White Paper

Cisco HyperFlex 2.0 All-Flash Storage and Microsoft SQL Server Best Practices

Contents

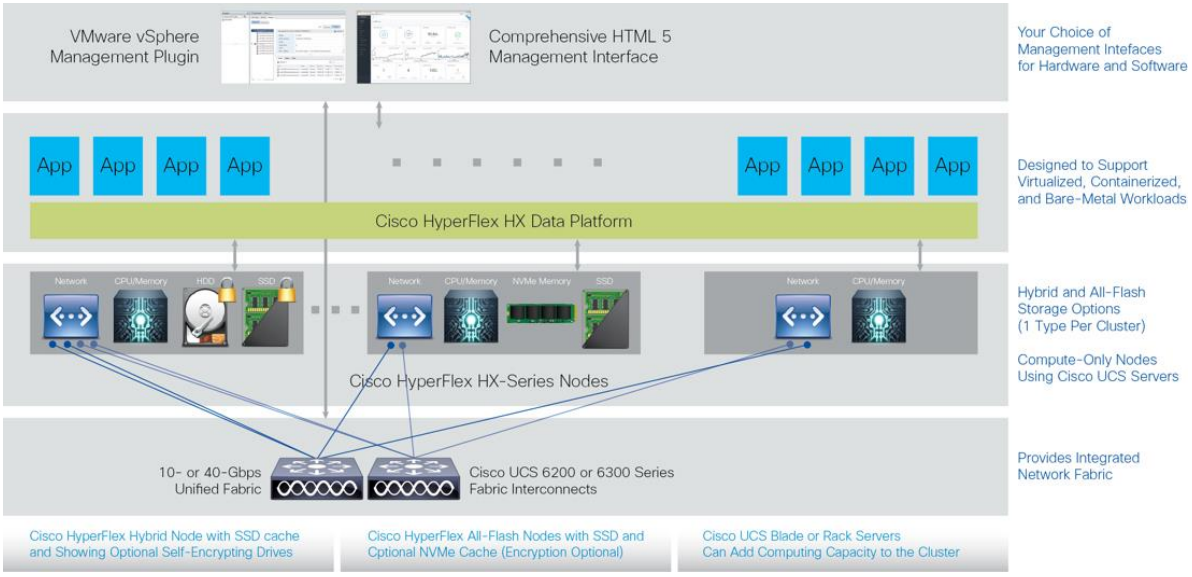
Executive Summary

Cisco HyperFlex™ systems unlock the full potential of hyperconvergence. The systems are based on an end-to-end software-defined infrastructure, combining software-defined computing in the form of Cisco Unified Computing System™ (Cisco UCS®) servers; software-defined storage with the powerful Cisco HyperFlex HX Data Platform, and software-defined networking with the Cisco UCS fabric that integrates smoothly with the Cisco® Application Centric Infrastructure (Cisco ACI™) solution. Together with a single point of connectivity and hardware management, these technologies deliver a preintegrated and adaptable cluster that is ready to provide a unified pool of resources to power applications as your business needs dictate.

Cisco HyperFlex HX Data Platform 2.0 All-Flash Storage

Cisco HyperFlex systems are designed with an end-to-end software-defined infrastructure that eliminates the compromises found in first-generation products. With all-flash memory storage configurations and a choice of management tools, Cisco HyperFlex systems deliver a preintegrated cluster that is up and running in less than an hour and that scales resources independently to closely match your Microsoft SQL Server requirements (Figure 1). For an in-depth look at the Cisco HyperFlex architecture, see the Cisco white paper Deliver Hyperconvergence with a Next-Generation Platform.

Figure 1. Cisco HyperFlex Systems Offer Next-Generation Hyperconverged Solutions with a Set of Features That Only Cisco Can Deliver



Microsoft SQL Server on Cisco HyperFlex Systems

Cisco HyperFlex HX-Series all-flash nodes fully support SQL Server. SQL Server isn't just a back-office server. It is the back-end database for many commercial and custom server applications with a variety of workloads and use cases. Customers run:

- Enterprise resource planning (ERP) applications
- Web-facing e-commerce online transaction processing (OLTP) applications

- Online Analytical Processing (OLAP) applications and data warehouses
- The back end for familiar applications such as VMware vCenter and Microsoft SharePoint Server

Microsoft SQL Server 2016

Microsoft SQL Server 2016 is the latest relational database engine release from Microsoft. SQL Server 2016 adds many new features and enhancements to the relational and analytical engines, including in-memory OLTP, analysis services, always-encrypted applications, greater availability, and improvements to the temporary database (tempdb) file.

Microsoft SQL Server on Cisco HyperFlex Systems: VMware Best Practices

When you implement a SQL Server database on a Cisco HyperFlex system, you should follow the best practices proposed by VMware at http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-best-practices-guide.pdf.

The following sections present the major design guidelines and configurations for a successful SQL Server implementation.

High Availability

High availability is critically important to database administrators, and they have many tools at their disposal to meet this challenge. The HX Data Platform has availability built in at the storage file system layer, with all data written in duplicate or triplicate (replication factor or 2 [RF2] or replication factor of 3 [RF3]). The HX Data Platform can promote a copy of the data to primary status if the primary storage controller that owns the primary copy is unavailable, without affecting the virtual machines. To protect against a node or hypervisor failure, you can configure VMware High Availability (HA) to bring up the virtual machines on other nodes in the cluster.

SQL Server itself has high availability options built in at the database layer. The features that are available depend on the edition of SQL Server deployed. These high-availability features are described in the following sections.

Note that shared-disk Microsoft Windows failover clusters are not supported on Network File System (NFS) data stores. NFS data stores are supported with non-shared disk Microsoft Windows failover clusters using SQL Server Always-On Availability Groups. NFS is used to connect to data stores on the HX Data Platform.

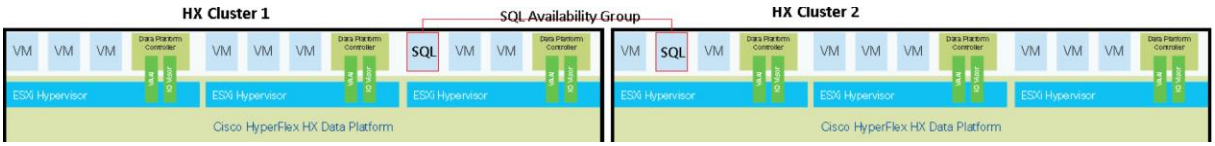
Microsoft SQL Server AlwaysOn Availability Groups

AlwaysOn Availability Groups are an enterprise alternative to database mirroring that uses no shared storage and was first implemented in SQL Server 2012. An AlwaysOn Availability Group is a set of databases that fail over together with one primary set and up to eight secondary sets (four secondary sets in SQL Server 2012). Two availability modes (synchronous and asynchronous) can be selectively applied to each secondary replica, with a maximum of two synchronous replicas.

Asynchronous-commit mode is used for disaster recovery, or for situations in which the increased transactional latency of synchronous-commit mode cannot be tolerated. It will always be behind the primary replica and thus requires a forced failover, which may cause data loss.

Synchronous-commit mode requires the secondary replica to harden the log to disk before the transaction is confirmed. Therefore, secondary replicas will be synchronized with the primary replicas when they are in a healthy state, allowing the configuration of two additional failover modes: automatic (enhancing high availability) and manual.

Figure 2. Microsoft SQL Server Availability Group Best Practice Is to Isolate at Least One Copy on a Separate HX Data Platform Cluster.



When you are deploying a highly available database, eliminating single points of failure can increase availability in the event of a cluster failure due to power, network, or building problems. A best practice is to stretch an availability group across multiple clusters when they are nearby, as shown in Figure 2. In the figure, one SQL Server virtual machine from the availability group is in HX Cluster 1, and one SQL Server virtual machine is in HX Cluster 2. For more information about how to create an availability group listener that spans multiple subnets, see this Microsoft Developer Network article. For the third or subsequent copies of an availability group, or for customers who must place all copies in a single HX Data Platform cluster, see the VMware Distributed Resource Scheduler Anti-Affinity Rules section of this document for information about increasing SQL Server availability.

Database Mirroring

Database mirroring is deprecated in future versions of SQL Server. It is configured on a per-database basis creating a one-to-one mirror of the database and requires a full recovery model. Operating modes include high-safety (synchronous) and high-performance (asynchronous) modes. Basic Availability Groups are considered the successor to database mirroring.

Log Shipping

Log shipping is the copying, or shipping, of transaction log backups to one or more separate secondary server instances. These backups are applied independently to each secondary database.

AlwaysOn Failover Clustering

Failover clustering requires shared storage between the SQL Servers using Microsoft Windows Server Failover Clustering. VMware does not support AlwaysOn Failover Clustering on NFS storage, which is the Cisco HyperFlex storage protocol. Therefore, SQL Server Failover Clustering Instances cannot be deployed on Cisco HyperFlex systems today.

Sizing Considerations

Several factors can influence size requirements.

Data Deduplication

Data deduplication is used on all storage in the cluster, including memory and solid-state disk (SSD) drives. By fingerprinting and indexing just these frequently used blocks, high rates of deduplication can be achieved with only a small amount of memory, which is a high-value resource in cluster nodes. Deduplication rates vary with SQL Server, but many customers see large deduplication with operating system and application binary files. Transaction log files and databases tend to achieve lower deduplication rates, in the 2 to 15 percent range.

Inline Compression

The HX Data Platform uses high-performance inline compression on data sets to save storage capacity without negatively affecting performance. Incoming modifications are compressed and written to a new location, and the existing (old) data is marked for deletion, unless the data needs to be retained in a snapshot. The data that is being modified does not need to be read prior to the write operation, which avoids typical read-modify-write penalties and significantly improves write performance. Overall clusterwide compression rates for SQL Servers average in the 30 to 35 percent range, although some options, such as OS-level file-system encryption and database encryption, significantly lower the achievable compression rates.

Thin Provisioning

The platform makes efficient use of storage by eliminating the need to forecast, purchase, and install disk capacity that may remain unused for a long time. Virtual data containers (data stores and Virtual Machine Disk [VMDK] files) can present large amounts of logical space to applications, whereas the amount of physical storage space that is needed is determined by the data that is written. You can expand storage on existing nodes and expand your cluster by adding more storage-intensive nodes as your business requirements dictate, eliminating the need to purchase large amounts of storage before you need it.

Sizing the Database and Log Files

User database files and transaction logs grow as data is written to the database, and the way that these files behave is influenced by the recovery model. Typically, production databases are backed up, and the recovery model determines which files must be backed up, and what the recovery-point objective (RPO) is. If a database file runs out of free space, it must grow, and this action will extend the file and write zeroing operations, potentially causing a pause in transactional I/O processing and affecting performance. Regardless of the recovery model, the transaction log size must be sufficient so that it does not run out of space and grow before the next backup interval or multiple backup intervals to a size that results in backup job failure, according to the service-level agreement (SLA) for the database.

After the backup operation is complete, transactions in the log are released, and more space is available for future write operations to the transaction log, enabling in most production databases a transaction log file size that never physically grows on disk. See the Microsoft Developer Network article Manage the Size of the Transaction Log File for information about how to monitor log space use.

With the simple recovery model, circular logging is enabled in the transaction log. Because the transaction log is not backed up at all with the simple recovery model, the database can be recovered only from a backup operation, with all transactions after that point lost. The full recovery model backs up both the transaction log and the database, allowing both a point-in-time restore operation and roll-forward recovery. For more information about backup and restore operations with SQL Server, see the document Understanding How Restore and Recovery of Backups Work in SQL Server.

The database files can be grown or shrunk. SQL Server database administrators (DBAs) use several methods for limiting the impact of database file changes, including these strategies:

- Make sure that the database file has sufficient free space so that the file does not need to grow.
- Monitor the files sizes or set alerts and then manually increase the file at some interval, preferably during off-peak hours.

- Configure auto-grow settings to a set size, rather than a percentage, and treat this configuration as a contingency strategy to protect against a monitoring failure. When auto-grow is enabled, be sure to set the maxsize value for the database so that you never encounter an out-of-disk space condition, in which the database has filled the entire disk (VMDK), thus affecting availability. Although setting maxsize only forces SQL Server to report that it is out of space before the file system limit is reached, is it still a best practice, and in the case of multiple files in the VMDK, it can prevent one file from starving the all the others for disk space. On traditional storage systems, this scenario can be catastrophic and can cause significant downtime. With the HX Data Platform, although growing the VMDK or data store is a trivial process if you should run out of space at either layer, this scenario can affect SQL Server availability.
- Although you might be tempted to manually shrink a database file, the performance impact is significant, and you should use this approach with caution. Auto-shrink is disabled by default with good reason. Enabling this option should be carefully considered, because auto-grow combined with auto-shrink can cause unnecessary overhead.
- You can use the Instant File Initialization feature to prevent SQL Server from filling database file extensions with zeroes and overwriting data on the disk, resulting in nearly instantaneous extensions. Instant File Initiation cannot be used with the Transparent Data Encryption (TDE) function, and it requires the SQL Server service account to be given additional permissions. Because deleted disk content would not be zeroed out and instead only overwritten as new data is written to the file, this deleted content could be accessed by an unauthorized entity. Carefully consider the security implications of enabling Instant File Initiation. Regardless, make sure that any detached data files and backups have restrictive discretionary access control lists (DACLs).

Selecting Database Files

At the smallest scale, a database has a single transaction log file (LDF) and database file (MDF). The database file is a part of the primary file group, to which more data files can be added. These additional database files (NDFs) can be placed in separate VMDKs to increase the **aggregate queue depth** of the database. Consider an additional separate database file, in a separate VMDK, in a round-robin configuration across four VMware Paravirtual Small Computer System Interface (PVSCSI) controllers for each 1000 I/O operations per second (IOPS) of database traffic.

Selecting Tempdb Files

Tempdb is a shared database file for temporary data for all databases in the SQL Server instance and can become a point of contention for many workloads. It has traditionally been isolated from user databases and transaction logs, with many database administrators taking additional careful steps to place the tempdb file and the temporary log (templog) file in isolated VMDKs. Most smaller SQL Server implementations do not require such careful isolation, but many SQL Server instances can benefit from having additional files added to the tempdb file group. With SQL Server 2016, the number of files is adjusted, with a default of eight or the number of cores assigned to the virtual machines, whichever is smaller. For other versions of SQL Server, you should follow that guidance.

A good indicator of contention on tempdb is PAGEIOLATCH_XX. When contention is identified, first add tempdb files to the file group. In rare cases, such as when you are bulk-inserting large amounts of data, separate the tempdb file into separate VMDKs spread across multiple PVSCSI controllers.

Selecting the Logical Unit Number Layout

In a virtual machine, the logical unit number (LUN), or logical disk, is a VMDK file stored in the Cisco HyperFlex data store. Usually after you add a disk to a Microsoft Windows virtual machine, the disk will be offline and must be brought online, initialized, and formatted before use.

Selecting the Globally Unique Identifier Partition Table

When initializing, the globally unique identifier (GUID) partition table (GPT) is preferred, as it has more file system redundancy in place and can be used for partitions larger than 2 terabytes (TB).

Selecting the Allocation Unit Size

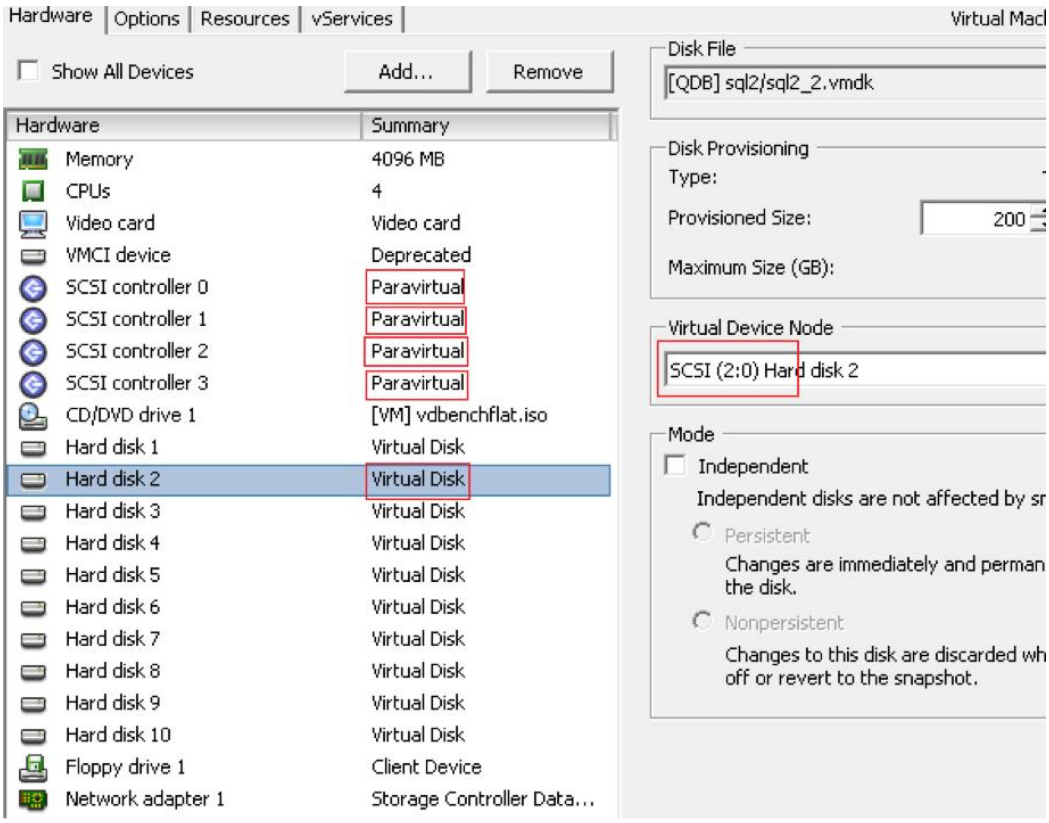
When formatting, Microsoft recommends a selecting 64-KB allocation unit size (AUS), also sometimes referred to as the cluster size. If you leave the setting at the default, the size will be 4 KB until you have very large partitions. The 64-KB AUS is recommended for any disk that will house tempdb, templog, user databases, and user transaction logs.

Isolating the Page File, OS, Tempdb, Database, and Log

Many customers run dozens to hundreds of databases. Most of these databases require few IOPS and do not warrant careful isolation of every workload and file type. As databases scale, or for those few databases that are tier-1 databases, isolation of everything can improve performance, or at least prevent file-system fragmentation. Optimally, you should isolate everything into separate VMDKs that are spread across the maximum of four PVSCSI controllers. The recommended approach is to set the page file size to 1 x RAM + 257 MB on an isolated VMDK on which a complete memory dump can be saved. For some customers, enough space for a kernel memory dump is sufficient; for kernel dump sizing, see the Microsoft blog Page File the Definitive Guide. For an in-depth discussion of file size, see the Microsoft blog Windows Page File and SQL Server.

You should spread the database data file and log files across multiple PVSCSI controllers for optimal performance. Each controller will start with the number 0 and increment up to 3. The second number after the colon is the LUN or disk number. For example, Figure 3 shows 2:0, indicating controller 2 on disk 0.

Figure 3. Four PVSCSI Controllers with Each VMDK Configured with Round-Robin Load Balancing Across the Controllers



Maximum Degree of Parallelism

The maximum degree of parallelism (MAXDOP) setting controls the number of processors that are used when running a query in a parallel plan. Historically, this setting was used both to prevent a query from taking over and thus increasing latency for other operations, and to keep the query confined to a non-uniform memory access (NUMA) node (that is, a physical processor). You can set this value on the Advanced page of the server properties. It is called out specifically in Transact-SQL (T-SQL) queries. With SQL Server 2016, this setting can be applied directly to a database with the scoped configuration function. If the server has two sockets and eight cores, set the maximum degree of parallelism to 8. However, the default setting of 0 should be sufficient for most deployments. Carefully consider changing this setting after testing the impact of changes on the production workload.

Microsoft SQL Server Maximum and Minimum Memory

Set the SQL Server memory configuration options to help ensure a healthy virtual machine. VMware recommends that the maximum memory should be set to a few gigabytes below what is assigned to the virtual machine so that the operating system and other applications in the virtual machine can function properly. The minimum memory should be set to a reasonable amount to prevent VMware ESX from taking memory from SQL Server when under memory pressure.

Locked Pages in Memory

Locking pages in memory prevents the system from paging the data to the paging file. For tier-1 databases, locking pages and setting the SQL Server minimum memory can prevent the balloon driver in ESX from inflating into the SQL Server's virtual machine memory space and thus affecting SQL Server performance.

In-Memory OLTP

The Cisco HyperFlex system starts with 128 GB on the Cisco HyperFlex HX220c All Flash Node and 256 GB on the Cisco HyperFlex HX240c All Flash Node. Both server configurations can be updated up to 1.5 TB of memory per node. For a typical 4-node deployment, 512 GB to 6 TB of RAM in the cluster is nearly an order of magnitude more memory than what was available just a few years ago. Starting with SQL Server 2016 SP1, in-memory OLTP has been added to the 64-bit SQL Server Standard, Web, and Express versions. SQL Server Enterprise is limited only by the OS, and SQL Server Standard can use a maximum of 64 GB of RAM with a 32-GB in-memory OLTP quota per database. When sizing the solution, consider purchasing additional memory to allocate to your SQL Servers to enable some of the many in-memory OLTP features. These features can dramatically improve performance, and in some cases by orders of magnitude.

One of these features is the use of memory-optimized tables. These tables are fully durable by default and meet atomicity, consistency, isolation, and durability (ACID) requirements. Because the table rows are read from and written to memory, the speed and response time is much faster than even with flash storage. When using memory-optimized tables, be sure to allocate enough memory to the virtual machine, and set the memory reservation value to more than what the tables require. In many cases, as a guideline, set the value to three or four times the amount of memory required to house the in-memory data to maintain both versioning and logging. For precise sizing of tables, indexes, versioning, variables, and growth, see the Microsoft Developer Network article Estimate Memory Requirements for Memory-Optimized Tables.

Trace Flags

Trace flags are used to change SQL Server behavior, and some, discussed here, can have a large impact. These trace flags are global. In SQL Server 2016, many of these settings can be made on each database using the alter database scoped configuration T-SQL command. Most trace flags behavior is now set by default in SQL Server 2016. Settings are listed here for older SQL Server versions.

Trace Flag 1118

Trace flag 1118 removes single-page allocations, which can reduce contention on the shared global allocation map (SGAM) page. When paired with settings to help ensure that tempdb and other databases have enough files and reduced PAGEIOLATCH contention, this flag setting can increase performance. This trace flag is unnecessary in SQL Server 2016 because tempdb has this function enabled by default. User databases also have this function enabled by default. It can be disabled on with the following T-SQL command:

```
ALTER DATABASE <dbname> SET MIXED_PAGE_ALLOCATION { ON | OFF }
```

Trace Flag 1117

Trace flag 1117 is used to grow data files uniformly. This function is enabled by default on tempdb in SQL Server 2016. User databases have this function disabled by default. It can be enabled with the T-SQL command shown here. If the database contains more than one file group, the AUTOGROW_ALL_FILES setting must be enabled for each file group. For best performance, user databases should **have more than one file** and should grow uniformly.

```
ALTER DATABASE <dbname> MODIFY FILEGROUP <filegroup> { AUTOGROW_ALL_FILES |
AUTOGROW_SINGLE_FILE }
```

Trace Flag 834

Trace flag 834 enables large pages in virtual machines with at least 8 GB of memory. It can increase performance and requires locked pages in memory to be assigned to the SQL Server service account. All buffer pool memory will be allocated during startup, which can increase the performance of some queries and increases the need to set the **SQL maximum memory**. An older VMware performance white paper recommends this trace flag. If you use this flag, be sure that the column store Index feature is not being used. Otherwise, severe performance problems can be triggered. Click here for a sample script Microsoft published to check for both trace-flag-834 and column-store indexes.

Microsoft SQL Server Startup Options

The SQL Server –E startup option allocates multiple extents per file group. This setting can improve performance in data warehouse environments that have few users. In some OLTP environments with many users, this setting can reduce performance.

Microsoft SQL Server Licensing

Microsoft licenses different versions of SQL Server, each with a few different editions, in a variety of ways, so always check with Microsoft for the exact terms of the license. Many editions of SQL Server are available: the Web edition, which is for service provider environments; the Express edition, which is free; the Developer edition, which is for nonproduction use; and the Standard and Enterprise editions, which are the most commonly used. The Standard edition can be licensed per physical server or per core. The Enterprise edition is licensed per core. Cisco HyperFlex 2.0 is deployed on VMware vSphere, so the example here focuses on SQL Server 2016 licensing in a virtualized environment.

Standard License

The Standard edition of SQL Server 2016 is licensed in two ways: with a SQL Server license and a client access license (CAL) for each user accessing the database; or with a license based on server cores, without the need for CALs. The Standard edition has some scalability limitations. It is limited to 4 CPU sockets, 24 CPU cores, 128 GB of memory, and a 2-node Basic Availability Group with only 1 availability database, which is like the deprecated Database Mirroring feature.

Enterprise License

The Enterprise edition of SQL Server 2016 is licensed by server cores: either all the cores in the physical server or all the cores in a virtual machine, with a 4-core minimum. Enterprise edition scalability is limited in most cases to operating system maximums. Software Assurance can be added to the Enterprise edition, providing more flexibility for moving licenses with License Mobility.

For each fully licensed active virtual machine, you can have one SQL Server license-free passive virtual machine, such as the target of an availability group, log shipping, or database mirroring. However, because the virtual machine is passive, it is standby only, becoming active only if the active virtual machine fails. If you allow a secondary availability group to be an active read-only database copy, or if you make a backup copy from a secondary instance, that instance must be fully licensed. All passive secondary virtual machines other than the first free one must be licensed. If SQL Server instances can failover independently, then each SQL Server OS environment (OSE) requires separate licensing. Carefully consider SQL Server licensing when designing your SQL Server deployment.

Data Protection

The Cisco HyperFlex system supports snapshots for quick backup and replication and clones for fast and efficient testing and development.

Snapshots

The HX Data Platform uses metadata-based, zero-copy snapshots to facilitate backup operations and remote replication: critical capabilities in enterprises that require always-on data availability. Space-efficient snapshots allow you to perform frequent online backups of data without the need to worry about consumption of physical storage capacity. The snapshot data can be moved to a backup repository, or if the snapshots are retained, they can be restored instantaneously.

- **Fast snapshot updates:** When modified-data is contained in a snapshot, it is written to a new location, and the metadata is updated, without the need for read-modify-write operations.
- **Rapid snapshot deletions:** You can quickly delete snapshots. The platform simply deletes a small amount of metadata, rather than performing a long consolidation process, as needed by solutions that use a delta-disk technique.

Many basic backup applications read the entire data set or all the changed blocks since the last backup at a rate that is usually as fast as the storage can provide or the operating system can handle. This process can have performance implications because the Cisco HyperFlex system is built on Cisco UCS with very fast 10 Gigabit Ethernet on each host, which could result in multiple gigabytes per second of backup throughput with just a few simultaneous backup jobs. These basic backup applications, such as Microsoft Windows Server Backup, should be scheduled during off-peak hours, particularly the initial backup operation if the application uses some form of change-block tracking.

Full-featured backup applications, such as Veeam Backup and Replication v9.5, can limit the amount of throughput that the backup application can consume, which can protect latency-sensitive applications during the production hours. With the release of Veeam v9.5 Update 2, Veeam is the first partner to integrate HX Data Platform native snapshots into its product. HX Data Platform native snapshots do not experience the performance penalty of delta-disk snapshots and do not require intensive disk I/O operations during snapshot consolidation when snapshots are deleted.

Particularly important for SQL Server administrators is the capability to take a Microsoft Volume Shadow Copy Service (VSS) quiesced snapshot that is application aware. The Veeam Explorer for SQL Server can provide transaction-level recovery within the VSS framework. The Veeam Explorer for SQL Server can restore SQL Server databases from the backup restore point, from a log replay to a point in time, and from a log replay to a specific transaction—all without taking the virtual machine or SQL Server service offline.

Clones

In the HX Data Platform, clones are writable snapshots that can be used to rapidly provision copies of the SQL Server infrastructure for test and development environments. These fast, space-efficient clones rapidly replicate storage volumes so that virtual machines can be replicated through just metadata operations. Disk space is consumed in the clones only when data is written or changed in the virtual machine. With this approach, hundreds of clones can be created and deleted in minutes. Compared to full-copy methods, this approach can save a significant amount of time, increase IT agility, and improve IT productivity. SQL Server administrators can easily clone a production database and test for compatibility with a patch or update to SQL Server, Windows, or a custom front-end application. They can then easily remove the clones after testing is complete.

Storage Configuration: Configuring Additional Data Stores

For most deployments, a single HX Data Platform data store is sufficient, resulting in fewer objects to manage. The HX Data Platform is a distributed file system that is not vulnerable to many of the problems that face traditional systems that require data locality. A VMDK does not have to fit within the available storage of the physical node. If the cluster has enough space to hold the configured number of copies of the data, the VMDK will fit. Similarly, moving a virtual machine to a different node in the cluster is a host migration; the data itself is not moved.

In some cases, however, additional data stores may be beneficial. For example, an administrator may want to create an additional HX Data Platform data store for logical separation. Because performance metrics can be filtered to the data store level, isolation of workloads or virtual machines may be desired. The data store is thinly provisioned on the cluster. However, the maximum data store size is set during data-store creation and can be used to keep a workload, a set of virtual machines, or end users from running out of disk space on the entire cluster and thus affecting other virtual machines.

Another good use for additional data stores is to assist in the throughput and latency on high-performance SQL Servers. If the cumulative IOPS of all the virtual machines on an ESX host surpasses 10,000 IOPS, the system may begin to reach that queue depth. In ESXTOP, you should monitor the Active Commands and Commands counters, under Physical Disk NFS Volume. A SQL Server that itself surpasses 10,000 IOPS should have more than one database file in the file group, and those files should be placed in two or more HX Data Platform data stores.

Virtual Machine Configuration

This section presents some best practices for configuring virtual machines.

SCSI Controller

Disk I/O is queued at many levels in the stack, and understanding where bottlenecks can occur can help you make design decisions. In a virtual machine, the factor with the biggest impact is the queue depth that is set on the SCSI controller. By default, a virtual machine will use the LSI Logic SAS SCSI controller, which has an unchangeable queue depth of 32. VMware instead recommends the PVSCSI controller, which has a default queue depth of 64. A virtual machine that is running SQL Server and that requires fewer than 1000 IOPS will probably be fine with the default settings, which simplifies hyperconvergence. Experienced SQL Server administrators, however, are more cautious and used to isolating everything to separate disks. The Cisco HyperFlex best practice for SQL Server is to use PVSCSI controllers.

Traditionally the PVSCSI controller was not supported as a boot device, and because a virtual machine can have a maximum of four SCSI controllers, one was used for boot, with up to three additional PVSCSI controllers for databases and transaction logs. With recent versions of Windows Server, you can change the original controller (SCSI controller 0) to PVSCSI after verifying that the driver is properly installed in Windows.

PVSCSI Queue Depth

For many of customers who deploy SQL Server, the default settings will be sufficient. After a database nears the 1000 IOPS threshold, however, consider increasing the PVSCSI queue depth from the default of 64 to 254, as noted in the VMware knowledgebase. You should increase the RequestRingPages and MaxQueueDepth values to 32 and 254 respectively. Because the queue depth setting is per SCSI controller, consider adding PVSCI controllers to increase the total number of outstanding IOPS that the virtual machine can sustain.

A good indicator that you do not have enough queue depth is latency that is more than 10 percent higher in the guest system than the amount visible in the Cisco HyperFlex performance chart available in the Cisco HyperFlex user interface or ESXTOP. To verify this value in an existing live environment, check Windows Performance Monitor to see whether the cumulative active queue depth of all the VMDKs on the controller is sustained at greater than the queue depth of the controller during intensive I/O processing. For example, if two database files in separate VMDKs each see sustained spikes of 80 in the queue while you are using the LSI SAS controller, you can switch to PVSCSI to double the controller queue depth (from 32 to 64). Placing each VMDK on a separate PVSCSI controller would again double the available maximum queue depth (from 64 total to 64 each, or 128). Changing the registry setting for the PVSCSI queue from 64 to 254 would change the maximum queue depth available to the database from 128 to 508 in this example.

VMDK Layout

Small SQL Servers can run sufficiently with a single VMDK, the C: drive, that contains everything. As the number of IOPS of a database scales up, isolating different workloads on their own VMDKs can increase performance. The best practice is to isolate all workloads on their own VMDKs. Create separate VMDKs for the operating system, paging file, SQL Server binary files, tempdb, tempdb transaction log, user databases, and user transaction logs. Be sure to follow the LUN layout guidance and to consider both the number of tempdb files and the number of user database files to be deployed.

VMXNET3 Network Interface Card

When a virtual machine starts to approach a gigabit per second in bandwidth, consider switching to VMXNET3 network interface cards (NICs) instead of the default VMware E1000 network card. VMXNET3 is designed for the best performance and requires VMware Tools to be installed. Enable receive-side scaling (RSS) on VMXNET3 NICs, which allows the network drivers to spread the incoming TCP traffic across multiple CPUs, in the guest virtual machine on the VMXNET3 adapter.

Virtual CPU NUMA

Virtual CPU NUMA (vNUMA) exposes NUMA, or non-uniform memory access topology, to the guest operating system. In multisocket motherboards, the memory DIMMs are assigned to a socket, so processes running on a CPU experience a performance penalty when they access memory that is assigned to the other socket. A simple guideline is to size the virtual machine with CPU cores with a quantity that is less than or equal to the number of CPU cores that are on one NUMA node on the physical CPU. In most cases, that is the number of cores on the processor, but not always. If more cores are required, use a multiple of the number of cores in the NUMA node.

Virtual Cores per Virtual Socket

When configuring a virtual machine, you can set both the number of virtual sockets (vSockets) and the number of virtual cores (vCores) per socket. VMware recommends setting the number of cores per socket to 1 and increasing the number of virtual sockets when allocating CPU for SQL Server virtual machines.

High-Performance VMware ESX Policy

The Cisco HyperFlex system sets the ESX power policy to High Performance. It is important for storage controller performance for this setting to remain at High Performance.

CPU and Memory Reservation

SQL Server is resource intensive, so you should use care to help ensure that the hypervisor, operating system, and SQL Server are not constantly battling over computing and memory resources. As mentioned earlier, SQL Server minimum and maximum memory can be set at the SQL Server layer, but often administrators deploy other virtual machines in a highly overprovisioned manner. Taking the precaution of setting memory and CPU reservation on important SQL Server virtual machines can protect them from unanticipated resource consumption by the smaller overprovisioned virtual machines. When performance is the primary goal for a virtual machine, set memory reservation equal to the provisioned memory, and reserve at least one CPU core's worth of megacycles in the virtual machine's resource allocation settings.

If the SQL Server User Right > Lock Pages in Memory option is set, be sure that the virtual machine memory reservation matches the amount of memory assigned to the virtual machine. VMware recommends the following formula to estimate the amount of memory to reserve for SQL Server in a virtual machine:

SQL Server maximum memory = Virtual machine memory – Thread stack – OS memory – Virtual machine overhead

Thread stack = SQL Server maximum worker threads x Thread stack size

Thread stack size = 1 MB on x86

= 2 MB on x64

= 4 MB on ia64

OS memory = 1 GB for every 4 CPU cores

VMware ESX Configuration

This section presents some best practices for configuring VMware ESX.

VMware vSphere Storage I/O Control

You can use VMware vSphere Storage I/O Control (SIOC) to prevent a noisy neighbor from consuming all the storage I/O space and starving other virtual machines in the cluster, which may require a fraction of the I/O space as a percentage of the total cluster IOPS. Unfortunately, SIOC does not differentiate on I/O size, but it can be enabled per data store and configured with a latency threshold that, when reached, will apply a simple quality-of-service (QoS) policy across the data store I/O processing.

VMware ESX reqCallThreshold Value

By default, in ESX the reqCallThreshold value is 8, meaning that the I/O in the virtual host bus adapter (vHBA) queue won't flush to a lower layer until the threshold is reached. Some latency-sensitive databases with VMware Version 11 hardware experience improved latency when this value is lowered. This value can be lowered in the VMware VMX file of the virtual machine or globally in ESX. For exact settings, see the VMware white paper Performance Best Practices for VMware vSphere 6.0.

VMware Distributed Resource Scheduler Anti-Affinity Rules

When using high-availability features in SQL Server, such as AlwaysOn Availability Groups, be sure to configure anti-affinity rules in VMware Distributed Resource Scheduler (DRS) to indicate that virtual machines participating in an Availability Group should be kept apart on different physical hosts. If both virtual machines in a two-copy Availability Group were homed to the same physical server, if that server fails all copies of the data will be moved with VMware HA to a new host, but a SQL Server outage would occur. With anti-affinity rules, VMware would work to isolate the virtual machines to separate hosts, and a host outage would not affect service availability because SQL Server would activate the passive copy of the databases in synchronous mode.

When you use log shipping, database mirroring, or asynchronous mode with availability groups, the activation is seldom automatic, but it is still a best practice to use anti-affinity rules in these cases.

Anti-affinity rules can also help balance a cluster when you are deploying larger virtual machines that consume a large percentage of the physical server space. For example, if you are deploying three large SQL Server virtual machines that each consume 40 percent of the memory or computing cycles on an ESX host, you can use anti-affinity rules to help ensure that, in most cases, no more than one of these virtual machines is ever on a single host. This configuration enables, for example, an SLA that requests that no more than 75 percent of the host resources be being used in production to handle a host failure without any impact on important services.

Conclusion

The Cisco HyperFlex HX Data Platform revolutionizes data storage for hyperconverged infrastructure deployments that support new IT consumption models. The platform's architecture and software-defined storage approach gives you a purpose-built, high-performance distributed file system with a wide array of enterprise-class data management services. With innovations that redefine distributed storage technology, the data platform gives you the hyperconverged infrastructure you need to deliver adaptive IT infrastructure.

Cisco HyperFlex systems in all-flash configurations lower both operating expenses (OpEx) and capital expenditures (CapEx) by allowing you to scale as you grow. They also simplify the convergence of computing, storage, and network resources. Size and acquire what you need this quarter, easily scaling the storage with automated rebalancing after you add disks to the converged nodes or add more converged nodes. If more computing resources are required, use both approved rack and blade Cisco UCS servers, adding them to the cluster as computing-only nodes.

The SQL Server best practices discussed in this document are guidelines for high-performance (greater than 500 IOPS) virtual machines. Most virtual machines containing SQL Server will work fine without the need for you to worry about a multitude of settings at the storage, ESX, virtual machine, or SQL Server instance layers.

Unlike with a lot of traditional storage systems, you can easily grow a VMDK or even the data store. Veeam integration with Cisco HyperFlex systems reduces the impact on the cluster by eliminating the need for delta-disk consolidation and enables additional recovery functions with VSS application-consistent snapshots. Native cloning is space efficient and fast and provides SQL Server administrators with quick access to production data for testing and to optimization without requiring additional storage capacity.

Both dedicated SQL Servers and applications that use SQL Server behind the scenes are excellent candidates for the high-performing Cisco HyperFlex all-flash system.

For More Information

- VMware SQL Server availability: http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-availability-and-recovery-options.pdf
- VMware support for Microsoft Cluster Service and Windows Server Failover Clustering with shared storage: https://kb.vmware.com/kb/2147661
- Microsoft SQL Server licensing: https://www.microsoft.com/en-us/sql-server/sql-server-2016-pricing
- SQL Server tuning for high-performance workloads: https://support.microsoft.com/en-us/help/2964518/recommended-updates-and-configuration-options-for-sql-server-2012-and-sql-server-2014-with-high-performance-workloads
- vSphere 6.0 performance best practices: http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-perfbest-practices-vsphere6-0-white-paper.pdf
- VMware SQL Server best practices: http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-best-practices-guide.pdf
- Cisco HyperFlex white paper: http://www.cisco.com/c/dam/en/us/products/collateral/hyperconverged-infrastructure/hyperflex-hx-series/white-paper-c11-736814.pdf

Author: Robert Quimbey

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C11-738817-00 03/17