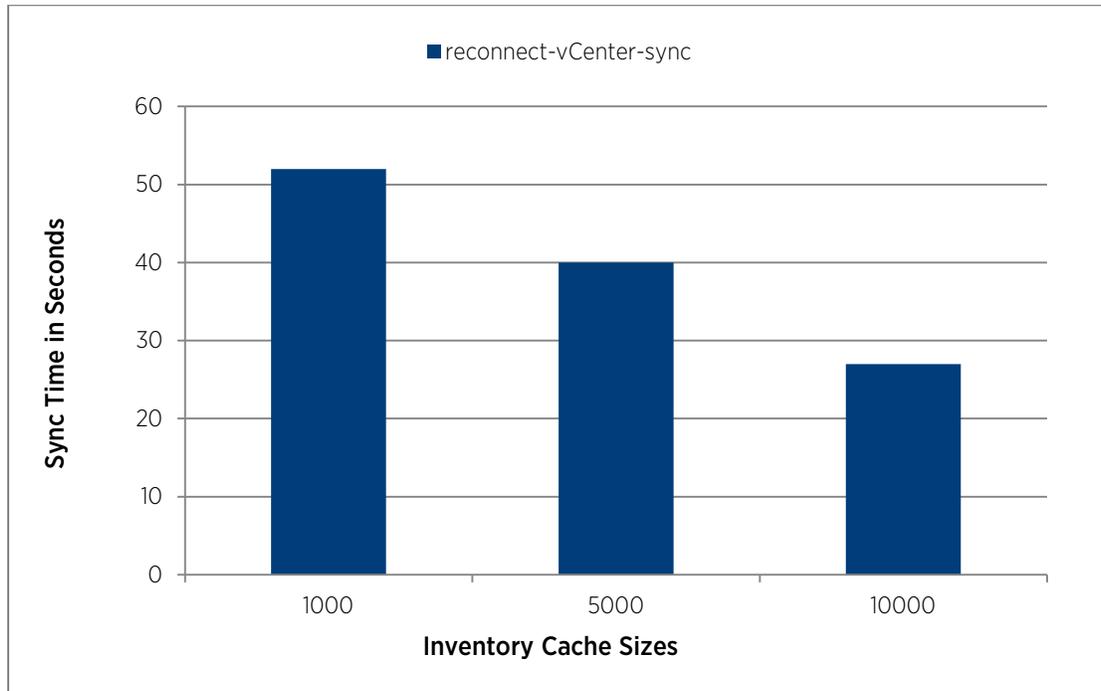
Figure 5. Sync time for varying inventory Cache sizes with 8000 inventory items

By default, each vCloud Director cell is configured for 5000 inventory items, (total inventory cache entries including hosts, networks, folders, resource pool, and so on). We estimate this sizing is optimal for 2000 virtual machines. Therefore, proper tuning of this inventory size will help boost performance. We recommend the following formula to help determine what number to use for the cache size:

Inventory Cache Size = 2.5 × (Total Number of VMs in vCloud Director)

It is assumed here that most virtual machines in vCenters that are managed by vCloud Director are the ones created by vCloud Director. If that is not the case, substitute the "Total number of VMs in vCloud Director" with "Total number of VMs in vCenters."

Best Practices for Inventory Sync

Properly increasing the inventory cache size will decrease the reconnect-vCenter-sync time.

Elastic Virtual Datacenter

Elasticity is an important aspect of cloud computing—physical resources such as CPU, memory, and storage need to grow as consumers require them, and they need to shrink so that the resources can be made immediately available elsewhere in the cloud environment. vCloud Director adds elasticity to the datacenter through a feature called elastic virtual datacenter (elastic vDC). Elastic vDC allows for an efficient utilization of vCenter resources. A provider vDC can have multiple resource pools and administrators can deploy or remove the resource pools on the fly as needed. These resources will be available to all of the organization vDCs associated with the provider vDC. Elasticity is only supported by the resource pool models Pay-As-You-Go and Allocation; Reservation is not supported.

To enable elasticity in a virtual datacenter, add multiple resource pools to a provider vDC. In vCloud Director, choose System → Manage & Monitor → Resource Pools.

This section presents experimental results from a number of case studies designed to demonstrate the performance of elastic vDC. Note that these latency and throughput numbers are only for reference. The actual numbers could vary with different deployments.

Test Environment

For the results in this section, we used the following test-bed setup.

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 2008, 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 two Dell PowerEdge R610 boxes with 8 Intel Xeon CPUs@2.40GHz, and 48GB RAM.

Software Configuration

vCenter: vCenter Server 5.1

vCenter Database: Microsoft SQL Server 2008

vCloud Director : vCloud Director Version 5.1

vCloud Director Database: Microsoft SQL Server 2008

Number of clusters: 1-16

Number of hosts in each cluster: 4; each host connects to an NFS data store

Methodology

We defined two workloads to test the feature of elasticity:

- Instantiation load: sequentially instantiating vApps
- Deployment load: sequentially deploying vApps

The instantiation and deployment operations go through the vCloud Director placement engine component, which determines the resources that the virtual machines should be associated with. The resources include resource pools, datastores, networks, and network pools. The placement engine is invoked during virtual machine creation (part of instantiation of the vApp), virtual machine deployment (which happens when a user powers on the vApp containing the virtual machine), and virtual machine update. During virtual machine creation, the placement engine identifies appropriate resources for the virtual machine based on its resource and capacity requirements. During virtual machine deployment, the placement engine validates that the currently assigned resources still have sufficient capacity to power on the virtual machine; otherwise, it attempts to migrate the virtual machine to a new set of resources more compatible with the requirements of the virtual machine. The placement engine can handle multiple simultaneous requests for creating and deploying virtual machines. (For more information on the placement engine component, refer to "About the vApp Placement Engine" in the *vCloud Director Administrator's Guide* [8]).

Results

Placement and Deployment Performance Regarding Various Resource Pool Numbers

In order to understand what the elastic vDC performance characteristics are at various resource pool sizes, we compared the end-to-end latency of the instantiation load and deployment load with several given numbers of resource pools.

When the provider virtual data center has different numbers of resource pools, we observed there is no significant latency when instantiating vApps and deploying vApps. In Figure 6 and Figure 7, we show the latencies for instantiating vApps and deploying vApps. The X-axis is the number of resource pools and Y-axis is the latency for doing the respective vApp operation. As can be seen, the latencies for instantiating vApps are almost the same when the provider vDC includes 1, 4, 8, and 16 resource pools. The difference between the latency results can be attributed to the noise in our measurements. We also see approximately the same result for the operations. So we can infer that there is no resource pool size impact on vApp instantiation and deployment performance.
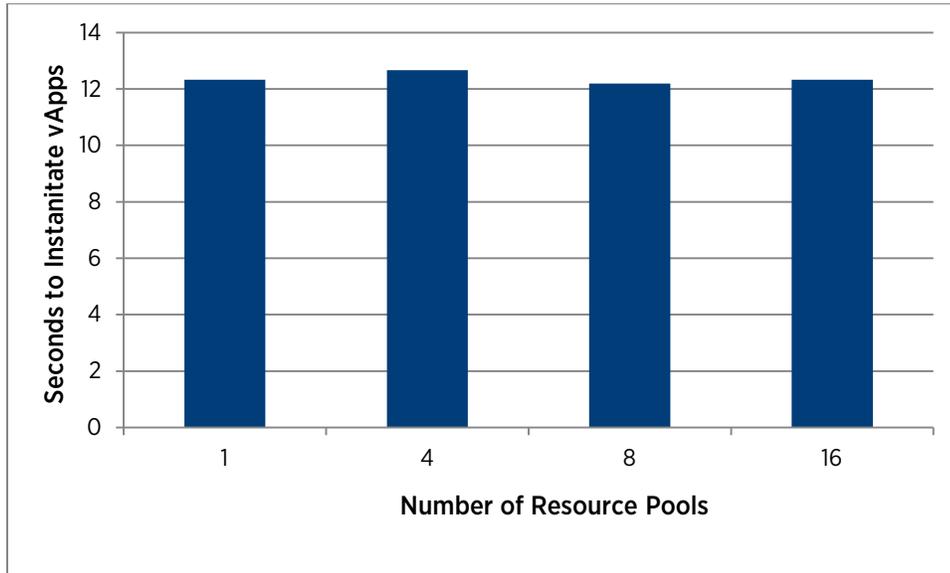
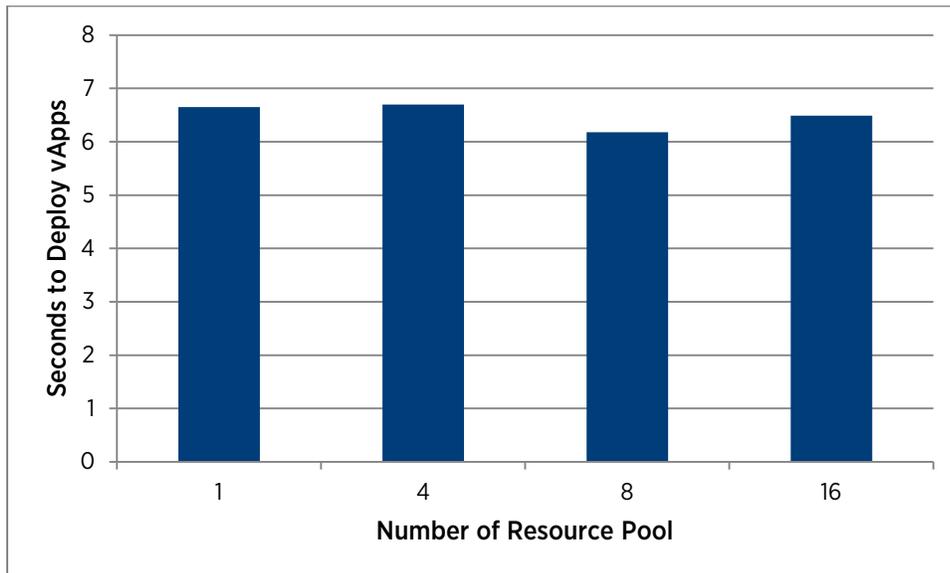Figure 6. Average time it takes to instatiate vApps at various resource pool sizes

Figure 7. Average time it takes to deploy vApps at various resource pool sizes

Deployment Performance Regarding Various Concurrent Users

The performance of operation concurrency is an importance aspect in a cloud environment—there may be multiple vCloud Director users operating at the same time in a single cloud. Here, we study an experiment which varied the number of vCloud Director users connected at the same time. We measured the end-to-end throughput of 1, 8, 16, 32, 64, and 128 concurrently connected users while they deployed vApps. The provider vDC includes 16 resource pools. Because deployment operations go through the vCloud Director placement engine component, when the number of concurrent clients increases, the placement engine spends more time determining the resources with which the virtual machines should be associated.

In Figure 8, we show the throughput for deploying vApps with a given number of concurrent users. The X-axis is the number of concurrent users and the Y-axis shows the throughput at operations per minute. For vApp deployment in Figure 9, the throughput is increased as the number of concurrent users increases until 128 concurrent users is reached, but then the trend of throughput growth gets slow during a high concurrent user number.
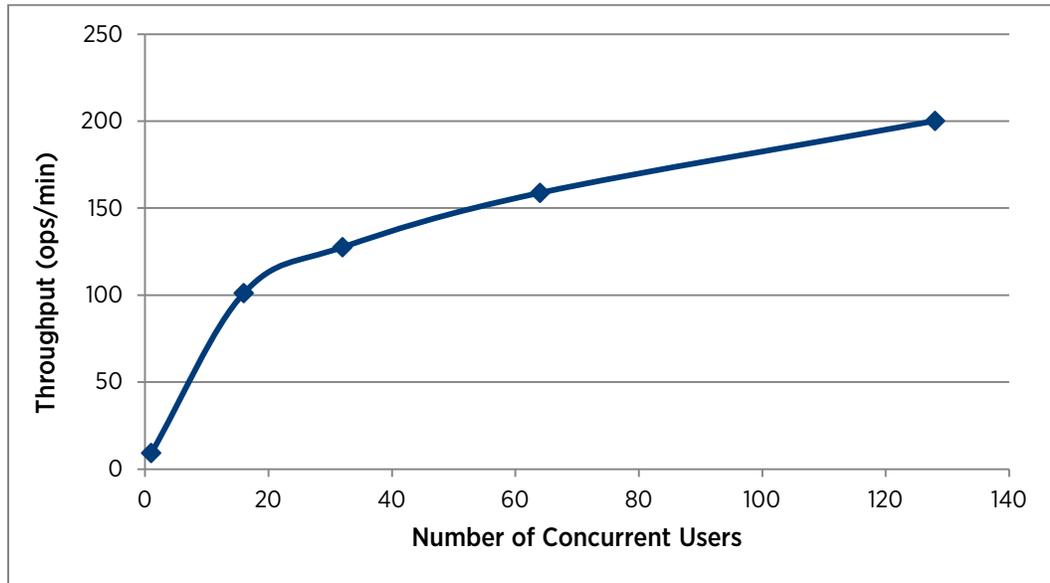

Figure 8. Throughput (operations/minute) of with users each deploying a vApp at the same time

Note that all tests are performed with a single cell and a single vCenter Server. More throughput can be achieved by adding more cells and more vCenter Server hosts. Related information can be found in the section "Sizing for Number of Cell Instances."

Placement and Deployment Performance Regarding Various VM Sizes in Each vApp

In this test, we measure the elastic vDC performance characteristics at various virtual machine sizes; this experiment compares the end-to-end latency of the instantiation load and deployment load with several given numbers of virtual machines in each vApp. The number of virtual machines in each vApp is 1, 8, 16, and 24. The provider vDC includes 16 resource pools.

When the test vApp had different numbers of virtual machines, we observed the latency increased when the number of virtual machines increased. In Figure 9 and Figure 10, we show the latencies for instantiating vApps and deploying vApps. The X-axis is th0e number of virtual machines and the Y-axis is the latency for doing the respective vApp operation. From these figures, we can see that the latency increases in a nearly linear pattern.
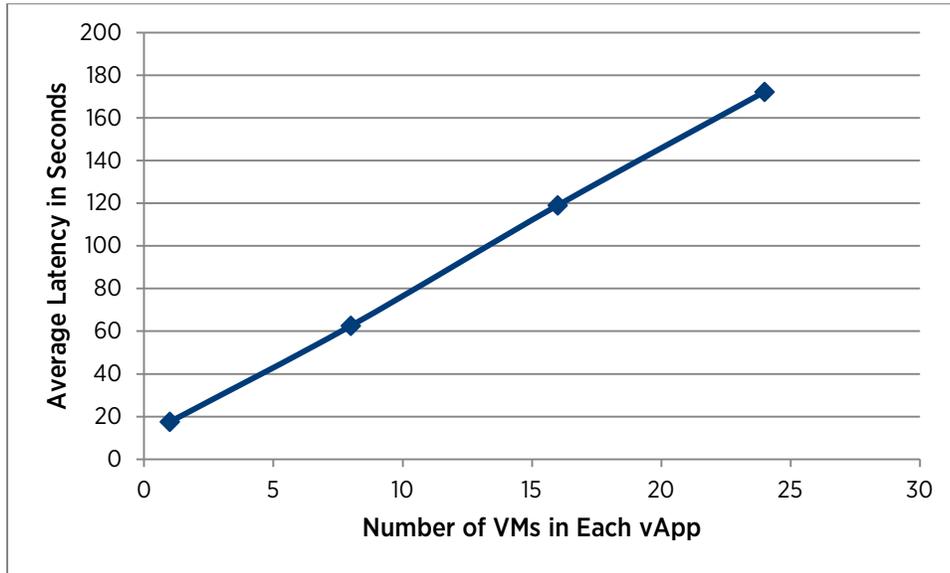
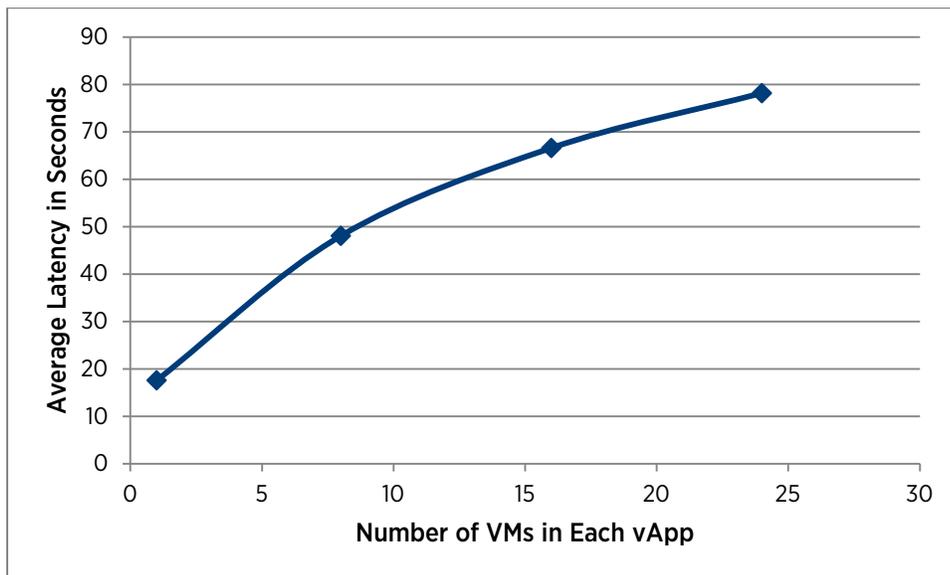Figure 9. Instantiation latency for a vApp with multiple virtual machines

Figure 10. Deployment average latency in seconds for a vApp with multiple virtual machines

Best Practices for Elastic vCD

- Remember that CPU and memory resources are not reserved for virtual machines that are powered off. Therefore, you might be able to create a large number of virtual machines but only power on a subset of them due to insufficient capacity in the provider vDC or organization vDC. Only storage resources are reserved for virtual machines during creation time. System administrators may want to consider this as part of their capacity planning. Add new resource pools to the provider vDC to resolve the problem of insufficient capacity at power on.
- Always keep sufficient CPU and memory headroom capacity available on each cluster that is part of the

provider vDC. When clusters are running very low on capacity, the system will run into deployment failures due to capacity fragmentation, memory overhead requirements, and so on. We recommend to keep at least 5% headroom capacity always available on the clusters.

- When new capacity is added to the provider vDC, vCloud Director does not perform automatic rebalancing of existing workloads to utilize the new capacity. Any future virtual machine creations and deployments will attempt to utilize the new capacity, but existing running virtual machines will not be migrated automatically. We recommend that administrators utilize the vCloud Director migrate functionality to migrate some of the workload from existing clusters to the new clusters to ensure a more balanced utilization of the overall capacity and to ensure that the headroom requirement on clusters is always satisfied. For more information on workload migration, refer to "Migrate Virtual Machines Between Resource Pools on a Provider vDC" in the *vCloud Director Administrator's Guide* [8].

Independent Disk

Independent disks are stand-alone virtual disks that can be created in organization vDCs. Administrators and users who have adequate rights can create, remove, and update independent disks, and attach or detach them to or from virtual machines. Refer to the *vCloud API Programming Guide 5* [5] for more information.

Test Environment

For characterizing vCloud Director Independent Disks operations, we use one vCloud Director cell and one vCenter server. Each server has a standalone database. The test bed settings are as follows:

- vCloud Director cell: 64-bit Red Hat Enterprise Linux 5, 4 vCPUs, 8GB RAM
- vCloud Director database: Microsoft SQL Server 2008 R2, 64-bit Windows Server 2008, 4 vCPUs, 8GB RAM
- vCenter Server: 64-bit Windows Server 2008 Enterprise Edition, 4 vCPUs, 8GB RAM
- vCenter Server database: Microsoft SQL Server 2008 R2, 64-bit Windows Server 2008 Enterprise Edition, 4 vCPUs, 8GB RAM

Methodology

We measure the throughput of 1, 8, 16, 32, 64, and 128 concurrently connected users while each user iteratively does one of the following single independent disk operations, one operation per experiment.

- Creating an independent disk in an organization vDC
- Attaching an independent disk to a virtual machine in the same datastore as the disk
- Attaching an independent disk to a virtual machine in a different datastore as the disk
- Detaching an independent disk from a virtual machine

Results

Creating an Independent Disk

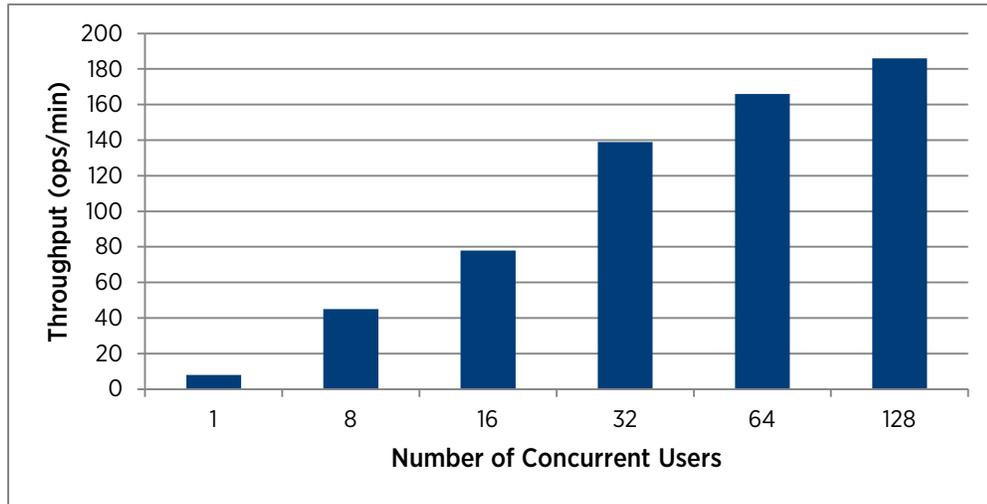Figure 11 shows the throughput of creating an independent disk in an organization vDC.

Figure 11. Creating an independent disk

As the number of concurrent users rises, the throughput of creating independent disks grows in a path that is nearly linear, and this indicates good performance.

Attaching an Independent Disk to a Virtual Machine

For this experiment, independent disks and virtual machines are created before the attaching operation. One independent disk can only be attached to one virtual machine. When attaching a disk to the virtual machine, the virtual machine is reconfigured to add the independent disk.

When attaching an independent disk to a virtual machine located in a different datastore, the disk will be relocated to the datastore where the virtual machine is located first, then the virtual machine is reconfigured to add the independent disk.

Figure 12 shows the throughput of attaching an independent disk to a virtual machine operation.
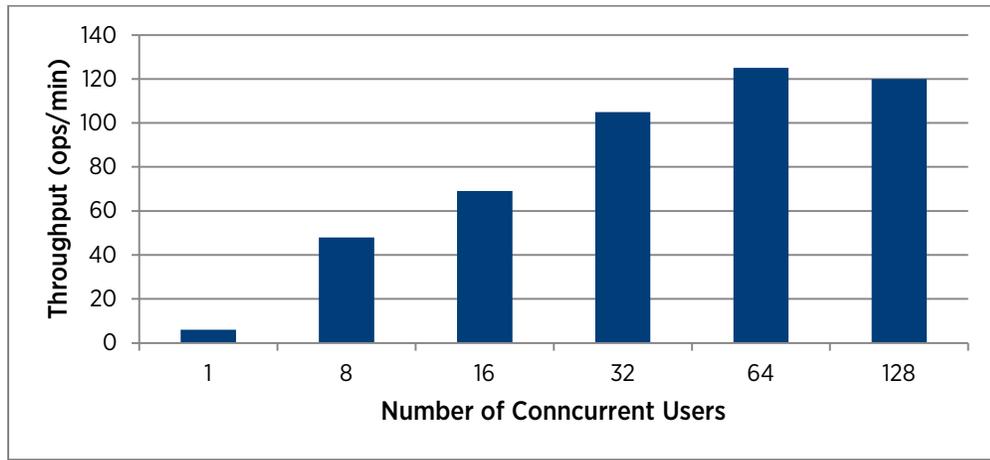

Figure 12. Attaching an indepedent disk to a virtual machine

Detaching an Independent Disk from a Virtual Machine

When detaching an independent disk from a virtual machine, the virtual machine will be reconfigured to remove the virtual disk to make the disk independent of the virtual machine, and the data on the independent disk is preserved.

Figure 13 shows the throughput of detaching an independent disk from a virtual machine operation.
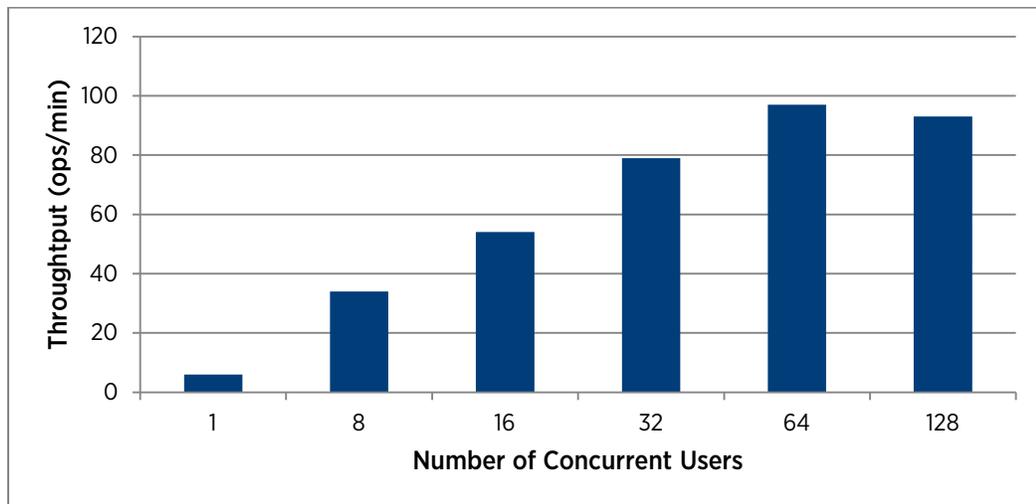

Figure 13. Detaching a disk from a virtual machine

From the results, we observe that independent disk operations achieve optimal throughput with 64 concurrent users and remains stable up to 128 concurrent users.

Best Practices

If you are using the REST API to connect to vCloud Director, we recommend that you provide locality parameters to provide hints that can help the placement engine optimize placing the virtual machine or the independent disk.

- If a vApp exists before creating a disk, when creating a disk, you can specify using the REST API the *href* of the vApp as the Locality property in the DiskCreateParams parameter.

- If the disk exists before creating a vApp, when instantiating a vApp from a vApp template, you can specify the disk reference as in the LocalityParams (in the REST API) to make the vApp placed close to the disk.

For more details about attaching and detaching independent disks, see "Attach or Detach an Independent Disk" in the *vCloud API Programming Guide* [5].

vCloud Director Networking

There are three categories of vCloud Director networks: external networks, organization vDC networks, and vApp networks. An external network provides virtual machines with network connectivity to the outside world. Organization vDC networks can be used by any vApp in the organization vDC. Organization vDC networks can be configured to provide direct or routed connections to external networks, or can be isolated from external networks and other organization vDC networks. A vApp network is a logical network that controls how the virtual machines in a vApp connect to each other and to organization vDC networks.

A vCloud Director Edge gateway, in essential, is a vShield Edge virtual appliance, acting as a virtual router for organization vDC networks. You can configure it to provide network services such as DHCP, firewall, NAT, static routing, VPN, and load balancing. Please refer to *vCloud API Programming Guide 5.1* [5] for more information.

Test Environment

This test uses a similar configuration that is described in "Independent Disk test bed settings" from the previous section.

Methodology

In vCloud Director 5.1, we measured the throughput of 1, 4, 8, and 16 concurrent users for the following vCloud Director networking operations:

- Creating an Edge gateway

- Creating a direct organization vDC network

- Creating a routed organization vDC network

- Creating an Isolated Organization vDC Network with DHCP service enabled

- Instantiating and deploying a vApp with routed vApp network, which connects to a routed organization vDC network

Results

Creating an Edge Gateway

When creating a vCloud Director Edge gateway, a vShield Edge virtual appliance is deployed and the supported advanced network services are configured. The process of creating an Edge gateway can be time consuming because of the multiple operations required.

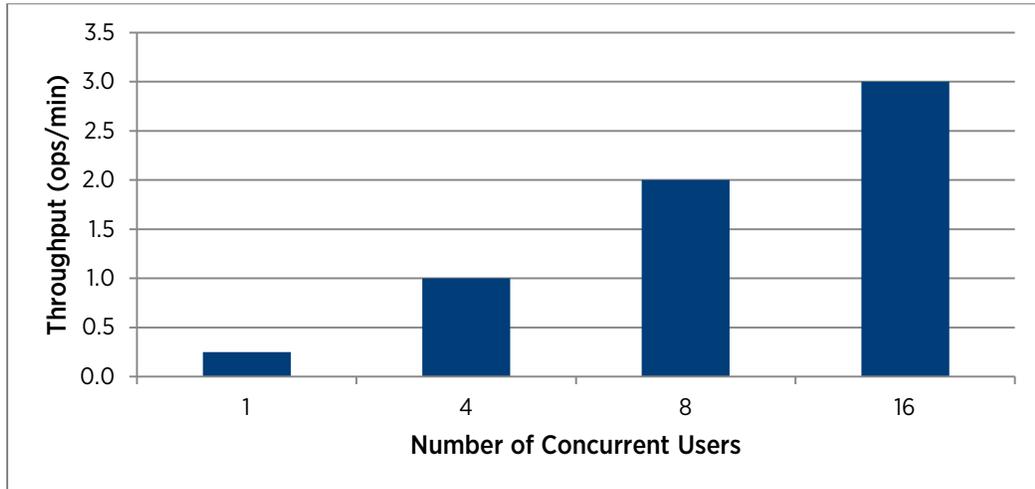Figure 14 shows the throughput of creating an Edge gateway operation.

Figure 14. Creating an Edge Gateway

The results show a linear progression—as more users are added to the workload, the throughput also increases. This is the growth we expect and it shows good performance.

Creating Organization vDC Networks

There are three types of networks that an organization vDC can have—direct, routed, and isolated.

- **Direct Organization vDC Network**
  A direct organization vDC network is directly connected to an external network. If a virtual machine is connected to a direct organization vDC network, the NIC card of the virtual machine will be attached to the portgroup of the external network.
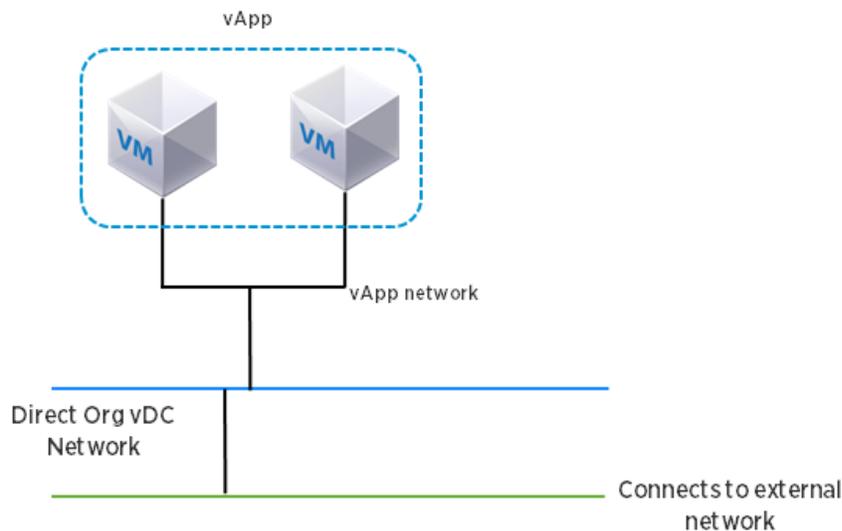

Figure 15. Direct Organization vDC Network

- **Routed Organization vDC Network**
  A routed organization vDC network uses an Edge gateway as a virtual router in the organization vDC network. It provides NAT, Firewall, DHCP, IPSec VPN, and static routing services to the vApps connected to it. The Edge gateway needs to be deployed and configured before creating a routed organization vDC network.

If a virtual machine is connected to a routed organization vDC network, a NIC card of the virtual machine will be attached to the port group of the organization vDC network. The NIC card of the vShield Edge appliance will also be attached to the port group of the organization vDC network; that is, both the vShield Edge appliance NIC card and the NIC card of the virtual machine are attached to the same port group that belongs to the organization vDC network. Another NIC of the vShield Edge virtual appliance will be connected to the port group of the external network.
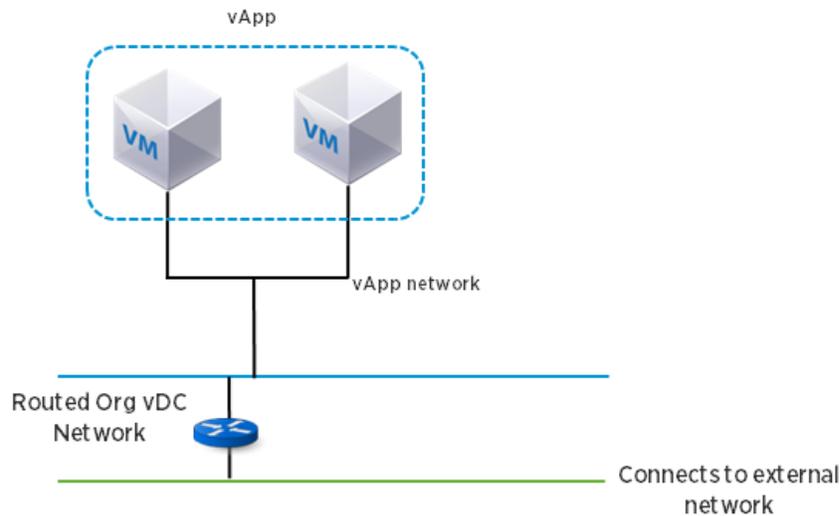

Figure 16. Routed Organization vDC Network

- **Isolated Organization vDC Network with DHCP Service Enabled**
  An isolated organization vDC network provides an internal network. Virtual machines connected to an isolated organization vDC network cannot communicate with any other network or the external network.
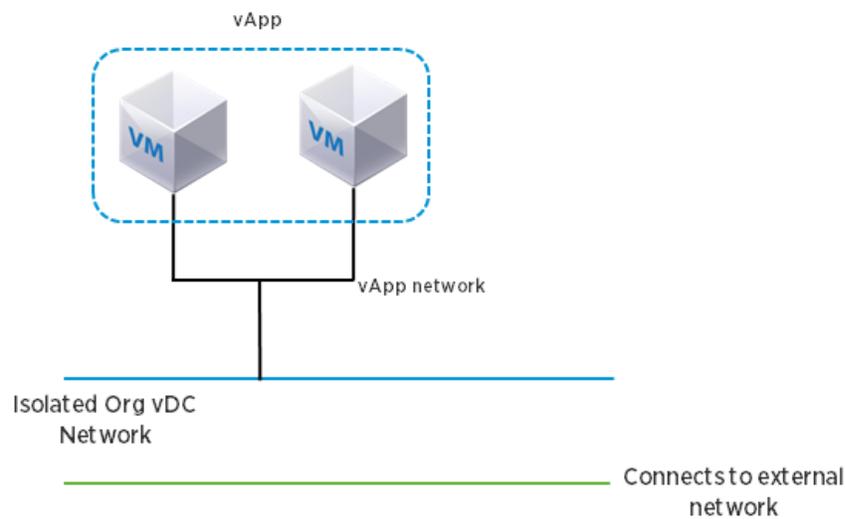

Figure 17. Isolated Organization vDC Network

When the DHCP service is enabled in the isolated organization vDC network, a vShield Edge virtual appliance is deployed and configured to provide the service.

Figure 18 shows the throughput of creating these three types of organization vDC network operations.
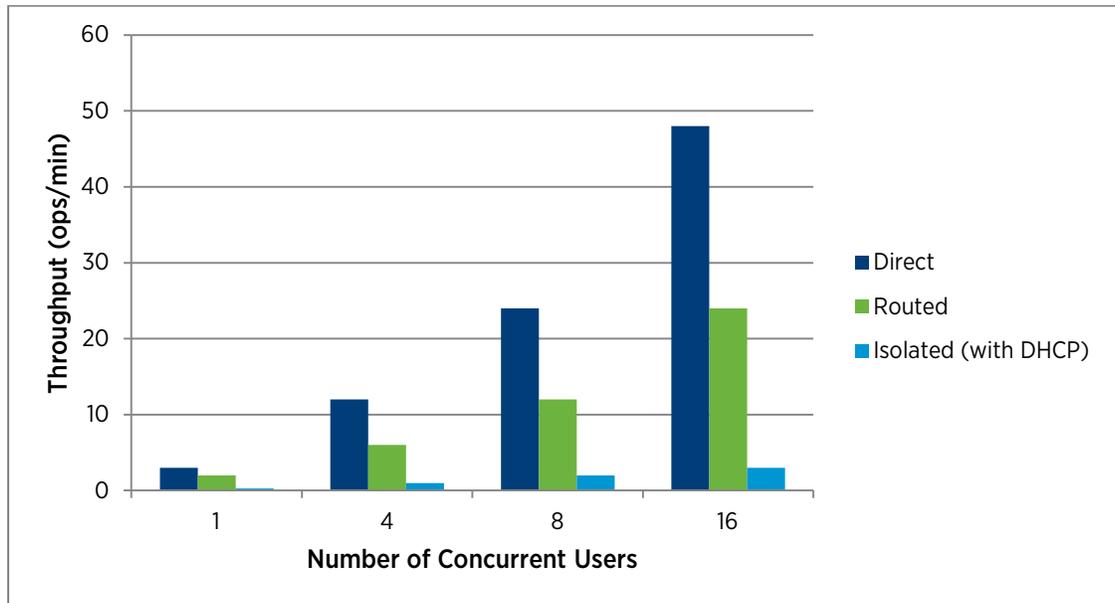
Figure 18. Creating an Organization vDC Network

Deploying a vApp with a Routed vApp Network

A routed vApp network provides advanced networking features like DHCP, Firewall, NAT, or static routing to the virtual machines in a vApp. The vShield Edge virtual appliance is deployed and the advanced network services are configured on a per-vApp network basis.

Figure 6 shows a vApp, vApp A, with a routed vApp network and connected to a routed organization vDC network.

Figure 19. vApp A with a routed vApp network and connected to a routed organization vDC network

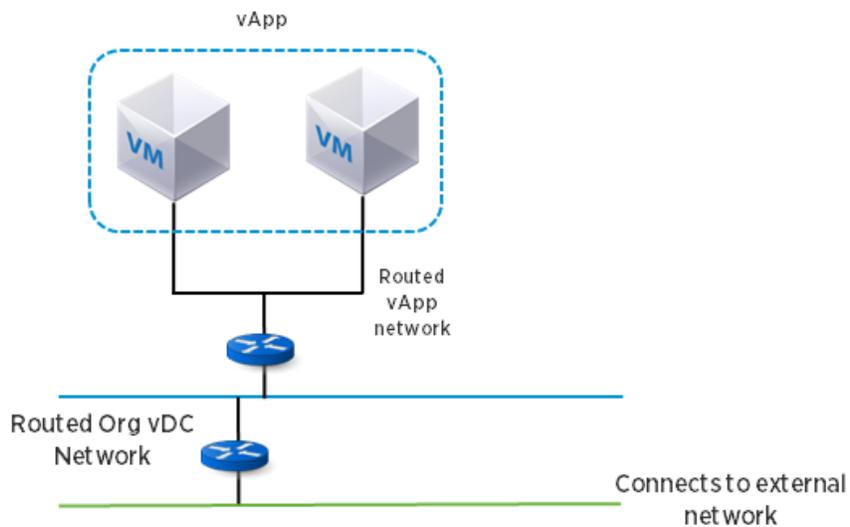Figure 20 shows the throughput of deploying a vApp with routed vApp network operation.
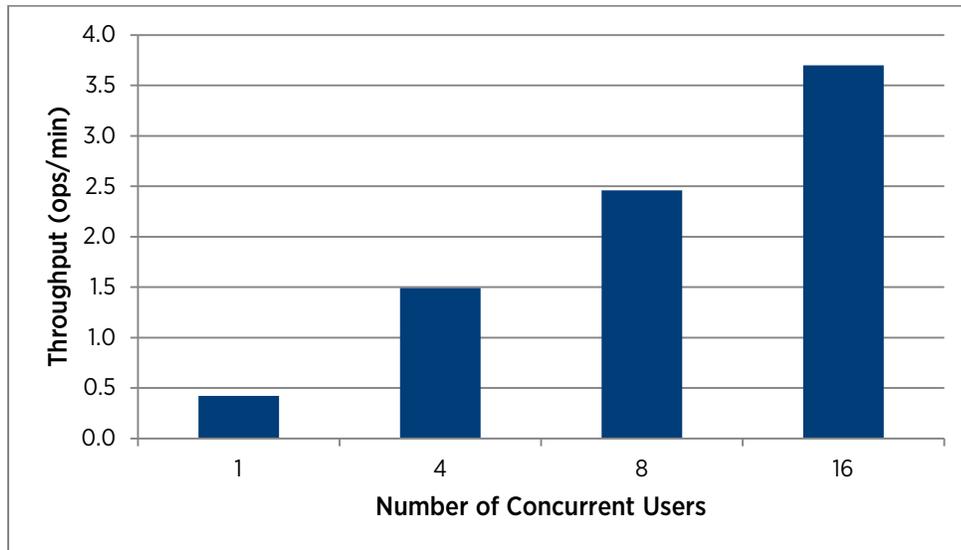

Figure 20. Deploying a vApp with routed vApp network

From the results, we can see deploying a vApp with a routed vApp network has a similar throughput result as creating an Edge gateway (shown in Figure 14). This is because a vShield Edge virtual appliance needs to be deployed and configured for the routed vApp network at deployment phase.

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 "Oracle Database" to make sure the database is configured for best performance.

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 5.0 and VC 5.1 , please refer to *Configuration Maximums for VMware vSphere 5.0* [3] and *Configuration Maximums for VMware vSphere 5.1*[4].

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 restarting the vCloud Director cell.

THREAD POOL	DEFAULT SIZE (FOR EACH CELL)	USAGE/INFORMATION	ADJUSTMENT PROCEDURE
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

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.

Especially when a vCloud cell is working in high concurrency (as described below), we recommend increasing **JVM heap size, DB Connection pool,** and **jetty connection** for better performance.

The default configuration of a JVM heap size is 2GB. When there are more than 128 concurrent user operations, we recommend increasing the JVM heap size to 3GB as below.

```
JAVA_OPTS:--Xms1536M -Xmx3072M -XX:MaxPermSize=768m
```

To modify JVM heap size:

1. Stop the cell.

2. Edit the `vmware-vcd-cell` file found in `<vCloud director install directory>/bin/vmware-vcd-cell.`

3. Configure java heap option, for example, increase it to 3GB as follows:

    ```
    JAVA_OPTS:--Xms1536M -Xmx3072M -XX:MaxPermSize=768m
    ```

4. Save the file.

5. Start the cell.

For high concurrency with 128 users or higher, it is also recommended to increase the jetty thread pool size and DB connection pool via editing `global.properties` file as shown in Table 4.

ADVANCED OPTION NAME	DESCRIPTION	DEFAULT VALUE	PROPOSED VALUE FOR HIGH CONCURRENCY
database.pool.maxActive	The max number of active database pool size	75	200
vcloud.http.maxThreads	The max number of http thread size	128	150
vcloud.http.minThreads	The min number of http thread	25	32
vcloud.http.acceptorThreads	The number of acceptor thread size	2	16

Table 4. Advanced options

To increase the database connection pool:

Edit the `global.properties` file found in `<vCloud director install directory>/etc/`

```
database.pool.maxActive = 200
```

To increase the jetty thread pool:

Edit the `global.properties` file found in `<vCloud director install directory>/etc/`

```
vcloud.http.maxThreads = 150
vcloud.http.minThreads =  32
vcloud.http.acceptorThreads = 16
```

vCenter configuration limits are very important because vCloud Director utilizes vCenter for many operations. Refer to *Configuration Maximums for VMware vSphere 5.0* [3] and *Configuration Maximums for VMware vSphere 5.1* [4].

Conclusion

In this paper, we discussed some of the features of the vCloud Director 5.1 release, performance characterizations including latency breakdown, latency trends, resource consumptions, sizing guidelines and hardware requirements, and performance tuning tips.

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] Configuration Maximums for VMware vSphere 5.0
http://www.vmware.com/pdf/vsphere5/r50/vsphere-50-configuration-maximums.pdf

[4] VMware vCenter 5.1 Configuration Limits
https://www.vmware.com/pdf/vsphere5/r51/vsphere-51-configuration-maximums.pdf

[5] vCloud API Programming Guide
 http://pubs.vmware.com/vcd-51/topic/com.vmware.ICbase/PDF/vcd_51_api_guide.pdf

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

[7] vCloud Director Installation and Upgrade Guide
http://pubs.vmware.com/vcd-51/topic/com.vmware.ICbase/PDF/vcd_51_install.pdf

[8] vCloud Director Administrator's Guide 5.1
http://pubs.vmware.com/vcd-51/topic/com.vmware.ICbase/PDF/vcd_51_admin_guide.pdf

[9] vCloud Director User's Guide 1.5
http://pubs.vmware.com/vcd-51/topic/com.vmware.ICbase/PDF/vcd_51_users_guide.pdf

[10] Installing and Configuring a vCloud Director 5.1 Database
http://kb.vmware.com/kb/2034540

Appendix

Rebuilding Cell Database Indexes

In order to rebuild indexes for Oracle and MS SQL server, we ran the following commands against the cell database. Please note that these scripts are used only in our internal testing, and are only for reference. Consult your latest documents or your database administrator for any updates or changes based on the version of the database you are using to perform the database operation appropriately. For vCloud Director installation details, refer to the *vCloud Director Installation and Upgrade Guide* [7].

For the Oracle Database:

```
SELECT 'ALTER INDEX ' || index_name || ' REBUILD; ' FROM User_indexes WHERE
tablespace_name = 'CLOUD_INDX' AND index_name NOT LIKE '%$$';

EXEC DBMS_STATS.GATHER_SCHEMA_STATS('VCLOUD', cascade=>true);
```

For the MS SQL Server:

```
USE [database-name]
GO
EXEC sp_MSforeachtable @command1="print '?' DBCC DBREINDEX ('?', ' ', 80)"
GO

USE [database-name]
EXEC sp_updatestats
GO
```

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.

Kimberly Wang is a senior member of technical staff and has been with VMware for 6 years. She works in the performance group for scalability and networking for VMware vCloud Director. Before joining the performance group, Kimberly worked in the VMware vSphere® SDK group.

Zhe Yang is a senior member of technical staff at VMware and mainly works on performance and scalability for VMware vCloud Director. Prior to VMware, Zhe was a senior engineer at Videro, Yahoo!, and Schlumberger. Zhe received her MS degree in computer science from the University of Minnesota.

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 MS in computer science from California State University, Fresno.

Xuwen Yu is a senior member of technical staff in VMware where he works on several projects including VMware vCloud Director, VMware vSphere Update Manager and VMware vSphere Host Profile/Auto Deploy. Xuwen received his PhD degree in computer science and engineering from the University of Notre Dame in 2009.

Acknowledgments

vmware®