Virtualization Explained:
Definitions You Need to Know

SearchServerVirtualization.com

By Stephen J. Bigelow
with Edward L. Haletky

Virtualization Explained:
Definitions You Need to Know

Table of contents

AMD-V

IT professionals should understand which AMD processors have AMD-V technology. But first, a quick history lesson:

Traditional computing does not allow software to readily share hardware resources. Virtualization overcomes that limitation and allows two or more virtual machines (VMs) to share the computing resources of a single physical server, and the VM monitor is central to any virtualization deployment. It provides the abstraction that separates software from the underlying hardware and manipulates the virtual workloads so that each workload can share the common processor, memory, I/O and local storage. Early virtualization efforts relied on software emulation to replace hardware functionality. But software emulation is a slow and inefficient process.

Many virtualization tasks were handled through software, so VM behavior and resource control were often poor, resulting in unacceptable VM performance on the server. By early 2005, processors still lacked the internal microcode to handle intensive virtualization tasks in hardware. Both Intel Corp. and Advanced Micro Dynamics Inc. (AMD) addressed this problem by creating a new set of processor extensions (similar to MMX, 3DNow! and others) that could offload the repetitive and inefficient work from the software. By handling these tasks through processor extensions, traps and emulation of virtualization, tasks through the operating system were essentially eliminated, vastly improving VM performance on the physical server.

AMD Virtualization (AMD-V) technology was first announced in 2004 and added to AMD's Pacifica 64-bit x86 processor designs. By 2006, AMD's Athlon 64 X2 and Athlon 64 FX processors appeared with AMD-V technology, and today, the technology is available on Turion 64 X2, second- and third-generation Opteron, Phenom and Phenom II processors.

AMD performance with hardware-assisted virtualization centers on memory management that builds on prior Direct Connect Architecture. In effect, the physical processor and the guest VM can communicate directly. This direct communication reduces the overhead often encountered with emulation and improves the way that memory space is handled, because the processor works with the virtual instance.

AMD-V introduced Rapid Virtualization Indexing (RVI) features that allow VMs to manage memory directly. RVI reduces the number of processing cycles needed by the hypervisor that would otherwise be wasted handling memory-related operations. In addition, RVI lets the processor switch between numerous virtual guest instances very quickly, further enhancing virtualization performance. Memory handling and switching is also boosted by tagged Translation Lookaside Buffer (TLB) capabilities that map memory space to the individual VM. This reduces memory management and speeds up the switching process between VMs.

AMD-V-supported processors also provide extended migration capabilities that allow virtualization platforms to easily and quickly migrate VMs across servers that run these processors. It's important to note that extended migration is not necessarily compatible with Intel processors, and this may cause performance issues (or outright failures) when migrating VMs between servers with different processor makers.

—Stephen J. Bigelow, Senior Technology Writer

C

Capacity planning

A capacity planning strategy attempts to predict the future load utilization of an IT environment -- which could include servers, storage or networks -- and then establishes a plan to ensure that enough computing resources will actually be available to handle the projected load. It has been an important practice for years, but the shift to virtualization has underscored a renewed need to have a capacity planning strategy.

With virtualization, the general approach to a capacity planning strategy is the same as for traditional physical environments. First, computing resources are monitored over time. Monitoring tools may determine CPU usage, memory usage, I/O loading, disk storage demands, network bandwidth and myriad other factors. Utilization trends are evaluated in the context of business goals to identify specific needs, which can then be translated into action items like server upgrades or additional server purchases.

For example, consider a transactional database for online orders running on avirtual machine (VM) hosted on a physical server. Suppose that monitoring reveals ongoing increases in CPU utilization and network latency at the server. This can suggest business growth through more orders, but it also indicates the eventual need for greater processing power to manage increasing volume, keep latency low and maintain a smooth user experience. This in turn may justify a server upgrade or replacement.

Capacity planning in a virtual data center is also driven by the need to manage new VM deployments. Since VMs are easily created and a physical server can host numerous VMs, careless IT administrators can quickly overwhelm their current computing capacity -- a phenomenon known as VM sprawl.

An excessive number of VM workloads (and poor VM workload distribution among servers) can easily choke off a server's performance. This not only compromises the performance of every VM on the server but can also cause stability problems and crash a VM -- or even the entire server and all of its VMs. In virtualization, a capacity planning strategy is often coupled with strong policies to ensure that each new VM is indeed necessary for the business and that adequate computing capacity is available to accommodate it in advance.

Furthermore, overloading a server may leave inadequate computing capacity in reserve, and the server might no longer be able to accept VMs failed over from other faulty servers. The result is invariably poor application availability. It's a common practice to load virtual servers to between 50% and 80% of their total computing capacity, leaving the remaining capacity free for failovers.

—Stephen J. Bigelow

Clone

A VM clone is a complete duplicate of a particular virtual machine (VM) at that precise moment in time.

Cloning a virtual machine is not intended for backup, disaster recovery (DR) or other data protection purposes. The most common use of a VM clone is in the mass deployment of standardized VMs. Cloning a virtual machine that is appropriate for a new workload and tailoring it for specific needs is more efficient than creating new VMs manually and then installing their operating systems and applications.

VM cloning is also useful for test and development. A workload may be cloned from the production environment and placed on another server in the lab, which allows development personnel to work with a real workload without interfering with production.

The traditional process of cloning a virtual machine normally starts by quiesceing and then creating a VM snapshot backup of the source. It is not typically necessary to snapshot the source VM's memory, so disabling that option can usually speed up the snapshot process. Once the snapshot is complete, the original VM should leave the quiesced state and return to normal operation with little (if any) perceivable disruption to users.

In a VMware environment, the next step is to create a new VM using the virtual infrastructure (VI) Client. When configuring the resources for this new VM, do not assign a physical NIC. Assign a virtual switch instead, and stick with the default hard drive size. Do not activate the new VM just yet.

Now, use the command line interface to delete the virtual disk files for the new VM. Open the folder where the new VM resides and delete all of the VMDK files, leaving you with a new (but empty) VM.

At this point, go to the folder containing the snapshot of the original VM. Copy the snapshot to the new VM folder using the command line interface and the "vmkfstools" command. That places the content of your original VM into the new VM instance. The VI Client will allow administrators to activate the new VM. As long as the new VM is not running on a real physical NIC, it will not interfere with the original production VM.

Perform a little housekeeping by deleting the snapshot, which is no longer needed. Deleting the snapshot frees up unneeded disk space and minimizes file clutter. The VM clone will now function exactly like the original workload, and it can be migrated to other servers if needed.

Today, virtualization platforms have automated much of the process of cloning a virtual machine.

—Stephen J. Bigelow

Clustered file system

Most operating systems include a file system that allows a computer to organize the files and data stored on that system. A file system manages the data storage and retrieval between the operating system (OS) and the storage sub-system. For example, a single personal computer with Windows will usually implement NTFS or FAT on the local disk, while Linux supports a variety of file systems such as ext2, ext3, ext4, XFS, JFS and others.

Storage access is complicated when multiple computers are organized into a cluster, so a Windows clustered file system or a Linux clustered file system may be needed to organize data storage and access across all of the cluster nodes.

The need for a clustered file system depends on the nature of the cluster that is deployed. Many clusters simply provide redundancy. If one server node fails, another server node in the cluster will take over and maintain availability of the workload. Each server operates autonomously in this common configuration, so clustering the file system is usually not necessary.

Some server clusters operate collaboratively, however, and aggregate the processing power of each server to tackle more demanding computing tasks. When servers work together collectively, they must often share access to the same files and data at the same time -- which is impossible for individual file systems. This situation requires deployment of a Windows clustered file system or a Linux clustered file system, depending on your environment.

The clustered file system arbitrates and controls each node's access to the shared storage resource (such as a SAN), preventing more than one node from writing data at the same time -- a catastrophic phenomenon called a "write collision." The actual storage is not important, and clustered file systems can support block-level storage, including SCSI, iSCSI, ATA over Ethernet, Fibre Channel, and Fibre Channel over Ethernet.

Not only does a clustered file system manage simultaneous file access to nodes, it also recognizes and handles node failures that threaten disk consistency. When a node writes bad data or stops sending its changes, it must be fenced off from the other nodes.

Windows clustered file systems can generally isolate a defective node logically, while Linux clustered file systems will employ utilities like STONITH -- which stands for "Shoot The Other Node In The Head" -- to actually power down the questionable node.

There are numerous clustered file systems available, and the choice normally depends on the underlying operating system. For example, Linux uses the Global File System (GFS), Oracle provides the Oracle Clustered File System (OCFS) and OCFS2, VMware offers the Virtual Machine File System (VMFS) for ESX servers and Sun Microsystems uses Lustre. Native Windows clustered file systems are available in Microsoft products such as Windows Server 2008 R2, and third-party clustered file systems for Windows are also available.

—Stephen J. Bigelow

Guests and hosts

A virtual machine (VM) typically has two components: the **host server** and the **guest virtual machine**. The host server is the underlying hardware that provides computing resources, such as processing power, memory, disk and network I/O, and so on. The guest virtual machine is a completely separate and independent instance of an operating system and application software. Guests are the virtual workloads that reside on a host server and share in that server's computing resources.

Virtualization platforms such as VMware vSphere, Microsoft Hyper-V and Citrix Systems' XenServer install a layer of abstraction between the host server and guest virtual machine. This is the virtual machine monitor (VMM), or hypervisor, that isolates each guest from another, enabling multiple guests to reside and operate on the host simultaneously. Abstraction also removes the direct relationship between software and the underlying hardware, so it is possible for a server to host guest virtual machines with different operating systems and application software requirements. For example, one VM with a Windows guest operating system can readily co-exist with a second VM running a Linux guest operating system on the same physical host server at the same time.

The only requirement is that a host server must meet or exceed the minimum hardware requirements for its guest VMs (workload). Since most traditional nonvirtualized servers are woefully underutilized, a host should supply ample computing resources for several guest virtual machines. But as additional guests are added to the host, computing resources -- such as memory or processing capacity -- may be exhausted. This can create performance problems in one or more guests or can crash guests -- even the entire host -- leading to numerous simultaneous workload outages, which can undermine user productivity. Guests are usually distributed among several available hosts to better distribute each workload's computing demands.

Although most virtualization relies on the concept of "isolated guest instances," some forms of virtualization do not create isolated instances. Instead they create separate areas within the same operating system. Examples of operating system virtualization" or OS virtualization, include Solaris Zones, Parallels Virtuozzo, Linux-VServer, OpenVZ and others. A "zoned" approach presents certain pros and cons. It is more efficient because there is only one iteration of the OS installed on the host server. However, every guest zone must run an application that supports the operating system; you cannot have one guest run Windows and another run Linux. Further, a security exploit in the operating system will compromise all of the zones, where an exploit in a fully isolated guest virtual machine would only threaten that one guest.

—Stephen J. Bigelow

I

Intel VT

Intel developed its virtualization hardware technology to provide hardware assistance to processors running virtualization platforms. To appreciate Intel Virtualization Technology (Intel VT) and its variations used in processors, chipsets and I/O, it's important to understand the traditional problems that limit virtualization performance in the first place.

Virtualization allows IT professionals to operate multiple virtual machines on one physical server. This requires the single physical system to create and maintain multiple virtual systems. The core of virtualization is the virtual machine monitor(VMM), which handles many of the tasks essential to successful virtualization. The VMM manages multiple instances of an operating system and switches instances so that each virtual machine can access the underlying processor, memory and local disk resources. It uses emulation to share the computing resources while keeping each of the virtual machines isolated from one another.

The problem with virtualization performance is that before 2005, major processors sorely lacked the internal instruction set needed to process virtualization tasks efficiently. Virtualization tasks were handled in software, which resulted in poor VM behavior, poor resource control and poor efficiency. In short, virtualization did not work well, and server performance suffered. Both AMD and Intel created processor extensions to address the added demands of virtualization. These extensions reduce (or even eliminate) the trapping and emulation of demanding instructions within the guest operating system, vastly improving the security and performance of virtual machine instances.

Intel VT, initially code-named "Vanderpool," appeared in plans for processors using their "Nehalem" architecture, which now encompasses a large number of processor families including versions of Pentium 4, the Celeron, Core i5, Core i7, Core Solo, Core 2 Duo, Core 2 Extreme, Core 2 Quad, Xeon and other processors. It's important to realize that not all recent Intel processors (nor every model within a particular family) necessarily support VT-x: It's considered a feature for high-end PCs and servers. A list of Intel processors with VT-x support can be referenced on Intel's website.

Intel has created a series of extensions to provide hardware for virtualization. The Intel VT-x extensions are probably the best recognized extensions, adding migration, priority and memory handling capabilities to a wide range of Intel processors. By comparison, the VT-d extensions add virtualization support to Intel chipsets that can assign specific I/O devices to specific VMs, while the VT-c extensions bring better virtualization support to I/O devices such as network switches.

Memory control and management play a huge role in hardware-based virtualization support. Intel VT-x extensions eliminate the software-based emulation often associated with VMM interventions, and Extended Page Tables add the memory management features needed to switch the processor control between various virtual machines.

Intel's VT FlexPriority feature helps manage processor interrupts. Since interrupts from certain devices and other applications can often impact the processor's performance, Intel includes a task priority register to gauge task priority. Only interrupts with a higher priority than the current task are handled immediately, and lesser interrupts are placed onto a stack for later handling as the workload permits.

The Intel VT FlexMigration feature allows virtualization platforms to easily and quickly migrate virtual machines across servers that run Intel VT-x supported processors. It's important to note that extended migration is not necessarily compatible with AMD processors, and this may cause performance issues (or outright failures) when migrating VMs between servers with differing processor makers.

—Stephen J. Bigelow

L

Live migration

Virtual machine live migration is a virtualization process that moves a virtual machine (VM) from one physical host server to another. It moves the memory and state of a VM without shutting down the application, so users will generally not detect any significant interruption in application availability.

The process captures the complete memory space occupied by the VM -- along with the exact state of all the processor registers currently operating on the VM -- then sends that content across a TCP/IP link to memory space on another server. Processor registers are then loaded, and the newly moved VM can pick up its operation without missing a step.

Most VM live migrations occur between similar hypervisors, so the migrated VM retains its name and other unique identifiers. Even though the VM is on a different server, it's the exact same machine as far as the users are concerned.

Live migration is a key benefit of virtualization, allowing workloads to move in real time as server or data center conditions change. Consider the impact on business continuity: A virtual server scheduled for maintenance can migrate its workloads to a spare server or to other servers that have extra computing capacity. Once the maintenance is complete and the server returns to service, these workloads can all migrate back to the original server without disruption.

Live migration helps server consolidation by allowing IT administrators to balance workloads across data center servers, ensuring that each server is used efficiently without being overtaxed. Live migration helps with disaster recovery too because VMs can just as easily be moved from one site to another, relying on spare servers at a remote site to receive and operate the migrated VMs.

All of the major virtualization software platforms include VM live migration tools. These include VMware vMotion (part of vSphere), Microsoft Live Migration (part of Hyper-V R2) and Citrix Systems XenServer live migration.

Migration tools typically allow administrators to prioritize the movement of each VM so that failover and failback processes occur in a predictable and repeatable manner. Mission-critical VMs usually take priority and are often moved to spare servers with ample computing resources.

Secondary VMs can be addressed next, although the migration software may be left to move noncritical VMs automatically based on the computing resources on each available server. Migration audits allow administrators to locate VMs and track their movements to refine and optimize ongoing migration behaviors.

Live migration works between almost all virtual host servers, but it's important to test migration behaviors between servers with various processor manufacturers. Processors from Intel and AMD both include extensions that provide hardware assistance for virtualization tasks, including migration. However, Intel VT and AMD-V processors use different architectures to facilitate migration, and moving VMs between Intel and AMD-based servers may result in unexpectedly poor migration performance.

—Stephen J. Bigelow

M

Memory overcommit

Memory overcommit (or overcommitment) is ahypervisor feature that allows a virtual machine (VM) to use more memory space than the physical host has available. For example, virtualization platforms like VMware ESX allow a host server with 2 GB of physical memory to run four guest machines, each with 1 GB of memory space allocated.

The idea of memory overcommit may seem dangerous, because a computer will crash if physical memory is exhausted. In actual practice, however, overcommitment of server computing resources is harmless -- most VMs use only a small portion of the physical memory that is allocated to them. For the previous example, a guest machine with 1 GB of physical memory allocated to it might only need 300-400 MB, leaving 600-700 MB of allocated space unused. If all four example machines use 300 MB, the physical server will have 800 MB of its original 2 GB left over.

Still, some VMs may need all (or even more) of the memory that they have been allocated, while some VMs may need considerably less. Hypervisors such as ESX can identify idle memory and dynamically reallocate unused memory from some VMs to others that need more memory. If none of the current guest machines need additional memory, any idle physical memory can be used to host additional guest machines if necessary.

Note that the concept of memory overcommit is not new. For example, operating systems like Windows have been able to operate and execute applications within the confines of limited computer memory for many years by swapping portions of data and program code content between memory and disk as needed. The technology is called virtual memory or memory paging.

Virtual memory or memory paging may prevent a virtual host server from crashing, but it imposes a significant performance penalty on the server, because disk access is far less efficient than with solid-state memory access. The performance hit would simply be too great for servers running multiple VMs.

Ultimately, there must be enough physical memory for optimum virtual server performance. IT professionals will need to exercise due diligence in their assessment of computing resource demands for each VM, then ensure that the prospective host server has adequate computing resources to meet that demand. When memory demands actually approach the server's available memory, an IT professional can choose to add memory to the server (upgrade) or distribute the VMs among various servers to balance computing resource needs (workload balancing).

—Stephen J. Bigelow

Network virtualization

Network virtualization is a versatile technology. It allows you to combine multiple networks into a single logical network, parcel a single network into multiple logical networks and even create synthetic, software-only networks between virtual machines (VMs) on a physical server.

Virtual networking typically starts with virtual network software, which is placed outside a virtual server (external) or inside a virtual server -- depending on the size and type of the virtualization platform.

Any virtual networking that takes place outside of a virtual server is called external network virtualization. This occurs when multiple physical LANs are aggregated into a single logical LAN, or when a single physical LAN is parceled out into multiple virtual LANs (VLANs). External network virtualization uses virtual network software such as Hewlett-Packard Virtual Connect and involves network switches, network adapters, servers, network storage devices and the Ethernet or Fibre Channel media that interconnects these hardware devices.

In internal network virtualization, virtual network software can emulate network connectivity within the server and allow VMs hosted on that server to exchange data. It might seem trivial, but the isolation that a virtual network provides can be useful. Eliminating the need to pass data on an external network can improve performance and bolster security for associated VMs.

Some virtual platforms support both internal and external network virtualization. VMware is one example; its platform supports internal network virtualization through the native hypervisor and uses additional software to support external virtualization.

Certain vendors tout virtualization and networking as a vehicle for additional services -- not just as a way to aggregate and allocate network resources. For example, it's common practice for a network switch to support security, storage, voice over IP (VoIP) and other advanced network services.

Regardless of the approach, managing virtual network software can be extremely challenging. It can be difficult, even impossible, to keep track of the multiple services and virtual networks running within the physical LAN. Careful documentation, clear workflow procedures and comprehensive management tools are vital for proper virtual networking management.

Several initiatives will influence the proliferation of network virtualization in the future. These include, but are not limited to, the Global Environment for Network Innovations, Future Internet Research and Experimentation, the AKARI Architecture Design Project and Federated E-infrastructure Dedicated to European Researchers Innovating in Computing Network Architectures (FEDERICA).

—Stephen J. Bigelow

P

P2V

Physical-to-virtual server migration is not black magic. To migrate to a virtual server, you copy the bits residing on a physical disk to a virtual disk, inject some drivers, and modify some other bits to support those drivers.

Sound simple? The challenge is in knowing how to perform a physical-to-virtual (P2V) server migration -- or, more to the point, when.

For some versions of Windows operating systems and some virtual server migration tools, it is possible to run a conversion while the physical host is running. Other tools require the host to be powered off and booted off special media , such as CD-ROM or iSCSI Initiator. I would allow for some downtime for this, just in case your operating system can't perform a live P2V migration. It is critical to always plan for possible downtime.

In either case, the process is the same:

1. For each local physical disk, first read the physical disk dimensions and used space by file system. Then, set the virtual disk file system dimensions, which cannot be less than the used space but can be less than the physical file system dimensions.
2. Create the virtual machine (VM) configuration, including the VM name (different than the physical machine name), the network connections, the number of virtual CPUs and the amount of memory to assign to the virtual machine.
3. Customize the IP address and other items as necessary using tools such as Sysprep that are embedded into the virtual server migration tool.

4. Copy the bits from one file system to the target file system on a virtual disk.
5. Inject the appropriate drivers into the guest operating system for the devices now in use. In general, these will be the necessary SCSI and networking drivers. In many cases, this is just a reassignment of how to access the disk.
6. Reboot the VM.

Now you're done with the steps for within the virtual server migration toolkit. But there are still some necessary administration steps.

1. Setup the IP address within the VM if not already done.
2. Install VMware Tools if using VMware, XenTools if using Xen, etc.
3. Test the application within the VM.
4. When ready, power down the physical server and move the VM to a live network.

Voila! You have just learned how to migrate to a virtual server.

While a P2V migration is not necessarily difficult, it is important to fully understand the operating system you are converting because such migrations sometimes transfer all the bits but fail to boot. Usually, this is due to a driver issue that can easily be fixed by booting from operating system rescue media.

—Edward L. Haletky, Contributor

Provisioning

Virtual machine provisioning, or virtual server provisioning, is a systems management process that creates a new virtual machine (VM) on a physical host server and allocates computing resources to support the VM. These computing resources typically include CPU cycles (or entire cores) and memory space, but can also involve I/O cycles and storage.

Although virtual machine provisioning can be accomplished manually, administrators generally prefer to automate server provisioning by creating a generic VM, called a VM template. This generic VM is loaded from storage (usually the corporate SAN) to the desired host server.

A corporation may maintain an extensive library of various VM templates -- each with a unique suite of computing resources -- that it can deploy in response to varied needs. For example, a Windows Server 2003 VM template may allocate 384 MB of memory and 10 GB of disk space, while a Windows Server 2008 VM template may allocate 512 MB of memory and 30 GB of disk space.

The more recent trend toward rapid virtual machine provisioning shifts the focus to the SAN, where a generic VM can be replicated and presented to a host server without the time and bandwidth needed to copy a blank VM to the host over the network. Administrators can also add or subtract desired elements of the new VM without having to create a larger number of complete VM templates.

Virtual server provisioning should be approached carefully. An application with inadequate computing resources can suffer from poor performance, poor availability or crash entirely, so it's vital for

administrators to allocate sufficient resources. Testing prior to rolling out the virtual application can help ensure these resources are adequate.

Server resources are finite, so a given server can only support a limited number of virtual machines. The exact number really depends on the age and sophistication of the physical server itself. Most administrators prefer to load a physical server at 50% to 80% of its total resource capacity. This includes a VM for the server's host operating system. The unused computing resources can be pooled and reallocated dynamically as workload demands change, or left in reserve to support additional VMs that are migrated to the server.

Virtual machines generally do not use all of the resources that are allocated to them, so virtual machine provisioning may also involve some amount of over-provisioning -- allocating more resources than the server has available. For example, "thin provisioning," a common practice in storage, is appearing in virtual platform features like memory overcommit. When implemented properly, over-provisioning makes it possible for a server to host more VMs than might otherwise be feasible.

Finally, administrators face a challenge of process control in virtual server provisioning. VMs are so simple to create that servers can easily be overwhelmed by an uncontrolled proliferation of virtual machines (a phenomenon called VM sprawl). To control VM sprawl, organizations should be able to justify the creation of new VMs, limit the actual creation rights to a few knowledgeable administrators and employ lifecycle management tools.

—Stephen J. Bigelow

Server consolidation

Server consolidation benefits users by allowing one physical server to host multiple virtual machine (VM) instances, which increases the effective utilization of server hardware.

Most traditional non-virtualized servers are only utilized at 5% to 10% of their total computing capacity. By adding a virtualization platform to the server (such as Citrix XenServer, VMware vSphere or Hyper-V in Windows Server 2008 R2), the server can operate its original workload as a virtual machine, and host additional virtual workloads simultaneously -- often increasing the total utilization of the physical server from 50% to 80% of its total computing capacity.

But increasing computing efficiency is only one of many server consolidation benefits. With more workloads running on less hardware, power and cooling demands are also lowered. This translates to lower operating costs for the business, and can also forestall capital-intensive facilities projects.

Server consolidation with virtualization allows the flexibility to seamlessly migrate workloads between physical servers -- literally moving workloads at-will or as-needed. For example, a traditional server

would have to be taken offline for maintenance or upgrades. With virtualization, all of the server's consolidated workloads can be migrated to a spare server or distributed amongst other servers, and then the original server can be shut down without any disruption to the workloads. Once the work is completed, the workloads can be migrated back to the original hardware. Workloads from a failing server can likewise be failed over or restarted on other servers, minimizing the effect of hardware problems.

Virtualization is also a boon to data protection, and workloads consolidated with virtualization can easily be copied with periodic point-in-time snapshots or replicated to off-site storage systems with little (if any) of the performance penalty experienced with traditional tape backup systems.

Even with a wealth of server consolidation benefits, however, successful server consolidation requires a careful server consolidation strategy. First, consolidation should be approached in phases. Start by virtualizing and consolidating non-critical or low-priority workloads. Administrators can gain valuable experience with server consolidation tools. Then With more experience, you can then systematically virtualize and consolidate more important workloads until you tackle the most mission-critical applications.

The distribution of those virtualized workloads can make a tremendous difference in the success of your consolidation project. Since each workload can demand different computing resources, it's important to measure the needs of each workload and allocate workloads so that the underlying host servers are not overloaded -- a process known as "workload balancing." For example, it's often better to distribute CPU-intensive workloads on different servers rather than putting them on the same server. This prevents resource shortages that can cause workload performance or stability problems.

—Stephen J. Bigelow

Snapshot

Virtual machines are dynamic entities. They exist and run within a portion of the memory space available on a physical host server.

In spite of this dynamic nature, however, virtual machines (VMs) rely on disk for data storage. For example, when a host server boots up, the guest virtual machine must be loaded from the enterprise storage, which is often a high-performance Fibre Channel storage-area network (SAN). But virtual machines also require data-protection technologies in case of system crashes and hardware failures. VM snapshot backup techniques are the most common means of VM protection.

Simply stated, a VM snapshot is a copy of the VM state as it exists in server memory at a particular moment, along with any settings and the state of any virtual disks assigned to the VM. The VM snapshot is saved to disk, typically the SAN.

A regular snapshot backup process can significantly reduce the recovery point objective (RPO) for the protected VM. That is, if you take a VM snapshot every 15 minutes, you stand to lose only up to 15 minutes of data in the event of a failure. If the VM snapshot occurs every five minutes, the RPO is reduced to five minutes, and so on. It's only theoretically necessary to keep the latest VM snapshot, adding less demand for storage.

It may actually take several minutes to write a VM snapshot. During this time, it is impossible to write to the virtual machine disk file (such as a VMware VMDK file). However, an additional VM file will record any differences between the current machine state and the machine state at the start of the VM snapshot. This disk file, called a delta disk file, allows users to continue accessing the VM during the snapshot backup process. It also creates a full and complete copy of the machine state at the moment that the snapshot backup is complete, and the main disk file is available to receive writes again.

Taking a VM snapshot is a simple matter for an administrator. It's usually a feature integrated into the virtualization platform. For example, taking a snapshot in a VMware vSphere environment is as simple as right-clicking the VM, selecting Snapshot, and then choosing Take Snapshot. In the VM snapshot dialog that appears, enter a name and description, capture the VM's memory state, quiesce the VM, and click OK. A snapshot backup can also be scheduled to occur at a certain frequency.

A snapshot backup is an important data-protection tool for VMs, but it should not be the only backup technique. The VM snapshot backup process treats virtual machine disk files as a single file, so the entire instance must be restored to recover a lost text or deleted note. This has long forced administrators to restore the VM snapshot to a lab or other nonproduction server to recover specific files within the VM. However, this is changing with the introduction of more robust and capable tools that can look inside VM snapshots and find specific data.

—Stephen J. Bigelow

Storage virtualization

Storage virtualization creates a layer of abstraction between the operating system and the physical disks used for data storage. The virtualized storage is then location-independent, which can enable more efficient storage use and better storage management.

For example, the storage virtualization software or device creates a logical space, and then manages metadata that establishes a map between the logical space and the physical disk space. The creation of logical space allows a virtualization platform to present storage volumes that can be created and changed with little regard for the underlying disks.

The most immediate benefit of storage virtualization is increased storage utilization, which can reduce wasted storage within the enterprise. For example, a logical unit number (LUN) provisioned on a storage area network (SAN) may allocate space that may not be used, or disks may be left unallocated -- lost and forgotten on storage arrays scattered across the data center. With virtualization, otherwise-unused storage can be cobbled together into viable LUNs and allocated to applications.

Data storage virtualization also supports migration and replication of LUNs between storage systems. This is particularly useful when one storage system must be taken offline for maintenance or replacement. By simply changing the mapping scheme, virtualization can move the location of data without disrupting disk I/O, allowing for efficient and nondisruptive data movement within an enterprise.

Storage management can be greatly simplified. Rather than managing multiple (often heterogeneous) storage subsystems, a virtualized storage environment can be managed through a single mechanism. In

addition, advanced storage management techniques such as thin provisioning and dynamic provisioning can readily be supported. thin provisioning allows the creation of a LUN that is larger than the physical disk space allocated to it. For example, a 1 GB LUN can be created with perhaps 100 MB of storage space to start. As the associated application uses the disk space, more disk space can be allocated periodically (up to the assigned amount) without having to recreate the LUN.

Dynamic provisioning is similar, allowing the size of a LUN to be grown or shrunk as needed, ensuring the size of a LUN is always appropriate for each application. But traditional storage management (such as RAID group creation and maintenance) is still required.

Host-based storage virtualization such as Symantec's Veritas Storage Foundation uses device drivers and additional software to virtualize the physical disks within a host system. Device-based storage virtualization such as Hitachi Data Systems' Universal Storage Platform incorporates and manages virtualization within the storage array itself. Network-based storage virtualization uses a server (appliance) or smart switch like an EMC Invista within a Fibre Channel or iSCSI SAN to implement the abstraction between the network I/O and storage controllers.

—Stephen J. Bigelow

V

Virtual hardware

A virtual hardware platform is a collection of computing resources allocated from a physical host server to a virtual machine (VM) during the virtualization process.

Virtualization creates a layer of abstraction between the application and the underlying computer hardware. This abstraction allows software -- the virtualization hypervisor -- to assign, control and monitor the computing resources that form the virtual hardware platform for a workload.

The concept of a virtual hardware platform is crucial to virtualization. It frees a workload from specific physical hardware devices so that the workload can run on any physical host with the proper computing resources available. It also allows key virtualization features, such as live migration, which moves workloads between physical servers with no downtime.

The process of creating a VM assigns a default virtual hardware platform to the VM. Virtual hardware assignments include memory, processor cores, optical drives, network adapters, I/O ports, a disk controller and one or more virtual hard disks (VHDs).

Perhaps the most interesting attribute of a virtual hardware platform is its versatility, because an administrator can adjust the levels of each resource -- adding more memory and additional processor cores, allocating another VHD or assigning more network adapter ports, for example. Increasing

resource levels will normally boost that workload's performance or responsiveness -- especially on older physical servers -- or allow the VM to support more users.

Conversely, an administrator can also remove virtual hardware from a VM. For example, an application may not be able to utilize a full 2 GB of memory or two processor cores. Removing excess resources will free those resources for allocation elsewhere, improving the performance of other busy workloads or increasing utilization on the physical server.

Generally speaking, an administrator will need to power down a VM before adding or removing resources.

It is sometimes possible to allocate more memory to a virtual hardware platform than what is actually available from the physical host. For example, it is possible to configure a VM with 16 GB of memory on a physical server with only 8 GB of RAM. This process, called memory overcommit, allows more VMs to reside on a server, because most VMs don't utilize all of their allocated memory space. When physical memory runs short, the virtualization platform implements a virtual memory swap file on the virtual disk.

The practice of memory overcommit is well accepted, but there is still a serious performance penalty when using a swap file. Therefore, it's best to ensure adequate memory on the server.

—Stephen J. Bigelow

Virtual machine

A virtual machine (VM) is a separate and independent software instance that includes a full copy of an operating system and application software. A physical server prepared with a server virtualization hypervisor such as Microsoft Hyper-V, VMware vSphere or Citrix XenServer, can host multiple VMs while maintaining logical isolation between each machine. Each instance can then share the server's computing resources -- dramatically increasing physical server hardware usage.

Server consolidation is the most compelling benefit of virtual machines. A typical non-virtualized application server may reach just 5% to 10% utilization. But a virtual server that hosts multiple VMs can easily reach 50% to 80% utilization. The net result is that more VMs can be hosted on fewer physical servers, translating into lower costs for hardware acquisition, maintenance, energy and cooling system usage.

Virtualization also facilitates VM creation and VM management. Unlike conventional servers that host a mixture of an OS, driver and application files, an entire VM exists as a single file, such as a VMDK file. A VM file can be created and duplicated as needed, proliferating virtual machines on servers across an enterprise. These golden images can be modified for each user or application.

Virtual machines in operation are saved to storage nondisruptively using periodic snapshots to a SAN. Troublesome or crashed VMs are quickly reloaded to the server directly from storage, which accelerates recovery times after a server or application crash.

VM tools assist server management by allocating and regulating the computing resources that each VM uses. For example, multiple CPU cores may be allocated to a CPU-intensive application in one VM while other noncritical VMs may share the same CPU core. Similarly, an administrator can reserve the minimum required amount of network bandwidth for a VM running a transactional application. A virtual file or print server may not have this reservation.

Administrators must balance the computing demands of each VM against the total computing resources that each virtual server provides. Instead of supporting multiple memory-intensive VMs on the same server and risking low-memory performance penalties, the VMs can be distributed across multiple physical servers.

The concept of VM workload balancing also relates to CPU and I/O-intensive VMs. Certain tools enable nondistruptive workload migration between servers so that VMs can move from server to server in real time. Migration continues until an acceptable balance of server consolidation and performance is achieved.

—Stephen J. Bigelow

Virtual machine monitoring

When someone asks if there is a way to monitor their virtual machine (VM), my first question is always, "What do you want to monitor?" The second question I ask is, "Do you have any existing server monitoring tools that you can use, and if so, how do they work?" These two questions lead administrators down different VM monitoring paths.

Take, for example, the simple act of monitoring the up/down state of service or virtual machine failure monitoring. There are hundreds of tools that do this with or without the use of agents. If they use agents, the agents install directly within the virtual machine. If they don't use agents, the tools tend to probe the VM for a state to determine if a Web server is running or if the machine can be pinged, for example.

In either case, the agents help you determine if the VM is up or down and whether its services are running. Remember, though, that you can incur license costs for probing a system and installing an agent.

In general, anything you can monitor within a physical system can be monitored within a VM, with one exception: performance. With virtual machine performance monitoring, it is actually better to monitor from outside the VM than from within. This is because data retrieved from within the VM may not be accurate. Although this data can be used as an estimate, you don't know if the raw numbers are correct. When a virtualization host over-commits CPU, the virtualization host scheduler divides the processor between each VM, but not necessarily in a contiguous fashion. In other words, a VM no longer spends all of its time on a CPU.

However, CPU-specific performance counters within a guest operating system are expected to read the CPU for data all the time. The counters then compute this data based on the fact that every nanosecond is spent running a CPU.

When CPUs are over-committed, a VM doesn't spend every nanosecond on a CPU -- some nanoseconds are spent idle, waiting for a CPU to run on. Therefore, there is often a discrepancy in the numbers that performance monitoring tools generate and from those within the VM. Since it's difficult for a VM to know when it has been made idle, this information is best retrieved from a hypervisor outside the VM. The hypervisor knows this information and can give accurate numbers for the VM in question.

Therefore, you need a VM performance monitoring tool that works with your hypervisor. There are several available for each hypervisor on the market. If you don't use one specific to your hypervisor, the data you receive will just be an estimate of your current VM performance.

—Edward L. Haletky

Virtual switch

The core to the virtual network is the virtual switch. The virtual switch has uplinks from physical switches via physical network interface cards (pNICs) and ports that attach to virtual machines. But what other features define the virtual switch?

Let's start with what a virtual switch is not. Technically speaking, KVM, XenServer, open source Xen and Hyper-V, plus all the hosted virtualization products, use a virtual bridge and not a virtual switch. There is not much difference between a bridge and a switch. They both send data between two different segments of a network and they both verify the data being sent.

However, virtual switch software has many more features available to it and can perform more complex operations, including the implementation of virtual LANs, EtherChannel and other more advanced networking tools.

In either case, the pNICs are placed in bridging mode, and you bridge from a physical switch to a virtual switch. This is just like putting an uplink cable between two physical switches. We are in essence uplinking from the physical switch to the virtual switch. In some cases, it is also possible to have virtual switches that are not connected to any pNIC, thereby creating fully virtualized networks.

Each virtual switch system for each hypervisor has different virtual switch security mechanisms. Currently, VMware vSphere has some of the more advanced capabilities, but this will change when the Xen OpenVSwitch is available.

Connected to the virtual machine side of each virtual switch is a virtual network interface card (vNIC). There can be one or more vNICs per VM. However, it is not possible to layer virtual switches, i.e., one virtual switch cannot directly attach to another virtual switch. To connect a virtual switch to another virtual switch from within the virtual network, you must place a VM between them. These VMs would have at least 2 vNICs and act either as a router, firewall or gateway device. By using VMs that act as these types of devices, it is possible to build very complex virtual networks.

The key to virtual switch configuration is that the switch connects on one side to the physical network through the pNICS, and on the other side a virtual switch connects to the virtual machines via the VM's vNICs. This creates the virtual network.

A full Layer 2 virtual switch (like VMware's vSwitch, Cisco Nexus 1000V and the OpenVSwitch) has its own security and mechanisms to support 802.3ad, 802.1q and other protocols used in today's switching networks.

—Edward L. Haletky

VMDK

Traditional non-virtualized servers and desktops will load and execute hundreds, even thousands, of individual files such as operating-system kernel files, device drivers, application components and data files. Virtualization abstracts the software from the underlying hardware, and places all of the constituent data for any given virtual machine (VM) into a single disk file.

The Virtual Machine Disk (VMDK) file format is the disk-format specification used with VMware virtual machine files. In essence, a file with a .VMDK file extension is a complete and independent virtual machine using VMware virtualization products or other platforms that support VMDK files such as Sun XVM, QEMU, or VirtualBox.

There are pros and cons to this "single file" VMDK disk. The principal advantages are simplicity and convenience. A single VMDK file, for example, is easy to move between servers using live migration features in the virtualization platform. Similarly, a single file can easily be protected with snapshots or continuous data protection (CDP) technology. Virtual machine files, in fact, are often copied to the SAN. There, additional resiliency practices such as replication to off-site disaster recovery facilities and RAID within the SAN-storage array can further protect the VMs. By maintaining VM files on a high-performance SAN, recreating VMDK files or restarting troubled VMs on other physical servers is a simple process. This can be crucial when a VMDK file is damaged or corrupted.

The traditional drawback to a "single file" VMDK disk is the extra effort necessary for recovering lost data. Before the introduction of specialized software tools, it had been impossible to recover only a part of the VM, such as a deleted Word document. The entire VM would have had to be restored, usually to a spare server instead of the actual production server, and the missing or corrupted file could then be retrieved. Today it can be quicker and easier to restore an individual file from a VM than it is to restore an entire VM and extract the desired file(s) from it. But regardless of the available tools, administrators must still develop a sound process for data recovery from VM files.

Since the VMDK format is a central part of VMware's virtual environment, it's critical for any third-party provisioning, management and backup tools to be fully interoperable with disks running VMDK. Third-party developers and organizations developing their own custom applications for a VMware environment can employ the VMware Virtual Disk Development Kit. This includes a C library and command-line utilities that allow developers to create and access VMDK files.

The VMDK file format competes with the Microsoft Virtual Hard Drive (VHD) disk format used with Virtual Server and Hyper-V hypervisors. This can become problematic when moving from VMware to Hyper-V. Since the disk formats are not directly compatible, existing virtual machines cannot operate under a different hypervisor. In many cases, virtual machines would need to be translated onto physical servers first (V2P) to effectively remove virtualization. The new hypervisor would then create new

virtual machines using the new disk format. Some third-party tools such as VMDK2VHD claim to convert VMDK to VHD files, smoothing the transition between hypervisors for IT professionals.

—Stephen J. Bigelow

X

Xen

The Xen open source virtual machine monitor was originally created through the University of Cambridge Computer Laboratory and developed through XenSource, Inc. Citrix Systems acquired XenSource in 2007, and Xen technology has since emerged in the free edition called XenServer, along with two paid enterprise-class products: Essentials for XenServer Enterprise and Essentials for XenServer Platinum. Other commercial implementations of Xen include Oracle VM and Sun xVM.

The Xen open source virtual machine monitor is designed for common Intel and IBM architectures, supporting x86 (32 and 64-bit), Itanium, and PowerPC-based systems. The Xen hypervisor loads and supports all of the subsequent operating systems (OSes) and workloads. This is referred to as a Type 1 or bare-metal hypervisor, which runs directly on the system's hardware and hosts OSes above it. The first guest is typically the system's host OS, which has management privileges and extensive control over the system's physical hardware (Xen nomenclature denotes this "domain 0" or "dom0").

This allows system administrators to access and control the server and other workloads from that first guest virtual machine (VM). Xen management tasks are usually simplified and automated through third-party tools like XenTools, Ganeti, MLN, HyperVM, Convirtue and others.

There are many operating systems that can serve as a host operating system (dom0). These include most distributions of Linux, Novell's SUSE Linux Enterprise Server release 10, Red Hat Enterprise Linux 5, Fedora, openSUSE 10.3, Ubuntu 8.04, NetBSD 3.x, Debian release 5 and others. Xen 3.0 and later can support Microsoft Windows and other proprietary OSes as guests, but the host server will require processors that support virtualization acceleration technologies such as Intel VT and AMD-V.

The hypervisor in the Xen open source virtual machine monitor is particularly noted for its virtual machine live migration capabilities, allowing administrators to move virtual workloads from one physical host to another without shutting down or even disrupting the workload being moved. Generally speaking, a live migration process copies the memory space used by the VM and then replicates that content on the destination server. Once the copy is complete, the two iterations are synchronized and processing is handed to the destination server, allowing the original server to delete the unneeded instance. The actual disruption involved in this handoff is usually less than half a second.

Live migration is one of the key benefits of server virtualization because it enables several critical benefits related to application availability. The workloads on a problematic server can often be migrated

(failed over) to other servers before a crash actually occurs, and this preserves the workload's availability. The ability to move workloads also makes it possible for administrators to balance the computing demands on each server. This optimizes system performance and helps to prevent VM (or even entire server) crashes due to inadequate computing resources. Finally, an administrator can migrate workloads to free up a server for maintenance or replacement without interrupting the workload's availability.

—Stephen J. Bigelow

Virtualization Explained:
Definitions You Need to Know

About the authors

Stephen J. Bigelow, a senior technology writer in the Data Center and Virtualization Media Group at TechTarget Inc., has more than 15 years of technical writing experience in the PC/technology industry. He holds a bachelor of science in electrical engineering, along with CompTIA A+, Network+, Security+ and Server+ certifications, and has written hundreds of articles and more than 15 feature books on computer troubleshooting, including *Bigelow's PC Hardware Desk Reference* and *Bigelow's PC Hardware Annoyances*.

Edward L. Haletky, a SearchServerVirtualization.com contributor, is the author of *VMware ESX Server in the Enterprise: Planning and Securing Virtualization Servers*. At Hewlett-Packard Co., he worked on the virtualization, Linux and high-performance computing teams. Haletky now owns AstroArch Consulting Inc. and is a champion and moderator for the VMware Communities Forums.