

## Network virtualization

The IaaS cloud computing model, the cloud provider runs a datacenter that offers "VMs for rent" along with dynamically allocated resources. The customer owns the VM and manages it as "its server" in the cloud. The meaning of the terms cloud provider and customer can differ, of course, depending on whether you're talking about a shared private cloud or a shared public cloud. Specifically, the following points apply:

- In the **shared private cloud scenario**, the cloud provider is the organization itself, which owns and operates its own datacenter, whereas the customers might be different business units, departments, or offices in different locations.
- In the **shared public cloud scenario**, the cloud provider is the hosting company, whereas the customers might be large enterprises, mid-sized companies, or even small businesses. The hosting company owns and manages the datacenter and may "rent out" servers to customers, offer colocation of customer-owned servers, or both.

In both scenarios, the cloud provider can provide the numerous benefits of cloud computing to its customers, but typically not without problems using today's technologies. For example, VLANs are typically used by cloud providers to isolate the servers belonging to one customer from those belonging to other customers and provisioned from the same cloud. VLANs accomplish this by adding tags to Ethernet frames. Then Ethernet switches can be configured to enforce isolation by allowing nodes that have the same tag to communicate with each other, but not with nodes having a different tag. But VLANs have several limitations:

- They have limited scalability because typical Ethernet switches support no more than 1,000 VLAN IDs (with a theoretical maximum of 4,094).
- They have limited flexibility because a single VLAN can't span multiple IP subnets.
- They have high management overhead associated with them because Ethernet switches need to be reconfigured each time a VLAN is added or removed.

Another problem that customers often experience when contemplating moving their computing resources to the cloud is IP addressing. The issue is that the customer's existing infrastructure typically has one addressing scheme, whereas the datacenter network has an entirely different addressing scheme. So

when a customer wants to move one of its servers into the cloud, typically by virtualizing the workload of the existing physical server so the workload can be run as a VM hosted within the cloud provider's datacenter, the customer is usually required to change the IP address of their server so it can fit the addressing scheme of the cloud provider's network. This can pose difficulties, however, because IP addresses are often tied to geographical locations, management policies, and security policies, so changing the server's address when its workload is moved into the cloud may result in routing issues, servers moving out of management scope, or security policies failing to be applied properly.

It would simplify cloud migrations a lot if the customer's servers could keep their existing IP addresses when their workloads are virtualized and moved into the cloud provider's datacenter. That way, the customer's existing routing, management, and security policies should continue to work as before. And that's exactly what Network Virtualization does!

### **How Network Virtualization works**

Network Virtualization is a new feature in Windows Server 2012 that lets you keep your own internal IP addresses when moving your servers into the cloud. For example, let's say that you have three on-premises physical servers having private IP addresses 192.168.33.45, 192.168.33.46, and 192.168.33.47, and you want to move these servers to the datacenter of a cloud provider called Fabrikam. These servers are currently in the 192.168.0.0/16 address space, and Fabrikam's datacenter uses 10.0.0.0/24 for its datacenter network's address space. If Fabrikam has Windows Server 2012 deployed in its datacenter, you're in luck because your servers can keep their existing IP addresses when their workloads are migrated into VMs running on Fabrikam host machines. This means that your existing clients, which are used to accessing servers located on the 192.168.0.0/16 subnet, will be able to continue doing so with no modifications needed to your routing infrastructure, management platform, or network security policies. That's Network Virtualization at work.

But what if another customer of Fabrikam uses the exact same subnetting scheme for its own virtualized workloads? For example, let's say that Northwind Traders also has been using 192.168.0.0/16 on its private network, and one of the servers it's moved into Fabrikam's datacenter has the exact same IP address (192.168.33.45) as one of the servers that you've moved into Fabrikam's

datacenter? No problem! Network Virtualization in Windows Server 2012 provides complete isolation between VMs belonging to different customers even if those VMs use the exact same IP addresses!

Network Virtualization works by allowing you to assign two different IP addresses to each VM running on a Windows Server 2012 Hyper-V host. These two addresses are:

- **Customer address (CA).** The IP address that the server had when it resided on the customer's premises before it was migrated into the cloud. In the previous example, this might be the 192.168.33.45 address for a particular server that the customer wants to move to the cloud.
- **Provider address (PA).** The IP address assigned by the cloud provider to the server once the server has been migrated to the provider's datacenter. In the previous example, this could be 10.44.2.133, or some other address in the 10.0.0.0/24 address space.

From the customer's perspective, communication with the migrated server is just the same as if the server still resided on the customer's own premises. This is because the VM running the customer's migrated workload can see and use its customer address and thus can be reached by other hosts on the customer's network. The VM cannot see or use its provider address, however, because this address is visible only to the hosts on the cloud provider's network.

Network Virtualization thus lets the cloud provider run multiple virtual networks on top of a single physical network in much the same way as server virtualization lets you run multiple virtual servers on a single physical server. Network Virtualization also isolates each virtual network from every other virtual network, with the result that each virtual network has the illusion that it is a separate physical network. This means that two or more virtual networks can have the exact same addressing scheme, yet the networks will be fully isolated from one another and each will function as if it is the only network with that scheme.

To make this all happen, Network Virtualization needs a way of virtualizing IP addresses and mapping them to physical addresses. Network Virtualization in Windows Server 2012 offers two ways of accomplishing this:

- **Network Virtualization Generic Routing Encapsulation (NVGRE).** In this approach, all the VM's packets are encapsulated with a new header before they are transmitted onto the physical network. NVGRE requires only one PA per host, which is shared by all VMs on that host.

- **IP rewrite.** This approach modifies the customer addresses of packets while they are still on the VM and before they are transmitted onto the physical network. IP rewrite requires a one-to-one mapping of customer addresses to provider addresses.

NVGRE is compatible with today's datacenter network hardware infrastructure and is the recommended approach for implementing Network Virtualization.

Because Network Virtualization is intended for datacenters, implementing it requires that you have a VM management framework in place. System Center Virtual Machine Manager 2012 Service Pack 1 provides such a framework and lets you use Windows PowerShell or WMI to create and manage virtual networks.

## **Benefits of Network Virtualization**

Network Virtualization is key to being able to build and provision multi-tenant cloud services, both for shared private clouds, where the "customers" are different business units or departments, and for public cloud scenarios, where the cloud provider offers "space to rent" to all comers. Network Virtualization lets you create multi-tenant networks where each network is fully isolated from all other networks, and it does this without any of the limitations of or overhead associated with the job of creating and managing VLANs. This means that cloud providers can use Network Virtualization to create as many networks as you want – thousands and thousands of them for example if you are a large hosting provider – and then move workloads anywhere you want without having to perform the arduous (and error-prone) task of reconfiguring VLANs.

Network Virtualization also provides greater flexibility for VM placement, which helps reduce overprovisioning and fragmentation of resources for the cloud provider. By enabling dynamic VM placement, the cloud provider can make best use of the compute, network, and storage resources within their datacenter and can monitor and control the provisioning of these resources more easily.

Regardless of whether you are a customer looking to migrate your server workloads into the cloud, an enterprise seeking to implement a shared private cloud for provisioning "servers for rent" to different divisions or locations, or a hosting provider wanting to offer cloud hosting services to large numbers of customers, Network Virtualization in Windows Server 2012 provides the foundation for achieving your goals. Table 2-2 summarizes the benefits of Network Virtualization to these different parties.

TABLE 2-2 – The benefits that Network Virtualization can provide to customers, enterprises, and hosting providers

Owner	Benefits
The customer who owns the workload that needs to be moved into the cloud	Seamless migration to the cloud Easy to move your three-tier topology to the cloud
An enterprise seeking to deploy a shared private cloud	Easy cloud bursting Preserve your VM settings, IP addresses, and policies Cross premises server-to-server connectivity
A hosting provider wanting to offer secure, multi-tenant "servers for rent" using a shared public cloud	Flexible VM placement requiring no network reconfiguration Create and manage large number of tenant networks

### Network Virtualization operational challenges

The Network Virtualization capabilities found in Windows Server 2012 provide a fresh approach to an old problem, and that is primarily that of operator density. Operators, or service providers, are no longer interested in 1:1 solutions. They want more virtual servers per physical server today the same way they wanted more subscribers for a given pool of dial-up modems at the early of days of the World Wide Web. Density typically came at the price of mobility and scalability. Today, of course, this is less of an issue, at least in datacenter virtualization scenarios, as we can have density pushing the limits of hardware while maintaining mobility and scalability.

There was always one difficult problem to solve: how to extend the mobility and scalability of a single datacenter to two or more. This was often required either as datacenters ran out of space or often after a merger or acquisition. Nearly every customer I have worked with in the past five years has or had a datacenter relocation or consolidation project. The problem was now: how do I move these servers to new datacenters while maintaining all the monitoring and security policies associated with their location? The answer usually consisted of storage and network architects sitting down and installing new network and storage equipment which really extended the network subnet(s) from one datacenter to another. This, in a way, was the precursor to Network

Virtualization, and we were able to learn a lot from this. Especially with respect to the newer problems we discovered as a result. Some of the problems we discovered included:

- **Application behavior.** Moving the VM from one datacenter to another typically introduced network latency. Some applications just did not behave well with the added latency.
- **Supportability.** It was now difficult for datacenter technicians to effectively know which datacenter a VM was located in by looking at its IP address.
- **Licensing.** It used to be that some vendors licensed their products to a single IP address. This proved challenging to customers, so certain vendors changed their licensing to be based on the MAC address of the host's NIC. This meant that moving the VM (while keeping its IP) to another datacenter meant it had to keep the same MAC address. Although this is typically possible within a single management domain, this is impossible to predict when that VM was being moved, for instance, to a service provider or a private cloud provider.

Looking back at these problems, I realized the key to avoiding them was to involve the application and server operations. Although this sounds incredibly trivial in theory, it is incredibly difficult to do in practice. How often do you get involved in a project involving Network Virtualization if you are the corporate custodian or owner of the HR application, for instance?

Server virtualization forced teams to learn to communicate with other teams. Network Virtualization will make that even more critical. When you decide to implement Network Virtualization features found in Windows Server 2012, consider adding teams with operational experience to your team and ensure key application support teams are also consulted.

### **Improved Live Migration**

Live Migration was introduced in Windows Server 2008 R2 to provide a high-availability solution for VMs running on Hyper-V hosts. Live Migration uses the Failover Clustering feature to allow running VMs to be moved between cluster nodes without perceived downtime or loss of network connection. Live Migration provides the benefit of increased agility by allowing you to move running VMs to the best host for improving performance, achieving better scaling, or ensuring optimal workload consolidation. Live Migration also helps increase productivity

and reduce cost by allowing you to service your host machines without interruption or downtime for your virtualized workloads.

Live Migration in Windows Server 2008 R2 required storing VMs on an Internet Small Computer Systems Interface (iSCSI) or Fibre-Channel SAN. In addition, Live Migration in Windows Server 2008 R2 supported performing only a single Live Migration at a time – multiple simultaneous Live Migrations were not supported.

Now Live Migration in Windows Server 2012 has been improved in several significant ways. First, Live Migrations can be performed much more quickly. In fact, you can even saturate a 10 GB network connection when performing a Live Migration between Windows Server 2012 Hyper-V hosts, something you couldn't do before with Windows Server 2008 R2 Hyper-V hosts.

A second improvement to Live Migration in Windows Server 2012 is that now you can perform multiple Live Migrations simultaneously within the same failover cluster. This means, for example, that if you needed to take down a particular cluster node for immediate servicing, you can migrate all running VMs from that node to a different node quickly and simultaneously in a single operation using either the GUI or a Windows PowerShell command. This can greatly simplify the task of performing maintenance on Hyper-V hosts within your environment.

A third improvement is that Live Migration is now possible even if you don't have a failover clustering infrastructure deployed. In the previous version of Windows Server 2008 R2, Live Migration required installing the Failover Clustering feature, and you also needed to ensure that **Cluster Shared Volume** (CSV) storage was enabled to ensure the **logical unit number** (LUN) on which your VM is stored could be accessed by any cluster node at any given time. With Windows Server 2012, however, you have two additional options for Live Migration that can be performed outside a failover clustering environment:

- You can store your VMs on a shared folder on your network, which lets you live-migrate between non-clustered Hyper-V hosts while leaving the VM's files on the share.
- You also can live-migrate a VM directly from one stand-alone Hyper-V host to another without using any shared storage at all.

Let's look at these two Live Migration options in a bit more detail.

### **Live Migration using a shared folder**

With Hyper-V in Windows Server 2012 you can now store all of a VM's files on a shared folder on your network provided the shared folder is located on a file server running Windows Server 2012 (see Figure 2-6). The reason the shared folder must be located on a file server running Windows Server 2012 is because this scenario is supported only through the new capabilities of version 3 of the **server message block** (SMB) protocol (SMB 3).

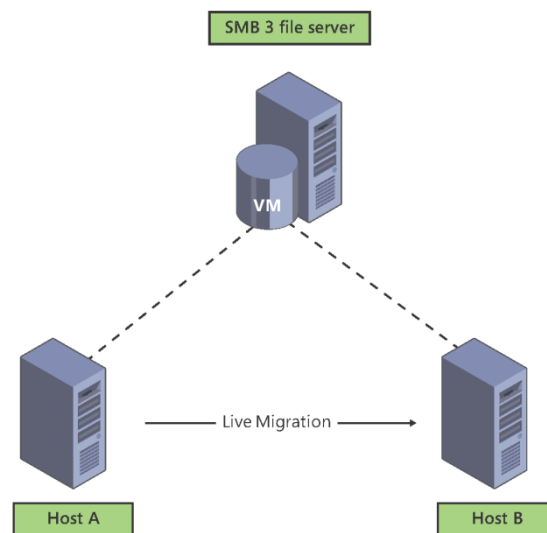


FIGURE 2-6 – Live Migration using SMB 3 shared storage but no clustering

Live Migration using SMB 3 shared storage does not in itself provide high availability unless the file share itself is also highly available. It does, however, also provide the benefit of enhanced VM mobility. And this added mobility can be achieved without the high costs associated SANs and their associated switching fabric. SANs also add extra management overhead in the form of provisioning and managing LUNs. But by simply deploying a Windows Server 2012 file server, you can centralize storage of the VMs in your environment without the added cost and management overhead associated with using a SAN.

Live Migration using SMB 3 shared storage does have a couple of requirements to get it to work, namely the permissions on the share must be configured appropriately, constrained delegation must be enabled in Active Directory directory service, and the path to the shared storage must be configured correctly in the VM's settings. But once everything is set up properly, the procedure for performing Live Migration is essentially unchanged from before.



## Live Migration without shared storage

Windows Server 2012 also allows you to live migrate VMs between stand-alone Hyper-V hosts without the use of any shared storage. This scenario is also known as Live Migration Without Infrastructure (or Shared Nothing Live Migration), and the only requirements are that the two hosts must belong to the same Active Directory domain and that they must be using processors from the same manufacturer (all AMD or all Intel, for instance). When Live Migration without infrastructure is performed, the entire VM is moved from the first host to the second with no perceived downtime. The process basically works like this (see Figure 2-7):

- 1) The Virtual Machine Management Service (VMMS; Vmms.exe) on the first host (where the VM originally resides) negotiates and establishes a Live Migration connection with the VMMS on the second host.
- 2) A storage migration is performed, which creates a mirror on the second host of the VM's VHD file on the first host.
- 3) The VM state information is migrated from the first host to the second host.
- 4) The original VHD file on the first host is then deleted and the Live Migration connection between the hosts is terminated.

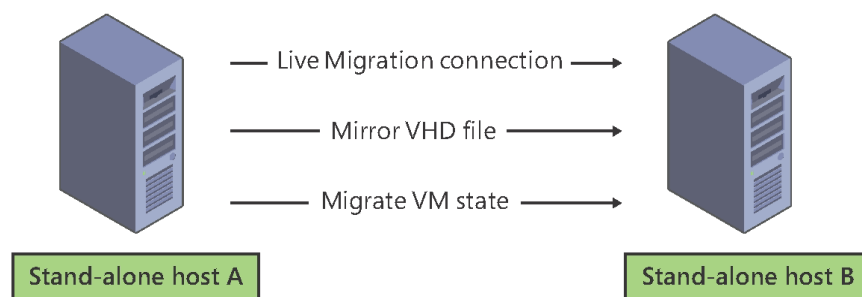


FIGURE 2-7 – How Live Migration without shared storage works in Windows Server 2012

## Performing Live Migration

Live Migration can be performed from the GUI or using Windows PowerShell, but first you need to enable Live Migration functionality on your host machines. This can be done by using the Hyper-V console to open the Hyper-V Settings dialog box, as shown in Figure 2-8.

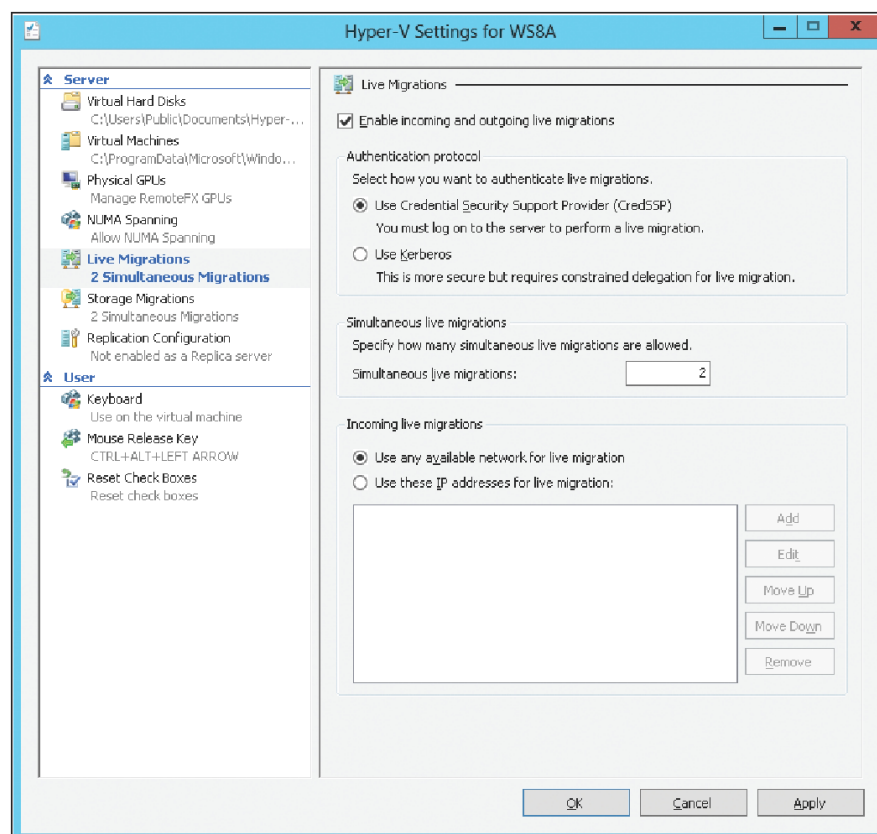


FIGURE 2-8 – Enabling Live Migrations in Hyper-V Settings.

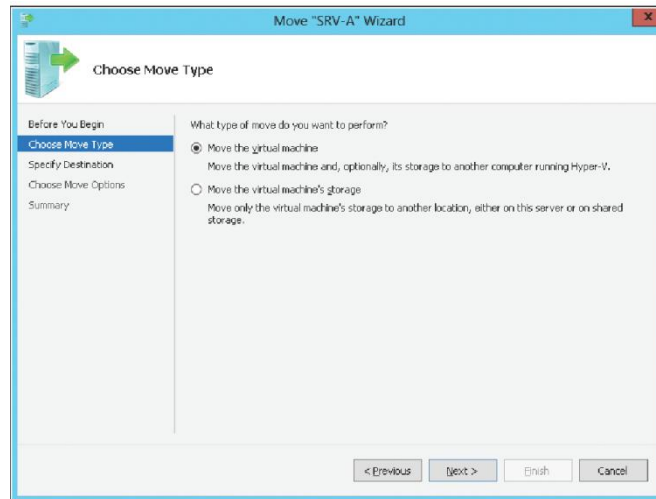
The tools that you can use to perform a Live Migration depend on the kind of Live Migration you want to perform. Table 2-3 summarizes the different methods for performing Live Migrations in failover clustering environments, Live Migrations using SMB 3 shares, and Live Migrations without infrastructure.

TABLE 2-3 – Methods for performing different types of Live Migrations

Type of Live Migration	GUI tools	Windows PowerShell cmdlets
VM is on a cluster node and managed by the cluster.	Failover Cluster Manager	Move-ClusterVirtualMachineRole Move-VM
VM is on an SMB 3 share.	Hyper-V Manager	Move-VM
VM is on a stand-alone host.	Hyper-V Manager	Move-VM

Windows Server 2012 gives you great flexibility in how you perform Live Migrations of running VMs, including moving different VM components to

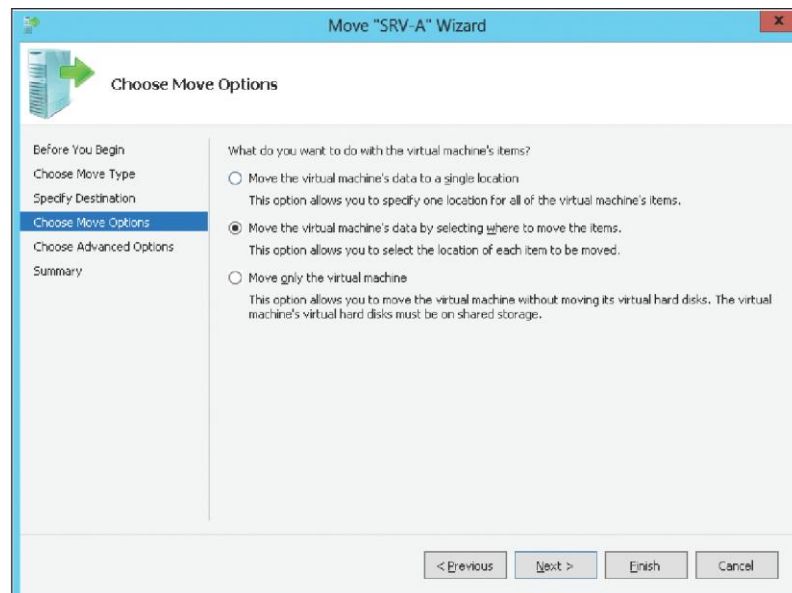
different locations on the destination host when performing Live Migrations with or without shared storage. To see this, right-click a running VM in Hyper-V Manager and select Move to start the wizard for moving VMs. The first choice you make is whether to move the VM (and, optionally, its storage) to a different host or to move only the VM's storage, as shown here:



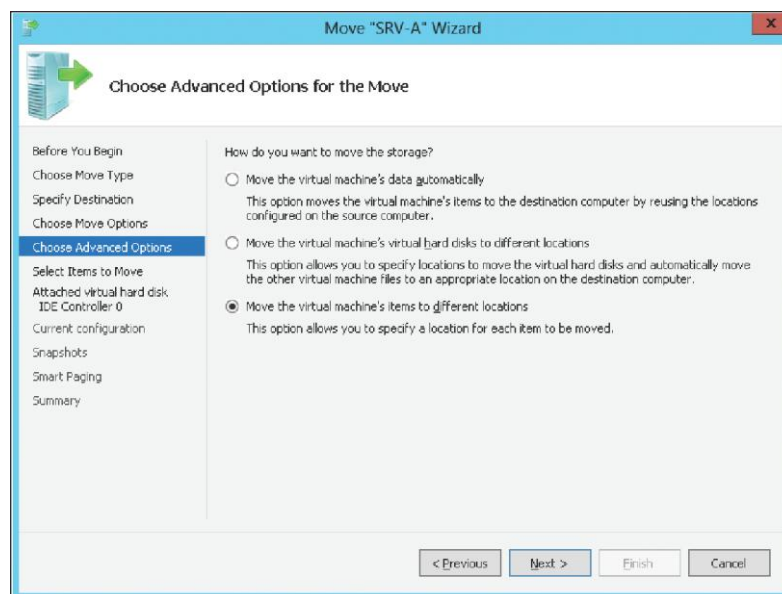
Moving the storage of a running VM is called **storage migration** and is a new capability for Hyper-V in Windows Server 2012. Once you've specified the name of the host you want to move the VM to, you're presented with three options:

- Moving all the VM's files to a single location
- Moving different files of the VM to different locations
- Moving all the VM's files except its VHDs

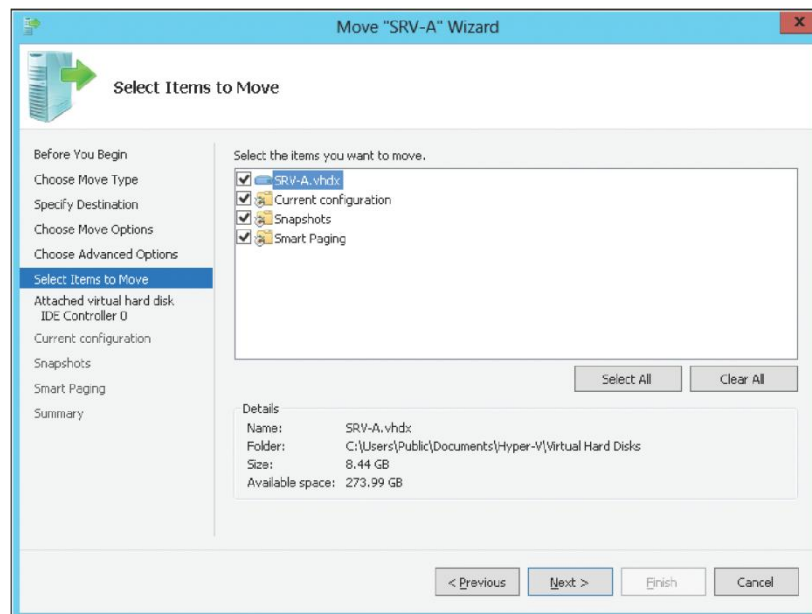
In each case, the target locations could be a shared folder on a Windows Server 2012 file server or a local directory on the destination host:



If you choose the second option of moving different files of the VM to different locations as shown here, you're presented with additional options for specifying how to move the storage:



Choosing to move the VM's items to different locations lets you specify which items you want to move, including the VHDs, current configuration, snapshot files, and smart paging files for the VM:



Additional wizard pages allow you to specify the exact way in which these items should be moved.

## Enhanced quality of service (QoS)

We looked at the new bandwidth management capabilities found in Hyper-V, which allows for guaranteeing a minimum amount of bandwidth and/or enforcing a maximum amount of bandwidth for each VM running on a host. This is just one example, however, of the powerful new bandwidth management capabilities built into Windows Server 2012. The term quality of service (QoS) refers to technologies used for managing network traffic in ways that can meet SLAs and/or enhance user experiences in a cost-effective manner. For example, by using QoS to prioritize different types of network traffic, you can ensure that mission-critical applications and services are delivered according to SLAs and to optimize user productivity.

As we've previously seen in the earlier section, Hyper-V in Windows Server 2012 lets you specify upper and lower bounds for network bandwidth used by VMs. This is an example of software QoS at work where packet scheduling is implemented by the operating system. But Windows Server 2012 also supports implementing QoS through the use of network adapter hardware that use Data Center Bridging (DCB), a technology that provides performance guarantees for different types of network traffic. DCB is typically found in 10 GbE network adapters and certain kinds of switching fabrics.

The enhanced QoS capabilities included in Windows Server 2012 are particularly useful in shared cloud environments, where the cloud provider wants to ensure that each customer (or business unit for shared private clouds) is able to access the computing, storage, and network resources they need and have paid for or been guaranteed. Customers (and departments of large enterprises) need predictable performance from applications and services they access from the cloud, and the enhanced QoS capabilities in Windows Server 2012 can help ensure this.

But these enhanced QoS capabilities also can provide benefits to the cloud provider. Previously, to ensure that all customers accessing a shared cloud have enough computing, storage, and network resources to meet their needs, cloud providers often overprovisioned VMs on the hosts in their datacenter by running fewer VMs on more hosts, plus extra storage and network resources to ensure that each customer has enough. For example, the cloud provider might use separate networks for application, management, storage, and Live Migration traffic to ensure that each type of workload can achieve the required level of performance. But building and managing multiple physical networks like this can be expensive, and the provider may have to pass the cost on to the customer to ensure profitability.

With the enhanced QoS capabilities in Windows Server 2012, however, cloud providers can ensure that SLAs are met while using their physical host, storage, and network resources more efficiently, which means cost savings from needing fewer hosts, less storage, and a simpler network infrastructure. For example, instead of using multiple overlapping 1 GbE networks for different kinds of traffic, the provider can use a single 10 GbE network backbone (or two for high availability) with each type of traffic carried on it being prioritized through the use of QoS policies.

From the perspective of enterprises wanting to build private clouds and hosting providers wanting to build public clouds, QoS allows replacing multiple physical networks with a single converged network carrying multiple types of traffic with each traffic type guaranteed a minimum amount of bandwidth and limited to a maximum amount of bandwidth. Implementing a QoS solution thus can save enterprises and hosting providers money in two ways: less network hardware is needed and high-end network hardware such as 10 GbE network adapters and switches can be used more efficiently. Note, however, that the converged fabric still needs to be carved up into Management and Production networks for security reasons.

The bottom line is that the old approach of overprovisioning the network infrastructure for your datacenter is inefficient from a cost point of view and now can be superseded by using the new QoS capabilities in Windows Server 2012. Instead of using multiple physical network fabrics like 1 GbE, iSCSI, and Fibre Channel to carry the different kinds of traffic in your multi-tenant datacenter, QoS and other enhancements in Windows Server 2012 now make it possible to use a single converged 10 GbE fabric within your datacenter.

## Implementing QoS

There are a number of different ways of implementing software-based control of network traffic in Windows Server 2012. For example:

- You can configure Hyper-V QoS as described previously by enabling bandwidth management in the settings of your VMs to guarantee a minimum amount of bandwidth and/or enforcing a maximum amount of bandwidth for each VM.
- You can use Group Policy to implement policy-based QoS by tagging packets with an 802.1p value to prioritize different kinds of network traffic.
- You can use Windows PowerShell or WMI to enforce minimum and maximum bandwidth and 802.1p or Differentiated Services Code Point (DSCP) marking on filtered packets.

There are additional ways of implementing QoS as well. The method(s) you choose will depend upon the network infrastructure you have and the goals that you are trying to achieve. See the "Learn more" section for more information about QoS solutions for Windows Server 2012.

In terms of which QoS functionality to use in a given scenario, the best practice is to configure Hyper-V QoS for VMs and then create QoS policies when you need to tag traffic for end-to-end QoS across the network.

## QoS and the cloud

If you are a hosting provider or a large enterprise that wants to deploy a shared private cloud that provides "servers for rent" to customers or business units, there are several ways that you can configure Hyper-V QoS to assign a minimum bandwidth for each customer or business unit that access applications and services from your cloud:

- **Absolute minimum bandwidth.** In this scenario, you could set different service tiers such as bronze for 100 Mbps access, silver for 200 Mbps access, and gold

for 500 Mbps access. Then you can assign the appropriate minimum bandwidth level for customers based on the level of their subscription.

- **Relative minimum bandwidth.** In this scenario, you could assign different weights to different customer workloads such as a weight of 1 for normal priority workloads, 2 for high-priority workloads, and 5 for critical-priority workloads. Then you could assign a minimum bandwidth to each customer based on their workload weight divided by the total weight of all customers accessing your cloud.

Note that minimum bandwidth settings configured in Hyper-V QoS are applied only when there is contention for bandwidth on the link to your cloud. If the link is underused, the configured minimum bandwidth settings will have no effect. For example, if you have two customers, one with gold (500 Mbps) access and the other with silver (200 Mbps) access, and the link between the cloud and these customers is underused, the gold customer will not have  $500/200 = 2.5$  times more bandwidth than the silver customer. Instead, each customer will have as much bandwidth as they can consume.

Absolute minimum bandwidth can be configured using the Hyper-V Settings in Hyper-V Manager, as shown previously in this chapter. Absolute minimum bandwidth also can be configured from Windows PowerShell by using the Set-VMSwitch cmdlet. Relative minimum bandwidth can be configured from Windows PowerShell only by using the Set-VMSwitch cmdlet.

As far as configuring maximum bandwidth is concerned, the reason for doing this in cloud environments is mainly because wide area network (WAN) links are expensive. So if you are a hosting provider and a customer accesses its "servers in the cloud" via an expensive WAN link, it's a good idea to configure a maximum bandwidth for the customer's workloads to cap throughput for customer connections to their servers in the cloud.

### **Data Center Bridging (DCB)**

Data Center Bridging (DCB) is an IEEE standard that allows for hardware-based bandwidth allocation for specific types of network traffic. The standard is intended for network adapter hardware used in cloud environments so that storage, data, management, and other kinds of traffic all can be carried on the same underlying physical network in a way that guarantees each type of traffic its fair share of bandwidth. DCB thus provides an additional QoS solution that uses



hardware-based control of network traffic, as opposed to the software-based solution described previously.

Windows Server 2012 supports DCB, provided that you have both DCB-capable Ethernet network adapters and DCB-capable Ethernet switches on your network.

## **Resource metering**

Resource metering is a new feature of Windows Server 2012 designed to make it easier to build solutions for tracking how cloud services are consumed. Such tracking is important in both enterprise and hosting scenarios. For example, if a hosting provider provides cloud-based applications and services to customers, the hosting provider needs a way of tracking how much resources those customers are consuming to bill them for their use of these resources. Similarly, if a large enterprise has deployed a shared private cloud that is accessed by different business units within the organization, the enterprise needs a way of tracking how much cloud resources each business unit is consuming. This information may be needed for internal billing purposes by the organization, or it may be used to help plan how cloud resources are allocated so that each business unit gets its fair share of the resources they need.

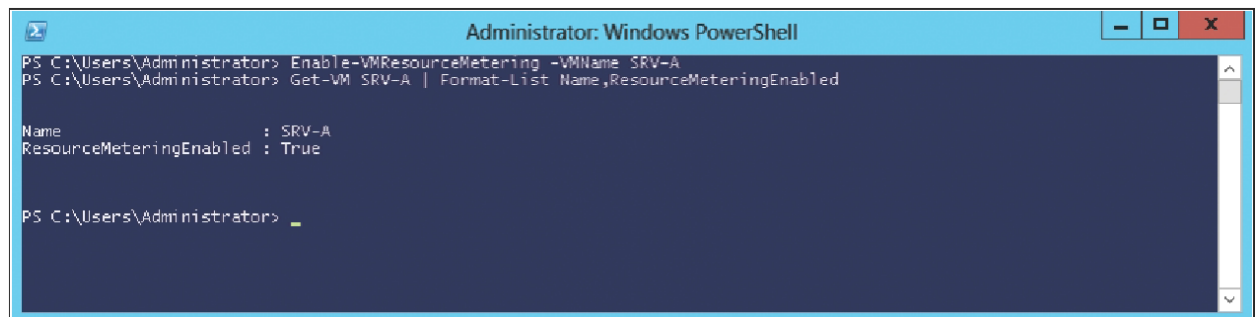
Previously, enterprises or hosting providers who deployed shared private or public cloud solutions using Hyper-V virtualization in Windows Server 2008 and Windows Server 2008 R2 had to create their own chargeback solutions from scratch. Such solutions typically were implemented by polling performance counters for processing, memory, storage, and networking. With the new built-in resource metering capabilities in Windows Server 2012, however, these organizations can use Windows PowerShell to collect and report on historical resource usage of the following metrics:

- Average CPU usage by a VM
- Average physical memory usage by a VM
- Minimum physical memory usage by a VM
- Maximum physical memory usage by a VM
- Maximum amount of disk space allocated to a VM
- Total incoming network traffic for a virtual network adapter
- Total outgoing network traffic for a virtual network adapter.

In addition, these metrics can be collected in a consistent fashion even when the VMs are moved between hosts using Live Migration or when their storage is moved using storage migration. And for billing of network usage, you can differentiate between billable Internet traffic and non-billable internal datacenter traffic by configuring network metering port ACLs.

## Implementing resource metering

As an example, let's use resource metering to measure resource usage for a VM on our Hyper-V host. We'll start by enabling resource metering for the VM SRV-A using the `Enable-VMResourceMetering` cmdlet, and then we'll verify that resource metering has been enabled by piping the output of the `Get-VM` cmdlet into the `Format-List` cmdlet:

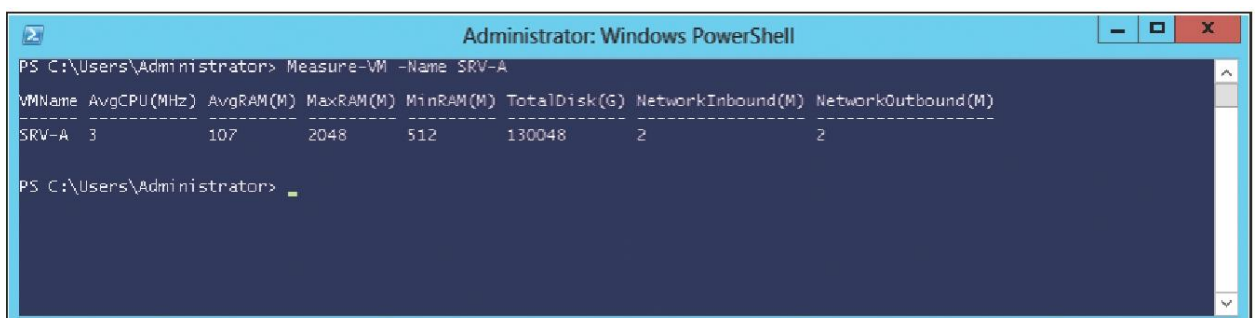


```
Administrator: Windows PowerShell
PS C:\Users\Administrators> Enable-VMResourceMetering -VMName SRV-A
PS C:\Users\Administrators> Get-VM SRV-A | Format-List Name,ResourceMeteringEnabled

Name           : SRV-A
ResourceMeteringEnabled : True

PS C:\Users\Administrators> _
```

Now we can use the `Measure-VM` cmdlet to report resource utilization data on our VM:

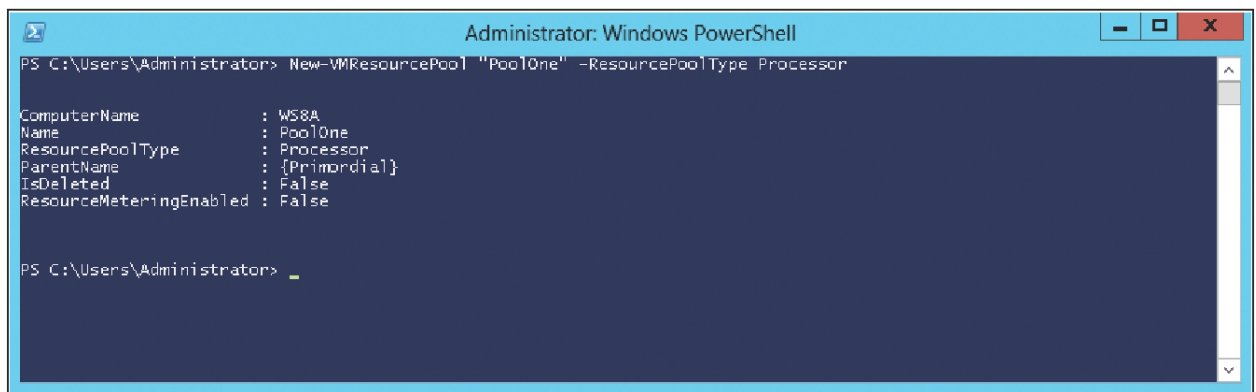


```
Administrator: Windows PowerShell
PS C:\Users\Administrators> Measure-VM -Name SRV-A

VMName AvgCPU(MHz) AvgRAM(M) MaxRAM(M) MinRAM(M) TotalDisk(G) NetworkInbound(M) NetworkOutbound(M)
-----
SRV-A   3           107       2048      512       130048       2                     2

PS C:\Users\Administrators> _
```

You also can create resource pools for reporting usage for different types of resources such as Processor, Ethernet, Memory or VHD. For example, you could create a new resource pool named `PoolOne` using the `New-VMResourcePool` cmdlet:



```
Administrator: Windows PowerShell
PS C:\Users\Administrator> New-VMResourcePool "PoolOne" -ResourcePoolType Processor

ComputerName      : WS8A
Name              : PoolOne
ResourcePoolType  : Processor
ParentName        : {Primordial}
IsDeleted         : False
ResourceMeteringEnabled : False

PS C:\Users\Administrator> _
```

Then, once you've enabled resource metering on the new pool using the `Enable-VMResourceMetering` cmdlet, you can use the `Measure-VMResourceMetering` cmdlet to report processor utilization for the pool. You also can use the `Reset-VMResourceMetering` cmdlet to reset the collection of resource metering data.

Resource metering data can be collected, retrieved and reported by combining different Windows PowerShell cmdlets using pipelines. To configure network metering port ACLs for differentiating different kinds of traffic, you can use the `add-VMNetworkAdapterACL` cmdlet.

## Increase scalability and performance

Building cloud solutions, whether with private or public clouds, requires investment of time, energy, and money. To ensure best return on your investment, you need to build your solution on a platform that can scale and perform well to meet the changing demands of your business. This means being able to take advantage of cutting-edge hardware that can provide extreme performance while handling the largest possible workloads. It means being able to use resources effectively at every level, while ensuring that SLAs can be met. It means reducing the chances of mistakes occurring when maintenance tasks are performed. And it means being able to monitor performance effectively to ensure computing, storage, and network resources are used with maximum efficiency.

Windows Server 2012 delivers a virtualization platform that can achieve the highest levels of performance while delivering extreme scalability that enables new scenarios for migrating massive workloads into the cloud. This section examines some new features in Hyper-V and in the underlying operating system that enable such increased scalability and performance.

## **Expanded processor and memory support**

Hyper-V in Windows Server 2008 R2 has been embraced by many organizations as a way of making more efficient use of physical server hardware through virtualization and consolidating server workloads. But limitations in the number of logical processors supported on the host and for VMs, together with limitations of how much physical memory can be supported on the host and assigned to VMs, has meant that Windows Server 2008 R2 lacked sufficient scalability for certain types of mission-critical business applications. For example, large database applications often require large amounts of memory and many logical processors when used to implement business solutions involving online transaction processing (OLTP) or online transaction analysis (OLTA). Until now, the idea of moving such applications into the cloud has been mostly a dream.

Windows Server 2012 changes all this in the following ways:

Through its increased processor and memory support on the virtualization host by enabling the use of up to 160 logical processors and 2 TB of physical memory per host system.

Through its increased virtual processor and memory support for VMs by enabling the use of up to 32 virtual processors and 1 TB of memory per VM.

### **Increased host processor and memory support**

The advent of Windows Server 2012 brings the expansion of processor and memory support in Windows Server 2012. In Windows Server 2008 R2, the host system had limitations of the amount of maximum logical processors (cores, Hyper-Threading, individual CPUs) and memory available for use between the host and the VM. To illustrate this point, note the following:

Windows Server 2008 R2 SP1 had support for up to:

- 64 logical processors per host.
- 1 TB of memory per host.
- 4 virtual processors per VM.
- 64 GB of memory per VM.

Windows Server 2012 now has support for up to:

- 320 logical processors per host.
- 4 TB of memory per host.
- 64 virtual processors per VM (up to a maximum of 2,048 virtual processors per host).
- 1 TB of memory per VM.

## **Virtual NUMA**

In addition to its expanded processor and memory support on hosts and for VMs, Hyper-V in Windows Server 2012 also expands support for Non-Uniform Memory Access (NUMA) from the host into the VM. NUMA allows the use of memory by processors to be optimized based on the location of the memory with respect to the processor. High-performance applications like Microsoft SQL Server have built-in optimizations that can take advantage of the NUMA topology of a system to improve how processor threads are scheduled and memory is allocated.

In previous versions of Hyper-V, VMs were not NUMA-aware, which meant that when applications like SQL Server were run in VMs, these applications were unable to take advantage of such optimizations. Because NUMA was not used in previous versions, it was possible for a VM's RAM to span NUMA nodes and access non-local memory. There is a performance impact when using non-local memory due to the fact that another memory controller (CPU) has to be contacted.

But with VMs now being NUMA-aware in Windows Server 2012, the performance of applications like SQL Server can be significantly better. Note, however, that NUMA support in VMs works in Hyper-V in Windows Server 2012 only when Dynamic Memory has not been configured on the host.

### **How it works**

Virtual NUMA presents a NUMA topology within a VM so that the guest operating system and applications can make intelligent decisions about thread and memory allocation that are reflected in the physical NUMA topology of the host. For example, Figure 2-9 shows a NUMA-capable four-socket host machine with four physical NUMA nodes labeled 1 through 4. Two VMs are running on this host, and two virtual NUMA nodes are presented within each VM, and these virtual NUMA nodes align with physical NUMA nodes on the host based on policy. The result is that NUMA-aware applications like SQL Server installed on the guest operating system of one of these VMs would be able to allocate its thread and memory resources as if it was running directly upon a physical server that had two NUMA nodes.

## Virtual NUMA and failover clustering

Virtual NUMA support also extends into high-availability solutions built using failover clustering in Windows Server 2012. This enables the failover cluster to place VMs more appropriately by evaluating the NUMA configuration of a node before moving a VM to the node to ensure the node is able to support the workload of the VM. This NUMA-awareness for VMs in failover clustering environments helps reduce the number of failover attempts which results in increased uptime for your VMs.

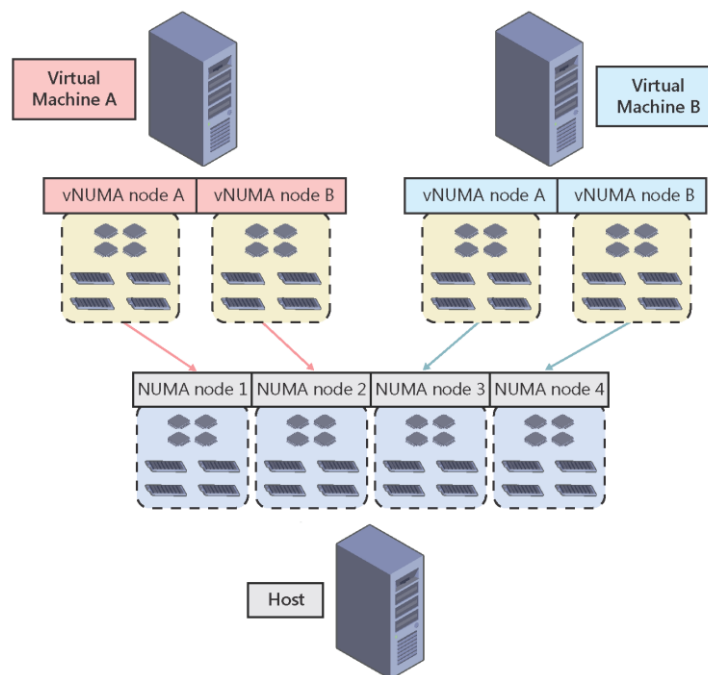


FIGURE 2-9 – Example of virtual NUMA at work

## Network adapter hardware acceleration

Besides the increased processor and memory support available for both hosts and VMs, Windows Server 2012 also supports various hardware acceleration features of high-end network adapter hardware to ensure maximum scalability and performance in cloud scenarios. As Figure 2-10 shows, most of these features can be enabled in the Hyper-V Settings of Hyper-V Manager, provided that your network adapter hardware supports these functionalities.

## Virtual Machine Queue (VMQ)

Virtual Machine Queue (VMQ) was first available for the Hyper-V role in Windows Server 2008 R2 for host machines that had VMQ-capable network adapter hardware. VMQ employs hardware packet filtering to deliver packets from an external VM network directly to VMs using Direct Memory Access (DMA) transfers. This helps reduce the overhead of routing packets from the host to the VM, which helps improve the performance of the host operating system by distributing the processing of network traffic for multiple VMs among multiple processors. Previously, all network traffic was handled by a single processor.

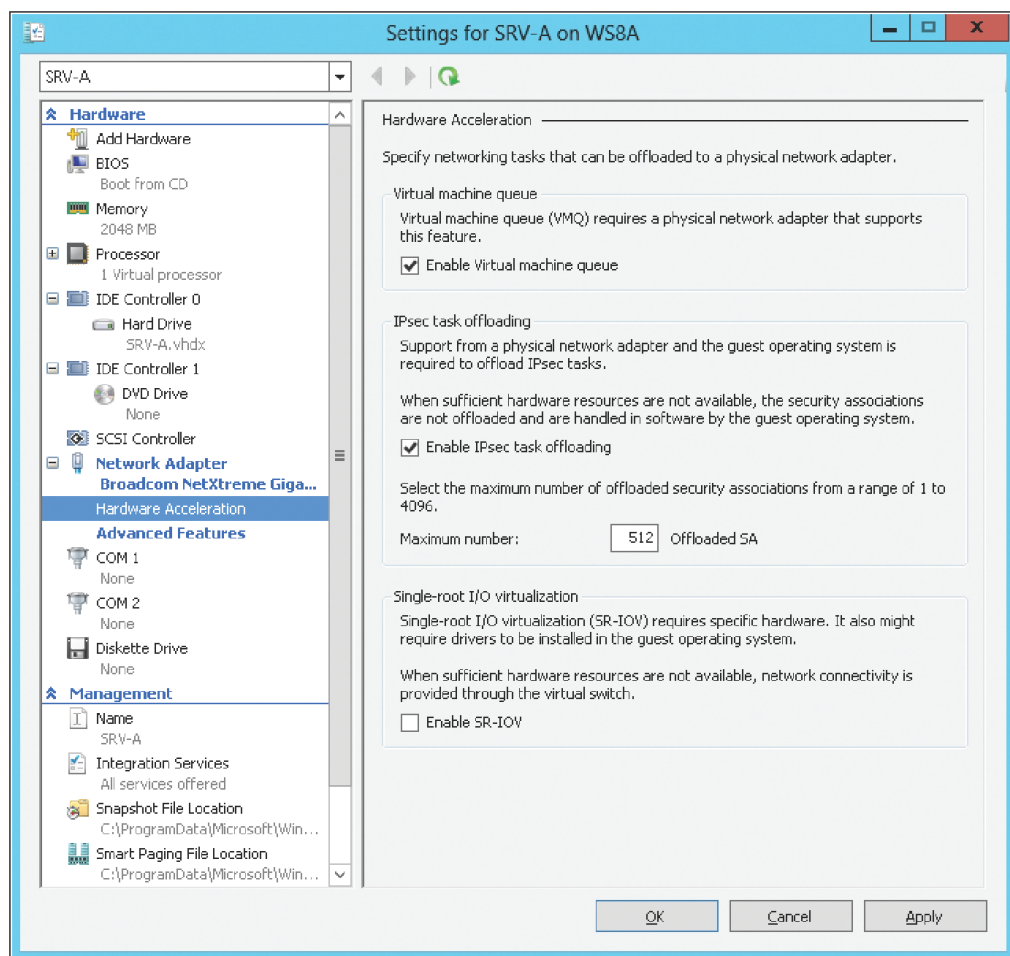


FIGURE 2-10 – Enabling use of the hardware acceleration capabilities of high-end network adapter hardware on Hyper-V hosts

NDIS 6.30 in Windows Server 2012 includes some changes and enhancements in how VMQ is implemented. For example, splitting network data into separate look-ahead buffers is no longer supported. In addition, support for Static VMQ has been removed in Windows Server 2012. Drivers using NDIS 6.3

will automatically access Dynamic VMQ capabilities that are new in Windows Server 2012.

Although in Windows Server 2008 R2 you had to use System Center Virtual Machine Manager to enable VMQ for a VM on a Hyper-V host, beginning with Windows Server 2012, you can enable VMQ directly from within the VM's settings exposed through Hyper-V Manager, as discussed previously. Windows Server 2012 also includes several new Windows PowerShell cmdlets, such as `Set-NetAdapterVmq`, `Get-NetAdapterVmq`, and `Get-NetAdapterVmqQueue`, that can be used to manage the VMQ properties of network adapters.

### **IPsec task offload**

Internet Protocol Security (IPsec) task offload was first available for servers running Windows Server 2008 that had network adapters that supported this functionality. IPsec task offload works by reducing the load on the system's processors by performing the computationally intensive job of IPsec encryption/decryption using a dedicated processor on the network adapter. The result can be a dramatically better use of the available bandwidth for an IPsec-enabled computer.

Beginning with Windows Server 2012, you can enable IPsec task offload directly from within the VM's settings exposed through Hyper-V Manager, as detailed previously. Windows Server 2012 also includes some new Windows PowerShell cmdlets, such as `Set-NetAdapterIPsecOffload` and `Get-NetAdapterIPsecOffload`, that can be used to manage the IPsec Offload properties of network adapters.

### **Single-root I/O virtualization**

Single root I/O virtualization (SR-IOV) is an extension to the PCI Express (PCIe) specification, which enables a device such as a network adapter to divide access to its resources among various PCIe hardware functions. As implemented in the Hyper-V role of Windows Server 2012, SR-IOV enables network traffic to bypass the software switch layer of the Hyper-V virtualization stack to reduce the I/O overhead in this layer. By assigning SR-IOV capable devices directly to a VM, the network performance of the VM can be nearly as good as that of a physical machine. In addition, the processing overhead on the host is reduced.



Beginning with Windows Server 2012, you can enable SR-IOV directly from within the VM's settings exposed through Hyper-V Manager, as shown in Figure 2-11. Before you can do this, however, the virtual switch that the VM uses must have SR-IOV enabled on it, and you also may need to install additional network drivers in the guest operating system of the VM. You can enable SR-IOV on a virtual switch only when you create the switch using the Virtual Switch Manager of Hyper-V Manager or by using the New-VMSwitch cmdlet when using Windows PowerShell. Windows Server 2012 also includes some new Windows PowerShell cmdlets, such as Set-NetAdapterSriov, Get-NetAdapterSriov, and Get-NetAdapterSriovVf, that can be used to manage the SR-IOV properties of network adapters, such as the number of virtual functions (VFs), virtual ports (VPorts), and queue pairs for default and non-default VPorts.

Note that only SR-IOV supports 64-bit guest operating systems (specifically Windows Server 2012 and 64-bit versions of Windows 8). In addition, SR-IOV requires both hardware and firmware support in the host system and network adapter. If you try to configure a guest operating system to use SR-IOV when either the hardware or firmware is not supported, the Network tab in Hyper-V Manager will display "Degraded (SR-IOV not operational)."

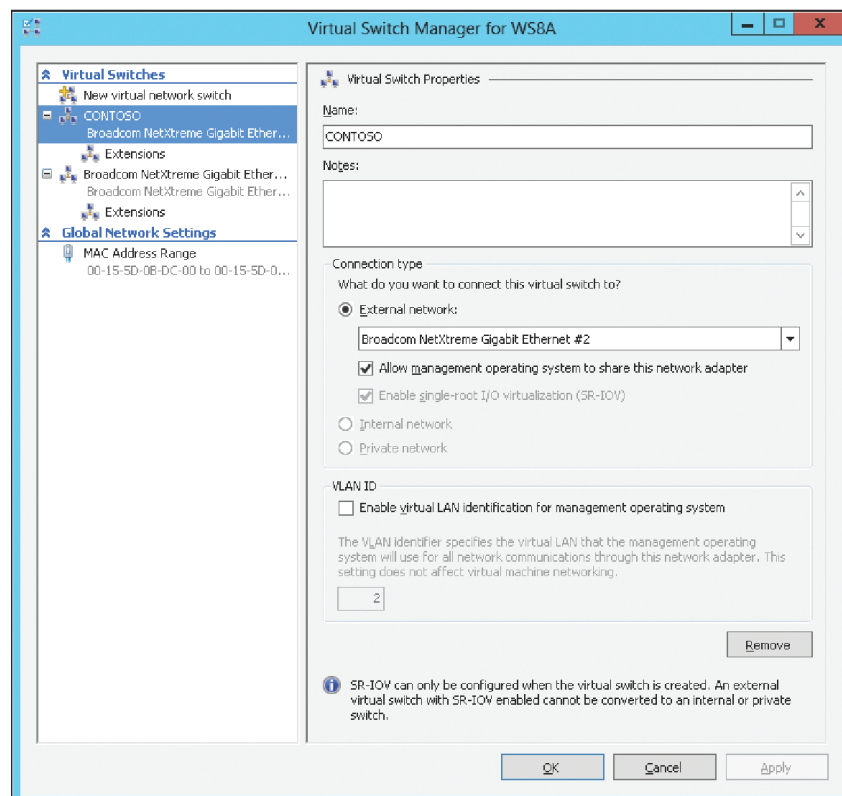


FIGURE 2-11 – SR-IOV must be configured on the virtual switch before it can be configured for the VM

### Offloaded Data Transfer (ODX)

Another performance and scalability improvement in Windows Server 2012 revolves around storage, in particular when storing VMs on storage arrays. Offloaded Data Transfer (ODX) is a feature of high-end storage arrays that uses a token-based mechanism to read and write data within and between such arrays. Using ODX, a small token is copied between the source and destination servers instead of routing data through the host (see Figure 2-12). So when you migrate a VM within or between storage arrays that support ODX, the only thing copied through the servers is the token representing the VM file, not the underlying data in the file.

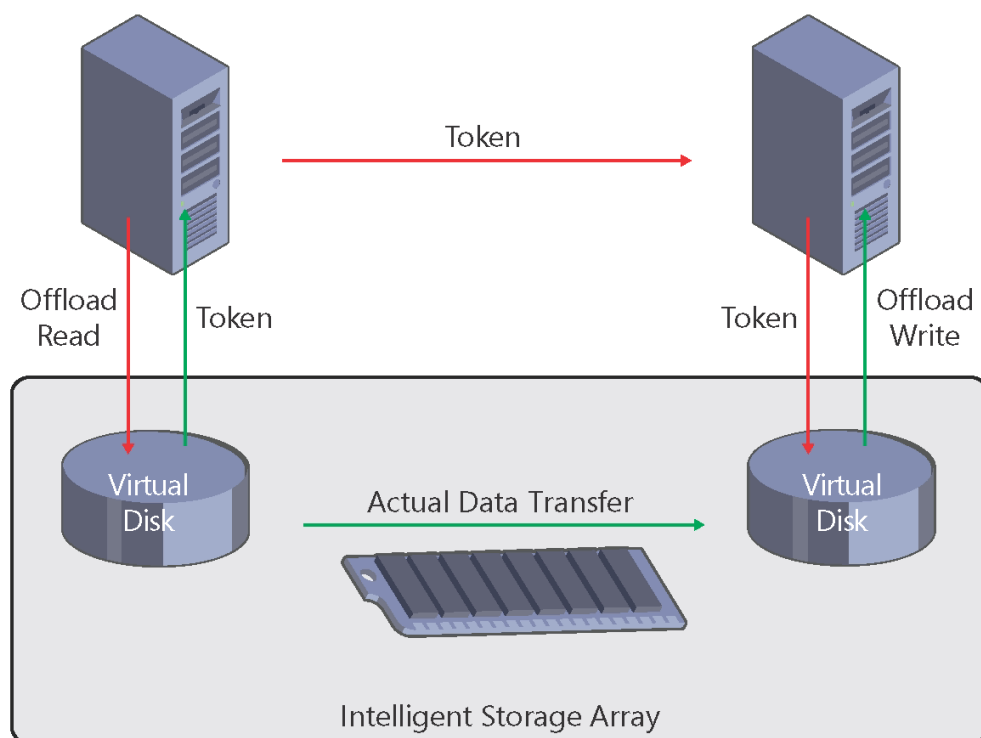


FIGURE 2-12 – How offloaded data transfer works in a Hyper-V environment

The performance improvement when using ODX-capable storage arrays in cloud environments can be astounding. For example, instead of taking about three minutes to create a new 10 GB fixed VHD, the entire operation can be

completed in less than a second! Other VM operations that can benefit just as much using ODX-capable storage hardware include:

- Expansion of dynamic VHDs.
- Merging of VHDs.
- Live Storage Migration.

ODX also can provide benefit in non-virtualized environments, such as when transferring large database files or video files between servers.

### **Support for 4 KB sector disks**

Windows Server 2012 now includes support for large-sector disks. These disks represent the newest trend in the storage industry whereby the old 512-byte sector format is being replaced by the new 4,096-byte (4 KB) format to meet demand for increased disk capacity. Hyper-V in Windows Server 2012 now supports hosting VHD files on disks that have either the native 4-KB format or the transitional 512-byte emulation (512e) mode.

### **Dynamic Memory improvements**

Dynamic Memory was introduced for Hyper-V in Windows Server 2008 R2 as a way of enabling virtualization hosts to make more effective use of physical memory allocated to VMs running on the host. Dynamic Memory works by adjusting the amount of memory available to the VM in real time. These adjustments in memory allocation are based on how much memory the VM needs and on how Dynamic Memory has been configured on the VM.

Dynamic Memory provides important scalability and performance benefits, especially for virtual desktop infrastructure (VDI) environments, where at any given time, a subset of the VMs running on the host tend either to be idle or to have a relatively low load. By using Dynamic Memory in such scenarios, you can consolidate greater numbers of VM on your Hyper-V hosts. The result is that you'll need fewer hosts for provisioning virtual desktops to your user population, which means you won't need to procure as much high-end server hardware. In other words, Dynamic Memory can help you save money.

### **Configuring Dynamic Memory**

Dynamic Memory is enabled on a per-VM basis. You can enable and configure Dynamic Memory in the Memory section of the VM's settings in Hyper-V Manager, as shown in Figure 2-13 below. You also can enable and configure Dynamic Memory using Windows PowerShell by using the Set-VM cmdlet, which can be used to configure the various properties of a VM. Note that you can enable or disable Dynamic Memory only when the VM is in a stopped state.

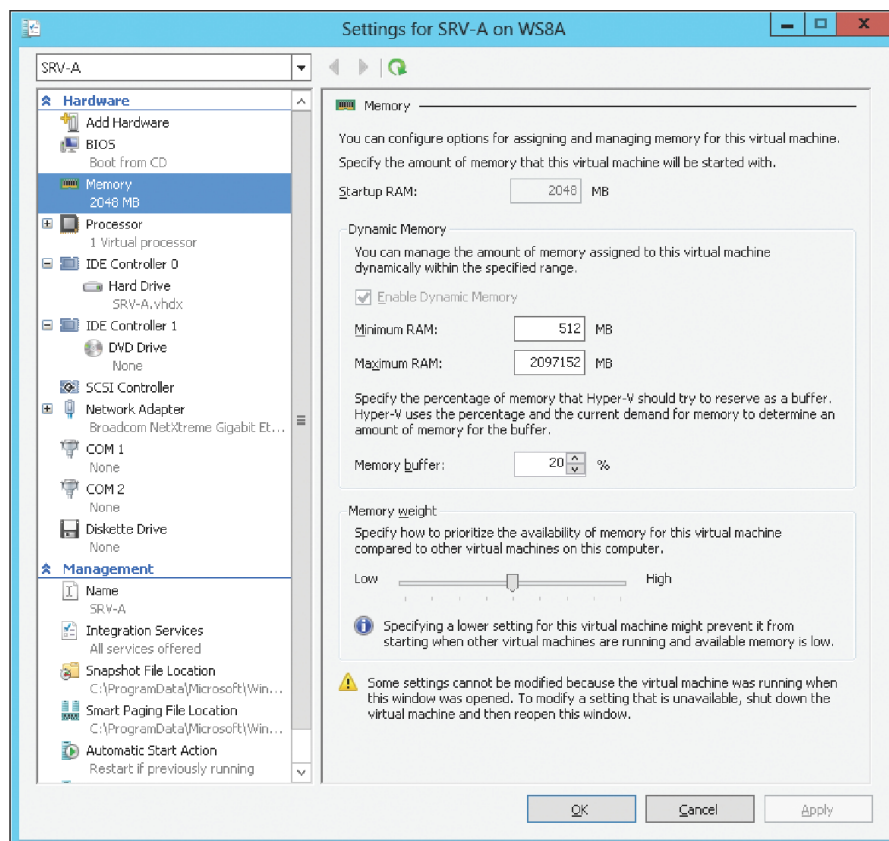


FIGURE 2-13 – Configuring Dynamic Memory for a VM.

Configuration options for Dynamic Memory for VMs on Hyper-V hosts running Windows Server 2008 R2 were as follows:

- **Startup RAM.** The amount of memory needed for starting the VM
- **Maximum RAM.** The maximum amount of memory that the VM can use
- **Memory buffer.** An amount of memory (as a percentage of the amount that the VM actually needs to perform its workload) that can be allocated to the VM when there is sufficient memory available on the host
- **Memory weight.** A parameter that determines how available memory on the host is allocated among the different VMs running on the host.

Configuration options for Dynamic Memory for VMs on Hyper-V hosts running Windows Server 2012 have been enhanced in several ways:

A new configuration setting called **Minimum Memory** allows you to specify the minimum amount of memory that the VM can use when it is running. The reason for introducing this new setting is because Windows generally needs more memory when starting than it does when idle and running. As a result of this change, you now can specify sufficient startup memory to enable the VM to start quickly and then a lesser amount of memory (the minimum memory) for when the VM is running. That way, a VM can get some extra memory so it can start properly, and then once it's started, Windows reclaims the unneeded memory so other VMs on the host can use the reclaimed memory if needed.

Another change in the way that Dynamic Memory can be configured in Windows Server 2012 is that now you can modify the maximum and minimum memory settings while the VM is running. In Windows Server 2008 R2, the maximum memory setting could be modified only when the VM was in a stopped state. This change gives you a new way of quickly provisioning more memory to a critical VM when needed

## Smart Paging

Specifying a minimum memory for a VM can enable Windows to reclaim some unneeded memory once the VM has started. Then this reclaimed memory can be reallocated towards other VMs on the host. But this raises a question: What if you start as many VMs as you can on a host, allow Windows to reclaim unneeded memory once the VMs are running, then start more VMs using the reclaimed memory, then again allow Windows to reclaim any additional unneeded memory, then try to start more VMs on the host... and so on? Eventually, you reach the point where almost all the host's memory is in use and you're unable to start any more VMs. But then you find that one of your running VMs needs to be restarted immediately (for example, to apply a software update). So you try and restart the VM, and it shuts down successfully but it won't start again. Why not? Because there's not enough free memory on the host to meet the Startup RAM criterion for that VM.

To prevent this kind of scenario from happening while enabling Dynamic Memory to work its scalability magic, Hyper-V in Windows Server 2012 introduces a new feature called Smart Paging (see Figure 2-14). Smart Paging allows a VM that's being restarted to use disk resources temporarily on the host as a source

for any additional memory needed to restart the VM successfully. Then, once the VM has started successfully and its memory requirements lessen, Smart Paging releases the previously used disk resources because of the performance hit that such use can create.

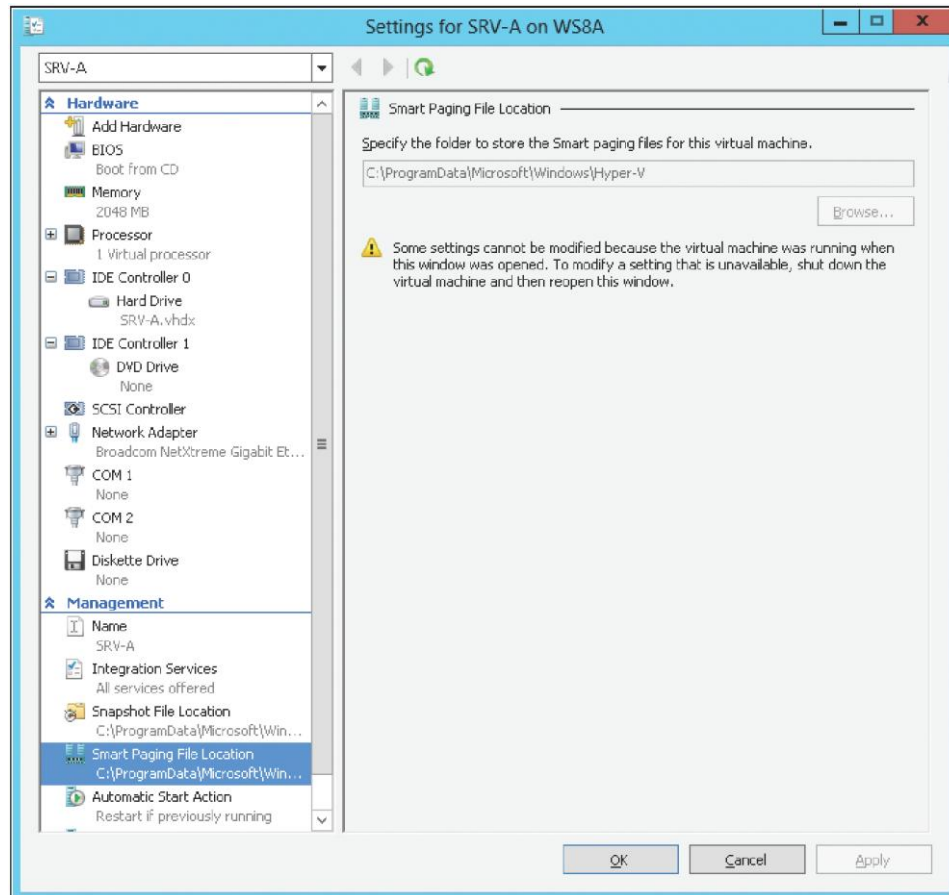


FIGURE 2-14 – Smart Paging works with Dynamic Memory to enable reliable VM restart operations

Smart Paging is used only when a VM is restarted and there is no free physical memory on the host and no memory can be reclaimed from other running VMs. Smart Paging is not used if you simply try and start a VM that's in a stopped state, or if a VM is failing over in a cluster.

### Viewing Dynamic Memory at work

Sometimes small changes make a big difference in the usability of a user interface feature. In the Hyper-V Manager of Windows Server 2008 R2, you could monitor in real time how much physical memory was allocated to each VM that

had Dynamic Memory enabled on it. This real-time allocation amount is called the assigned memory. In addition, you could monitor the memory demand (the total committed memory) and the memory status (whether the current amount of memory assigned to the VM as a buffer is sufficient) for the VM. The problem, though, was that these real-time measurements were displayed as columns in the Virtual Machines pane of Hyper-V Manager, which meant that you had to scroll horizontally to see them.

Hyper-V in Windows Server 2012 adds a series of tabs to the bottom central pane, and by selecting the Memory pane, you can view the assigned memory, memory demand, and memory status for the selected VM quickly (see Figure 2-15).

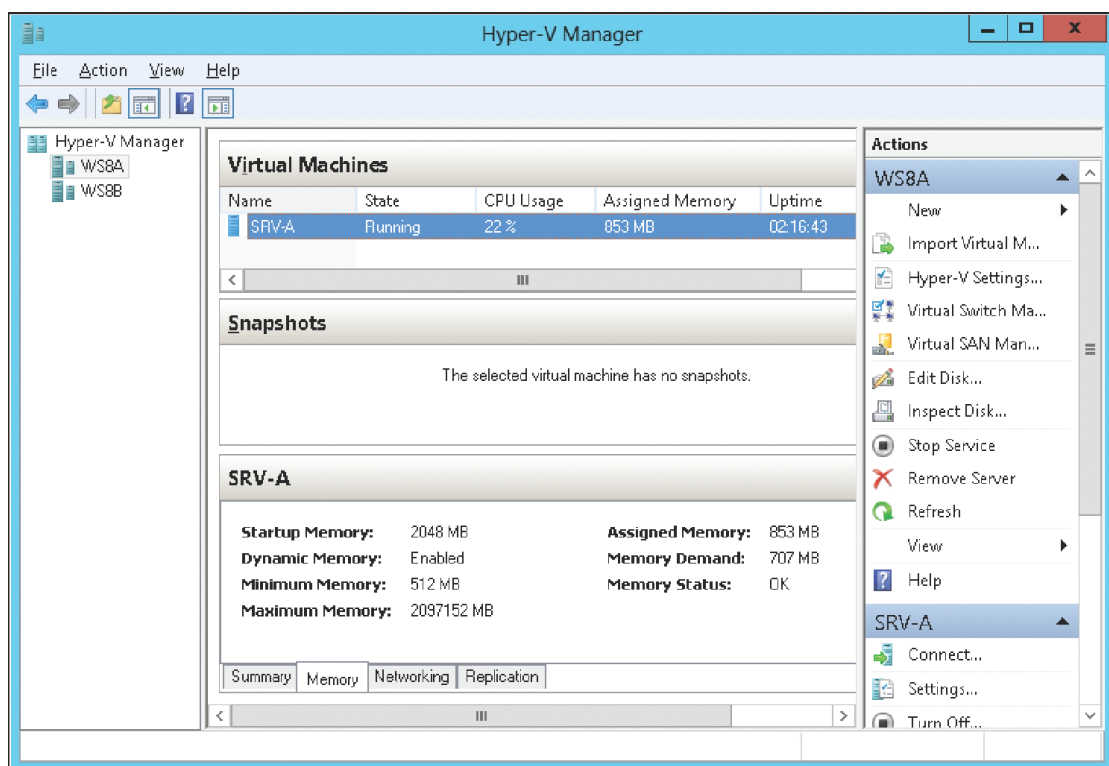
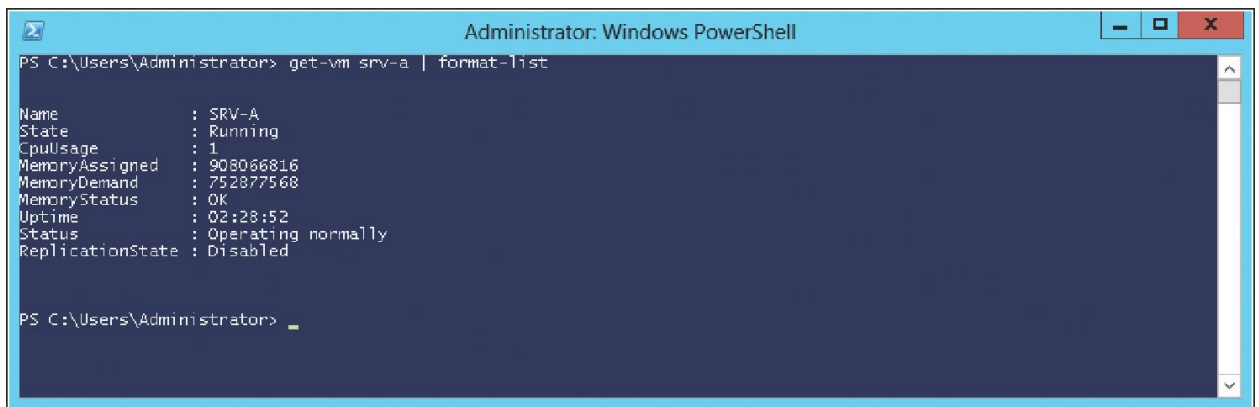


FIGURE 2-15 – Using Hyper-V Manager to display real-time changes in memory usage by a VM with Dynamic Memory enabled.

You also can use the Get-VM cmdlet in Windows PowerShell to display these same real-time measurements, as shown in Figure 2-16.

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command prompt shows the command "get-vm srv-a | format-list" being executed. The output displays various properties of the VM SRV-A, including its state (Running), CPU usage (1), memory assigned (908066816), memory demand (752877568), memory status (OK), uptime (02:28:52), status (Operating normally), and replication state (Disabled).

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> get-vm srv-a | format-list

Name           : SRV-A
State          : Running
CpuUsage       : 1
MemoryAssigned : 908066816
MemoryDemand   : 752877568
MemoryStatus   : OK
Uptime        : 02:28:52
Status        : Operating normally
ReplicationState : Disabled

PS C:\Users\Administrator> _
```

FIGURE 2-16 – Using Windows PowerShell to display real-time changes in memory usage by a VM with Dynamic Memory enabled.

## Virtual Fibre Channel

Existing technologies often present obstacles when considering the migration of your server workloads into the cloud. An example of this might be if you have an AlwaysOn failover cluster instance running on SQL Server 2012 that's configured to use a Fibre Channel SAN for high performance. You'd like to migrate this workload into the cloud, but Hyper-V in Windows Server 2008 R2 does not support directly connecting to Fibre Channel from within VMs. As a result, you've postponed performing such a migration because you want to protect your existing investment in expensive Fibre Channel technology.

Virtual Fibre Channel removes this blocking issue by providing Fibre Channel ports within the guest operating system of VMs on Hyper-V hosts running Windows Server 2012. This now allows a server application like SQL Server running within the guest operation system of a VM to connect directly to LUNs on a Fibre Channel SAN.

Implementing this kind of solution requires that the drivers for your HBAs support Virtual Fibre Channel. Some HBAs from Brocade and QLogic already include such updated drivers, and more vendors are expected to follow. Virtual Fibre Channel also requires that you connect only to LUNs, and you can't use a LUN as boot media for your VMs.

Virtual Fibre Channel also provides the benefits of allowing you to use any advanced storage functionality of your existing SAN directly from your VMs. You can even use it to cluster guest operating systems over Fibre Channel to provide high availability for VMs.



## SMB 3

Windows Server 2012 introduces SMB 3, version 3 of the Server Message Block (SMB) protocol to provide powerful new features for continuously available file servers. SMB is a network file sharing protocol that allows applications to read and write to files and to request services from services over a network. (Note that some documentation on TechNet and MSDN still refer to this version as SMB 3.)

The improvements in SMB 3 are designed to provide increased performance, reliability, and availability in scenarios where data is stored on file shares. Some of the new features and enhancements in SMB 3 include:

- **SMB Direct.** Enables using network adapters capable of Remote Direct Memory Access (RDMA) such as iWARP, Infiniband, or RoCE (RDMA over Converged Ethernet) that can function at full speed and low latency with very little processor overhead on the host. When such adapters are used on Hyper-V hosts, you can store VM files on a remote file server and achieve performance similar to if the files were stored locally on the host.

SMB Direct makes possible a new class of file servers for enterprise environments, and the new File Server role in Windows Server 2012 demonstrates these capabilities in full. Such file servers experience minimal processor utilization for file storage processing and the ability to use high-speed RDMA-capable NICs including iWARP, InfiniBand, and RoCE. They can provide remote storage solutions equivalent in performance to Fibre Channel, but at a lower cost. They can use converged network fabrics in datacenters and are easy to provision, manage, and migrate.

- **SMB Directory Leasing.** Reduces round-trips from client to server because metadata is retrieved from a longer living directory cache. Cache coherency is maintained as clients are notified when directory information changes on the server. The result of using SMB Directory Leasing can be improved application response times, especially in in branch office scenarios.
- **SMB Encryption.** Enables end-to-end encryption of SMB data to protect network traffic from eavesdropping when travelling over untrusted networks. SMB Encryption can be configured either on a per-share basis or for the entire file server. It adds no cost overhead and removes the need for configuring IPsec and using specialized encryption hardware and WAN accelerators.
- **SMB Multichannel.** Allows aggregation of network bandwidth and network fault tolerance when multiple paths become available between the SMB client and the SMB server. The benefit of this that it allows server applications to

take full advantage of all available network bandwidth. The result is that your server applications become more resilient to network failure.

SMB Multichannel configures itself automatically by detecting and using multiple network paths when they become available. It can use NIC teaming failover but doesn't require such capability to work. Possible scenarios can include:

- Single NIC, but using Receive-Side Scaling (RSS) enables more processors to process the network traffic.
  - Multiple NICs with NIC Teaming, which allows SMB to use a single IP address per team.
  - Multiple NICs without NIC Teaming, where each NIC must have a unique IP address and is required for RDMA-capable NICs.
- 
- **SMB-specific Windows PowerShell cmdlets.** Provides Windows PowerShell cmdlets and WMI objects to manage SMB file servers and SMB file shares.
  - **SMB Scale Out.** Allows you to create file shares that provide simultaneous access to data files with direct I/O through all the nodes in your file server cluster. The result is improved use of network bandwidth and load balancing of the file server clients, and also optimization of performance for server applications. SMB Scale Out requires using CSV version 2, which is included in Windows Server 2012, and lets you seamlessly increase available bandwidth by adding cluster nodes.
  - **SMB3 Secure Dialect Negotiation.** Helps protect against man-in-the-middle attacks, where eavesdroppers attempt to downgrade the initially negotiated dialect and capabilities between an SMB client and an SMB server.
  - **SMB Transparent Failover.** Allows administrators to perform hardware or software maintenance of nodes in a clustered file server without interruption to server applications storing their data on file shares. If a hardware or software failure happens on a cluster node, SMB clients will reconnect transparently to another cluster node with no interruption for server applications storing data on these shares.

SMB Transparent Failover supports both planned failovers (such as maintenance operations) and unplanned failovers (for example, due to hardware failure). Implementing this feature requires the use of failover clustering, that both the server running the application and the file server are running Windows

Server 2012, and that the file shares on the file server have been shared for continuous availability.

- **VSS for SMB file shares.** Allows SMB clients and SMB servers supporting SMB 3.0 to take advantage of the Volume Shadow Copy Service (VSS) for SMB file shares.

The implementation of SMB 3 in Windows Server 2012 also includes new SMB performance counters that can provide detailed, per-share information about throughput, latency, and I/O per second (IOPS). These counters are designed for server applications like Hyper-V and SQL Server, which can store files on remote file shares to enable administrators to analyze the performance of the file shares where server application data is stored.

### **Benefits for Hyper-V**

These new capabilities of SMB 3 mean that Hyper-V hosts can store VM files, including the configuration, VHD, and snapshots in file shares on Windows Server 2012 file servers. You can implement this kind of solution for stand-alone Hyper-V servers. You also can implement it for clustered Hyper-V servers where file storage is used as shared storage for the cluster.

The benefits that enterprises can experience from these scenarios include simplified provisioning, management and migration of VM workloads, increased flexibility, and reduced cost.

### **SMB and Windows PowerShell**

You can view and manage many SMB 3 capabilities by using Windows PowerShell. To see what cmdlets are available for doing this, you can use the Get-Command cmdlet, as shown in Figure 2-17.

Capability	Name	ModuleName
CIM	Block-SmbShareAccess	SmbShare
CIM	Close-SmbOpenFile	SmbShare
CIM	Close-SmbSession	SmbShare
CIM	Get-SmbClientConfiguration	SmbShare
CIM	Get-SmbClientNetworkInterface	SmbShare
CIM	Get-SmbConnection	SmbShare
CIM	Get-SmbMapping	SmbShare
CIM	Get-SmbMultiChannelConnection	SmbShare
CIM	Get-SmbOpenFile	SmbShare
CIM	Get-SmbServerConfiguration	SmbShare
CIM	Get-SmbServerNetworkInterface	SmbShare
CIM	Get-SmbSession	SmbShare
CIM	Get-SmbShare	SmbShare
CIM	Get-SmbShareAccess	SmbShare
CIM	Get-SmbWitnessClient	SmbWitness
CIM	Grant-SmbShareAccess	SmbShare
CIM	Move-SmbWitnessClient	SmbWitness
CIM	New-SmbMapping	SmbShare
CIM	New-SmbShare	SmbShare
CIM	Remove-SmbMapping	SmbShare
CIM	Remove-SmbShare	SmbShare
CIM	Revoke-SmbShareAccess	SmbShare
CIM	Set-SmbClientConfiguration	SmbShare
CIM	Set-SmbServerConfiguration	SmbShare
CIM	Set-SmbShare	SmbShare
CIM	Unblock-SmbShareAccess	SmbShare
CIM	Update-SmbMultiChannelConnection	SmbShare

FIGURE 2-17 – Windows PowerShell cmdlets for managing SMB features and infrastructure

For example, Figure 2-18 shows how to use the Get-SMBServerConfiguration cmdlet to determine whether SMB Multichannel is enabled on a file server running Windows Server 2012.

## Improved VM import

The process used for importing VMs onto Hyper-V hosts has been improved in Windows Server 2012. The goal of these improvements is to help prevent configuration problems from happening that can prevent the import process from completing successfully.

In Hyper-V on Windows Server 2008 R2, when you imported a VM onto a host, the VM and all its files were copied to the host, but they weren't checked for possible configuration problems. However, Hyper-V on Windows Server 2012 now validates the configuration of VM files when they are imported to identify potential problems and, if possible, resolve them.

An additional enhancement to the process of importing VMs in Hyper-V on Windows Server 2012 is that now you can import a VM after manually copying the VM's files to the host. In other words, you don't have to export a VM from one host before you can import it into another host – you can simply copy the files from the first host to the second one and then initiate the import process.

## VHDX disk format

VHDX is the new default format for VHDs in Hyper-V in Windows Server 2012. This new format is designed to replace the older VHD format and has advanced capabilities that make it the ideal virtual disk format going forward for virtualized workloads. Some of the features of this new format include the following:

- It supports virtual disks up to 64 TB in size, so you'll be able to use it to virtualize even the largest database workloads and move them into the cloud.
- It aligns to megabyte boundaries to support large sector disks (4 KB sector disks), so you can take advantage of new low-cost commodity storage options.
- It uses large block sizes to provide better performance than the old format could provide.
- It includes a new log to protect from corruption due to power failure, which means the new format has much greater resiliency than the old format.
- You can embed custom user-defined metadata into VHDX files; for example, information about the service pack level of the guest operating system on the VM.