

Chapter 1

Introduction to Hyper-V

With the release of Windows Server 2008, Microsoft has included a built-in virtualization solution, Hyper-V. Hyper-V is a role of Windows Server 2008 that lets administrators create multiple virtual machines. A *virtual machine* is a separate, isolated environment that runs its own operating system and applications.

Virtual machine technology isn't new—it's been available from Microsoft in both Virtual PC and Virtual Server since late 2003 and from other vendors since the 1970s. By including it in the operating system, Microsoft has made an extremely feature-rich product available at no extra cost.

Hyper-V takes the concept of virtualization to the mainstream IT environment by including it in the operating system. Previous Microsoft virtualization solutions ran on top of the operating system—a significant difference from the way Hyper-V is designed. Inclusion in the operating system also provides a seamless management experience when paired with the System Center family of products.

In this chapter, we'll review the following elements of Hyper-V:

- ◆ Key scenarios for Hyper-V
- ◆ Hyper-V architecture
- ◆ Hyper-V features
- ◆ Hardware and software requirements

Scenarios for Hyper-V

Hyper-V was developed with a several key scenarios in mind. When Microsoft started developing Hyper-V, the development team spent a great deal of time meeting with customers who were using virtualization—small businesses, consultants who implement virtualization on behalf of their customers, and large companies with multimillion dollar IT budgets. The following key scenarios were developed as a result of those meetings; they represent customer needs, demands, and wants.

Server Consolidation

Systems are becoming increasingly powerful. A couple of years ago, it was rare to find a quad-processor server at a price most customers could afford. Now, with major processor manufacturers

providing multicore functionality, servers have more and more processing power. Multicore technology combines multiple processor cores onto a single die—enabling a single physical processor to run multiple threads of execution on separate cores. Virtualization and multicore technology work great together: If you’re combining multiple workloads onto a single server, you need to have as much processing power as possible. Multicore processors help provide the optimal platform for virtualization.

Businesses are increasingly likely to need multiple systems for a particular workload. Some workloads are incredibly complex, requiring multiple systems but not necessarily using all the power of the hardware. By taking advantage of virtualization, system administrators can provide a virtualized solution that better utilizes the host hardware—thus allowing administrators to get more out of their expenditure.

Workloads aren’t the only driving item behind virtualization. The power and cooling requirements of modern servers are also key driving factors. A fully loaded rack of servers can put out a significant amount of heat. (If you’ve ever stood behind one, you’re sure to agree—it’s a great place to warm up if you’ve been working in a cold server room.) All that heat has to come from somewhere. The rack requires significant power.

But for companies in high-rise buildings in the middle of major cities, getting additional power is incredibly difficult, if not impossible. In many cases, the buildings weren’t designed to have that much power coming in—and the companies can’t add more power without extensive retrofitting. By deploying virtualization, more workloads can be run on the same number of servers.

Testing and Development

For people working in a test or development role, virtualization is a key to being more productive. The ability to have a number of different virtual machines (VMs), each with its own operating system that’s ready to go at the click of a mouse, is a huge time-saver. Simply start up whichever VM has the operating system. You no longer need to install a clean operating system. Also, by using the snapshot functionality, users can quickly move between known states in the VM.

With Hyper-V’s rich Windows Management Interface (WMI) interfaces, testing can be started automatically. By scripting both Hyper-V and the operating system to be tested, testers can run a script that starts the VM, installs the latest build, and performs the necessary tests against it.

A Hyper-V virtual machine is also portable. A tester can work in the VM; if an issue is found, the tester can save the state of the VM (including the memory contents and processor state) and transfer it to the developer, who can restore the state at their convenience. Because the state of the VM is saved, the developer sees exactly what the tester saw.

Business Continuity and Disaster Recovery

Business continuity is the ability to keep mission-critical infrastructure up and running. Hyper-V provides two important features that enable business continuity: live backup and quick migration.

Live backup uses Microsoft Volume Shadow Services functionality to make a backup of the entire system without incurring any downtime, as well as provide a backup of the VM at a known good point in time. The system backup includes the state of all the running VMs. When a backup request comes from the host, Hyper-V is notified, and all the VMs running on that host

are placed into a state where they can be backed up without affecting current activity; they can then be restored at a later time.

Quick migration is the ability to move a VM from one host to another in a cluster using Microsoft Failover Cluster functionality. During a quick migration, you save the state of the VM, move storage connectivity from the source host to the target host, and then restore the state of the VM. Windows Server 2008 added support for the virtual-machine resource type to the Failover Clustering tool, enabling you to make a VM highly available using functionality included with the operating system. For more information about both of these features of Hyper-V, refer to Chapter 7, “Backup and Recovery.”

Disaster recovery is becoming a requirement for increasing numbers of businesses. With natural disasters such as Hurricane Katrina fresh in the minds of system administrators, enterprises are seeking ways to keep their businesses running throughout such events. You must consider more than just big disasters, though—small disasters or even simple configuration issues can lead to a mission-critical service being unavailable. Hyper-V includes support for geographically dispersed clusters (a new feature of Windows Server 2008).

Dynamic IT

Microsoft’s idea of a dynamic IT infrastructure involves self-managing dynamic systems—systems that adjust automatically to the workload they’re running. By using Hyper-V in conjunction with the systems-management functionality present in the System Center family of products, enterprises can take advantage of the benefits of virtualization to meet the demands of a rapidly changing environment.

Now that we’ve covered Hyper-V’s key targeted scenarios, let’s review the architecture of Hyper-V to see how Microsoft has implemented support for them.

Architecture

Before we examine the architecture of Windows Server 2008 with the Hyper-V role, it’s useful to understand how Windows Server 2008 works without this role.

As shown in Figure 1.1, Windows Server 2008 operates in both kernel mode and user mode. Kernel mode (also known as Ring 0) is where the Windows kernel lives, as well as all the device drivers for the hardware installed in the system. User mode (Ring 3) is where applications are run. This ring separation is a key feature of the x86 architecture—it means that a rogue application shouldn’t be able to take down the operating system.

A default installation of Windows Server 2008 doesn’t include any active roles or features. Windows Server 2008 was designed to be as secure as possible. As part of the development process, Microsoft worked with and received feedback from many users about how they deploy servers. A frequent customer request was an easy way to deploy a server to perform a particular task—for example, a file server or print server. That’s where the concept of a role or feature came into play.

Now that you understand the meaning of roles and features in Windows Server 2008, let’s talk about the Hyper-V role. We’ll cover installation of the role in Chapter 2, “Installing Hyper-V and Server Core.”

FIGURE 1.1
Simplified architecture for a clean install of Windows Server 2008

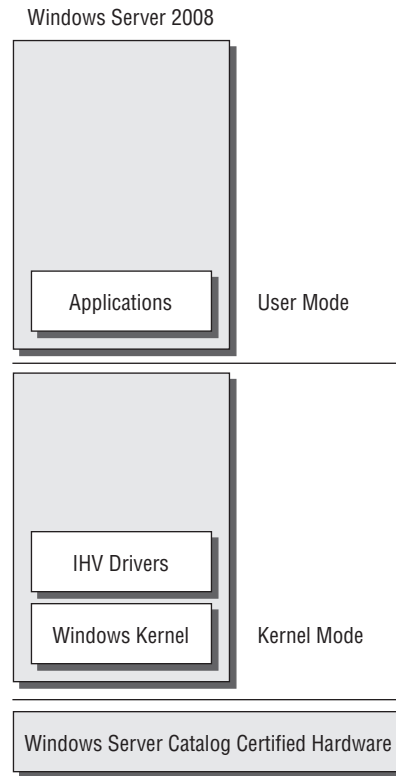


Figure 1.2 shows that once the role is installed, some pretty significant changes happen to the installed copy of Windows Server 2008.

A *role* in Windows Server 2008 is a task for the server, whereas a *feature* can (and often does) supplement a role. A great example of this role/feature distinction is a web server. IIS functionality is a role of Windows Server 2008, and features that go hand in hand with IIS include Network Load Balancing and Windows PowerShell. Each of those features can be installed on an as-needed basis.

Looks quite a bit different, doesn't it? Let's break down each of the changes.

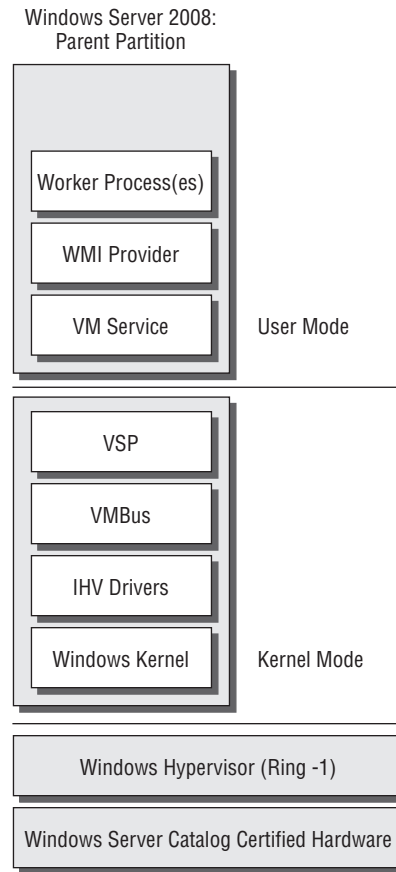
Parent Partition

The installation of Windows is now running on top of the Windows hypervisor, which we'll describe later. One of the side effects of running on top of the hypervisor is that the installation is technically a VM. We'll refer to this as the *parent partition*.

The parent partition has two special features:

- ◆ It contains all the hardware device drivers, as well as supporting files, for the other VMs. We'll look at the functions of each of those drivers later in the chapter.
- ◆ It has direct access to all the hardware in the system. In conjunction with the virtualization service providers, the parent partition executes I/O requests on behalf of the VM—sending disk traffic out over a fibre channel controller, for example.

FIGURE 1.2
Simplified architecture for Windows Server 2008 with the Hyper-V role added



The following best practices provide a secure and stable parent partition, which is critical to the VMs running on the host:

- ◆ Don't run any other applications or services in the parent partition. This may seem like basic knowledge for system administrators, but it's especially crucial when you're running multiple VMs. In addition to possibly decreasing stability, running multiple roles, features, or applications in the parent partition limits the amount of resources that can otherwise be allocated to VMs.
- ◆ Use Windows Server 2008 in the Core role as the parent partition. We'll discuss Windows Server Core in Chapter 2.

WINDOWS HYPERVISOR

The Windows hypervisor is the basis for Hyper-V. At its heart, the hypervisor has only a few simple tasks: creating and tearing down partitions. (A partition is also known as the basis for a VM) and ensuring strong separation between the partitions. It doesn't sound like much, but the hypervisor is one of the most critical portions of Hyper-V. That's why development of the

hypervisor followed the Microsoft Security Design Lifecycle process so closely—if the hypervisor is compromised, the entire system can be taken over, because the hypervisor runs in the most privileged mode offered by the x86 architecture.

One of Microsoft’s design goals was to make the Microsoft hypervisor as small as possible. Doing so offered two advantages:

- ◆ The Trusted Computing Base (TCB) is smaller. The TCB is the sum of all the parts of the system that are critical to security. Ensuring that the hypervisor is small reduces its potential attack vectors.
- ◆ The hypervisor imparts less overhead on the system. Because all VMs (as well as the parent partition) are running on top of the hypervisor, performance becomes a concern. The goal is to minimize the hypervisor’s overhead.

KERNEL-MODE DRIVERS

A Windows kernel-mode driver is one of two types of drivers in Windows. Kernel-mode drivers execute in Ring 0. Because this type of driver is executing in kernel mode, it’s crucial that these drivers be as secure as possible: An insecure driver, or a crash in the driver, can compromise the entire system.

Hyper-V adds two kernel-mode drivers:

VMBus VMBus is a high-speed in-memory bus that was developed for Hyper-V. VMBus acts as the bus for all I/O traffic that takes place between the VMs and the parent partition. VMBus works closely with the virtualization service provider and virtualization service client, which we’ll describe later in this chapter.

Virtualization Service Provider (VSP) The Virtualization Service Provider (VSP) enables VMs to securely share the underlying physical hardware by initiating I/O on behalf of all VMs running on the system. It works in conjunction with the hardware vendor drivers in the parent partition—which means that no special “virtualization” drivers are necessary. If a driver is certified for Windows Server 2008, it should work as expected with Hyper-V. Each class of device has a VSP present—for example, a default installation of Hyper-V has a networking VSP as well as a storage VSP. The VSPs communicate with the matching Virtualization Service Client (VSC) that runs in the VM over VMBus. We’ll cover the VSC when we look at the different types of VMs.

USER-MODE APPLICATIONS

User-mode applications are, strangely enough, applications that run in user mode. They execute in Ring 3, which is where all unprivileged instructions are run. Many of the applications that run in Windows are user-mode applications—for example, the copy of Notepad that you use to look at a text file is executing in user mode.

Hyper-V has a number of different user-mode applications:

Virtual Machine Management Service (VMMS) The VMMS acts as the single point of interaction for all incoming management requests. It interacts with a number of processes, two of which we’ll refer to here.

WMI providers Hyper-V has a rich set of WMI interfaces. They provide a way to manage the state and health of the VMs as well as get settings information and some performance information. All the WMI interfaces are fully documented on <http://msdn.microsoft.com>.

Worker processes When a VM is started up, a worker process is created. The worker process represents the actions that are taking place in the virtual processor, as well as all emulated devices and the virtual motherboard. Each VM that is running on a host has a worker process.

Now that we've shown you what's happening in the parent partition, let's look at the VMs. After you create a VM and power it on, you can install a wide variety of x86/x64-based operating systems. Even though these are VMs, they can run the same operating systems without modification as a physical computer. But operating systems that are supported by Microsoft include new synthetic drivers, which work in conjunction with the matching VSP running in the parent partition.

Let's first examine how a fully emulated operating system handles I/O.

Virtual Machine

A VM can have two different types of devices: emulated and synthetic. Although synthetic devices are encouraged due to their superior performance, they aren't available for all operating systems. Emulated devices are present in Hyper-V mainly for backward compatibility with non-supported operating systems. VMs running certain distributions of Linux have synthetic device support as well. Let's examine each type of device.

EMULATED DEVICES

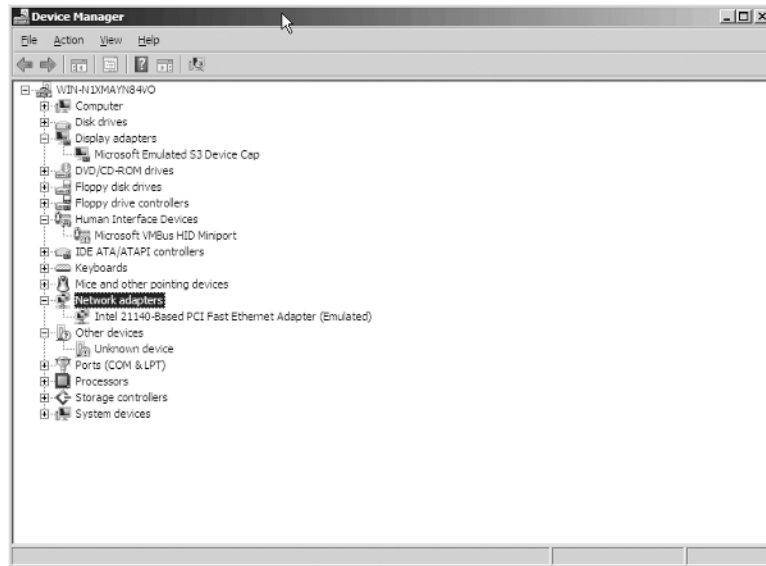
Emulated devices in Hyper-V exist primarily for backward compatibility with older operating systems. In an ideal world, all applications would run on the latest version of the operating system they were designed for, but that's far from reality. Many companies have systems in production that run on older copies of operating systems because one of their applications doesn't run on anything newer. An older operating system may not be supported under Hyper-V, which means it can't take advantage of the high-performance I/O. That's not a total loss, however: If you consolidate those older systems onto a newer Hyper-V host, the advantages of moving to a more up-to-date hardware platform can provide a performance boost.

Emulated devices have another key role. During the installation of the VM, operating systems don't have support for the synthetic devices that may be installed in the VM. For that reason, you must use emulated devices—otherwise, the operating-system installation can't function. For Hyper-V, it's easy to move from emulated to synthetic devices.

The emulated devices presented to a VM are chosen for their high degree of compatibility across a wide range of operating systems and in-box driver support. As you can see in Figure 1.3, the video card is based on an S3 video card, and the network card is an Intel 21140-based Ethernet adapter.

Emulated devices under Hyper-V don't perform as well as the new synthetic devices. Thanks to part of the work that was done to harden the entire virtualization stack, emulated devices execute in the worker process—specifically, in user mode in the parent partition.

FIGURE 1.3
Device Manager
for a Windows
Server 2008 virtual
machine, showing
emulated devices



How does I/O happen with emulated devices?

Figure 1.4 goes into considerable detail about how emulated storage requests are handled. Emulated networking is handled in a similar fashion. I want to point out a few specific items:

- ◆ Context switches are used. A *context switch* is a switch from executing a particular processor instruction in kernel mode to user mode. When paired with virtualization, a context switch is an “expensive” operation. There’s no money involved, but the CPU cost for such an operation is very high. That time could be spent doing other tasks.
- ◆ The path that the data packet traverses is long, especially compared to the synthetic case (which we’ll review next).
- ◆ The path illustrated in the figure is repeated hundreds of times for a 10 kilobyte write to disk. Imagine if you’re doing a large SQL transaction that involved writing hundreds of megabytes to disk, or running a popular website being served up from IIS running in the VM. You can see that it won’t scale well.

SYNTHETIC DEVICE DRIVERS

Synthetic devices provide much higher performance than their emulated counterparts. By taking advantage of VMBus, synthetic devices can execute I/O transactions at a much faster rate.

Synthetic devices, such as the Microsoft Virtual Machine Bus Network Adapter shown in Figure 1.5, don’t have real-world counterparts. They are purely virtual devices that function only with Hyper-V—loading the drivers on a physical system does nothing. These new synthetic devices rely on VMBus.

FIGURE 1.4
I/O for emulated
storage devices

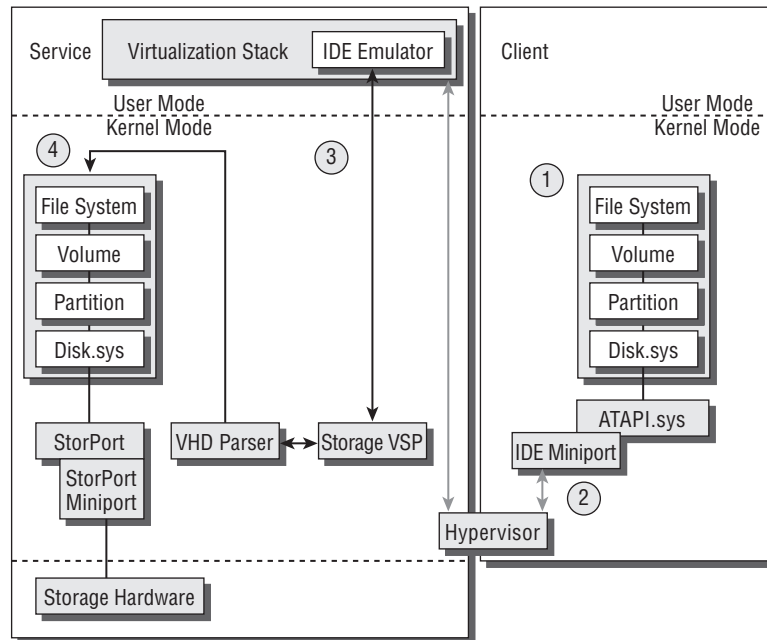
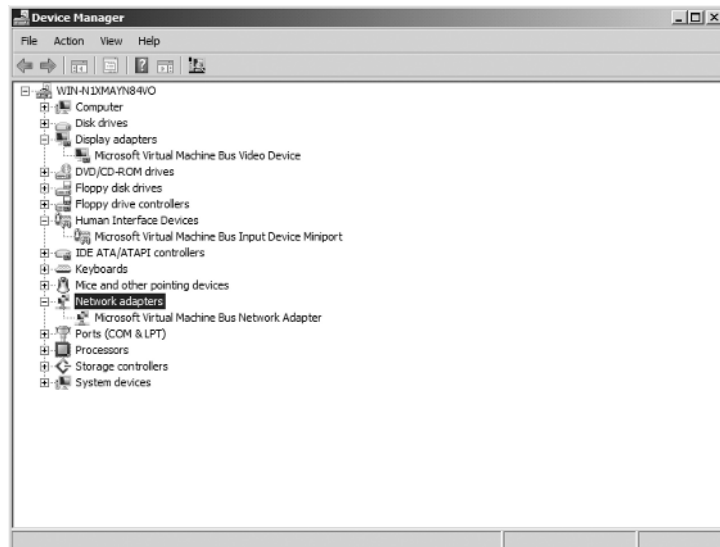


FIGURE 1.5
Device Manager
for a Windows
Server 2008 virtual
machine, showing
synthetic devices

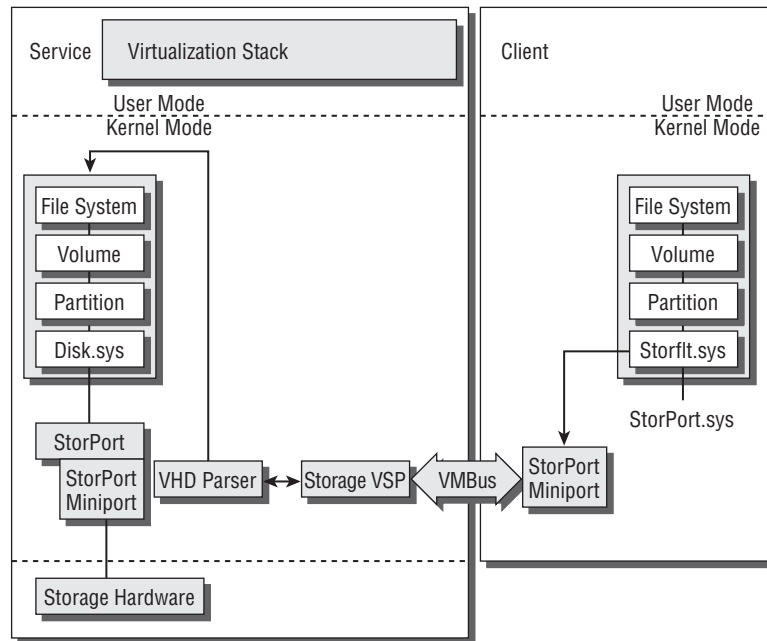


Synthetic device drivers are available only for operating systems that are supported by Microsoft. (For reference, a list of supported operating systems for Hyper-V is available at www.microsoft.com/virtualization.) If you're running an operating system in the VM that isn't supported by Microsoft, you'll need to use the emulated devices in the VM.

Much like the emulated storage request chart shown earlier in Figure 1.4, Figure 1.6 presents a lot of data. Here are a few key differences:

- ◆ In the beginning, the data path is similar to the emulated data path. However, the synthetic storage device in Hyper-V is a SCSI-based device—so the last driver it hits before getting put on VMBus is the StorPort driver.
- ◆ When a packet makes it to the miniport driver, it's put on VMBus for transport to the Storage VSP in the parent partition. Because VMBus is a kernel-mode driver, no context switches are necessary.
- ◆ After the data packet crosses over to the parent partition, the correct destination is determined by the VSP, which routes the packet to the correct device. In Figure 1.6, the destination is a virtual hard disk (VHD) file.

FIGURE 1.6
I/O for synthetic
storage devices
using VMBus



Installing Synthetic Device Drivers

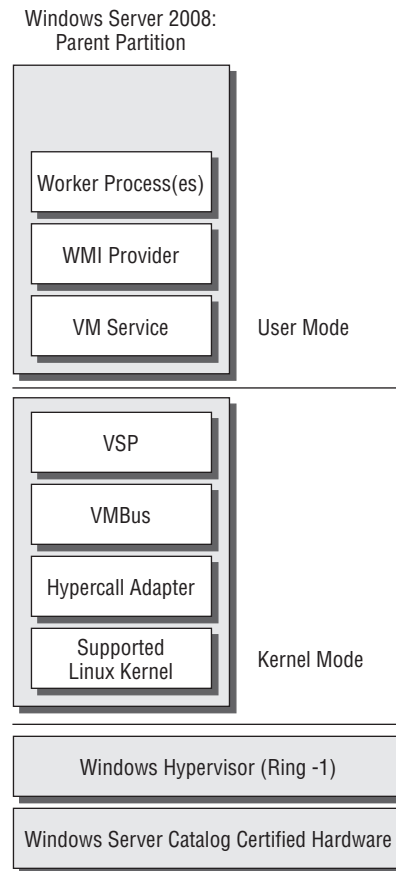
It's easy to install synthetic device drivers in the VM. After you've installed the operating system, select Action ➤ Insert Integration Services Setup Disk. An installer launches and automatically installs the drivers for you. When you reboot, the VM can take advantage of the new architecture.

NOTE A special synthetic driver deals with the boot process: Optimized Boot Performance. Because the synthetic drivers rely on VMBus, you can't boot off hard drives that are connected to the SCSI controller. All isn't lost—during the boot process, after the VMBus driver is loaded, all the IDE boot traffic is automatically routed through the same infrastructure that is used for SCSI traffic. This means the boot process and all disk traffic (reads and writes) perform at the same accelerated speed.

LINUX DEVICE DRIVERS

No, that's not a typo—certain distributions of Linux are supported under Hyper-V. Not only is the operating system supported, but a full set of device drivers also enable synthetic device support under Linux (see Figure 1.7). The drivers include the Hypercall adapter—a thin piece of software that runs in the Linux VM and increases performance by translating certain instructions to a format that Hyper-V can understand.

FIGURE 1.7
Synthetic device
support under
Linux



Features

Now that we've gone over both the scenarios and architecture of Hyper-V, let's dive into some of the features of Microsoft's virtualization platform:

32-bit (x86) and 64-bit (x64) VMs Hyper-V provides support for both 32-bit as well as 64-bit VMs. This lets users provision both architectures on the same platform, easing the transition to 64-bit and providing legacy 32-bit operating systems.

Large memory support (64 GB) within VMs With support for up to 64 GB of RAM, Hyper-V scales out to run the vast majority of enterprise-class workloads. Hyper-V can also use up to a total of 1 terabyte (TB) of RAM on the host.

SMP virtual machines Symmetric Multi Processor (SMP) support allows VMs to recognize and utilize four virtual processors. As a result, server applications running in a Hyper-V VM take full advantage of the host system's processing power.

Integrated cluster support for quick migration and high availability (HA) Windows Server 2008 Hyper-V and HA go hand in hand. As we'll discuss later in the book (see Chapter 8, "High Availability"), it's easy to create a failover cluster of VM hosts that your VMs can live on. After you set up the failover cluster, you can quickly and easily move a VM from one host to the other from the Failover Cluster Manager or from other management tools (such as System Center Virtual Machine Manager).

Volume Shadow Service integration for data protection Hyper-V includes a Volume Shadow Services (VSS) provider. As we discussed earlier, in the list of scenarios, VSS lets backup applications prepare the system for a backup without requiring the applications (or VMs) to be shut down.

Pass-through high-performance disk access for VMs When a physical volume is connected directly to the VM, disk I/O-intensive workloads can perform at their peak. If the Windows Server 2008 system can see the volume in the Disk Management control panel, the volume can be passed through to the VM.

Although you'll see faster performance with pass-through disk access, certain features (such as snapshots, differencing disks, and host-side backup) that you get from using a VHD file aren't available with pass-through disks.

VM snapshots Snapshots let administrators capture a point in time for the VM (including state, data, and configuration). You can then roll back to that snapshot at a later point in time or split from that snapshot to go down a different path. The snapshot is a key feature for the test and development scenario, because it lets users easily maintain separate points in time. For example, a user may install an operating system inside a VM and take a snapshot. The user can perform a number of tasks and then take a second snapshot. Then, the user can return to either of those snapshots later, saving configuration time and effort.

New hardware-sharing architecture (VSP/VSC/VMBus) By using the new VMBus communication protocol for all virtual devices, Hyper-V can provide higher levels of performance than were previously seen with Microsoft virtualization products.

Robust networking: VLANs and NLB Virtual Local Area Network (VLAN) tagging—also referred to as the IEEE standard 802.1q—provides a secure method for multiple networks to use the same physical media. Hyper-V supports VLAN tagging (802.1q) on the virtual network interfaces and specifies a VLAN tag for the network interface.

Network Load Balancing (NLB) support in Hyper-V allows VMs to participate in an NLB cluster. An NLB cluster is different from a failover cluster, such as those used for VM quick migration. NLB clusters are configured with front-end nodes that handle all incoming traffic and route it to multiple servers on the back-end.

DMTF standard for WMI management interface The Distributed Management Task Force (DMTF) is a standards body that provides a uniform set of standards for the management of IT environments. Microsoft has worked closely with the DMTF to ensure that all the management interfaces for Hyper-V adhere to the standards, allowing management tools from multiple vendors to manage the system.

Support for full or Server Core installations Hyper-V can run on a full installation of Windows Server 2008 as well as the Server Core option. We'll discuss Server Core in more depth later.

Now that we've gone through the list of Hyper-V features, let's look at the system requirements.

ADVANTAGES OVER VIRTUAL SERVER

Windows Server 2008 Hyper-V has a number of advantages over Virtual Server 2005 R2 SP1:

- ◆ Support for SMP and 64-bit VMs. Virtual Server was limited to 32-bit uni-processor virtual machines.
- ◆ Support for more than 3.6 GB of RAM per VM.
- ◆ Support for mapping a logical unit number (LUN) directly to a VM.
- ◆ Increased performance from VSP/VSC architecture.
- ◆ Hyper-V management via a MMC-based interface instead of the web-based console.

However, it's impossible for users who have only 32-bit hardware in their environment to move to Hyper-V (because it's a feature of the 64-bit version of Windows Server 2008).

Requirements

Because Hyper-V is included as a role of Windows Server 2008 x64 Edition, it inherits the same hardware requirements. However, a few areas require special attention for Hyper-V.

Hardware Requirements

Some of the requirements for Hyper-V are hard requirements, such as the type of processor, whereas others are best practices to ensure that Hyper-V performs optimally.

PROCESSOR

Hyper-V requires a 64-bit capable processor with two separate extensions: hardware-assisted virtualization and data-execution prevention.

Hardware-assisted virtualization is given a different name by each vendor—Intel calls it Virtualization Technology (VT), and AMD calls it AMD Virtualization (AMD-V). Almost all processors now ship with those features present, but check with your processor manufacturer to make sure.

Although the functionality is required in the processor, it's also required to be enabled in the BIOS. Each system manufacturer has a different way of exposing the functionality, as well as a different name for it. However, most, if not all, manufacturers provide a way to enable or disable it in the BIOS. You can enable it in the BIOS, but some systems don't enable the feature unless there's a hard-power cycle—shutting off the system completely, for example. *We recommend that the system be completely powered off.*

Data-execution prevention (DEP) goes by different names depending on the processor manufacturer—on the Intel platform, it's called eXecute Disable (XD); and AMD refers to it as No eXecute (NX). DEP helps protect your system against malware and improperly written programs by monitoring memory reads and writes to ensure that memory pages marked as *Data* aren't executed. Because you'll be running multiple VMs on a single system, ensuring stability of the hosting system is crucial.

STORAGE

As we talked about earlier, Hyper-V's architecture lets you use standard Windows device drivers in conjunction with the VSP/VSC architecture. As such, any of the storage devices listed in the Windows Server Catalog will work with Hyper-V. These include SCSI, SAS, fibre channel, and iSCSI—if there's a driver for it, Hyper-V can use it. Of course, you'll want to take some considerations into account when planning the ideal Hyper-V host. We'll talk about those more in Chapter 10, "Automating Common Tasks."

Here are some of the areas where extra attention is necessary:

Multiple spindles and I/O paths Most disk-intensive workloads, such as database servers, need multiple spindles to achieve high performance. Hyper-V's storage architecture enables those workloads to be virtualized without the traditional performance penalty. When multiple disk-intensive workloads share the same disk infrastructure, they can quickly slow to a crawl.

Having multiple disks (as well as multiple I/O paths) is highly recommended for disk-intensive workloads. Even two workloads sharing a host bus adapter with a single fibre channel can saturate the controller, leading to decreased performance. Having multiple controllers also can provide redundancy for critical workloads.

Disk configurations for optimal performance Hyper-V has a number of different ways to store the VM's data, each with its own pros and cons:

- ◆ Pass-through disks:
 - ◆ Pros: Pass-through disks generally provide the highest performance. The VM writes directly to the disk volume without any intermediate layer, so you can see near-native levels of performance.
 - ◆ Cons: Maintaining the storage volumes for each VM can be extremely challenging, especially for large enterprise deployments.

- ◆ Fixed virtual hard disks:
 - ◆ Pros: These are the best choice for production environments using VHD files. Because you allocate all the disk space when you create the VHD file, you don't see the expansion penalty that occurs with the dynamically expanding VHD.
 - ◆ Cons: Because all the space for the VHD is allocated at creation, the VHD file can be large.
- ◆ Dynamic virtual hard disks:
 - ◆ Pros: A dynamically expanding VHD expands on demand, saving space on the system until it's needed. Disks can remain small.
 - ◆ Cons: There is a small performance penalty when a disk is expanded. If large amounts of data are being written, the disk will need to be expanded multiple times.

Snapshots Snapshots are extremely useful in the test and development environment. However, what can be helpful in one environment can be harmful in another. You shouldn't use snapshots in a production environment because rolling back to a previous state without taking the proper precautions can mean data loss!

NETWORKING

Much like storage, networking with Hyper-V inherits the rich driver support of Windows Server 2008. Many of the caveats for storage apply to networking as well—ensure that multiple NICs are present so a single interface doesn't become the bottleneck.

The following list identifies areas where you should pay special attention with networking:

- ◆ Hyper-V supports Ethernet network adapters, including 10, 100, 1000, and even 10Gb-E network adapters. Hyper-V can't use ATM or Token Ring adapters, nor can it use wireless (802.11) adapters to provide network access to the VMs.
- ◆ During the Hyper-V role installation (which we'll cover in Chapter 2), you can create a virtual network for each network adapter in your system.
- ◆ We recommend that you set aside a single NIC to manage the host. That NIC shouldn't be used for any VMs (no virtual switch should be associated with it). Alternatively, you can use out-of-band management tools to manage the host. Such tools typically use an onboard management port to provide an interface to the system.

Software Requirements

Hyper-V is a feature of Windows Server 2008 x64 Edition only. There's no support for Hyper-V in the x86 (aka 32-bit) Edition or the Itanium versions of Windows Server 2008. The x64 Edition is required for a couple of reasons:

Kernel address space The 64-bit version of Windows Server 2008 provides a much larger kernel address space as compared to the 32-bit edition. This directly translates into the support of larger processes, which is crucial for virtualization.

Large amount of host memory Hyper-V supports up to 1 TB of RAM on the host. x86 versions of Windows Server 2008 support only up to 64 GB of RAM on the host, which would severely limit the number of VMs you could run.

We’re frequently asked to explain the differences with Hyper-V between versions of Windows Server 2008. There’s no difference—the features of Hyper-V are the same, regardless of whether you’re running the Standard, Enterprise, or Datacenter product. However, differences in the versions of Windows Server 2008 affect key virtualization scenarios:

Processor sockets Windows Server 2008 Standard Edition is limited to four sockets, whereas Enterprise Edition supports eight sockets.

Memory Windows Server 2008 Standard Edition supports up to 32 GB of RAM, and Windows Server 2008 Enterprise Edition supports up to 2 TB of RAM.

Failover clustering Windows Server 2008 Standard Edition doesn’t include the failover-clustering functionality required for quick migration.

Virtual image use rights As summarized in Table 1.1, Windows Server 2008 includes the rights to run virtual images of the installed operating system. The number of those virtual images is tied to the edition.

TABLE 1.1 Virtual Image Usage Rights

EDITION	VIRTUAL IMAGE USAGE RIGHTS
Standard Edition	1
Enterprise Edition	4
Datacenter Edition	Unlimited

Summary

Are you intimidated yet? In this chapter, we’ve provided a great deal of information about Hyper-V. From its scenarios to its architecture to its features, we’ve laid the groundwork. In the upcoming chapters, we’ll go into depth about many of the items we touched on here. Keep reading to find out why you should deploy Hyper-V in your environment.