

MOVING SQL SERVER TO A VIRTUAL PLATFORM

BY ROBERT L. DAVIS

TABLE OF CONTENTS

Summary	3
Choosing Virtual	4
Non-Critical Workloads	4
Small Footprint Servers	5
Production Workloads	6
Best Practices for SQL Server on a Virtual Platform	7
Licensing Considerations	9
VM Licensing vs Server Licensing	10
Conclusion	11
SQL Diagnostic Manager for SQL Server	12

SUMMARY

This whitepaper is a guide for database administrators (DBAs) on moving SQL Server from a physical environment to a virtual environment.

Virtualization has become a major topic for Database Administrators. In the early days of virtualization, SQL Server was not considered suitable for virtualizing. However, the virtual platforms have matured, and virtualizing SQL Server is a widely accepted practice. Virtualization is now a critical part of enterprise environments to varying degrees.

Virtualization also plays a major role in consolidation, especially in large enterprises. Virtual machines (VMs) offer a way to merge many small physical servers into a few large servers while still maintaining separation of the workloads. A well tested, successful approach to virtualization is to start slow and grow your virtual platform. Combine this approach with proper planning and good standards and practices to ensure a successful move to VMs.



CHOOSING TO VIRTUALIZE

There are several ways that DBAs are working virtual machines into their environments. Non-critical workloads, small footprint servers, and even critical production workloads are running on virtual platforms. Some decisions to virtualize should be easy, while others should require careful deliberation and testing.

NON-CRITICAL WORKLOADS

The earliest, and easiest, implementations of VMs were for non-critical workloads. Non-critical workloads allow us to start small and scale up if needed. DBAs new to virtualization often start by virtualizing development, test, and other non-production SQL Servers. It is acceptable for these environments to endure reduced performance in favor of lower operational costs and simplified management.

There are benefits to virtualizing non-production workloads other than financial, such as snapshot environments and quick delivery of new servers.

The ability to snapshot a server at a specific point in time allows testing of deployments or configuration changes and then reverting back fast to a known good state. To revert complex environments to a known state may often take many hours or even days, depending on the size of the environment. This can decimate the time required for multiple runs of a test process.

One challenge of working with physical servers is it can take a long time to deliver new servers. With physical environments, you have to plan projects that require new servers many months in advance. In large environments, many infrastructure teams purchase hardware in blocks are set times in the year. It is not uncommon for the delivery of new servers to require a lead time of six months or more. A virtual platform, on the other hand, can deliver a new virtual machine (VM) in a matter of minutes. Automation tools like Microsoft's System Center Orchestrator have made the rollout of new virtual machines with SQL Server and any other software needed as simple as clicking a button.

SMALL FOOTPRINT SERVERS

Another straightforward decision to make in the early days of virtualization was to move production servers with small footprints onto VMs. Most businesses have several internal applications that must be segregated from other workloads because they contain sensitive data such as employee and financial information. These applications often have a small user base but high criticality for being available during business hours. The applications were often placed on hardware that was much more powerful than was needed. Another prime example of a small workload server would be a disaster recovery (DR) server used only for DR purposes and not as a highly available fail-over server.

A common alternative to virtualization for small footprint servers is to combine multiple instances of SQL Server onto the same base server. This approach requires careful management, or it can lead to major problems. SQL Server believes it is the most important application on the server and does not concern itself with the resource needs for other applications.

If you choose to go the multi-instancing route, it is crucial that you segregate the SQL Server instances as much as possible. Here are some tips for successful multi-instancing of SQL Server:

- Include the additional overhead of operating SQL Server when calculating the amount of memory that needs to be left free for the operating system and non-buffer-pooled processes
- Limit MAX SERVER MEMORY for the instances to ensure that there is enough memory for each instance to get its full allocation and enough memory is left free.
 - If you experience failures to allocate memory, it is an indicator that the max memory setting is not configured appropriately.
- Use separate dedicated drives for each instance of SQL Server to ensure that a sudden, high input/output (IO) workload on one instance cannot hinder performance of the other instances.
- Use Windows System Resource Manager to manage central processing units (CPU) per instance.
 - Deprecated in Windows Server 2012 in favor of using virtualization

Virtualization still requires careful planning, but it is segregated by nature and a lot of the things you have to take care to implement with multi-instancing are inherent in VMs. You are still combining multiple servers onto a single large piece of hardware, but each SQL instance would be contained within its own operating system. Each VM is assigned its own block of memory and CPUs, and each VM has its own file system on virtual hard drives (VHDs).

VMs have a certain amount of segregation built-in. However, they are still sharing the same resources on the host server. Careful planning is required for implementing multiple VMs to make sure that one VM cannot overload the system and cause performance problems for the other VMs on the same host. I will cover best practices for virtual platforms a little later in this paper.

PRODUCTION WORKLOADS

Both Microsoft and VMWare, as well as other smaller virtualization vendors, are pushing how the virtual platform performs. In a few years, we have gone from thinking that VMs were not suited for critical production workloads to using it for critical, heavily transactional production systems. The platform has matured at a rapid rate and has seen adoption in production grow at a high rate.

At the same time that the virtual platform was maturing, so was the knowledge base of virtualization administrators. A lot of mistakes were made early in the virtual platforms, but the industry has learned from those mistakes and created better virtualization products and better practices for ensuring a successful implementation.

Virtual platforms can provide VMs with large amounts of virtual CPUs (vCPUs) and memory, and they provide new options for availability that you cannot get without virtualization. Hyper-V and VMWare both provide technology for moving a live VM from one host server to another without downtime. This ability to move a live server across the host server is called Live Motion in Microsoft's Hyper-V and vMotion in VMWare. This can provide high levels of availability at the VM level without having to implement additional logic at the SQL Server or application level.

Technologies like Windows Failover Clustering have been extended to support virtualization to provide different options for incorporating clustering into your high availability plan. Clustering can be implemented for the VM host or at the VM level, called guest clustering. Guest clustering requires that all VM nodes exist on the same VM host. Unlike normal SQL Server clustering, guest clustering of SQL Server does not eliminate the single point of failure of running on a single server. It provides protection against failure of a single VM or SQL Server instance, but if the VM host goes offline, the entire cluster goes offline. Guest clustering has not received widespread adoption because it is often considered that the complexity of implementing guest clustering outweighs the benefits it provides.



It is important that you test and validate that the platform is suitable for your workload as you consider moving a critical production server onto a virtual platform. Migrating to a VM is not as simple as assigning it the same number of vCPUs and amount of random access memory (RAM) as the current physical hardware. In most cases, the VM will share a VM host with other VMs and care must be taken to ensure that best practices are followed. Any change in the platform should be well tested before it is put into production use, and this is particularly true when moving to a VM for the first time.

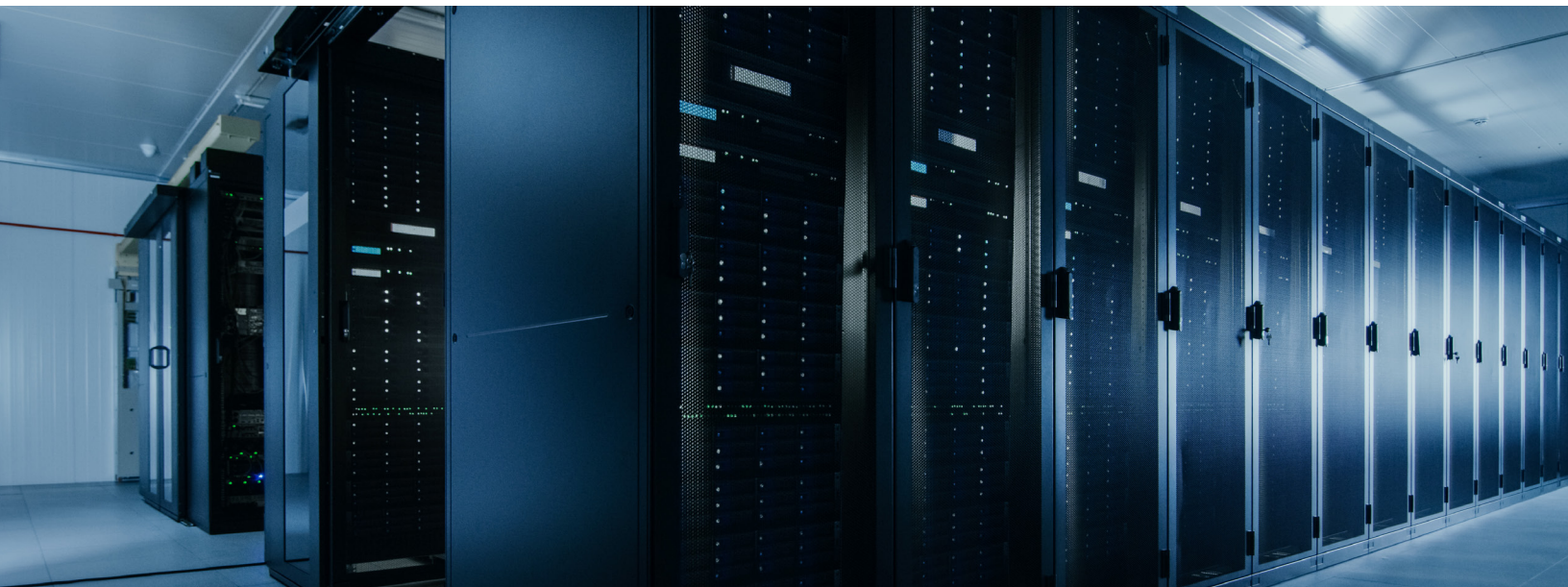
BEST PRACTICES FOR SQL SERVER ON A VIRTUAL PLATFORM

For running SQL Server on a virtual platform, it is not sufficient to follow simply best practices for virtualization. There are additional considerations that must be made for SQL Server that may not apply to VMs that host other applications or workloads. For example, it is sometimes considered okay to over-commit resources for VMs that support variable workloads, like web applications, but I recommend never over-committing resources to SQL Server VMs.

Non-critical or light workloads do not require the same level of diligence and strict adherence to best practices. For the sake of brevity, I will focus the best practices discussion on critical production SQL Server workloads. The following best practices will help ensure a successful virtual platform for critical production SQL Server instances.

- Configure the host appropriately: Ensure that the VM host is configured properly for handling large workloads.
- High performance power plan: Enable high performance plan on the VM host and in the VM.
 - You may save energy costs by segregating VMs that can use a balanced power plan to a different host so that the host can be throttled. SQL Server VMs should never have the CPU power throttled.
- Adequate drive arrays or LUNs on host: Ensure adequate drive arrays or LUNs on the host so that different servers can be placed on separated drives as needed.
 - Low activity or non-critical VMs can share host drives as the IO capabilities of the drive permits, but care should be taken to monitor the IO impact and move files if needed.
 - Critical, high-activity SQL Server VM files should not share drives at the host level because it is easy for a single over-loaded system to overwhelm other systems on the same host drive.
- Do not over-commit: Do not over-commit or over-allocate resources to SQL Server VMs.
 - If you license the VM host for unlimited SQL Server VMs, remember that “unlimited” is a licensing term only. It does not apply to the actual capabilities of your VM host.

- Monitor the host: Monitor the VM host and the individual VMs. If a performance issue cannot be explained by the counters on the VM, look at the counters on the VM host.
 - For example, if CPU usage is high on the VM host, even a lightly used VM can show signs of CPU pressure, though the local CPU counters may be low.
- Segregate SQL Server file types on host: Follow SQL Server best practices regarding storage for SQL Server VM files.
 - VM files (VHDs) for database data files and log files should be segregated into different files, which are segregated into different dedicated host drives.
 - VHDs hosting tempdb files should be separated.
 - VHDs for the backup drive should be separated.
 - VHDs for the system drive and SQL installation drive (C and D, respectively) should be separated from the other VHDs but may be co-hosted with other system drive VHDs from other VMs at the host level.
 - Any other SQL Server files you would segregate on a physical system, such as the database files for a heavily used replication distributor, should be separated at the VM host level as well.
- Segregate SQL Server file types on VM: Even if the initial design is to host the VM files on a combined drive on the host, the SQL Server files should still be segregated to different VHDs. This is required in case the need arrives to separate the files to different drives later. This would enable the files to be migrated without reconfiguring the VM or SQL Server.
- Network interface controller (NIC) Teaming: Use Windows built-in NIC teaming with resiliency to provide maximum throughput and reliability of network throughput.
 - Consider compressing the tabular data stream (TDS) or network packets to reduce the networking load.
- Use most current operating system (OS): Windows keeps making strides in its support for virtual platforms. Though it is always wise to use the most current version of the operating system, it is even truer for virtual platforms.
 - Non-critical and small footprint workloads may be alright on older versions of the OS, but you would really do yourself a disservice if you skimp on the OS for critical production workloads.



- Commit resources for SQL Server workloads: Ensure that critical servers have resources committed to them. Do not make the VMs compete for resources.
- Disable memory ballooning: Memory ballooning tries to reclaim memory from SQL Server by falsely signaling low memory to SQL Server to prompt it to release memory. You should disable ballooning for critical production SQL Servers.
- Enable Lock Pages in Memory: Experts disagree on the topic of granting Lock Pages in Memory permissions to SQL Server for physical servers. For virtual servers, it is important that SQL Server can lock pages so that it can protect itself from outside pressure.
- Do not cross non-uniform memory access (NUMA) boundaries: For NUMA (non-uniform memory access) architecture, memory is local to each CPU. If you need to cross NUMA boundaries to access memory for a different node, there is a small performance hit.

If the number of vCPUs you are allocating is less than the number of CPUs per NUMA node, the allocation should come from a single node. If you need to allocate multiple nodes, then present them at the VM layer as separate NUMA nodes the same as they are on the host.

- Used fixed size, static VHDs: Assign fixed sized VHDs. Do not use dynamic disks or thin provisioning for SQL Server files. SQL Server is a heavy disk IO application and introducing repeated waits for IO while storage makes provisioned space available can have a large impact on performance.

Besides not using thin provisioning at the VM layer, storage from a storage area network (SAN) or other storage device should not use thin provisioning either.

- When in doubt, test it: Any time there is a configuration where you are unsure of the correct setting, test it before going into production. This advice is a good best practice for just about everything related to SQL Server.
- Follow standard SQL Server best practices: Except where noted differently here, follow standard best practices for configuring SQL Server. If the VM and MV host are configured properly, you should be able to configure and maintain SQL Server the same way you would on a physical server.

LICENSING CONSIDERATIONS

SQL Server licensing is a complex subject, and at best, it is a moving target. When in doubt about SQL Server licensing, I highly recommend contacting a licensing professional to make sure you have the most current standards. Because of the complexity of SQL Server licensing, I am only going to cover the key components that pertain directly to SQL Server in a virtual platform.

Besides the well-known switch to per-core licensing for SQL Server introduced with SQL Server 2012, Microsoft made changes recently to the rules for fail-over rights for SQL Server. Introduced with SQL Server 2014, you can no longer receive a free license for a passive fail-over simply by licensing the active server. Fail-over rights now require Software Assurance (SA). Likewise, beginning with SQL Server 2012, mobility rights also require an SA agreement.

VM LICENSING VS. SERVER LICENSING

Outside the specific scenarios where you may want to use Server and clients access license (CALs), there are two basic ways to license SQL Server in a VM. You can license the total number of cores on the VM host for unlimited SQL Server VMs or you can license individual VMs for the number of cores that are presented to them.

The advantages of licensing all cores on the VM host are that it simplifies managing SQL Server licenses. You can create, destroy, and move VMs from server to server without the need to track individual licenses. This includes rights to move a VM from one server to another using vMotion or Live Migration without purchasing extra licenses as long as all hosts involved are licensed for all cores.

To license all cores of the host requires licensing Enterprise Edition and Software Assurance (SA). Though downgrade rights would allow you to run Standard Edition under this model, you would still have to pay for all Enterprise Edition licensing. Clearly, you would only use this option under the following conditions:

- You needed to use Enterprise Edition of SQL Server.
- The host would be dedicated to VMs that run SQL Server only. To use it for non-SQL Server workloads would be an expensive usage for the core licenses.
- The number of VMs that would be hosted exceeds the licensing requirements for per-VM licensing.

One of the enormous advantages of licensing individual VMs is that you can license Standard Edition or Enterprise Edition for individual VMs. This allows you to license whichever edition you need and can save you a lot of money. A major downside to per-VM licensing is that you must license a minimum of 4 cores for a server. Licenses are purchased in blocks of two after an initial purchase of four core licenses for a VM. If you have a lot of small footprint or non-critical SQL Server instances that only need one or two cores, purchase at least four core licenses for each VM.

Here are some other key things to consider when weighing the choice of per-VM licensing:

- VM licenses tied directly to the VM host server. If a VM that is migrated to a different host, the SQL licenses do not move with the server and the VM has to be licensed again on the new server.
 - Mobility rights are conveyed with SA. If you have an SA agreement, the licenses for the VM can be moved with the VM.
- VMs that are licensed individually using the per-core model can mingle freely with VMs licensed under the Server client access license (CAL) model.
- VMs that host non-SQL Server workloads can freely mingle on the same VM host as SQL Server VMs.
- Hyper-threading is free in individual-VM licensing. When using hyper-threading, you only buy core licenses for the number of hardware threads. You do not pay extra licenses for the extra threads seen because of hyper-threading.

It seems pretty clear that the default licensing model for licensing VMs should be individual VM licensing. To license the VM host for all cores is an expensive way to license VMs and is appropriate only for specific situations where you need to use all cores for SQL Server VMs running under Enterprise Edition.

CONCLUSION

To decide to move SQL Server to a virtual platform the first time can be a nervous time, but it doesn't need to be. With proper planning and best practices to guide you, it is pretty straightforward to have a successful migration to virtual machines. You can use the methods laid out in this paper to wade into virtualization slowly with non-critical workloads and small footprint servers before moving on to considering critical production workloads.

You can move SQL Server to a virtual platform with ease using these guidelines. Don't make the same common mistakes that others have made. Use the best practices outlined here to guide you and help you avoid these pitfalls. The licensing information provided should make the complex licensing rules a lot clearer and help you make the right licensing decisions for virtualizing SQL Server.

SQL DIAGNOSTIC MANAGER FOR SQL SERVER

Reduced availability and performance of Microsoft SQL Server can severely affect the critical applications that it supports.

SQL Diagnostic Manager provides robust SQL Server monitoring functionality that covers how the entire SQL Server environment performs and provides the most comprehensive diagnostics on the market.

- Monitor physical, virtual, and cloud environments.
- Track queries and plans to fix blocks and locks.
- Alert predictively and avoid false alerts.
- View expert advice with executable scripts.

Start for FREE

