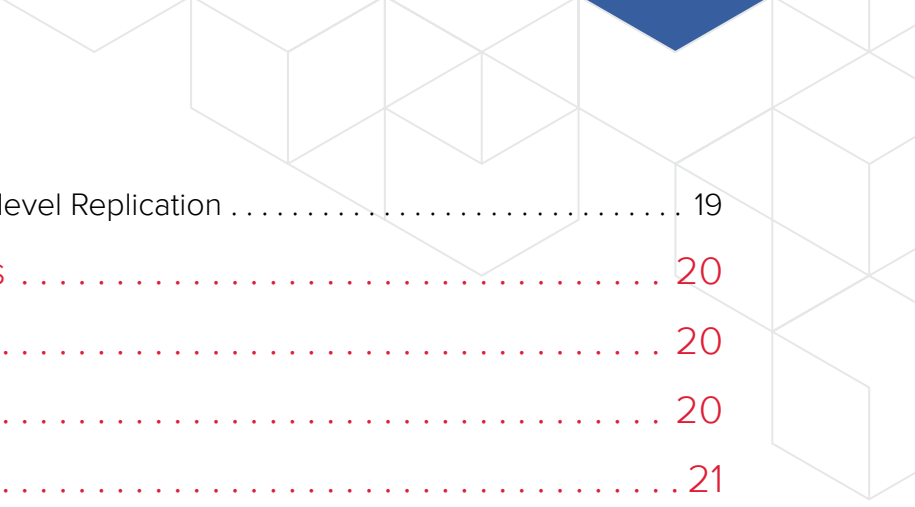


SQL Server Best Practices on Tintri – Stand-Alone Instances

Version 1.0

Contents

Intended Audience	4
Executive Summary	4
Assumptions	4
Microsoft SQL Server Overview	4
Typical I/O characteristics of a SQL Server database	4
Virtualizing SQL Server Databases – Overview.....	5
Summary of Tintri-specific storage recommendations:.....	7
Configuration Details.....	8
Virtual Machine Settings.....	8
Virtual Disks & Virtual SCSI Adapters	8
vDisk Sizing	10
VMware vCenter Settings.....	11
HA Cluster Settings	11
VM and Application Monitoring	12
DRS Cluster Settings	13
Use vApps to control startup order	14
Guest OS Settings	14
Disk Partition Alignment.....	14
VMware Tools	15
Page File Settings.....	15
SQL Server & Database Configuration Settings.....	16
Data Protection.....	17
Backups	17
SnapVM™ - Tintri Snapshots	17
Crash-Consistent Tintri Snapshots	18
VM-Consistent Tintri Snapshots	18



ReplicateVM™ – Tintri VM-level Replication	19
Maintenance Activities	20
Scheduling	20
Understand	20
Analyze	21
Monitor	21
Performance Testing	21
Dependencies	22
Rinse & Repeat	22
References	22
Tintri Links	22
SQL Server Specific Links	22
Other Links	23

Intended Audience

This SQL Server Best Practices Guide on Tintri is intended to assist SQL DBAs, IT Administrators and Architects who are responsible for the running SQL Database servers within their virtual infrastructures powered by Tintri VMstore™ storage appliances. In-depth knowledge of Virtualization technologies is not a pre-requisite and this guide is intended to be a useful resource for SQL DBAs, regardless of whether their existing SQL servers are virtual or physical.

Executive Summary

As virtualization first made its appearance into corporate datacenters, many admins were skeptical that mission critical servers, such as Microsoft SQL database servers, could be successfully virtualized without suffering from a substantial loss in performance compared to physical server deployments. Virtualization technology has matured significantly over the years and mission critical servers can safely be [deployed as VMs](#). The paper However, caution should be taken to ensure your VMs are setup correctly, just as you would ensure that physical servers and their components were configured for performance and scalability.

One can find many best practice documents and also tips and tricks for virtualizing SQL Server instances and to get the most out of a SQL Server VM. While most advice is helpful and applicable, some simply no longer applies when virtualizing your SQL Servers on a Tintri VMstore, especially those that contain advice regarding storage configuration. A Tintri VMstore is smart storage that sees, learns, and adapts in a virtualized world, and the goal of this document is to provide you with information to take full advantage of everything a Tintri VMstore has to offer for virtualizing Microsoft SQL Server deployments.

Assumptions

This document assumes you are working with a fully configured, highly-available Virtual Infrastructure. Configuration of the hosts, networking, storage, and virtual infrastructure components are out of scope of this document. Screenshots in this document are based on VMware vSphere 5.1, but similar options may exist for other versions of vSphere, or different hypervisors altogether.

Microsoft SQL Server Overview

Microsoft SQL Server is a relational database solution accommodating both management and analysis for data to day operations and data-warehousing solutions. Microsoft SQL Server has evolved over the course of the last 15 years, starting with SQL Server 2000 the focus enveloped Business Intelligence including transform, extract and ETL, additionally Online Analytical Processing (OLAP) and Reporting Services were included. SQL Server 2012 introduced application level HA with AlwaysOn Capabilities including Server Failover Instances and Availability Groups. SQL Server 2014, the latest version, includes in-memory Online Transaction Processing (OLTP) databases.

There are primarily three file types for SQL server Databases including:

- Primary Data Files – MDF extension – each database requires at least one MDF
- Secondary Data Files – NDF extension – Not necessarily required, databases can have none or many secondary files
- Log files – LDF extension – Hold all the transactional information needed to recover a database

Typical I/O characteristics of a SQL Server database

There are two types of generic SQL Server database workloads, OLTP and OLAP and these workloads typically produce different IO access patterns.

OLTP databases produce numerous concurrent transactions with random reads and writes (IOPS). Typical OLTP database workloads characteristics are:

- Both reads and writes issued against data files are generally random in nature
- Read activity is constant

- Write activity to the data files occurs during checkpoint operations
- Log writes are sequential
- Log reads are sequential

OLAP databases and data warehousing systems generally produce sequential reads and a small amount of write activity, except for when batch updates are taking place. Bandwidth is far more critical than IOPS in this scenario.

Microsoft best practices describe OLAP database workloads as the following:

- Sequential reads and writes – bulk insert operations and index scans
- High TempDB activity
- Varied I/O size typically larger than 8K
- Column store indexing IO size typically more than 256KB

Summary of typical I/O patterns for different types of database workloads:

IO Type	OLTP	OLAP
Data Files	<ul style="list-style-type: none"> • High read I/O – typically 70/30 to 90/10 R/W I/O workloads • Small random I/O – typically 8KB-64KB 	<ul style="list-style-type: none"> • Larger block size sequential I/O • Mostly read I/O workload except in batch upload
Log Files	<ul style="list-style-type: none"> • Smaller block size sequential IO workload • Mostly Writes 	
TempDB Files	<ul style="list-style-type: none"> • Small I/O, typically <64KB • Mostly random I/O with occasional sequential I/O • 50/50 R/W 	

Virtualizing SQL Server Databases – Overview

When deploying Microsoft SQL Server within Virtual Machines, you should adhere to vendors' best practices, most specifically those made available from Microsoft and VMware. It is important to keep in mind that "Best Practices" are generalized to meet most common use cases and recommendations are made without specific knowledge of unique variations in your environment, like the fact that you are deploying on Tintri storage. Information in this document will contradict some recommendations made within other best practice docs, but keep in mind that this document is list of recommendations made for the most typical deployment scenarios when using Tintri VMstore storage system, which may not be fully applicable in a customized environment.

With the exception of many storage-specific recommendations, which cater to more traditional storage technologies, VMware's "SQL Server on VMware Best Practices Guide" provides an excellent base for successfully virtualizing SQL Server and is applicable when deploying SQL Server on Tintri VMstore users as well:

http://www.vmware.com/files/pdf/solutions/SQL_Server_on_VMware-Best_Practices_Guide.pdf

As a general overview, VMware's blog included a great "cheat sheet" reference for SQL:

<http://blogs.vmware.com/apps/2012/12/best-practices-for-virtualizing-sql.html>.

We've taken the image from their site and omitted the storage section, which we'll replace with storage recommendations that are tailored specifically for Tintri VMstores:







SQL	<p>Virtual CPU</p>  	<ul style="list-style-type: none"> • Understand existing workload, avg., and peak • Properly manage pCPU allocation <ul style="list-style-type: none"> - For Tier 1 workload, avoid pCPU over-commitment - For lower-tiered databases workload, reasonable over-commitment can increase aggregate throughput, and maximize license savings; leverage vMotion, DRS for resource load balancing • Monitor to optimize <ul style="list-style-type: none"> - Host level: %RDY, %MLMTD, and %CSTP - VM Level: Processor Queue Length • Keep NUMA node size in mind <ul style="list-style-type: none"> - For smaller VM, allocate vCPUs to <= cores in a physical node - For wide VM (ESX 5), align vCPUs to physical NUMA boundaries, enable vNUMA on ESX to allow SQL Server NUMA optimization
	<p>Virtual Memory</p>  	<ul style="list-style-type: none"> • Do not over-commit memory until VC reports that steady state usage is below the amount of physical memory on the server • For critical SQL Server workloads, consider setting memory reservation = allocated memory to avoid ballooning or swapping • If over-committing memory, monitor SWAP /MB: r/s, w/s and MCTLSZ • Use Active memory counter with caution; always confirm usage by checking memory counter in perfmon • Do not set memory limits for memory-intensive applications
	<p>Networking</p>  	<ul style="list-style-type: none"> • Separate management traffic (vMotion, FT logging, VMKernel) and virtual machine traffic • Separate NICs assigned to separate vSwitches • Use VLAN Tagging (802.1q) • Allocate at least two NICs per vSwitch to leverage NIC teaming capabilities • Use the VMXNET3 para-virtualized NIC. VMXNET3 is optimized for virtual environments and designed to provide high performance • Consider installing MS KB2344941 if using Windows 2008 R2-based templates • Follow the networking design guidelines in VMworld 2010 session TA8595 - Virtual Networking Concepts and Best Practices - this includes designs to efficiently manage multiple networks and redundancy of network adaptors on vSphere hosts

Figure 1 - Overview of VMware's suggested best practices (excluding Storage).

Summary of Tintri-specific storage recommendations:

Tags	Recommendation
Virtual Disks & Virtual SCSI Adapters	"DO: Use many virtual disks configured on multiple Paravirtual SCSI adapters (PVSCSI)," on page 9.
Virtual Disks & Virtual SCSI Adapters	"DO: Locate all virtual disks within the same Tintri datastore." on page 9.
Virtual Disks & Virtual SCSI Adapters	"DO: Use Thin-provisioned disks" on page 9.
vDisk Sizing	"DO: Use unique, slightly different sized disks to avoid confusion when creating virtual disks." on page 11.
vDisk Sizing	"DO: Ensure to provision additional capacity when creating virtual disks as there is no wasted space with thin provisioned disks." on page 11.
vDisk Sizing	"DO: Monitor disk capacity of the entire VMstore to prevent running out of space." on page 11.
vDisk Sizing	"DO: Monitor guest file system capacity and extend vDisks as need to prevent running out of space." on page 11.
HA Cluster Settings	"DO: Set VM Restart Setting to High on SQL servers to ensure these VMs are prioritized in the event of a host failure." on page 12.
DRS Cluster Settings	"DO: Create a rule to prevent SQL server VMs that are members of the same AlwaysOn Availability Group (AAG) from running on the same host." on page 13.
Use vApps to control startup order	"DO: vApps are an easy way to ensure the correct startup/shutdown order, so use them for multi-tiered SQL Server applications." on page 14.
VMware Tools	"DO: Make sure VMware Tools is installed within your VM." on page 15.
Page File Settings	"DO: Make sure the page file is on the disk intended for it" on page 16.
Page File Settings	"DO: Configure Paging File as "System Managed" to automate page file size increase when additional vRAM is added to VM." on page 16.
VM-Consistent Tintri Snapshots	"DO: Use Tintri native snapshots for protecting SQL Server instances." on page 19.
ReplicateVM" – Tintri VM-level Replication	"DO: Use Tintri ReplicateVM for data protection and disaster recovery for protecting SQL Server instances." on page 19.
Scheduling	"DO: Schedule maintenance tasks within time periods that are least obtrusive to your business operations." on page 20.
Scheduling	"DO: Coordinate maintenance tasks with backup & snapshotting schedules." on page 20.
Understand	"DO: Ensure that unnecessary maintenance tasks no longer run for saving precious resources." on page 20.
Analyze	"DO: Use VMstore UI to monitor at VM and virtual disk level IO patterns to uncover and troubleshoot performance issues." on page 21.
Performance Testing	"DO: Run periodic performance tests to evaluate changes over time and proactively address concerns." on page 21.
Dependencies	"DO: Look at the environment holistically – host, networking and storage – when troubleshooting." on page 22.

More detail on each of these is provided within the Configuration Details section of this document.

Configuration Details

Traditionally, the success of a mission-critical system depended on how it was built from the ground-up, including hardware configuration settings. In virtual instances of these critical servers, such as Microsoft SQL Database servers, many of the best practices from the physical world still apply. This section is a list of best practices that still make sense in a virtual world, and others that are unique only to virtual infrastructures.

Virtual Machine Settings

The Virtual Machine settings can make a big difference on the performance and scalability of your SQL servers. From the Guest's perspective (i.e. Windows), the VM configuration is experienced the same as if it were a physical server... the OS sees RAM, CPUs, NICs, etc. This section contains advice how to configure the various elements that make up your Virtual Machine.

Virtual Disks & Virtual SCSI Adapters

When it comes to SQL servers (physical or virtual), best practices associated with traditional storage would have you separate disks according to their role, which each disk with its own inherent IO behavior, performance needs, and requirements for isolation to ensure adequate and predictable response times. These different dedicated drives are typically divided into: Operating System (OS), Page File, Database Files, Transaction Logs, Temporary DB/Logs, and Backup. The storage underpinning each of these is typically presented as a number of different LUNs, each of which is configured with unique RAID types, performance policies, mappings to physical drive spindles, etc. For a Virtual Machine, each virtual disk (.vmdk file) associated with the different data types would reside in different datastores so a typical VM ends up spanning many datastores and sometimes across many storage arrays.

Using traditional storage, multiple datastores are required, each tailored to their specific IO pattern:

- **Data.** Each different database performs either random or sequential I/O access patterns
- **Log.** Log access is mostly sequential. The database writes data to the log, and then writes the data into the database.
- **Temporary Database (TempDB).** TempDB is used when the system needs to perform temporary database operations. TempDB access is assumed to be highly Random.
- **Backup.** Database backup files are written to this separate backup vDisk, typically into a single .BAK file per database. A typical use case is the export of a database into a .BAK file, which is then copied to a Development or Test system to import the latest production data. Backup devices are accessed sequentially.

The recommendation to separate various data types into multiple virtual disks is applicable when virtualizing SQL Server on Tintri VMstore. **However, unlike traditional storage where you would need to make sure the virtual disks are located in multiple datastores spanning multiple arrays at times, we recommend to put all the virtual disks on the same Tintri datastore.** This deployment greatly reduces administrative complexity and Tintri's architecture was built from the ground up to dynamically optimize IO at a per-VMDK level of granularity.

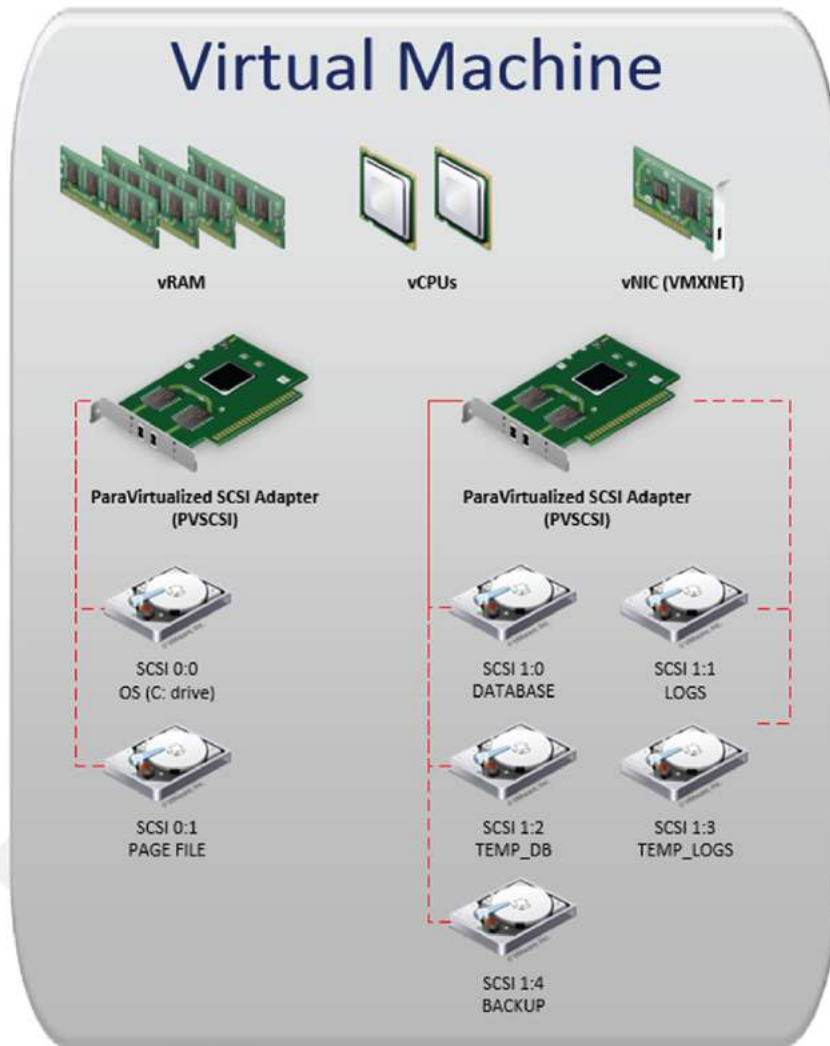


Figure 2 - A logical representation of an optimized Microsoft SQL Server VM.

DO: Use many virtual disks configured on multiple Paravirtual SCSI adapters (PVSCSI).

DO: Locate all virtual disks within the same Tintri datastore.

DO: Use Thin-provisioned disks

For more information on how to configure a VM to use PVSCSI adapter, refer to: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1010398

On the surface, this may seem over-complicated, especially if we compare this configuration to the most basic VM that is configured with a single vDisk. Here are some benefits of this deployment model:

- **Better Visibility** – Granular “per-vDisk” performance, latency, space, and other real-time & historical metrics available for analysis.

- **Better Performance** – SCSI Disks and Controllers (both physical and virtual) each have a finite queue depth for IO. Multiple vDisks provide the OS (Windows) with more IO queues, which subsequently allow more IOs to be handled in parallel which would otherwise be limited by a single queue in a single disk configuration. For more detailed information on this, refer to [Michael Webster's excellent blog post on the topic](#).
- More options for data protection.
- Option to pin a vDisk to flash (if desired; not recommended as a best practice).
- Make clones of database disks that leverage Tintri VAAI plugin to mount to dev & test VMs.

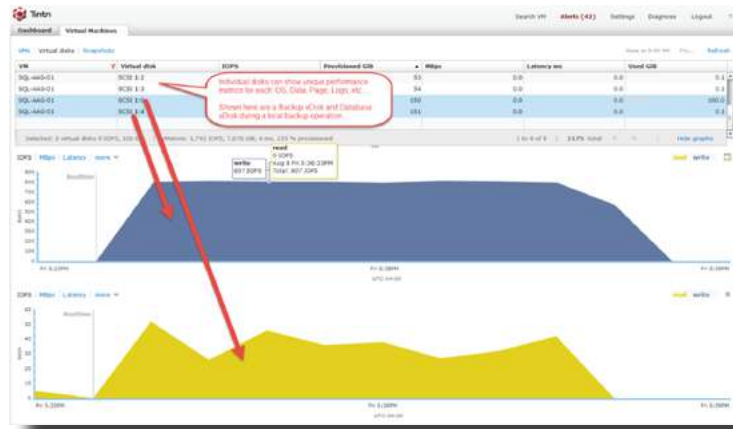


Figure 3 - Better Visibility within a VM by using multiple vDisks.

vDisk Sizing

Managing a VM with many vDisks can get confusing at times during operations such as adding/removing disks, or performing storage vMotion operation on only select disks (as shown in the screenshot below). Some operations are relatively harmless, such as extending the wrong vDisk to increase capacity when another is low on capacity, but other operations, such as deleting the wrong vDisk from a VM, can be catastrophic. A very simple solution to make your life easier: **Use slightly different sized disks (see screenshot below) to avoid confusion.**

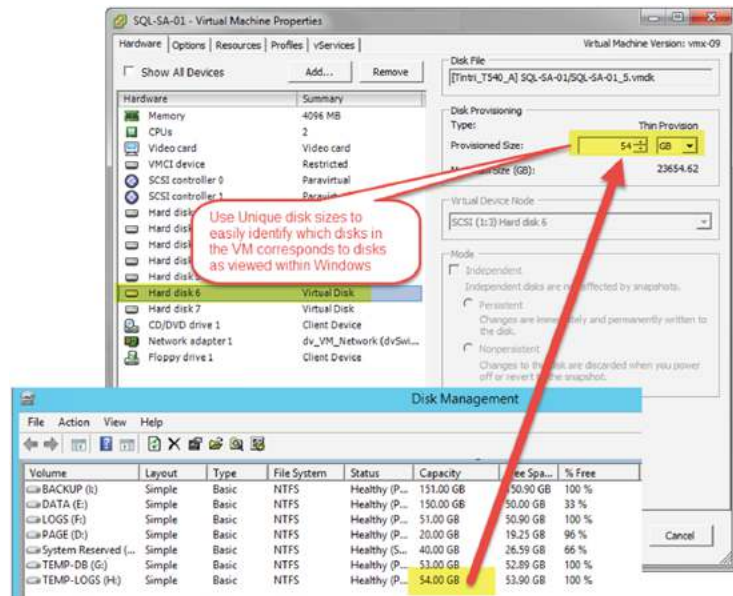


Figure 4 - Unique disk sizes, shown from the perspective of vCenter and within Windows.

For example, if you were going to make your DB disk and LOGS disks both 100 GB each, instead, make the LOGS disk 101 GB so that it is easier to differentiate within the OS, and throughout the various views within vCenter and Tintri VMstore UI.

DO: Use unique, slightly different sized disks to avoid confusion when creating virtual disks.

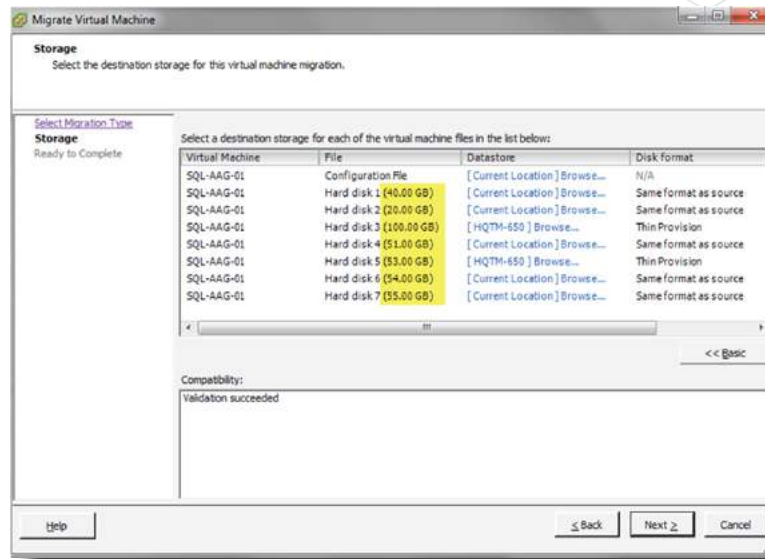


Figure 5 - Storage vMotion dialog box showing vdisks are all sized slightly different for easier identification.

When using Thin Disks, the relatively small amounts of additional capacity assigned to each vDisk to make them a unique size doesn't actually "cost" any additional capacity, unless the additional capacity is written to within the VM.

When sizing vDisks, give VMs breathing room and provision enough additional capacity on each to ensure virtual disks do not run out of capacity within the Guest. As you near capacity, extend the drives to allow for additional capacity well before you run out. Extending a disk does not require downtime, but running out of space may result in application downtime.

DO: Ensure to provision additional capacity when creating virtual disks as there is no wasted space with thin provisioned disks.

Leverage a strong monitoring solution to allow you to safely overprovision space within your VMstore with thin disks, while giving yourself a lot of warning when capacity thresholds near maximum. Monitor at the datastore level, as well as within the guest operating systems. Within VMware vCenter Operation Management Suite (vCOPs) "Guest File System (%)" is the metric to keep an eye on, ensuring it never reaches 100%.

An example monitoring and remediation approach may be to start vDisks at 50% capacity and alert when 75% is reached. At that time, extend the disk with enough space to bring it back to 50% utilized. Remember, there's really no extra "cost" to assigning additional capacity that is never used by the guest.

DO: Monitor disk capacity of the entire VMstore to prevent running out space.

DO: Monitor guest file system capacity and extend vDisks as need to prevent running out of space.

VMware vCenter Settings

HA Cluster Settings

For the highest levels of application availability, we recommend using Microsoft SQL AlwaysOn Availability Groups (AAG) to achieve Database-level high-availability. This paper focuses on stand-alone instances of Microsoft SQL server, refer to our SQL Server Best Practices on Tintri using AlwaysOn Availability Groups document for more info.

Virtual Machine start order is configured on standalone VMware hosts, and does not translate well into a highly-available cluster of hosts, which aren't intended to ever be in a powered-off state. Unlike a standalone host, a cluster is intended to always be on and available. VM start order can still be used for HA situations where a host fails, prioritizing when VMs are brought back up first.

Set VM Restart Setting to High on SQL servers to ensure these VMs are prioritized in the event of a host failure. SQL servers are often pre-requisites for other systems in multi-tiered applications, and should be brought back online before other tiers are brought up (ex. Web application with an IIS front-end and a SQL backend).

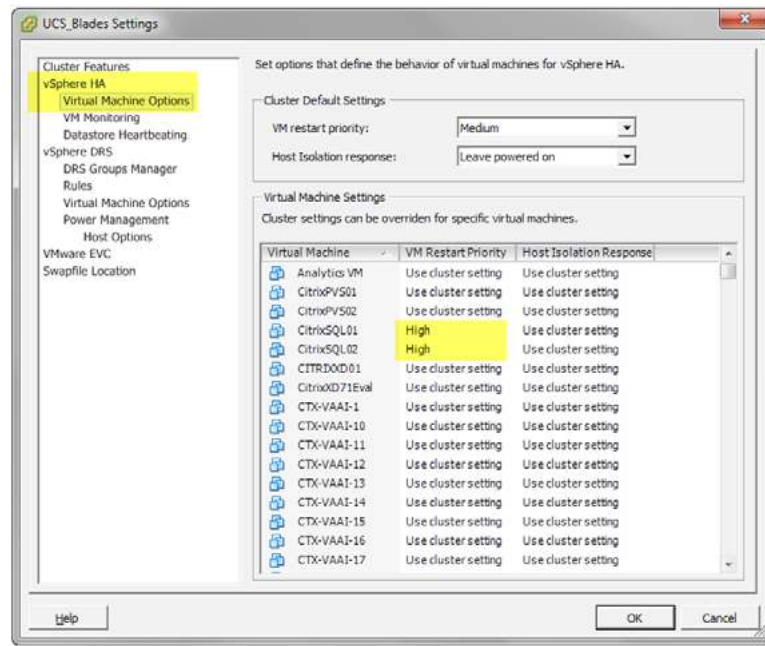


Figure 6 - vSphere Cluster High Availability settings.

DO: Set VM Restart Setting to High on SQL servers to ensure these VMs are prioritized in the event of a host failure.

VM and Application Monitoring

Even if you have a SQL VM that is up and running doesn't mean that your instance of SQL server is available. By Default, VMware Tools don't monitor whether your SQL services are running and accepting client connections, however it does provide an API that can be leveraged to extend VM monitoring functionality to monitor applications within a VM.

With vSphere Enterprise Plus licensing, VMware has a plug-in for the vSphere Web Client called "vSphere App HA" which can monitor SQL availability and is deployed using a Virtual Appliance (OVA format). In the event that a SQL instance is detected as unavailable, options exist to automatically restart SQL services, or reset the VM. Additional information on App HA can be found [here](#). At the time of this writing [App HA Documentation](#) indicates the following versions of SQL are supported for application-level availability monitoring: 2005, 2008, 2008R2, and 2012.

Alternative options to monitor SQL services are available using 3rd party products that leverage VMware Application Monitoring APIs. If running one of these, configure VM Monitoring settings on your cluster to enable this higher level of availability, as shown here:

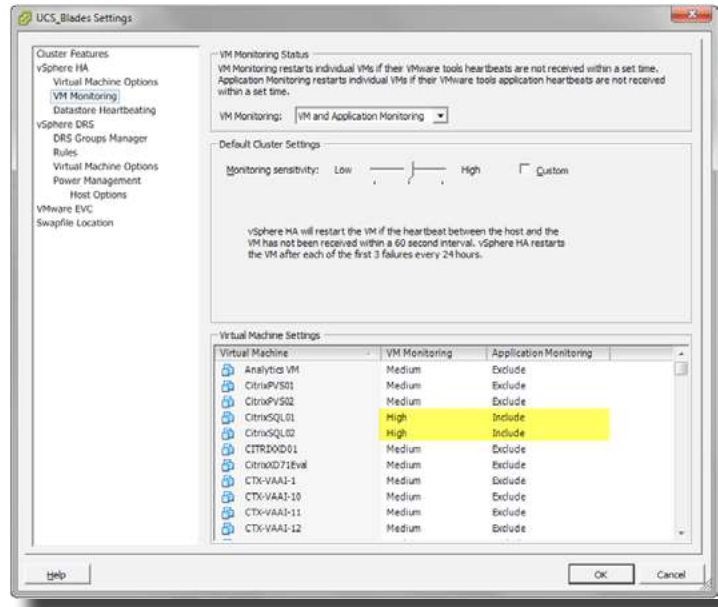


Figure 7 - vSphere HA VM Monitoring settings.

DRS Cluster Settings

Create a rule to prevent SQL servers that are members of the same AlwaysOn Availability Group (AAG) from running on the same host. The idea of having 2 or more servers responsible for keep a single logical database running in a highly available fashion is undermined if both of those servers go down at the same time due to a single point of failure (same host failure). Avoid situations like this by implementing DRS rules for your highly-available VMs.

DO: Create a rule to prevent SQL server VMs that are members of the same AlwaysOn Availability Group (AAG) from running on the same host.

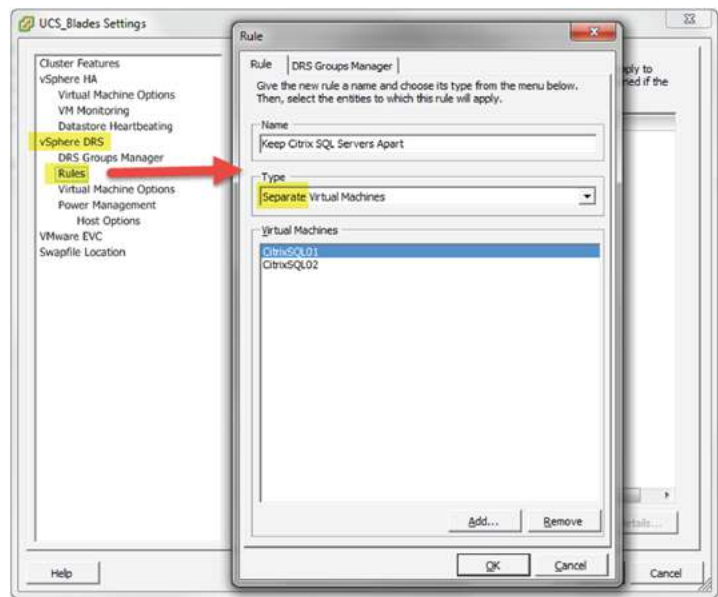


Figure 8 - DRS Rules to prevent VMs from sharing the same host.

Use vApps to control startup order

Consider grouping multi-tiered applications into vApps so that VM Startup order can be controlled to ensure an application that is spread over multiple VMs boots up correctly.

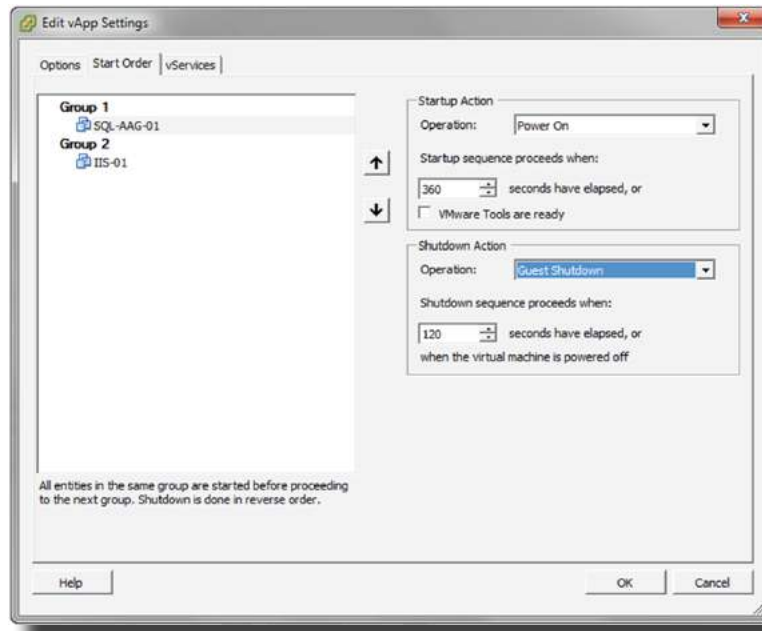


Figure 9 - A simple 2-Tier application within a vApp.

In the example shown in the screenshot above, we have a simple 2-Tier web app which requires SQL to be started and running before IIS Web Services are brought online. vApps are an easy way to ensure the correct startup/shutdown order.

DO: vApps are an easy way to ensure the correct startup/shutdown order, so use them for multi-tiered SQL Server applications.

Guest OS Settings

Disk Partition Alignment

There are a lot of best practice recommendations available that reference Disk Partition Alignment as a source of poor performance when configured incorrectly. However, one of the many benefits of a Tintri VMstore is that it auto-aligns disk partitions. If you're reading through one of the many SQL Best Practice guides available, when you get to the section on partition alignment, skip it... **Disk Partitions are always aligned on Tintri VMstore systems.**

Wondering if any of your existing SQL server VMs (or others) is misaligned? In the Virtual Machine view of the Tintri VMstore UI, you can add new columns to the headers at the top with a simple right-click of the mouse. Find "Aligned IO %" in the list of possible headers and use it to sort to find your VMs that would have run into performance problems elsewhere other than on Tintri.

Again, regardless of the number reported, no action is required on Tintri; all virtual disks are already auto-aligned.

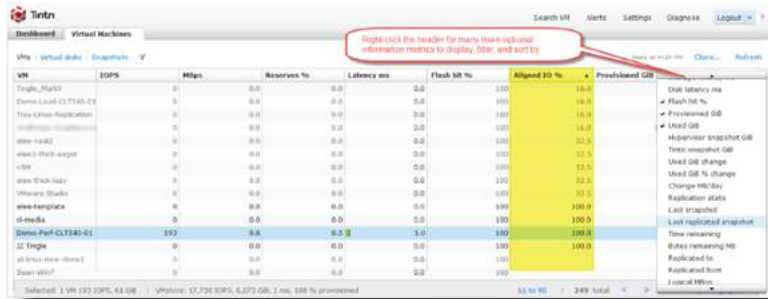


Figure 10 - The “Aligned IO %” field has been added to show which VMs are being aligned automatically.

VMware Tools

Make sure VMware Tools is installed within your VM. This will ensure you have the right drivers installed for virtual components, ensuring optimal performance and stability. VMware Tools also allows the hypervisor to communicate with the guest in order to properly quiesce the disks, a pre-requisite for some backup software, as well as for taking “VM-Consistent” Tintri snapshots.

DO: Make sure VMware Tools is installed within your VM.

Page File Settings

Make sure the page file is on the disk intended for it. This can be achieved by setting all disks to “None” in the page file options (System Properties – Advanced – Performance Options – Advanced – Virtual Memory: Change), and set your disk designated for PAGE to “System Managed”.

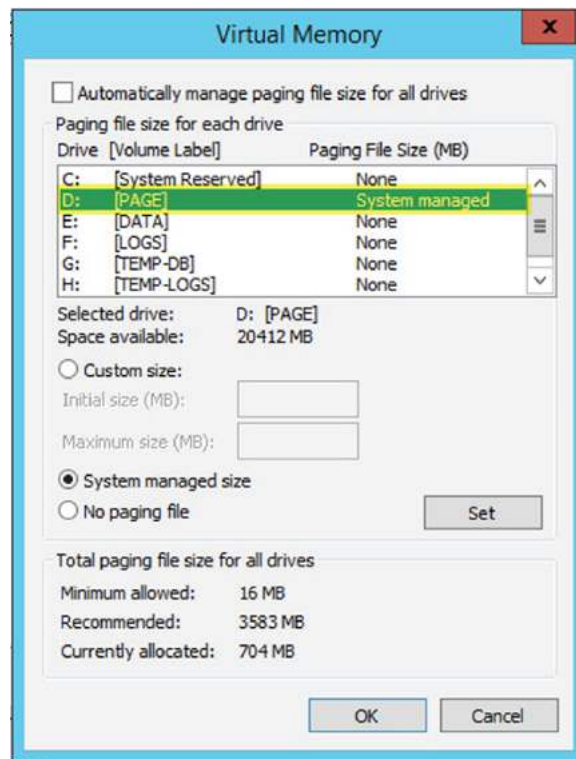


Figure 11 - Page File Settings.

You can statically set a page file, but be sure to follow Best Practices for windows page files and ensure

it is large enough. If you later decide to increase the memory allocated to the VM, but sure that the underlying disk for the page file has enough free space to accommodate 110% of the amount of RAM assigned to the VM. If you've selected "System Managed", no additional change is required to the page file after increasing the memory in the VM. However, if you're statically assigned a page file size, remember to go back and adjust it to reflect the increase in VM memory.

DO: Make sure the page file is on the disk intended for it.

DO: Configure Paging File as "System Managed" to automate page file size increase when additional vRAM is added to VM.

SQL Server & Database Configuration Settings

There are many configurable settings for SQL Server instances and databases that can lead to better performance, compatibility, security, etc. Of all the many options available, there aren't any that we recommend specifically for Tintri VMstores, other than ensuring that your database files use the specific vDisks that were allocated for each (Data, Logs, etc.).

During setup, you are prompted to provide default locations for Data, Logs, TempDB, TempLogs, and Backups. In the event you installed SQL before you added additional drives, it is not too late to modify these default locations. They can be accessed using SQL Server Management Studio, and are found by right-clicking the SQL Server instance (top node on left), selecting Properties, and then Database Settings, as shown here:

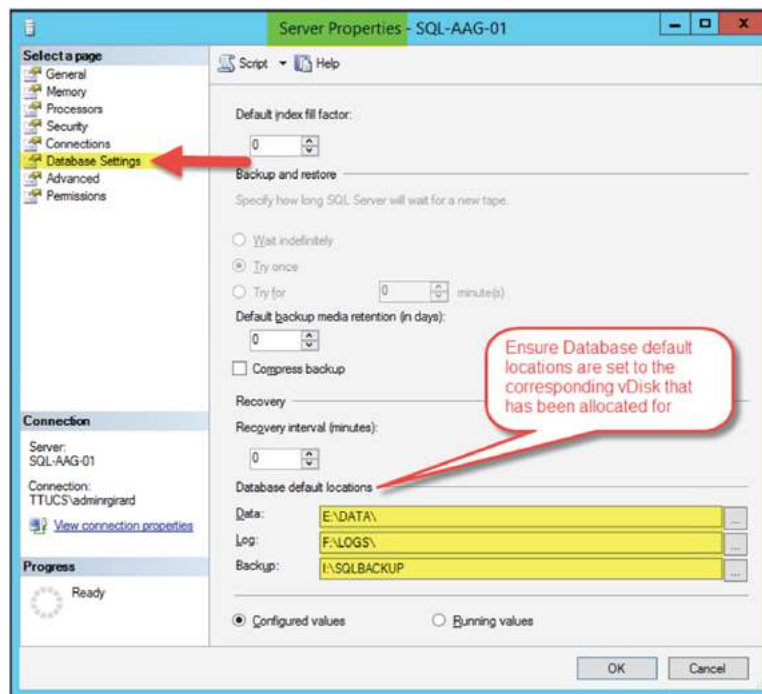


Figure 12 - Modify default locations to use specific disks.

Moving the TempDB files requires a little more effort, but is still a relatively simple procedure, which is well documented in this MSDN article: [Move System Databases](#).

Below is a brief example of the procedure; be use the correct Paths and Drive letters that match your unique environment.

```
USE master ;
GO
ALTER DATABASE tempdb
MODIFY FILE (NAME = tempdev, FILENAME = 'G:\TEMPDB\tempdb.mdf') ;
GO
ALTER DATABASE tempdb
MODIFY FILE (NAME = templog, FILENAME = 'H:\TEMPLOGS\templog.ldf') ;
GO
```

In order for the new TEMPDB locations to take effect, the “SQL Server (INSTANCE)” service must be restarted. After the service restart, new files are created in the locations specified above.

Note: The original TEMPDB log and database files will remain in place and may be manually deleted once the new files are configured to be active.

Data Protection

Backups

When backing up SQL server, it is important to get application-aware backups. This will ensure a clean copy of your data, as well as truncate database logs, which will grow indefinitely in their default settings if backups aren't taken at regular intervals. Please refer to the Tintri Backup Best Practices links found at the end of this document for more information on backing up VMs and Applications on Tintri VMstores.

SnapVM™ - Tintri Snapshots

When we think of snapshots, VMware admins often think of VMware snapshots, and SQL DBAs tend to think of SQL Snapshots. For most scenarios where VMware snapshots would be used, we recommend using Tintri snapshots instead as they have several advantages without the negative performance repercussions that VMware snapshots can have. If you are must create VMware-based snapshots, do not let them persist for long periods of time as VM performance will degrade as more data change is held in the snapshot delta file over time. More information on VMware snapshots and their inherent limitations can be found in [VMware KB Article 1025279](#).

Tintri Snapshots are quick and very space-efficient to create, clone and replicate. Tintri snapshots also don't have any negative performance repercussions if left on a VM for long periods of time, nor do they interfere with backup operations, which may leverage VMware-based snapshots. When taking a Tintri snapshot, either through a repeating schedule, manually, or through a cloning operation, there are two options you should familiarize yourself with that can have an impact on your SQL instances.

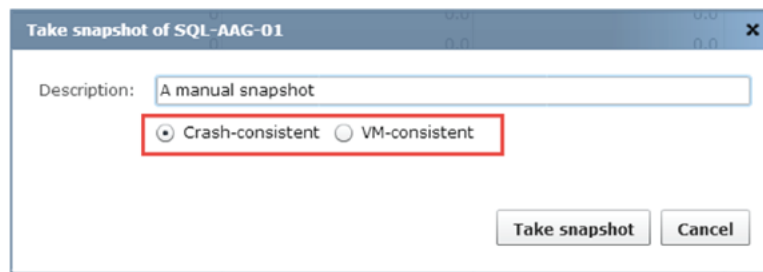


Figure 13 - Snapshot Dialog - Choose crash-consistent or VM-consistent.

Crash-Consistent Tintri Snapshots

The first option (default) is Crash-consistent. Modern operating systems and applications, such as Windows with SQL Server, are designed to be crash-consistent. As a result, crash-consistent snapshots are very quick and convenient and have a very high success rate of being integral when cloned into a VM and powered on.

For SQL Server specifically, there is a possibility that the most recent transactional data may be present in memory only if it hasn't yet been committed to disk through a [SQL Database CHECKPOINT](#) operation. And if the transactions don't make it to disk, they will not be present in a crash-consistent snapshot. Most of the time, this is acceptable as the likelihood of this happening is low. According to Microsoft's documentation on [SQL Recovery Interval Server configuration](#), only up to one minute of data may be lost, however in our tests, we weren't able to witness any data loss, even when snaps were taken immediately following data change.

VM-Consistent Tintri Snapshots

For VM-Consistent snapshots invoked from the Tintri VMstore UI, a VMware snapshot is taken with the "Quiesce Guest File System" option selected, then a Tintri snapshot is made, and finally the VMware snapshot is deleted afterwards to ensure there is no future performance degradation as data changes.

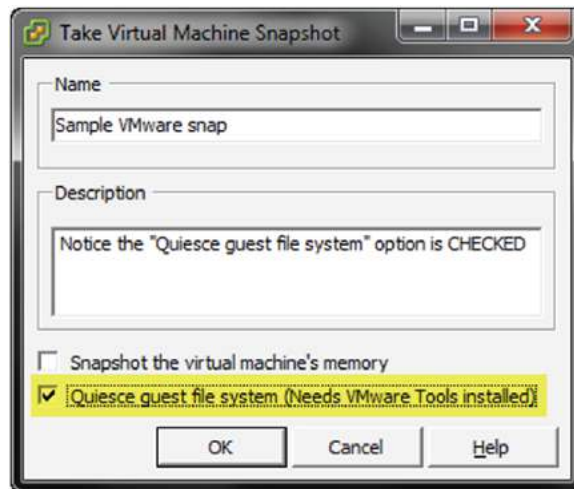


Figure 14 - VMware Snapshot dialog, showing Quiesce Guest File System option.

The Quiesce Guest File system option sends a call from the hypervisor into VMware Tools within the guest that leverage Microsoft VSS (Volume Shadow Services) to freeze IO within SQL and commit all data to disks. In addition to this, custom scripts can be written and used within the VM for "Freeze" and "Thaw" actions. Some DBA's may want to perform a specific action in the database upon freeze to send signals to other dependent systems, or perform other actions on the data, specific to the applications that depend on certain databases.

To leverage VMware Tools Freeze and Thaw scripts, create a folder called "backupscripsts.d" within the VMware Tools installation folder, typically found here: C:\Program Files\VMware\VMware Tools\backupScripts.d

A sample .bat file¹ to be placed in this folder may look like this:

```
echo off
if "%1" == "freeze" goto doFreeze
if "%1" == "thaw" goto doThaw
msg Administrator "other: %1"
goto EOF

:doFreeze
call c:\vmtools-scripts\freeze-script.bat
goto EOF

:doThaw
call c:\vmtools-scripts\thaw-script.bat
goto EOF

:EOF
```

Figure 15 - Sample Freeze/Thaw Script.

More information on [Customizing VMware Freeze/Thaw scripts](#) can be found [here](#).

There are some side-effects that may come as a result of the VMware snapshot that uses the Quiesce Guest File System option, so test your VMs diligently before performing these operations on Production servers during business hours. More information can be found in this VMware KB Article #1015180. Specifically, search for “quiesc” in that KB for specific vSphere gotchas around the quiesce option.

Given that there may be side-effects associated with VM-consistent snapshots as a result of the VMware quiesce file system option, and our own tests have had excellent results with crash-consistent snapshots, we recommend you use Crash-consistent snapshots (which is why it is the default).

DO: Use Tintri native snapshots for protecting SQL Server instances.

ReplicateVM™ – Tintri VM-level Replication

ReplicateVM is a Tintri technology that allows per-VM replication between VMstores. It leverages Tintri snapshots and performs WAN-efficient replication between sites. ReplicateVM is an excellent option to build into Data Protection and Disaster Recovery planning. When coupled with the Tintri Automation Toolkit, the results can be very powerful for a simple, quick and effective disaster recovery solution.

ReplicateVM can operate seamlessly to protect your SQL VMs (and all other VMs). While there is nothing extra that needs to be done within SQL, ReplicateVM does leverage Tintri snapshots, so you be sure to test whether crash-consistent snapshots will work for you, or if VM-consistent snapshots are required. With Tintri replication, multiple snapshots are replicated to the remote side, allowing you to choose which snapshot to clone. This is effective in protecting you in scenarios where data corruption may take place on the production VM, and the data corruption replicates to the remote site. Depending on how much time passes between noticing the corruption and failing over, you may choose not to fail to the latest one (T0) or two (T-1) snapshots, but instead fail to the 3rd oldest snapshot: T-2.

More information can be found on [ReplicateVM here](#).

DO: Use Tintri ReplicateVM for data protection and disaster recovery for protecting SQL Server instances.

¹ The script sample below is provided as is and requires customization for specific environments.

Maintenance Activities

Database maintenance is an important element of a SQL Server's overall health. Maintenance activities range from rebuilding Indexes to scheduled data imports for other sources to performing internal consistency checks. Most maintenance tasks are specific to your data and applications, and aren't covered in detail in this document. However, we will cover some overarching principles related to maintenance tasks and offer ideas to leverage Tintri VMstore analytics to help manage maintenance tasks. For more detailed information related to [SQL Maintenance Plans](#), click [here](#).

Scheduling

Scheduled maintenance tasks may have varying levels of obtrusiveness; attempt to schedule these tasks within time periods that are least obtrusive to your business operations. Some operations, such as rebuilding indexing, can have a heavy negative performance impact if your SQL database goes too long without. However, running and re-running these operations every 15 minutes can have an even worse overall impact on your server's health and performance. Find a balance that works best for your organization, starting at weekly intervals, and adjust accordingly to maximize performance.

DO: Schedule maintenance tasks within time periods that are least obtrusive to your business operations.

When it comes to scheduling, look holistically at the bigger picture and coordinate backup, snapshot, and replication schedules with your internal SQL Maintenance activities. Avoid taking backups or snapshots mid-way through maintenance activities and attempt to make them before and/or after. Having a roll-back point prior to a maintenance tasks can help if something fails during maintenance, such as data corruption. And having a roll-back point after a maintenance tasks allows recovery to a point-in-time where the database is well-tuned and optimized. The Tintri Automation Toolkit offers PowerShell cmdlets to facilitate granular automation around snapshotting and other per-VM information & actions.

DO: Coordinate maintenance tasks with backup & snapshotting schedules.

Understand

This may sound obvious, but seek to understand existing maintenance tasks being run and why they are run. It is common in high-paced IT environments to have database tasks scheduled and running consistently but any employees whom may have been involved with their setup may be long gone. It's not uncommon to run obsolete tasks simply because everyone is too busy to check if the original business (or other) drivers for the tasks still apply.

Involve the line of business stakeholders in your investigations and you may be surprised to find out that some maintenance tasks, which consume precious resources are no longer necessary.

DO: Ensure that unnecessary maintenance tasks no longer run for saving precious resources.

Analyze

Scheduled disk activity has a way of jumping out at you when looking at graphs. Unlike IO that results from sporadic user access, scheduled tasks have a very distinct pattern. By using the Tintri VMstore UI to spot these trends, as shown below, continue the investigation within the guest to discover the source of the traffic pattern.



Figure 16 - Example of scheduled behavior.

In some cases, the disk activity seen is necessary and is a result of the application's design. Equipped with new information, design enhancements can be made, so be sure to share the results with the app's developer whom may not realize that their current code produces the results witnessed. And in other cases, you may discover than a scheduled task you didn't know about, which can lead to new discoveries about the inner workings of the various VMs that make up your infrastructure.

DO: Use VMstore UI to monitor at VM and virtual disk level IO patterns to uncover and troubleshoot performance issues.

Monitor

With automated tasks, it's easy for a task to repeatedly fail and go unnoticed in a busy, complex environment. Develop good monitoring habits to ensure that any tasks that don't run as planned are investigated and resolved. In some cases, these may become apparent in the graphs if normal activity suddenly changes.

Performance Testing

Before making changes intended to improve performance, test the proposed changes and evaluate whether the results match expected outcomes prior to introducing the changes to production. One of the many benefits virtualization and Tintri has to offer is the ability to quickly clone VMs and run them in isolated network. Take advantage of this and create pre-prod environments comprised of space efficient Tintri CloneVM™ to test configuration changes to your VM. Once you are satisfied with the results, follow your organization's change control procedures and repeat the changes in production to minimize surprises associated with implemented untested changes. Consider using a tool such as [HammerDB](#) to generate load on your SQL VM, which will take into account all configuration and environmental settings that apply to a specific database within a specific instance of SQL. Use tests to develop a performance baseline and re-run the tests periodically to ensure that performance remains acceptable as data and changes accumulate as the environment continues to evolve. During these tests, leverage the Tintri VMstore UI and/or [Tintri Global Center](#) to look for bottlenecks that may result in undesired latency attributable to host, network, and storage layers, all from a single pane.

DO: Run periodic performance tests to evaluate changes over time and proactively address concerns.

Another performance tool, [SQLIO](#), is great for generic tests on storage and virtual environments, but produces simulated results designed to mimic SQL Server IO patterns. Unfortunately, [SQLIO](#) has a limited ability to predict actual real-world performance because it doesn't use your SQL server binaries or the unique data, code and configuration that is accessed by real applications and users.

Dependencies

In attempting to understand maintenance operations, identify incoming and outgoing dependencies to form a comprehensive, holistic view of how one particular SQL server fits into the broader IT environment as a whole. VMware's vCenter Operations Management suite (vCOPs Advanced Edition and higher) includes [Virtual Infrastructure Navigator \(VIN\)](#), a tool that is great for identifying and mapping out dependencies between your SQL servers and other application servers. Other tools are available from 3rd party software vendors as well which can simplify overall management & complexity.

You may identify scheduled task behavior using the Tintri VMstore UI, but may not be able to find a scheduled task on that VM that explains patterned behavior. In these cases, look to servers that interact with the SQL VM in question, as they may have the scheduled job that produces the patterns observed.

DO: Look at the environment holistically – host, networking and storage – when troubleshooting.

Rinse & Repeat

The life cycle of maintenance tasks listed above typically work hand-in-hand with each other. Go through each of the stages often and use the results from each to improve the others. Over time, you'll have a strong understanding of "what normal looks like" under the hood of your infrastructure, which makes it easier to spot abnormalities caused by change. Identifying the abnormalities will help you respond quicker to little issues, ideally before they become big problems.

References

Tintri Links

- [Tintri VMstore Overview](#)
- [Tintri NFS and vSphere Best Practices Guide](#)
- [Tintri ReplicateVM](#)
- [Tintri Support Portal \(login required\)](#)
- [Additional Tintri Resources](#)
- [Tintri Backup and Recovery Best Practices](#)
- [VDP Advanced Best Practices on Tintri](#)
- [Veeam Best Practices on Tintri](#)
- [CommVault Simpana Best Practices on Tintri](#)

SQL Server Specific Links

- [Official Microsoft Documentation for SQL 2014](#)
- [SQL Server on VMware Best Practices Guide](#)
- [SQL Server 2014 Maintenance Plans](#)
- [What's New in SQL 2014](#)
- [SQL Server Database Checkpoints](#)
- [SQL Recovery Interval Server configuration](#)
- [Move System Databases](#)

Other Links

- [Virtualizing Business-Critical Applications on VMware vSphere](#)
- [Configuring Database Servers for Optimal Performance](#) – an excellent blog post authored by Michael Webster that provides detail on virtualized SCSI Disks and Controllers.
- [VMware KB Article 1015180](#) – VMware Snapshots, including information on Quiescing the Guest File System
- [VMware KB Article 1025279](#) – Best Practices for VMware-based snapshots, including information on limitations and restrictions
- [VMware App HA](#) – Allows for application availability monitoring in vSphere 5.1 and 5.5 (Enterprise Plus licensing required)
- [Custom VMware Freeze/Thaw scripts](#)
- [HammerDB](#) – Load generation and performance assessment
- [VMware Infrastructure Navigator](#)

