



Hewlett Packard
Enterprise

HPE Nimble Storage Deployment Considerations for Microsoft SQL Server

Contents

Introduction.....	5
Types of SQL Server Workloads.....	5
OLTP Workloads.....	5
DSS or DW Workloads	5
OLAP Workloads.....	5
Optimizing SQL Server Performance.....	7
Memory Management for SQL Server.....	7
tempdb Settings.....	7
Trace Flag 1117.....	7
Trace Flag 1118.....	8
Volume Maintenance for the SQL Server Service Account.....	8
MAXDOP.....	8
Cost Threshold for Parallelism.....	8
Connecting to HPE Nimble Storage Arrays.....	9
HPE Nimble Storage Windows Toolkit.....	9
Multipath I/O.....	9
FC Connectivity.....	9
iSCSI Connectivity.....	10
Volume Formatting.....	10
Performance Policies, Compression, and Deduplication.....	11
Default Performance Policies.....	11
Custom Performance Policies.....	11
Compression.....	12
Deduplication.....	12
Storage Layout for SQL Server Volumes.....	13
tempdb Placement.....	13
OLTP Workload Considerations.....	13
DSS/DW and OLAP Workload Considerations.....	14
System Databases.....	14
Backups, Snapshots, Replication, and Volume Collections.....	15
Backup Considerations.....	15

Snapshots.....	15
Replication.....	16
Volume Collections	17
SQL Server High Availability on HPE Nimble Storage Arrays.....	18
Always On Failover Cluster Instance.....	18
Always On Availability Groups.....	18
Cloning SQL Server Databases	19
Clone a Database.....	19
Restoring SQL Server Database Volumes.....	20
Restore Database Volumes.....	20
Considerations for Virtualizing SQL Server.....	21
Datastore and Guest Storage Considerations for VMware vSphere.....	21
Guest Storage Considerations for Hyper-V.....	22
About the Author.....	23
Version History.....	24

© Copyright 2018 Hewlett Packard Enterprise Development LP. All rights reserved worldwide.

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel[®], Itanium[®], Pentium[®], Intel Inside[®], and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

Microsoft[®] and Windows[®] are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe[®] and Acrobat[®] are trademarks of Adobe Systems Incorporated. Java[®] and Oracle[®] are registered trademarks of Oracle and/or its affiliates.

UNIX[®] is a registered trademark of The Open Group.

Publication Date

Friday March 9, 2018 12:13:15

Document ID

bay1491596371227

Support

All documentation and knowledge base articles are available on HPE InfoSight at <https://infosight.hpe.com>. To register for HPE InfoSight, click the *Create Account* link on the main page.

Email: support@nimblestorage.com

For all other general support contact information, go to <https://www.nimblestorage.com/customer-support/>.

Introduction

This guide provides design and deployment guidance for running Microsoft SQL Server data management software on HPE Nimble Storage arrays from Hewlett Packard Enterprise (HPE). Readers are assumed to have a working knowledge of the following products and technologies:

- SQL Server
- The operating system (OS) on which they are deploying their solution
- Basic HPE Nimble Storage array operations

Although the guide focuses on SQL Server 2016, which is the latest edition of the Microsoft database platform, many of the core design considerations are also applicable to SQL Server 2014 and SQL Server 2012.

SQL Server databases can be deployed on HPE Nimble Storage arrays in many different ways. Configuration options change based on datacenter infrastructure, server hardware, and virtualization platforms. The document reviews some of the design considerations, parameters, and best practices that SQL Server database administrators, storage architects, and infrastructure administrators should evaluate. The guide also covers minor differences in deployment considerations for HPE Nimble Storage all-flash arrays and adaptive flash arrays where applicable.

Types of SQL Server Workloads

Understanding the characteristics of SQL Server workloads is essential for the proper deployment of SQL Server on HPE Nimble Storage arrays. Workload characteristics play a role in the design of the storage layout and in the configuration of performance policies, snapshots, and replication schedules. Typically, SQL Server handles the following types of workloads:

- Online transaction processing (OLTP) workloads
- Decision support system (DSS) or data warehouse (DW) workloads
- Online analytical processing (OLAP) workloads

OLTP Workloads

OLTP databases are often tied to business processes, with updates and inserts being performed in high frequency by end users. As a result, OLTP workloads are characterized by small datasets and a large number of simple queries. A common measure for OLTP workloads is the number of transactions per second.

OLTP workloads usually present a random, small-block, and mixed read/write workload to storage. Read/write ratios can vary from 60/40 to 90/10. OLTP workloads are often latency sensitive.

DSS or DW Workloads

Data warehouses are the aggregation of tiers of business data to run analytics, plan strategies, improve processes, and support decisions. Data often comes from OLTP databases in the business or from external sources, and it is refreshed and imported through batch or ETL (extract, transform, load) processes.

Refreshes of DW data are typically characterized by long-running batch jobs that present a sequential, large-block, write-intensive workload to storage. By contrast, reporting and data-mining processes to extract data from the warehouse are presented as intensive, large-block, sequential-read workloads.

OLAP Workloads

Whereas data warehouses store historical data, OLAP systems retrieve and manipulate data through complex queries and scans, organizing the data into different multidimensional views or reports for in-depth analysis. Data is commonly stored in multidimensional cubes, and OLAP systems are often a component of a data-driven DSS.

Workloads from OLAP queries to their source databases are characterized as large-block, sequential-read workloads for data mining, modeling, and analysis. In addition, ad-hoc queries from OLAP systems can spawn concurrent sequential workloads, which result in a random workload to the underlying storage system. While transactions per second are a common measure for OLTP workloads, response time and throughput are essential for OLAP systems.

Optimizing SQL Server Performance

When deploying SQL Server on HPE Nimble Storage arrays, consider modifying certain SQL Server settings to optimize the performance and scalability of the system. These settings are best explored on systems that are dedicated to SQL Server.

SQL Server 2016 comes with many enhancements in functionality and performance, so some tuning options are no longer required for SQL Server 2016. These options are noted where applicable.

For detailed information about SQL Server deployment and configuration, see [Server Configuration Options \(SQL Server\)](#).

Memory Management for SQL Server

By default, SQL Server automatically manages the memory that the database engine requires to execute queries, tasks, and optimizations. Depending on the workload, it may be necessary to control memory allocations or reservations for SQL Server to improve performance. Some settings to consider for tuning are the maximum and minimum memory for SQL Server and the lock pages in memory option. Also consider enabling trace flags, such as flag 834.

Microsoft recommends tuning memory options only after thorough testing. For more information, see [Server Configuration Options \(SQL Server\)](#).

BEST PRACTICE Evaluate whether a workload requires tuning the SQL Server memory settings. Do not edit memory options unless thorough testing reveals a performance benefit to the environment.

tempdb Settings

The SQL Server tempdb database is a global resource for all users, databases, and processes of SQL Server. Historically, the configuration of tempdb has been a manual process, leading to many unanticipated performance bottlenecks due to resource contention. To avoid performance problems caused by tempdb, consider the following Microsoft best practices:

- Separate tempdb data files into their own volume. Because tempdb is available as a global resource, growth patterns may be unanticipated and affected by user or application access.
- Create multiple tempdb files to avoid waits on tempdb resources. For installations running SQL Server versions earlier than SQL Server 2016, match the number of data files to the number of logical processors when that number is smaller than or equal to eight. SQL Server 2016 automatically configures multiple tempdb files during installation.
- Ensure that all tempdb data files are of equal size, and preallocate the required space to the files.
- Consider enabling trace flags 1117 and 1118.

BEST PRACTICE Follow Microsoft best practices for tempdb settings. Use multiple files for tempdb, equal to the number of logical processors when that number is smaller than or equal to eight. SQL Server 2016 automatically configures tempdb with multiple files during installation, based on the detected number of cores on the system.

Trace Flag 1117

In SQL Server 2014 and SQL Server 2012, trace flag 1117 forces all files in a filegroup to grow uniformly. This tuning option is especially important for tempdb databases because the Microsoft recommendation is to use multiple data file deployments. By growing files evenly, the Microsoft proportional fill algorithm targets all files uniformly rather than focusing on files with more free space.

In SQL Server 2016, trace flag 1117 has been replaced by the **ALTER DATABASE** command with the **AUTOGROW_ALL_FILES** option. In SQL Server 2014 and SQL Server 2012, trace flag 1117 is a global setting that affects all databases.

BEST PRACTICE Carefully consider whether to use trace flag 1117 because it affects all databases. Instead, consider preallocating space to tempdb files to accommodate typical workloads.

Trace Flag 1118

In SQL Server 2014 and SQL Server 2012, trace flag 1118 enables the allocation of full extents to each database object, thereby eliminating the contention on SGAM pages.

In SQL Server 2016, all allocations in tempdb and user databases use uniform extents, and trace flag 1118 is no longer required.

BEST PRACTICE Enable trace flag 1118 in SQL Server 2014 and SQL Server 2012.

Volume Maintenance for the SQL Server Service Account

Giving the SQL Server service account permissions to perform volume maintenance tasks through the Windows local security policy enables SQL Server to leverage instant file initialization (IFI). IFI allows SQL Server to immediately use space that it has allocated for data files rather than waiting for a zero-out process to complete. This capability greatly reduces the time required for database provisioning, database expansion, and database restores.

IFI does not apply to transaction logs. SQL Server 2016 has the option of enabling IFI as part of the installation.

BEST PRACTICE Enable IFI for SQL Server.

MAXDOP

The **max degree of parallelism** option (MAXDOP) controls the number of processors that are used to execute a query in a parallel plan. The default value for Microsoft installations is 0, meaning that all processors can be used.

Microsoft recommends changing this setting to a value that more closely matches the workloads in the environment. The guidance is to start with a value at or below the number of logical processors in a single NUMA node, not to exceed 8 without testing. Microsoft also offers guidelines for higher and lower values.

In SQL Server 2014 and SQL Server 2012, **max degree of parallelism** is a global setting. In SQL Server 2016, **max degree of parallelism** can be set per database.

BEST PRACTICE Observe the Microsoft recommendations for **max degree of parallelism**. SQL Server 2016 offers the flexibility of setting the value at the database level rather than affecting the entire instance.

Cost Threshold for Parallelism

The **cost threshold for parallelism** option controls the tipping point for SQL Server to determine whether a query benefits from going parallel over serial. The default setting on SQL Server is 5, which is very low.

If needed, this number can be increased to prevent unnecessary parallelization of SQL Server queries for OLTP workloads. For OLAP and DW environments, a low setting is good for larger queries that benefit from parallelization.

BEST PRACTICE Examine execution plans for queries to determine whether **cost threshold for parallelism** must be increased or decreased.

Connecting to HPE Nimble Storage Arrays

For detailed instructions on how to prepare the HPE Nimble Storage array for connection with the SQL Server host, see the *GUI Administration Guide* for the NimbleOS version that is running on the array. The guide is available on [HPE InfoSight](#).

Begin preparing the host by ensuring that all necessary updates have been applied to the Windows OS.

HPE Nimble Storage Windows Toolkit

To facilitate connectivity and volume management and to leverage application-consistent snapshots for SQL Server, install the HPE Nimble Storage Windows Toolkit (NWT) on the host. NWT is an essential plugin that provides the following capabilities and tools:

- HPE Nimble Storage Volume Shadow Copy Server (VSS) services for SQL Server
- iSCSI and Fibre Channel (FC) connection management
- The HPE Nimble Storage Device-Specific Module (NDSM) for multipath I/O (MPIO)
- The HPE Nimble Storage Setup Manager (NSM) tool

Also included in the toolkit is a Windows PowerShell module with cmdlets for managing volume attributes, cloning, and connectivity from the Windows host.

BEST PRACTICE Install the NWT on all Windows hosts that run SQL Server, including virtual machines. The NWT can be downloaded from [HPE InfoSight](#).

Multipath I/O

Enable multipath with either iSCSI or FC connectivity. Before installing NWT, add the MPIO feature to the Windows host by using Server Manager or PowerShell; in PowerShell, use the **Add-WindowsFeature** cmdlet.

During the NWT installation, the appropriate NDSM for MPIO is installed on the host. A minimum of two paths, one to each controller, is required for availability. However, more than one path per controller is desirable. HPE recommends dedicating paths according to the protocol:

- For FC, dedicate each set of paths on a physical fabric or vSAN.
- For iSCSI, use separate, private subnets that are dedicated to iSCSI connectivity.

BEST PRACTICE For data resiliency and performance, enable multipath on all production systems.

FC Connectivity

For FC connections to the HPE Nimble Storage array, consider the host bus adapter (HBA) queue depth setting. This setting determines the maximum number of I/O operations that can be serviced by a connection at one time.

Many HBAs have default queue depth settings that are too low to accommodate SQL Server I/O requirements. Default values and modification parameters are dependent on the HBA brand and model. A low queue depth setting can result in excess I/O queuing, incurring latency and poor performance for SQL Server.

A typical default queue depth setting for an HBA is 32. However, for highly transactional OLTP workloads that require maximum I/O, consider a higher queue depth in the range of 64–256. OLAP and DW workloads have a lower requirement for HBA queuing, and a setting of 32–64 might suffice. Perform thorough testing to determine the best queue depth for the SQL Server environment.

BEST PRACTICE For FC connectivity, increase the HBA queue depth to accommodate SQL Server I/O requirements. Start with a value of 64, and then test to determine whether increases are required.

iSCSI Connectivity

For iSCSI connections to the HPE Nimble Storage array, use dedicated network interfaces from the Windows host. For each iSCSI interface, unbind the **Client for Microsoft Networks** and the **File and Printer Sharing for Microsoft Networks** components and remove the setting for **Register this connection's addresses in DNS**.

For maximum performance, use 10 GbE connectivity and jumbo frames for iSCSI traffic. Jumbo frames can be configured on each adapter through Device Manager. Jumbo frames are especially important for databases that require more throughput—for example, OLAP and DW databases.

Use HPE Nimble Storage Connection Manager (NCM), installed by NWT, to establish iSCSI connections to volumes on the array. NCM automatically gathers interface, subnet, and volume information and then calculates the optimal number of connections for iSCSI.

For more information about preparing the SQL Server host to connect to HPE Nimble Storage arrays, see the *Windows Integration Guide* on [HPE InfoSight](#).

BEST PRACTICE For iSCSI connectivity, use dedicated ports on the Windows hosts, and configure each interface with an IP address on a private, dedicated subnet. Enable jumbo frames to provide maximum throughput for large-block workloads.

Volume Formatting

Volumes for SQL Server data, transaction log, and tempdb files must be formatted with a 64 KB NTFS allocation unit size. SQL Server binaries and system databases can use the default 4 KB allocation unit. When formatting volumes that will contain large files, typically over 1 TB, use the **/L** option to enable large file record segments (FRS).

BEST PRACTICE Format SQL Server volumes for data, transaction log, and tempdb files by using NTFS with the 64 KB allocation unit size. When formatting volumes that will contain files larger than 1 TB, enable FRS.

Performance Policies, Compression, and Deduplication

Before deploying SQL Server on HPE Nimble Storage arrays, evaluate the characteristics of the workloads that the system will handle and the requirements for performance and database capacity. Each of these elements plays a role in the design of the underlying storage for SQL Server, including optimization settings that can be controlled at the volume level or through global performance policies.

Default Performance Policies

Performance policies contain application-specific storage parameters, such as compression, volume block size, caching (on adaptive flash arrays), and deduplication (on all-flash arrays). During volume creation, the appropriate policy is applied to the volume.

HPE Nimble Storage arrays have preconfigured policies for SQL Server. These policies are based on the analysis of data gathered from [HPE InfoSight](#), the HPE Nimble Storage cloud-based analytics platform that ingests more than 30 million data points daily from customer arrays. The preconfigured performance policies for SQL Server have the settings listed in the following table.

Table 1: Performance policy settings

Volume Type	Block Size (KB)	Caching (Adaptive Flash Array Only)	Deduplication (All-Flash Array Only)	Compression
DATA	8	Enabled	Disabled	Enabled
LOG	4	Disabled	Disabled	Enabled

On adaptive flash arrays, caching is disabled for the transaction log performance policy because the I/O traffic of transaction logs normally has sequential-write characteristics, so random read caching has limited value. It is best to leave caching off for any volume that uses transaction log-type I/O to prevent the adaptive flash array cache from being populated with data that will not be read back (called a *dirty cache*).

The default performance policies work best for OLTP workloads.

Custom Performance Policies

Custom performance policies may be necessary to accommodate particular workloads or deployments of SQL Server. For example, for OLAP and DW workloads, a block size of 8 KB for the data files may not yield optimal throughput. HPE recommends that a custom performance policy with a larger block size of 16 KB or 32 KB be created for these data files to better align I/O and increase throughput.

When considering the block size for a volume, examine both the OS-level allocation size and the block size that is generated by the application. To prevent misalignment, make the volume block size no larger than the application I/O size and also no larger than the NTFS cluster size:

`NIMBLE_VOL_BLOCK_SIZE ≤ APP_IO_SIZE`

`NIMBLE_VOL_BLOCK_SIZE ≤ NTFS_CLUSTER_SIZE`

If in doubt, set the volume block size to a lower value, such as 4 KB. A low value might be suboptimal for peak throughput performance but will prevent misalignment.

For more information about custom performance policies, see the *GUI Administration Guide* on [HPE InfoSight](#).

BEST PRACTICE Use custom performance policies when the default policies are not optimal. Test for optimal performance during database and volume creation because block sizes cannot be changed on existing volumes. If in doubt, use the default SQL Server policies. The default policies work for all workloads.

Compression

Inline compression is enabled by default on all HPE Nimble Storage arrays. Data gathered from HPE InfoSight analytics shows that compression for SQL Server data and transaction logs results in data reduction savings of 2.5x to 3x. Disabling compression provides no noticeable performance gain, and HPE recommends that compression be enabled on all arrays.

SQL Server introduced native compression with the 2008 release. Since then, subsequent releases have further enhanced the feature. SQL Server supports compression for rowstore tables and indexes and for columnstore tables and indexes. However, native SQL Server compression cannot compress data in large object pages (LOB)—for example, (max) columns, XML, ntext, image, and several others. Inline compression on the HPE Nimble Storage array effectively compresses LOB data and can be used in conjunction with native SQL Server compression.

For databases that use SQL Server Transparent Data Encryption (TDE), use SQL Server page and row compression for data reduction because the HPE Nimble Storage array cannot compress that data. Keep in mind that backups for TDE-enabled databases are also encrypted, so space savings are limited to the native SQL Server backup compression. Consider using HPE Nimble Storage SmartSecure encryption for volumes in which encryption is required because data is compressed inline before encryption.

BEST PRACTICE Use array inline compression on all SQL Server volumes.

Deduplication

Inline deduplication is enabled by default on all-flash arrays. Deduplication of SQL Server data and log files provides varying degrees of benefit, depending on the data that is stored in the database and on the deployment. Although deduplication is disabled on the performance policy, it can be enabled on a per-volume basis.

For volumes that store SQL Server databases running in Always on Availability Groups, deduplication provides data reduction if the replicas are on the same array as the source database.

BEST PRACTICE Enable deduplication only if testing shows gains in the environment or if database copies are present, as in Always on Availability Groups. Keep in mind that most data reduction savings come from compression, not deduplication.

Storage Layout for SQL Server Volumes

In general, using multiple volumes for SQL Server deployments offers several advantages:

- The ability to apply custom performance policy settings to specific volumes
- The ability to control cache use for volumes on adaptive flash arrays
- The choice of limiting replication of certain volumes, such as the tempdb volume
- A gain in performance for I/O-intensive databases, especially if the queue depth of a device is at risk of becoming saturated
- The ability to group databases in separate volume collections and apply custom backup and replication schedules
- The ability to quickly recover specific databases with volume-based restores

Before designing the storage layout, carefully consider the administrative overhead of managing multiple volumes. The ideal solution provides a balance between optimization and administration.

tempdb Placement

The extensive use that OLTP, DSS/DW, and OLAP workloads make of the tempdb database can be the source of performance bottlenecks if storage for tempdb is not properly defined.

For workloads that use tempdb extensively, HPE recommends that tempdb data files be separated into their own volume and that multiple filegroups be used to prevent resource contention. For more information about tempdb, see [tempdb Settings](#) on page 7.

Choose a performance policy with a 4 KB volume block size and compression enabled for the tempdb volumes. The tempdb database has a high percentage of writes because this database is used as scratch space for SQL Server. For this reason, like transaction log files, tempdb may not benefit from caching on adaptive flash arrays. HPE recommends that tempdb volumes be tested with caching off and with caching on to determine efficiency. For all-flash arrays, deduplication should be disabled.

Log files for tempdb can be colocated with tempdb data files. Do not replicate or create snapshots of tempdb volumes because the database is recreated during a SQL Server restart.

BEST PRACTICE For databases that make extensive use of tempdb, separate tempdb data files into their own volume. Caching might be of no benefit to tempdb volumes, depending on the workload.

OLTP Workload Considerations

Typical OLTP databases see fairly random read and write patterns to the data file, whereas the transaction log sees mostly sequential-write traffic. Reads to the transaction log are increased if the database performs frequent transaction log backups, transaction log shipping, or some forms of database mirroring.

Traffic to the data files is typically between 8 KB and 64 KB, although it can span a wide range of block sizes. For example, SQL Server 2016 has increased the number of contiguous 8 KB pages from 32 to 128 for lazy writer, checkpoint, create index, and bulk insert workloads. This increase results in a 1,024 KB write to data files. By contrast, log files see traffic in the range of 512 bytes to 60 KB.

For OLTP databases, HPE recommends that data files and transaction log files be placed on separate volumes. For data files, choose a performance policy with an 8 KB volume block size and compression enabled. On adaptive flash arrays, ensure that caching is enabled. A volume block size of 8 KB aligns with OLTP I/O patterns of varying sizes, and caching optimizes random read performance.

I/O-intensive OLTP workloads on adaptive flash arrays might require data volumes to be pinned to cache to achieve low latency with 100 percent cache hits and all-flash-type performance. If the device queue depth is

a bottleneck, consider splitting the database into several files and placing the files on different volumes. This approach increases the overall queue depth, allowing for more outstanding I/Os.

When splitting databases into smaller files, ensure that all files are of equal size. SQL Server uses a proportional fill algorithm to write to SQL files, preferring writes to files with more free space. For files to be written to in a parallel fashion, size all files equally and consider using trace flag 1117. For SQL Server 2016, use the **Alter Database** command with the **AUTOGROW_ALL_Files** option.

It is not necessary to allocate storage specifically for each database in an instance, unless databases require very different caching, block size, performance, or snapshot settings. Several data files and transaction logs pertaining to multiple databases can coexist on the same volume. However, be cautious when colocating databases because this layout affects volume-level restore decisions.

BEST PRACTICE For optimum performance, place SQL Server data files and log files on separate volumes, and use dedicated performance policies that are geared towards OLTP workloads. Carefully consider colocating databases on the same volume because a volume-level restore affects all databases on the volume.

DSS/DW and OLAP Workload Considerations

Typical OLAP queries and batch or ETL jobs against DW databases see long-running, sequential, and large-block read or write patterns to the data file. Traffic to the data files is typically larger than 64 KB, although it can span a wide range of block sizes. Data file I/O can range from 8 KB to 8 MB, depending on the workload and on the use of new features such as columnstore indexes in a data warehouse.

As with OLTP databases, HPE recommends that data files and transaction log files for OLAP and DW databases be placed in separate volumes. Create a custom performance policy for data files that specifies a larger block size (either 16 KB or 32 KB) with compression enabled. When SQL Server is deployed on adaptive flash arrays, enable caching on data volumes because ad-hoc queries present a more random I/O pattern to the storage.

Very large OLAP and DW databases benefit from the use of multiple filegroups spread across different volumes in conjunction with table partitioning. Filegroups allow the overall increase of queue depth and parallel efficiency on the server, while table partitioning allows the organization of data so that only relevant information is accessed or loaded.

BEST PRACTICE For optimum performance, place SQL Server data files and log files in separate volumes, and use dedicated performance policies that are geared toward larger-block workloads. Consider the use of multiple filegroups across different volumes, and enable table partitioning.

System Databases

SQL Server includes the following system databases: master, model, msdb, resource, and tempdb. Storage layout considerations for tempdb are reviewed in [tempdb Placement](#) on page 13. The resource database is a hidden database that cannot be backed up, and the default installation location is with the SQL Server binaries.

System databases can be placed on an HPE Nimble Storage volume during the SQL Server installation, or they can be moved to an HPE Nimble Storage volume after the installation. All system databases, with the exception of tempdb, can be colocated on the same volume, including their log files.

Choose a performance policy with a 4 KB volume block size and compression enabled for the volume that houses the master, model, msdb, and resource system databases. On adaptive flash arrays, choose a policy with caching disabled because I/O traffic to these databases is limited.

BEST PRACTICE On adaptive flash arrays, apply a performance policy with caching disabled to the volume that houses system databases other than tempdb.

Backups, Snapshots, Replication, and Volume Collections

SQL Server databases are often tied to core business applications and are invaluable for day-to-day operations. When designing a data protection strategy for SQL Server databases that maximizes performance and scalability, factor in the recovery time objective (RTO) and the recovery point objective (RPO) for each business process that is tied to the databases.

To improve availability and data protection for SQL Server databases, follow the HPE recommendations for backups, snapshots, replication, and volume collections.

Backup Considerations

SQL Server databases support three types of recovery model: simple, full, and bulk-logged. Recovery models dictate how a transaction is logged, whether the transaction log needs to be backed up, and what type of restore options are available.

Databases in the simple recovery model write to a transaction log; however, after the transaction is committed and written to the data file, space is reclaimed and made available for other transactions. Databases in simple recovery support full and differential backups, but transaction log backups cannot be performed. In the event of a disaster, all transactions that were committed after a backup are lost and must be redone.

Databases in the full recovery model log all transactions to the transaction log where the data is kept until the transaction log is backed up or truncated. Recovery is possible to any point in time as long as backup files are available. The risk of data loss is minimized under this recovery model. Databases in full recovery support full, differential, file or filegroup, transaction log, and copy-only backups. A typical enterprise backup strategy is to create full backups weekly, differential backups daily, and transaction log backups every five minutes.

The bulk-logged recovery model should be used only for bulk operations, such as executing a refresh of DW data. When the database is in bulk-logged recovery model, bulk operations are minimally logged, but other transactions are fully logged. This model provides better performance with minimal space consumption for the transaction log file. The bulk-logged recovery model is used only intermittently and is not the normal recovery model of a database. Typically, after bulk operations are complete, the database is switched to full recovery and a full database backup is performed.

For backups of SQL Server data to an HPE Nimble Storage volume through native tools, HPE recommends using SQL Server backup compression. In addition, format the volume in Windows with a 64 KB allocation size, and apply a performance policy that has a 32 KB block size and compression enabled, with caching disabled on adaptive flash arrays. The level of data reduction with compression is limited because backups are compressed by SQL Server. For all-flash arrays, do not enable deduplication. To achieve maximum data protection, write native SQL Server backups to a secondary array.

BEST PRACTICE Databases in the full recovery model require native SQL Server backups or third-party integration for transaction log processing. Use SQL Server compression for backups to HPE Nimble Storage volumes, and apply a large-block 32 KB performance policy on the volume.

Snapshots

HPE Nimble Storage arrays can take near-instant, application-consistent snapshots of SQL Server database volumes by leveraging the HPE Nimble Storage VSS requestor and the HPE Nimble Storage VSS hardware provider that are installed as part of the NWT.

HPE Nimble Storage snapshots leverage redirect-on-write technology: Active blocks on the volume are frozen, and new writes are redirected to other blocks. Frozen blocks are tracked as the data changes. When a snapshot is deleted, the pointers to the frozen blocks are also deleted.

Because HPE Nimble Storage snapshots require a minimal amount of space, they can be taken multiple times a day. This frequency helps to reduce operational risk and meet the RTO. Multiple snapshots also provide additional restore and recovery points that become important for the RPO. Depending on the backup requirements for the database, snapshots of underlying volumes can be scheduled as frequently or infrequently as required.

The NWT must be installed on all systems that require application-consistent volume snapshots, with the exception of VMware virtual machines running SQL Server. These machines can leverage the VMware Tools VSS components for application consistency. All HPE Nimble Storage snapshots are array driven, meaning that the array acts as the requestor for the VSS snapshot.

VSS snapshots require that the volumes that contain the SQL Server database data and log files be in a volume collection. The volume collection allows VSS to coordinate a freeze in I/O of the appropriate volumes, at which point the HPE Nimble Storage array takes a consistent snapshot of all volumes in the group. Because more than one volume is involved in the snapshot process, the resulting dataset is known as a *snapshot collection*.

When smaller, less resource-intensive databases are colocated on the same volumes, monitor the VSS snapshot process for timeout problems. For the VSS snapshot process to occur, I/O to all databases stored on the underlying volumes must be temporarily frozen, at which point the HPE Nimble Storage provider initiates the snapshot. However, this process may time out if too many databases need to be quiesced or if the number of available worker threads are not enough to quiesce each database. Microsoft recommends creating snapshots of fewer than 35 databases at one time.

For deployments in which SQL Server database data and log files are colocated on the same volume (for example, SQL Server system databases), volumes can be protected as standalone volumes. Volume protection can be configured during volume creation or at a later time.

For databases in simple recovery for which transaction logs do not need to be backed up or truncated, HPE offers a full backup solution.

For databases in full recovery model, HPE Nimble Storage arrays can add a layer of protection to native SQL Server backups by performing application-consistent, copy-only backups. Copy-only backups do not interfere with native backups because they do not break the LSN chain. Transaction log backup and truncation must occur through native SQL Server backups or a third-party integration, such as the Veeam or Commvault backup solutions.

To create a copy-only backup, set the **Verify Backups** option to **No** during the creation of a volume collection or the manual creation of a snapshot collection. If **Verify Backups** is set to **Yes**, SQL Server will not perform a database consistency check (DBCC). Instead, the **Yes** setting marks the backup as a full backup within SQL Server, thereby interfering with the native SQL Server backup operations.

For SQL Server virtual machines that use VMware Tools VSS components, backups are always copy-only. For more information about HPE Nimble Storage snapshots and VSS, see the *Concepts Guide on [HPE InfoSight](#)*.

BEST PRACTICE For databases in the simple recovery model, use HPE Nimble Storage application-consistent snapshots as a full backup solution. For databases in the full recovery model, leverage snapshots as an extra layer of data protection. When more than 35 databases are colocated on the same volumes, watch for VSS timeouts.

Replication

HPE Nimble Storage replication makes it possible to copy a volume or a set of volumes to another HPE Nimble Storage array, called the replication partner. Replicated volumes contain the contents of a volume at the time when the replica was created or last updated, as well as a configured number of retained snapshots. HPE Nimble Storage replication is based on snapshots, and it is automatically initiated or updated according to volume collection schedules. Snapshots that are created ad-hoc can also be replicated if necessary.

Before initiating replication, create any custom performance policies on the replica; otherwise, replication might fail. For more information about replication, see the *GUI Administration Guide* on [HPE InfoSight](#).

BEST PRACTICE Leverage HPE Nimble Storage replication to duplicate SQL Server database volumes to another site for disaster recovery or to another on-site array for the backup of primary array data.

Volume Collections

Volume collections are consistency groups that allow the coordination of snapshot schedules and the replication of multiple volumes in a group. Volumes that contain SQL Server data and log files for a particular database should be grouped into the same volume collection. Volumes can be assigned to a volume collection at creation, or they can be added to a particular volume collection at a later time.

To create an application-consistent snapshot of a database, install NWT or VMware Tools on the SQL Server host, and place all HPE Nimble Storage volumes that house database and log files for particular databases in the same volume collection. In addition, populate application synchronization settings to include either the SQL Server host or, if the SQL Server is a virtual machine, the VMware vCenter host, along with the type of synchronization required—either Microsoft VSS or VMware vCenter.

Snapshot schedules are attached to volume collections. Carefully consider the schedules assigned to volume collections, and ensure that snapshots are created at the appropriate intervals to meet RTO and RPO requirements.

During the creation of a volume collection, set the **Verify Backups** option to **No** for copy-only backups. For full backups, set the option to **Yes**. Be cautious with this setting, because unexpected VSS full backups break the LSN chain of native SQL Server backups and may put data at risk.

Volume replication is also configured within volume collections. HPE recommends that all critical production databases be replicated to a secondary storage array. A separate schedule can be created to control replication, or all snapshots can be replicated.

BEST PRACTICE Place database and log file volumes for SQL Server databases in the same volume collection to enforce application consistency. Create the appropriate snapshot schedule to meet RTO and RPO objectives. Configure replication to secondary storage arrays for redundancy.

Example of Volume Collection Design to Coordinate Snapshot Schedules

Consider a scenario in which `SQL_ServerA` has eight user databases attached to the SQL Server instance. One of the user databases is made up of four filegroups, located on two HPE Nimble Storage volumes, volume A and volume B. Transaction log files for the user database are on volume C. The remaining seven databases are very small, with limited I/O traffic, so these databases are colocated with data files on volume D. Their transaction log files are on volume E.

In one use case for `SQL_ServerA`, volumes A, B, C, D, and E, are placed in a single volume collection to which a snapshot schedule is assigned. During the snapshot process, VSS is leveraged to briefly pause SQL Server I/O to the volumes, so the HPE Nimble Storage array can take an application-consistent snapshot.

Another viable solution for `SQL_ServerA` is to create two volume collections with different snapshot schedules. The larger database that is housed on volumes A, B, and C is backed up with copy-only backups every 30 minutes; native SQL Server backup tools are used to manage other backups. Meanwhile, the smaller databases are in simple recovery model, and a full backup of these databases is created every 15 minutes.

SQL Server High Availability on HPE Nimble Storage Arrays

HPE Nimble Storage arrays support both forms of SQL Server Always On deployments: Always On Failover Cluster Instances and Always On Availability Groups.

Always On Failover Cluster Instance

A failover cluster instance (FCI) uses Windows Server Failover Clustering (WSFC) in conjunction with shared storage to create high availability at the SQL Server instance level. Two or more nodes make up the FCI, with one node actively owning the resources. If an active node fails, resources automatically fail over to another node in the cluster. Because the FCI sits behind a virtual interface for client connectivity to SQL Server, failures are seamless to the end user.

When an FCI is deployed with HPE Nimble Storage arrays, no changes to the storage layout or performance policies are necessary. With volume collections, use the FCI virtual IP address or registered DNS as the synchronization host for application-consistent snapshots. This approach ensures that the node that owns the HPE Nimble Storage volumes is used as the VSS synchronization host.

BEST PRACTICE When creating application-consistent snapshots of FCI volume collections, use the IP address or registered DNS of the FCI as the synchronization host.

Always On Availability Groups

Always on Availability Groups (AG) are an alternative to database mirroring, providing high availability at the database level instead of at the instance level. AG replicas are hosted on different SQL Server instances, and replication between primary and secondary replicas is handled within SQL Server.

AG deployments require WSFC; however, no shared storage is required. Replication between databases is synchronous or asynchronous, and failovers can be configured automatically or manually. Like an FCI, AG can use a virtual interface, or *listener*, as the client access point. Unlike an FCI, in which the passive node cannot be used for SQL Server operations, AG secondary replicas can be configured for read-only access and be used for backups of the replicated SQL Server database. An FCI can serve as one of the AG replicas.

When an AG is deployed with HPE Nimble Storage arrays, no changes to the storage layout or performance policies are necessary. With volume collections, use any of the AG replicas as synchronization hosts for application-consistent snapshots. Snapshots must be configured as copy-only backups.

BEST PRACTICE When creating application-consistent snapshots of AG volume collections, use one of the AG replicas as the synchronization host and configure the snapshots as copy-only backups.

Cloning SQL Server Databases

SQL Server database clones can serve many purposes in the enterprise. Clones can be used for development and testing, report offloading, DBCC offloading, or database restores. The HPE Nimble Storage zero-copy cloning technology allows for near-instant presentation of clones to hosts, with minimal footprint, because space is allocated only for changed data.

BEST PRACTICE Use HPE Nimble Storage zero-copy clones as a simple, reliable, and efficient way to create clones of a SQL Server database.

Clone a Database

To clone a SQL Server database, use an existing snapshot collection for the specific point in time required or create a new snapshot collection. The procedure assumes that snapshot collection schedules have already been established, and it uses an existing snapshot collection for cloning.

Procedure

- 1 Log in to the HPE Nimble Storage array GUI.
- 2 Choose **Manage > Protection > Volume Collections**.
- 3 Select the volume collection that houses the files for the database.
- 4 Click the **Snapshots** tab and select the snapshot collection that will be the source for the clone.
- 5 Click **Clone** and specify names for the cloned volumes. Click **OK**.

Cloned volumes inherit the access control list (ACL) from their parent. To mount the clones to another host, edit the clones to clear the inherited ACL and assign a new one.

- 6 On the host, for iSCSI arrays, use NCM to connect the volume to the host.
FC arrays do not require an explicit connection to volumes. The volumes should appear automatically; if they do not, use Disk Management to rescan the disks.
- 7 Open PowerShell and use the HPE Nimble Storage PowerShell cmdlet to clear the flags and attributes that were set on the volumes during the VSS snapshot process.

Example using a disk ID:

```
PS C:\Users\Administrator> Set-NimVolume -DiskID 3 -Online $true -ReadOnly $false -Hidden $false -ShadowCopy $false
```

After the cmdlet is executed, the flags and attributes are reset on the volume, and the volume becomes visible to the host.

- 8 Mount the databases from the volumes to SQL Server. If the database is being attached to the original SQL Server, mount the database with a different name.

Restoring SQL Server Database Volumes

If necessary, the volumes that house SQL Server databases can be restored to a specific point in time from available snapshot collections. Restores of HPE Nimble Storage volumes make sense for large databases that have dedicated volumes.

When several databases are colocated on the same volume and the need arises to restore a single database, use a clone or a native SQL Server restore for the recovery process so as not to affect the other databases on the volume. Zero-copy clones record only changed blocks. Although the volumes appear as full copies to the Windows host, in reality no space is allocated to clones, but only to newly written blocks.

BEST PRACTICE To restore databases that are housed on dedicated volumes, use available snapshot collections. To restore a database that is colocated with other databases on the same volume, use an HPE Nimble Storage clone or a native SQL Server restore for the recovery process.

Restore Database Volumes

To perform a full restore of volumes that house SQL Server data files, use an existing snapshot collection for the specific point in time required. The procedure assumes that snapshot collection schedules have already been established, and it uses an existing snapshot collection for recovery.

Procedure

- 1 Disconnect the database that is housed on the volumes that will be restored from SQL Server.

As an alternative, stop the SQL Server instance.

- 2 Log in to the HPE Nimble Storage array GUI.
- 3 Choose **Manage > Protection > Volume Collections**.
- 4 Select the volume collection that houses the files for the database.
- 5 Click the **Snapshots** tab and select the snapshot collection that will be the source for the restore.
- 6 Click **Restore**. When prompted to take the volumes offline, agree.
- 7 Bring the volumes back online.
- 8 On the host, for iSCSI arrays, use NCM to refresh volume connections and reconnect the volumes to the host.

FC arrays do not require an explicit connection to volumes. The volumes should appear automatically; if they do not, use Disk Management to rescan the disks.

- 9 Open PowerShell and use the HPE Nimble Storage PowerShell cmdlet to clear the flags and attributes that were set on the volume during the VSS snapshot process.

Example using a disk ID:

```
PS C:\Users\Administrator> Set-NimVolume -DiskID 3 -Online $true -ReadOnly $false -Hidden $false -ShadowCopy $false
```

- 10 Verify that the volumes are online and that SQL Server data is present.

- 11 Take whichever action is appropriate:

- If you stopped the SQL Server instance but did not disconnect the databases, restart the instance. The databases should all be available and online. Otherwise, start the SQL Server instance and attach the database.
- If you did not stop the SQL Server instance, mount the databases from the volumes to SQL Server.

Considerations for Virtualizing SQL Server

Both Microsoft and VMware have published extensive guidance on best practices for deploying SQL Server in virtual environments. For detailed information, see the following guides:

- [Best Practices for Virtualizing and Managing SQL Server](#)
- [Architecting Microsoft SQL Server on VMware vSphere](#)

The allocation of resources to virtual machines and the choice of storage and network adapters are important considerations for virtualizing a SQL Server environment. In addition, different requirements and recommendations apply to the Microsoft and VMware virtualization platforms.

CPU and Memory Sizing

Carefully consider the workloads that will be virtualized, and allocate only the amount of vCPU and vRAM that is required. Avoid overallocating vCPU and vRAM to guests that do not require the resources for performance because doing so affects the virtual environment as a whole.

For smaller SQL Server virtual machines in a VMware environment, allocate vCPUs in a number that is equal to or smaller than the number of cores in each physical NUMA node. Consider using memory reservations to avoid the possibility of ballooning or swapping.

Enable CPU and memory hot-add for virtual machines in a VMware environment for ease of allocation if the virtual machine should require more resources.

Virtual Disks

Use the VMware Paravirtualized SCSI (PVSCSI) controller for virtual disks that contain data, log files, or the tempdb database. The PVSCSI is well suited for I/O-intensive applications, and it has queue depths that are better suited for SQL Server workloads. Leverage more than one PVSCSI device (up to four), and spread virtual disks across the devices to reduce the chance of bottlenecks at the vSCSI layer. If necessary, increase the VMware PVSCSI controller per-device and per-controller queue depths.

Microsoft recommends using fixed VHDX or virtual SCSI adapters for virtual disk connectivity.

Networking

Use the VMware VMXNET3 network adapter on virtual machines. This guideline is especially important if in-guest iSCSI is leveraged to connect to volumes on HPE Nimble Storage arrays. The VMXNET3 is designed for performance, supporting features such as multi-queue support and receive-side scaling.

Datastore and Guest Storage Considerations for VMware vSphere

VMware vSphere can be used to deploy SQL Server in several ways:

- Deploy the Windows OS on a datastore, while the SQL Server data disks are on separate VMFS datastores.
- Deploy the Windows OS on a datastore, while the SQL Server data disks are pRDM or vRDM devices.
- Deploy the Windows OS on a datastore, while the SQL Server data disks are volumes directly attached through in-guest iSCSI.
- Deploy the Windows OS on a datastore colocated with SQL Server data disks.
- Deploy SQL Server by using VMware VVols.

These are all valid configurations, but keep the following key concepts in mind:

- For snapshots, vCenter is the synchronization host unless the virtual machine is using RDMs, in which case the virtual machine acts as the synchronization host.

- Use VMware performance policies for VMFS datastores to achieve operational efficiency and better interoperability with the VMware architecture (vMotion, snapshots, DRS, and so on). Do not use SQL Server performance policies on VMFS datastores unless the datastore is dedicated to SQL Server.
- For RDM and in-guest iSCSI volumes, use SQL Server performance policies.
- Application-consistent snapshots for hosts that use vCenter as the synchronization host require a VMware Tools installation.
- Install NWT on all virtual machines that use in-guest iSCSI or RDMs so that the Windows guest can properly create application-consistent snapshots of SQL Server volumes.

Guest Storage Considerations for Hyper-V

Microsoft recommends using in-guest connectivity to deploy SQL Server in a Microsoft Hyper-V environment. Three deployment options are available:

- Guest iSCSI
- Hyper-V virtual Fibre Channel (vFC)
- SQL Server over SMB

For the guest iSCSI and Hyper-V vFC options, the recommendations are as discussed in this guide because these types of virtual deployment closely mimic a physical deployment.

For the SQL Server-over-SMB option, Microsoft recommends using Scale-Out File Server (SOFS). In this scenario, the database uses SMB3 to connect to the SOFS, and network adapters with RDMA capability, using SMB Direct, are required. For more information about using SOFS with HPE Nimble Storage arrays, see [Microsoft Scale-Out File Services on Windows Server 2012 R2 Deployment Considerations](#).

About the Author

Dianne Gonzalez



Senior Technical Marketing Engineer
Hewlett Packard Enterprise

Dianne Gonzalez is a Senior Technical Marketing Engineer with the HPE Nimble Storage Technical Marketing team. She has over 16 years of experience in infrastructure administration, data center design, and application delivery. Her career spans several industries, such as government, education, healthcare, medical research, and technical sales. At HPE, Dianne is focused on developing technical solutions for Microsoft applications, including cloud and container design and implementation.

Version History

Version	Release Date	Description
1.0	May 2017	Initial release