

SQLskills Immersion Event

IE3: High Availability & Disaster Recovery

Module 11: Virtualization

Jonathan Kehayias
Jonathan@SQLskills.com



Overview

- Why virtualization is popular
- How virtualization works
- Considerations for SQL Server
- HA/DR options
- Monitoring
- Appendix: x86 virtualization types



2

© SQLskills, All rights reserved.
<http://www.SQLskills.com>



Why Virtualization is Popular

Lower TCO for the Business

- Higher hardware utilization means there are less servers in the data center equating to lower power consumption and cooling costs
- Virtual machines abstract hardware away from the operating system making hardware upgrades fast and simple
- VM Server templates make new server deployments possible in a few minutes compared to a few days to weeks when new hardware must be acquired, configured, and installed

Why Virtualization is Popular

High Availability is Built-In

- Virtual machines are protected from most hardware failures and can be powered up on another VM host in the event of a hardware crash
 - Storage is the exception to this, if the disks hosting the VM totally crash, even in a SAN, the VM image will be lost and will require rebuilding or restoring from backups
- VMotion and Live Migration features allow load balancing of virtual machines in real time to minimize performance impact across the enterprise, but also facilitates planned VM host maintenance during normal business hours

Why Virtualization is Popular

Repositions IT as a Utility instead of a Cost Center

- For many businesses, IT is a tolerated cost center that is necessary for the business to operate
 - Department budgets rarely account for or plan for application upgrades and hardware replacement, forcing IT staff to deal with antiquated hardware and software installations well beyond the end of support agreements
- Virtualization changes IT into a shared-cost utility by:
 - Removing hardware costs from project implementation planning and costs
 - Requires a charge-back system for utilization to departments similar to household power-utility usage
 - Results in better utilization of hardware, centralized storage, and manageable scalability across the enterprise

Determining VM Candidates

Hardware Considerations

- CPU
 - Number of processors installed
 - Percent utilization (min, max and avg.)
- RAM
 - Total installed
 - Amount of available or unused memory in Windows
 - Page file usage
- Disk
 - Number and size of drives installed
 - Available free space per drive

Determining VM Candidates

Common Misconceptions About SQL Server

- **CPU**
 - Generally have dual or quad sockets with multi-core processors installed
 - Average 15-25% CPU usage with 5-10% during off hours
 - SQL doesn't really need all the CPUs it has and could be virtualized
- **RAM**
 - Generally have 64GB+ RAM, not uncommon for 128-256GB+
 - Memory usage is generally high
 - Must be a problem with SQL Server
 - If we double the RAM, SQL can still have what it needs and we can use the extra to run other VMs on the same host
- **Disk**
 - Excessive free space on the SAN/DAS for projected database growth can be reclaimed for VMs
 - Thin-provisioning storage and/or VMDK will allow growth as needed over time



7

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

Determining VM Candidates

Debunking Misconceptions around SQL Server

- **CPU**
 - SQL Server is designed to make use of additional CPUs for parallelism and reducing the CPU count can impact performance
 - Percent utilization should be low on database servers that have been appropriately tuned
- **RAM**
 - SQL Server uses memory to cache data as an optimization to minimize the amount of disk I/O that has to be done by the database engine; we can always use more memory if they will give it to us
- **Disk**
 - Capacity and free space are not as important as performance (IOPS), throughput (MB/sec), and latency
 - Sharing storage with other applications, or between data and log files can severely degrade performance



8

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

Determining VM Candidates

Other Considerations

- **High availability**
 - SQL HA options like failover clustering and database mirroring require redundant hardware that is not being utilized unless a failure occurs
 - Virtualization provides HA while utilizing the hardware for running other servers at the same time
 - Note: This is not the same as the HA provided by SQL Server HA solutions which we will cover later in this module as well as in other modules we've covered already this week
- **SQL Server 2012 licensing changes**
 - Core based licensing requires all vCPUs to be licensed
 - No hyperthreading reduction factor
 - Minimum 4 core licensing requirement

The Simple Truth

Virtualization is Cheaper

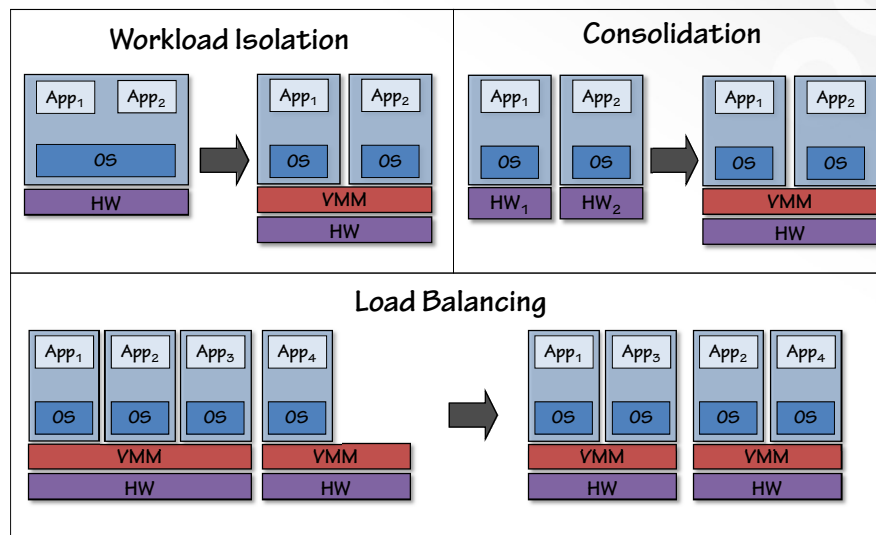
- **Despite what your IT administrators or managers believe, virtualization does not make things faster**
 - Performance benefits are often stressed by IT admins and managers because of the newer hardware they use for the virtual infrastructure
 - Given the same hardware capabilities, a physical implementation will always be faster because there is no hypervisor overhead
- **Virtualization makes things cheaper but good performance under virtualization is not going to be cheap**
 - Cheaper can mean:
 - Less hardware
 - Less administration
 - Less licensing
 - Cheaper can also mean performance problems if the right hardware is not used for the load being generated
 - 1Gb/sec iSCSI and virtualization works great for applications but not SQL
 - Memory overcommit and ballooning works great for applications but not SQL

Other Factors

Recent Hypervisor Changes

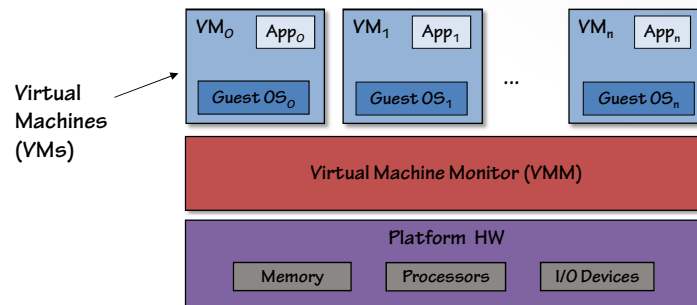
- Changes in hypervisor limitations in recent years have changed the landscape for virtualizing SQL Server
- Limitations on vCPUs expanded
 - 64 vCPUs on Hyper-V 2012 and ESX 5.1
- Memory limitations expanded
 - 1TB on Hyper-V 2012 and ESX 5.1
- Introduce the ability to pass vNUMA configuration through to VM hardware configuration
 - Affects SQLOS configuration for memory nodes, lazy writers, and connection distribution for higher scalability
- Hyper-V 2012 Virtual Fibre Channel allows N_Port ID Virtualization (NPIV) to share a single physical interface with multiple virtual HBAs for direct SAN access inside a VM

Virtualization Usages



How Virtual Machines Work

Virtual Machine Monitor (VMM)

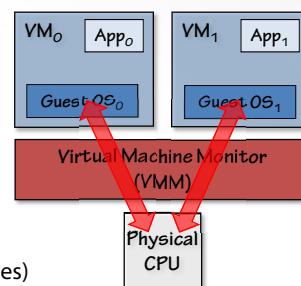


- **VMM is a layer of system software**
 - Enables multiple VMs to share platform hardware
 - Allows apps to run without modifications

How Virtual Machines Work

CPU Virtualization

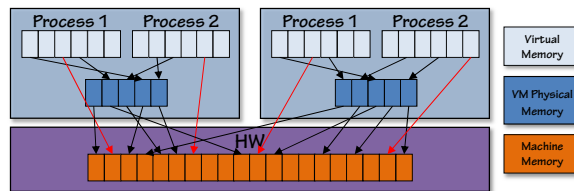
- **To virtualize a CPU the VMM must control:**
 - Access to privileged state
 - Exceptions (page faults, etc.)
 - Interrupts and interrupt masking
 - Address translation (through page tables)
 - CPU access to I/O (through I/O ports or MMIO)
- **This is accomplished through a process known as 'Ring De-privileging' by the VMM**



How Virtual Machines Work

Memory Virtualization

- The Memory Management Unit (MMU) must be virtualized to support the guest OS
- Guest OS continues to control the mapping of virtual memory to the guest physical memory addresses but guest does not have access to host machine memory
 - The VMM uses the Translation Look-aside Buffer (TLB) to map virtual memory directly to machine memory to avoid multi-level translation for every access
 - VMM updates shadow page tables to enable direct lookups



Memory Configuration

Understanding SQL Server Memory Requirements

- One of the focuses of virtualization is to reduce the amount of unused memory that exists inside of the datacenter
- For SQL Server most people don't know or understand how to determine what the actual memory usage or memory requirements for SQL Server is:
 - SQL Server:Memory Manager\Total Server Memory
 - SQL Server:Memory Manager\Target Server Memory
 - Total Server Memory is the amount of memory consumed by the buffer pool in kilobytes
 - Target Server Memory is the amount of dynamic memory the server can actually consume under its workload
 - If Target Server Memory exceeds the Total Server Memory, the SQL Server can use more memory if it is allocated to it

Memory Configuration

Implications of VMware Memory Overcommit

- VMware allows you to power on virtual machines on a single host with more virtual RAM allocated to them than actually exists as physical memory on the host
 - One of the reasons behind this is that many application servers have RAM allocations to them that go unused but need to be available for scenarios where demand requires additional memory temporarily
 - Another reason for this is advance features like memory de-duplication which will be covered later in this module
- When actual memory demand physically exceeds the available machine memory on the host VMware responds by ballooning each of the guest VMs to reclaim memory for allocation to other VMs by the hypervisor
- If ballooning fails to reclaim sufficient memory fast enough, the hypervisor resorts to disk paging of memory allocations

Memory Configuration

VMware Balloon Driver

- Balloon driver in VMware, installed by the VMware tools, fakes memory pressure in the guests to prevent actual memory pressure on the host
- The default SQL Server 'min server memory' setting is 0 which allows it to trim memory usage when the OS flag for memory pressure gets set however, SQL Server caches data pages in memory using the buffer pool and trimming of the memory usage will result in higher disk I/O to compensate
- In certain scenarios where the SQL Server process fails to respond fast enough to the memory pressure caused by ballooning of the VM, OS level paging of the SQL Server process to the page file can occur resulting in severely degraded performance and instability

Memory Configuration

VMware Recommended Best Practices

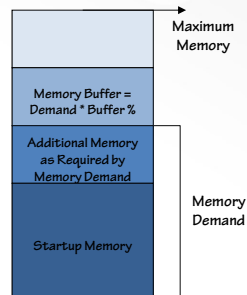
- **Don't overcommit memory if memory intensive applications are being virtualized**
- **Configure a memory reservation for SQL Server VMs if memory overcommit is being used**
 - Use Lock Pages in Memory to prevent hard OS paging of the SQL Server process
 - Set min and max server memory configuration options appropriately to control the size of the buffer pool that SQL Server manages
 - Keep in mind that the VMware reservation is for the entire server overall, not just SQL Server buffer pool
 - Anything above the reservation is not guaranteed, usually with memory over commit it is not going to be there
- **Use Large Pages, enabled by trace flag –T834 to limit TLB misses**
 - Consequence is that buffer pool must be allocated contiguously

VMware Memory De-Duplication

- **Hypervisor attempts to de-duplicate and compress memory used by virtual machines to allow further scalability**
 - If you run 10 Windows Server 2008 servers, there are 10 copies of basic DLLs in memory
 - Memory usage in the server will reduce over time when the server is running
 - Allows over-commit without actually over-committing resources
- **Performance degradation can occur during monthly patch windows due to the added hypervisor CPU overhead to identify duplicate memory pages across the VMs**

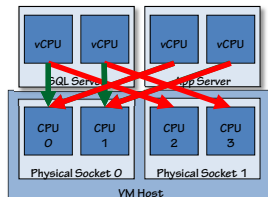
Hyper-V Dynamic Memory

- Requires Windows Server 2008 R2 SP1
- Removes the requirement to appropriately size VM memory requirements
 - Useful for VMs where necessary memory requirement is unknown and ability to dynamically grow/shrink VM memory allocation is beneficial
- SQL Server
 - Hot-Add memory support in Enterprise Edition allows SQL to make use of extra physical memory added after SQL Server starts
 - Requires Enterprise Edition+ of SQL Server 2005+
 - Startup RAM = 1-2GB + Min Server Memory
 - Maximum RAM > Max Server Memory
 - Memory Buffer >= 5%
 - Lock Pages in Memory recommended
 - Dynamic Memory not for Standard Edition



SQL VM CPU Allocations

- Shared CPUs can result in co-scheduling problems at the physical processor cache level
 - SQL running on CPU 0 and 1
 - SQL goes idle
 - App server runs on Core 0 and 1
 - SQL needs to run at same time
 - Scheduled on Core 2 and 3 = cache misses
 - Scheduled on Core 0 and 1 = waits for time slices on CPU to execute
 - Could result in cross NUMA problems if VMs span NUMA configurations



CPU Co-Scheduling

- **SMP virtual machines provide the illusion of a dedicated multi-processor system**
 - To accomplish this the vCPUs must be co-scheduled together to avoid guest OS problems (BSOD in Windows) and ensure guest OS performance of items like spin locks
- **The hypervisor tracks cumulative skew per vCPU for SMP VMs**
 - Ensure that all of the vCPUs are being scheduled and making progress concurrently with minimal skew
 - When the cumulative skew exceeds the configured threshold co-scheduling occurs
- **Co-scheduling requires the same number of physical CPUs be available for preemption as the number of virtual CPUs assigned to the VM for the VM to be scheduled for execution**

CPU Co-Scheduling Types

- **Strict Co-scheduling**
 - When any vCPU exceeds the skew threshold, all of the vCPUs for the VM are forced to stop to prevent further skew from accumulating
 - The next time a vCPU for the VM is scheduled, all of the vCPUs must be scheduled simultaneously to remove the skew that has accumulated
- **Relaxed Co-scheduling**
 - Instead of all vCPUs being scheduled simultaneously, only those that are skewed must be scheduled
 - Ensures that when any vCPU is scheduled, all other vCPUs that are "behind" will also be scheduled, reducing skew

CPU Co-Scheduling Problems

- **Multi-core systems have shared L2 and L3 caches on die between processor cores**
 - Load balancing and co-scheduling across physical sockets results in CPU cache misses and access latency for cache warm up depending on if it requires additional memory access or not
 - Hypervisors attempt to mitigate against the cold CPU cache scenario by throttling migrations and waiting for the original physical CPUs to become available
- **VMware is specifically NUMA aware and will schedule within the assigned NUMA node if the VM meets the requirements for NUMA optimizations**

Hyper-V/XenServer Scheduling

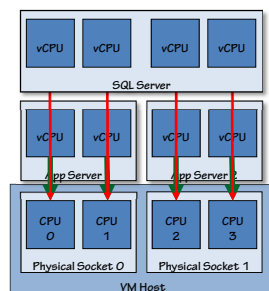
- **Credit scheduler implementation that is based on VM weight and cap**
 - Each vCPU is allocated credits based on its relative weight to the other VMs running on the host
 - As a vCPU runs, it consumes its allotted credits
 - A system-wide accounting thread recomputes how many credits each active VM has earned and bumps the credits periodically
- **Scheduling**
 - Each CPU manages a local queue of runnable vCPUs sorted by the priority representing whether the vCPU has or has not exceeded its fair share of CPU during the accounting period
 - vCPUs are scheduled in a round-robin fashion based on their relative weight
 - When a CPU doesn't find a vCPU with available credits in its local run queue, it will look on other CPUs for one to load balance the scheduling

VMware Scheduling

- **CPU Scheduler controls scheduling, preemption, and migration**
 - When the running vCPU exceeds its execution quantum (50ms) it changes to a WAIT state from RUN
 - CPU Scheduler checks the local queue for the next READY vCPU to run
 - If no local vCPU is found the CPU Scheduler will search other processor queues, in the same NUMA node if NUMA based, for READY vCPUs to run
 - Evaluates whether the cost to migrate CPU cache is more efficient over waiting
 - If no READY vCPUs are found IDLE vCPUs will be scheduled
- **Share based round-robin scheduling**
 - Allocates CPU resources to vCPUS based on the VM resource specifications for shares, reservations, and limits
 - Priority is based on the ratio of consumed CPU resources to the entitlement

VMware CPU Ready Time

- **Large VMs with small VMs can result in CPU Ready time accumulations**
 - App Server 1 running on CPU0 and 1
 - SQL Server needs co-scheduling for all four vCPUs and waits in a Ready state
 - App Server 2 can schedule on CPU2 and 3
 - SQL Server will preempt all four pCPUs pushing other VMs to Ready state



SQL VM CPU Allocation Considerations

- **How many virtual cores do we actually need?**
 - Consider co-scheduling issues under the hypervisor?
- **How many idle cores are available on the host VM?**
 - Co-scheduling on idle cores will be at full speed until another VM begins to execute and begins to take slices on the CPUs making those SQL threads slower to execute
- **How fast is each physical CPU core?**
- **Is hyper-threading being used on the physical server?**
- **How much of each physical core does the VM get % wise for execution?**
- **Did the last execution run on the same physical cores?**

Guaranteeing VM CPU Resources Hyper-V CPU Reserve

- **Percentage of the total possible CPU usage of a virtual machine**
 - Will block a virtual machine from starting if the reserve can not be honored by the hypervisor under peak load; total of all reserves on running VMs is 100%
 - Only enforced when CPU resource contention occurs
 - Default value is 0% or disabled; if all reserve is taken by active VMs a 0% reserve can still be powered on
- **May be used to enforce SLAs and guarantee a minimum amount of CPU usage**

Guaranteeing VM CPU Resources

Hyper-V CPU Reserve: Limitations

- **Reduces environmental flexibility**
 - VMs may fail to start in a failover if sufficient reserves are not available on the failover node
 - To start a new VM on a server with reserves maxed out, requires starting the new VM without a reserve or reconfiguring all of the existing VMs to reduce their reserve
- **Bin-Packing Problem**
 - 4 CPU host
 - VM with 4 vCPUs and a 50% reserve – 50% of host resources
 - VM with 2 vCPUs and a 80% reserve – 40% of host resources
 - Only one of these can be running at a time

Guaranteeing VM CPU Resources

VMware CPU Reservation

- **Minimum service level guarantee in MHz for the VM**
 - Admission control checks performed at VM startup and when changes to reservations are made to ensure that the VM will be able to get this guarantee
- **The reservation applies even when the host system is overcommitted and under CPU resource pressure**
- **Limits the flexibility of the environment**

Configuring VM CPU Priority

Hyper-V CPU Relative Weight

- **Number between 1 and 10,000**
 - Set to 100 by default
 - Higher numbers have higher priority
- **Does not guarantee a level of service**
 - Guarantees that higher VMs will receive more resources under pressure than VMs of lower priority only
- **You need to establish a common scale for configuring relative weight in the enterprise**
 - One admin using weights of 200 for important and 100 for standard
 - Another using weights of 1,000 for important and 500 for standard

Configuring VM CPU Priority

VMware CPU Shares

- Shares are a ratio based entitlement to the VM that is directly proportional to the total number of shares that have been issued to all VMs on the host server
- The number of shares for an individual VM is an abstract number with no real meaning on its own; only the VM to host total ratio matters for determining CPU resource allocations above the reservation amounts
- Shares are only used when VMs are actually competing for the same resources on the host

VM Storage Planning

- **Virtual SQL Server require the same I/O considerations as physical SQL Servers**
 - Pre-deployment I/O testing is even more critical for virtualized SQL Servers than it is for physical servers because of the limitations imposed by the hypervisor for key factors like multi-pathing
 - Focus should be on storage performance (IOPS), throughput (MB/sec) and latency to ensure that the VM configuration will meet the SQL Server requirements
- **Be sure to test the correct I/O patterns for device workloads**
 - Log files 60K sequential I/O
 - Data files 64K random I/O

VM Storage Planning

- **Physical segregation of LUNs per VMDK should be enforced where necessary to ensure that data files and log files are physically isolated I/O wise and not on shared LUNs by the hypervisor**
- **Raw Disk Mappings**
 - RDMS are necessary for clustered configurations under VMs
 - May allow an easier migration path from VM to physical if the VM requirements exceed the current limitations of a single virtual machine under the selected hypervisor
 - May provide an easy physical to virtual migration if planned correctly and the virtual disk can be mounted to allow the database to be attached

VM Storage Planning

Partition Alignment

- Partition alignment still matters!
- VMFS volume alignment
- CSV volume alignment

Virtualization High Availability Basics

- Clustered VM hosts provide high availability against hardware failure of the physical host for any guest running inside of the cluster
- Host failure may result in temporary guest failure, but the guests can be restarted on another host in the cluster restoring service immediately
- VMotion and Live Migration can be used to migrate guests between hosts to perform host maintenance at any time

Virtual High Availability (Reality)

- **The high availability provided by virtualization doesn't begin to compare to the high-availability solutions that are provided by SQL Server**
 - Downtime is still required for SQL Server service pack and cumulative update application
 - SQL Server 2008 rolling update application to failover clustering and database mirroring configurations will keep the instance available while the passive node or mirror is patched prior to failover
 - Automatic failover in the event of a host hardware failure is not guaranteed if guest reservations exceed the available reservations on the failover host
- **There is still a case for SQL Server based high availability if business requirements or SLAs dictate**

Virtual Machine Failover Clustering

- **Supported on SVVP certified hypervisors following vendor recommended practices for MSCS cluster configurations**
- **Adds an additional layer of complexity into troubleshooting cluster failover problems if they should occur**
 - May require coordination of Microsoft PSS with VM platform vendor and/or storage vendor to pinpoint the source of cluster failover problems
- **Over a dozen SQLskills customers in the last six months have engaged for validation or troubleshooting of VM-based cluster configurations**

VMware Host Counters (ESXTOP)

- **%RDY time**
 - If physical CPUs are busy: over-commitment, contention between VMs
 - If physical CPUs are idle: affinity conflicts or co-scheduling requirement problems
- **%CSTP**
 - Co-scheduling overhead: consider reducing the number of vCPUs or moving to less busy host
- **%WAIT or %IDLE**
 - Workload is not CPU intensive
 - Increasing entitlements will not improve performance
- **%MLMTD**
 - CPU limit set too low?

VMware PerfMon Counters

- VMware ESX 3.5 (manual download)
- VMware vSphere/ESX 4+ the counters are built-in
- No additional permissions are required
- Most VMware admins don't know they exist
 - VMware admins can't prevent you from seeing these counters from the guest if the VM tools are installed

New VMware PerfMon Counters

- **% Processor Time**
 - This is the host level % usage
- **Host Processor Speed (MHz)**
 - DRS or VMotion can move the guest from slower host to faster host and vice versa so you need to know when this occurs
- **Limit (MHz)**
 - VM admin can limit your CPU access to a slower amount of processor speed
- **Reservation (MHz)**
 - VM admin can guarantee you a set amount of CPU access

VMware Memory Counters

- **Memory Limit (MB)**
 - Maximum amount of memory the VM can use despite the configured memory for the VM
- **Memory Reservation (MB)**
 - Guaranteed amount of memory for the VM
- **Memory Ballooned (MB)**
 - Should always be 0 for SQL Server
- **Memory Swapped (MB)**
 - Should always be 0 for SQL Server
 - This is disk swapped memory for the VM and should be a flashing red error on the host in vCenter

Hyper-V Host Counters

CPU Usage Monitoring

- Under Hyper-V, the host operating system is logically another guest OS in the environment
 - Standard performance counters like “\Processor(*)\% Processor Time” only report information about the host OS usage and the totals for the physical hardware
- To measure the total physical processor utilization of the host operating system and all guest operating systems, the “\Hyper-V Hypervisor Logical Processor(_Total)\% Total Run Time” counter must be used
- “\Hyper-V Hypervisor Logical Processor(_Total)\% Total Run Time” < 75%

Hyper-V Host Counters

CPU Usage Monitoring

- All of the system interrupts are scheduled on logical processor 0 (LP0) in Hyper-V
- LP0 “%Total Run Time” should be monitored to ensure that it is not 100% utilized which can be signs of I/O bottlenecks

Hyper-V Host Counters

Memory Usage

- “\Memory\Available Mbytes” > 25%
 - Total amount of free memory available on the system for powering on additional VMs or memory growth under Dynamic Memory
- “\Memory\Pages/sec” < 500
 - Measures the rate at which pages are read from or written to disk to resolve hard page faults

Review

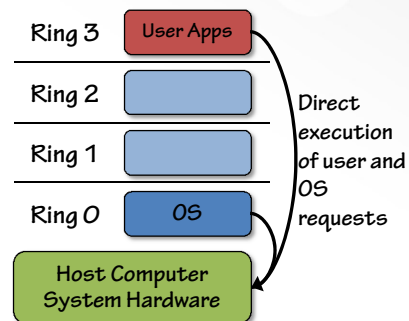
- Why virtualization is popular
- How virtualization works
- Considerations for SQL Server
- HA/DR options
- Monitoring

Appendix

- **x86 virtualization**
 - Levels of privilege
 - Full virtualization using Binary Translation
 - Para-virtualization: OS-assisted virtualization
 - Hardware-assisted virtualization

X86 Architecture Levels of Privilege

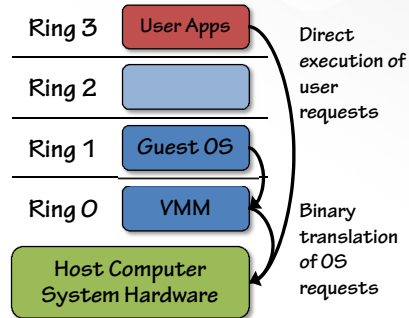
- The x86 architecture offers 4 levels of privilege to operating systems and applications to manage access to the hardware
- User applications typically run in Ring 3
- Since the operating system needs direct access to memory and hardware it executes at Ring 0
- Virtualization places a virtual layer under the operating system to manage the hardware



X86 Virtualization

Full Virtualization using Binary Translation

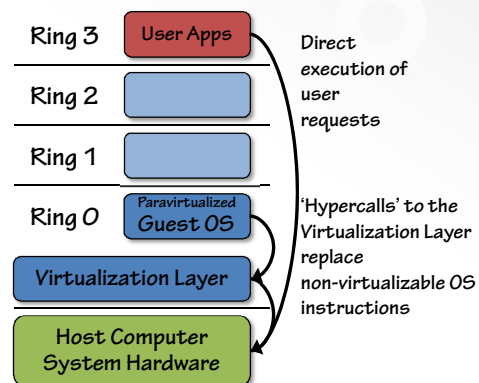
- Virtual Machine Monitor (VMM) provides each VM with physical system services
 - Virtual BIOS, CPUs, memory management, and devices
- User Level code continues to be directly executed on the physical processor for high performance
- VMM translates kernel code to replace non-virtualizable instructions with new instructions providing the same effects
- Guest OS is not aware it is virtualized
- VMware Server and Microsoft Virtual Server



X86 Virtualization

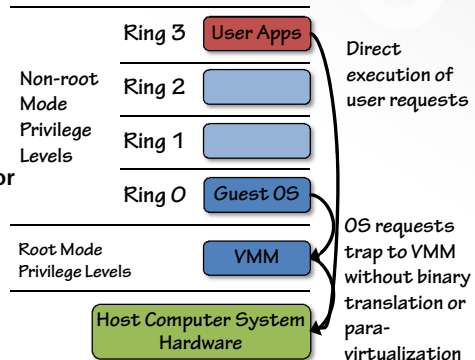
Para-virtualization: OS-Assisted Virtualization

- Guest OS and hypervisor communicate to improve performance
- OS kernel is modified to replace non-virtualizable instructions with hyper-calls directly to the hypervisor virtualization layer
- Hypervisor provides interfaces for kernel operations such as memory management, time keeping, and interrupt handling
- Can not support unmodified OS (Windows 2000/XP)
- VM tools and device drivers provide para-virtualization



X86 Virtualization Hardware-Assisted Virtualization

- Hardware allows the VMM to run in a new root mode below Ring 0
- Intel Virtual Machine Control Structures (VT-x) and AMD Virtual Machine Control Blocks (AMD-V)
- Privileged and sensitive calls automatically trap to the hypervisor, eliminating the need for binary translation or para-virtualization
- Required for 64 bit guest support
- VM tools and device drivers still installed on the guests but not to provide para-virtualization



Questions?