Hewlett Packard Enterprise

Technical white paper

Check if the document is available in the language of your choice.

BEST PRACTICES FOR ARCHITECTING MICROSOFT SQL SERVER ON HPE STORAGE

CONTENTS

Executive summary ... 3
HPE and Microsoft partnership ... 3
Target audience ... 3
Introducing Microsoft SQL Server with OLTP workloads ... 3
SQL Server instances and databases ... 3
SQL Server database file and log file ... 4
SQL Server checkpoints ... 5
SQL Server I/O characteristics ... 6
SQL Server data compression ... 8
Virtualized SQL Server ... 8
General best practices for deploying SQL Server on HPE Storage arrays ... 9
Choose the right storage subsystem for SQL Server workload ... 9
Keep HPE Storage firmware and software up to date ... 9
Considerations for virtualized SQL Server ... 12
Enable instant file initialization ... 13
SQL Server high availability and copy data management ... 13
Space-saving data compaction features and encryption ... 15
HPE DPP II: SQL Server workload and performance analysis tool ... 16
Integrated data protection ... 17
HPE Primera and HPE 3PAR product families ... 17
HPE Nimble Storage product family ... 19
Best practices for connecting HPE Nimble Storage arrays to SQL Server ... 19
HPE Nimble Storage dHCI ... 20
Best practices for connecting HPE Nimble Storage dHCI to SQL Server ... 20
HPE MSA product family ... 22
HPE Apollo 4200 product family and Azure Stack HCI ... 24
Best practices for connecting HPE Apollo 4200 Gen10 and ASHCI to SQL Server ... 24
Summary ... 24

EXECUTIVE SUMMARY

Hewlett Packard Enterprise (HPE) is a global, edge-to-cloud Platform-as-a-Service company; HPE Storage, one of the global business units of HPE, has a portfolio of intelligent storage built for the hybrid cloud world to help customers unlock the full potential of their data.

Microsoft enables digital transformation for the era of an intelligent cloud and an intelligent edge. The latest version of Microsoft SQL Server 2019 challenges the status quo of relational databases by making it possible to ingest, manage, store, and analyze any type of data from anywhere.

The focus of this paper is the components of SQL Server related to data storage. This paper helps enable HPE partners and customers familiar with HPE Storage or SQL Server to understand the fundamentals of SQL Server dealing with data storage and I/O, especially the I/O characteristics that impact the storage subsystem. This paper discusses the best practices for designing, implementing, and optimizing a customer's SQL Server environment with HPE Storage products, and includes the following concepts:

- SQL Server represents more than just a relational database management system. It provides built-in reporting services, analysis services, integration services, machine learning services, data virtualization (PolyBase), and others. This paper focuses primarily on SQL Server under online transaction processing (OLTP) workloads plus limited data analytic scenarios (such as the SQL Server columnstore index).
- SQL Server can be deployed in various forms, including as an application running in Microsoft Windows and Linux® operation systems, on physical servers or virtual machines (VMs), on-premises or in an Azure cloud. SQL Server 2017/2019 added support for container platforms. This paper's scope is SQL Server running on either a physical or a virtualized Windows Server operation system.

This paper explains the main OLTP workload profile with SQL Server and summarizes the best practices for integrating HPE Storage products with SQL Server. It does not replace the installation/deployment guides for related Microsoft products or HPE Storage products.

HPE and Microsoft partnership

HPE and Microsoft have more than 30 years of partnership across product and market development. The companies continue their ongoing collaboration across multiple businesses, with more than 29,000 joint reseller partners and hundreds of thousands of shared customers around the globe.

In addition, this partnership reaches 135 countries with a service desk that provides support in 43 languages. In 2018, Microsoft named HPE Global Technology Alliance Partner of the Year.

Target audience

The intended audience for this white paper includes solution architects from HPE and the partner community working on SQL Server and HPE Storage products. This paper also targets SQL Server database administrators, infrastructure administrators (server admins and storage admins), and others interested in best practices when architecting, designing, and implementing SQL Server on HPE arrays.

INTRODUCING MICROSOFT SQL SERVER WITH OLTP WORKLOADS

Instances and databases are part of the SQL Server architecture, regardless of the type of workload—OLTP, OLAP, or reporting—that SQL Server is performing. The I/O characteristics of the workload that SQL Server is carrying are different. The OLTP type of workload is most relevant to HPE Storage.

SQL Server instances and databases

The differences between an instance and a database must be clearly stated.[1] Some of the features and settings such as buffer pool (basically another name for RAM in SQL Server) apply to an instance, and others such as data and log files are associated with each database.

- Instance
  - An instance of the database engine is a copy of the sqlservr.exe executable that runs as an operating system service. Each instance manages several system databases including master, model, msdb, and TempDB, and one or more user databases.
  - An instance of the database engine operates as a service that handles all application requests to work with the data in any of the databases managed by that instance. Each instance manages its own resources like RAM-based buffer pool.
  - Multiple instances of the database engine can be running on single operation system. One instance can be the default instance. These instances could be of different versions or same versions, for example, one instance of SQL Server 2012 plus two instances of SQL Server 2016 running on same Windows Server 2019. Usually the database/system admin would set a limit to each instance in such situations. For example, in a x86 server with 192 GB RAM, 100 GB could be set as the maximum RAM instance-A can allocate and 50 GB for instance-B at most.

[1] The text in this section is excepted from the Microsoft document Database Engine Instances (SQL Server).

- Database
  - A database in SQL Server is made up of a collection of tables that stores a specific set of structured data. A table contains a collection of rows, also referred to as records, and columns, also referred to as attributes. Each column in the table is designed to store a certain type of information, for example, dates, names, dollar amounts, and numbers.
  - Within a database, there are one or many object ownership groups called "schemas." Schema defines the data format of a database. Within each schema there are database objects such as tables, views, and stored procedures. Some objects such as certificates and asymmetric keys are contained within the database but are not contained within a schema.
  - SQL Server databases are stored in the file system in files.
  - Files can be grouped into filegroups.

NOTE
For more information about SQL Server instances and databases, refer to the Microsoft documents Database Engine Instances and Databases.

SQL Server database file and log file

Each SQL Server database contains at least two files: a data file (*.mdf) and a log file (*.ldf). Some large databases may contain one or many additional secondary data files (*.ndf) but only one primary data file (usually *.mdf) is allowed per database. Primary database files and secondary database files contain customer data. SQL Server stripes data blocks (called "pages") to all database files for better performance (similar to a round-robin method). In addition to a customer's data blocks, the primary database file also contains database configuration information.

The log file is irreplaceable in SQL Server and other relational database products. The log file holds critical information to guarantee the data consistency of the database in order to recover the database in the event of an incident like a system crash or power outage. Log file performance also impacts and limits the number of transactions a database can handle concurrently.

Unlike data files, which are usually continuously growing in size, log files generally are smaller because periodically a new log file is created and the old log file is moved to a safe place for backup or archive. Because log files capture all changes made to a database, even if a disastrous event occurs such as the loss of all data files, as long as there is a previous database backup available plus all log files since that previous backup, DB administrators can restore the entire database to most its current status.

NOTE
mdf/ndf/ldf are default and recommended filenames for SQL Server files. However, the use of other filenames and extensions for SQL Server such as "customer.db1" is technically possible. In addition to *.mdf/*.ndf/*.ldf files, another type of file used in SQL Server called "filestream" is used to store unstructured data.

Database data file and log file I/O profiles

In addition to capacity, the I/O profiles for database data files and log files are also dramatically different.

In a SQL Server environment with an OLTP workload, each data file is composed of pages; each page is exactly 8K bytes. Because the SQL Server database engine only reads and writes data in server memory (RAM), a buffer manager module within SQL Server moves data between disks and RAM before the database engine performs execute create, read, update, or delete (CRUD) operations to a page.

In order to minimize the number of physical disk read I/O and improve disk transfer efficiency, SQL Server usually performs a prefetch operation (also called "read ahead"), which means the pages are loaded into the buffer pool based on data structure and past record. For pages that are required but are not in the buffer pool already, SQL Server waits until those pages are copied from disk to buffer pool (RAM) before the CPU processes them. The speed of disk I/O is usually 1000 times slower than DRAM (most disk I/O access latency is longer than 100 microseconds but the latency of DRAM is less than 100 nanoseconds), so this is a very expensive wait.

During prefetch, SQL Server reads not only a single 8 KB page in one I/O request but reads continuous pages to improve I/O efficiency. Read I/O block sizes can range from 8 KB to 512 KB or even larger in a SQL Server OLTP environment. In a good implementation of SQL Server, a 95% or more buffer cache hit ratio can be observed. SQL Server was designed to leverage all available system memory as a buffer pool to improve the hit ratio, so HPE usually does not recommend installing other applications in same host as SQL Server because they compete for system resources, particularly DRAM. In a configuration with multiple SQL Server instances on the same operating system, there will be one buffer pool per instance. All instances compete for DRAM, so some manual settings of min/max memory usage per instance are necessary.

After a page has been updated in RAM (called "dirty pages"), SQL Server needs writes it back to disk for data persistence. This usually happens during a checkpoint, which is a SQL Server feature that creates a clean point-of-time image on disk. Similar to a prefetch operation, if there are adjacent pages to be updated, SQL Server tries to write multiple pages in one I/O operation to disk. For example, if eight continuous pages need to be updated, a single 64 KB block write containing all eight pages is executed, which is more efficient compared to writing to disk eight times with a single 8 KB page in each I/O operation. As a result, the less often SQL Server writes pages to disk, the better chance it can combine multiple small block write I/O into one large block write I/O. This is referred to as "lazy write" behavior in SQL Server.

The data transfer of data files is always aligned to one or many 8 KB pages, for example, 8 KB, 64 KB, and 512 KB. Based on the nature of OLTP, there are always more read activities than writes. For example, a bank system might need to read an account's balance before proceeding with a withdrawal or transfer. It is not surprising that in an OLTP SQL Server environment, the data file access pattern is dominated by read I/O with block size ranging from 8 KB to 512 KB or even larger.

The database log file, on the other hand, contains no page concept and the access pattern is dramatically different from data files. There is only one active log file per SQL Server database, which records all database modifications in strict order. This is a common requirement to meet the atomicity, consistency, isolation, and durability (ACID) properties of a relational database. In theory, this serialized recording of all changes enables a database to be recovered in the event of a disaster and rollback to any point-in-time.

Similar to the buffer pool for data files, there is a buffer area in DRAM for log files, but at 60 KB, it is fairly small. Every time an application asks for a commit of a transaction in SQL Server, for example to finish a bank transfer between two accounts, the database needs to record this modification in a persistent place that is not vulnerable to events like a power failure or a software or operating system crash. Because it would take too long to write one or multiple 8 KB pages to the disk, SQL Server updates the related pages in DRAM, records the changes in log files, and flushes the entire log file buffer pool to disk.

SQL Server checkpoints

SQL Server uses the lazy write design to avoid frequent writes to disk. However, it cannot keep all dirty pages in memory forever. Even though the log file keeps the database consistent, too many dirty pages means the database recovery time after a system crash or power outage would be very long. Checkpoints were designed to solve this problem. During the checkpoint, all dirty pages in the buffer pool for data files, as well as the log file buffer, are flushed to disk. The key facts are:

- SQL Server generates a checkpoint every minute by default unless it finds that the write I/O latency is more than 50 milliseconds. This means the storage is too busy or too slow. In this case, SQL Server will delay the next checkpoint accordingly.
- Checkpoints put I/O stress on the storage subsystem with a mix of large and small block write I/O, mostly random. Storage devices with a large persistent write cache and faster media can cope with the I/O stress well.
- A checkpoint can be manually triggered by the T-SQL command `checkpoint`. Many of the backup solutions run a checkpoint to get a clean database image on disk, which makes recovery smooth.

NOTE
Some snapshots are taken from inside the storage array without triggering a SQL Server checkpoint. These copy images are usually named “crash-consistent” rather than “application-consistent,” which means an application-aware preparation was conducted before the snapshot or copy was created. Crash-consistent copy images can be used for SQL Server recovery if they contain legitimate log or data files. If there are updates not reflected in the data files (because the latest checkpoint to flush dirty pages was not executed) but the updates are contained in log files, SQL Server would take the time to clean up the data files before they can be recovered to online status.

SQL Server I/O characteristics

For SQL Server under an OLTP workload, the I/O behavior is largely shaped by lazy write, prefetch, transaction commits, and checkpoints.

Server RAM capacity is growing but faster storage media with lower I/O latency is still critical. Due to the decreasing cost of DRAM, the capacity of DRAM per SQL Server has been growing much faster than the growth of data size in OLTP SQL Server. It is not uncommon to have a SQL Server configured with hundreds of GBs of DRAM hosting an OLTP SQL Server with several TBs of data, or, for example, RAM capacity approaching 20% of data size. The majority of DRAM is used as a buffer pool to minimize random read I/O from disk. SQL Server tends to generate more large block sequential read I/O to prefetch data into a DRAM buffer pool. The buffer cache hit ratio (BCHR) often reaches 95% or even 99%. However, less than 5% of disk I/O is generated by random reads. Because the speed of disk I/O is more than 1000 times slower than DRAM (most disk I/O is faster than 100 microseconds, DRAM is slower that 100 nanoseconds), the improvement of disk I/O is still apparent in average I/O response time.

The importance of disk I/O latency regarding log files must be stressed because there is only one active log file per DB with single thread access because of the ACID requirement. More DRAM in the server would not be helpful because the log buffer pool is fairly small (60 KB per DB) and it must be flushed to disk for every transaction commit. The key to log file performance is either faster media for a direct-attached storage environment or a large amount of persistent write cache (usually battery-backed RAM with mirror protection) to stage write I/O. Merging small block log writes into large block I/O writes to back-end storage media in an external storage array also aids performance.

This 60 KB log buffer area in RAM would be filled quickly because it captures all changes to a database. As a result, the write I/O sizes of log files are dominated by sequential writes with block sizes from 512 bytes to 60 KB. Although there is more read I/O than write I/O to log files during a normal OLTP workload, a delay of write I/O to the log file will delay completion of the transaction.

For a single SQL Server instance running on an HPE server, SQL Server 2016 introduced a solution to put the tail of the log file (the last 20 MB of a log file) on NVDIMM-N in HPE servers. This solution achieves up to four times more transactions per second because NVDIMM-N provides write performance similar to RAM but also contains persistent media to retain the data during a power outage. However, for a highly available design that requires the data to reside on and be synchronized between more than one server, this advantage fades quickly because latency is introduced when data travels across the network.

A widely implemented practice to minimize write I/O latency in log files is to use separate devices for log files and data files. For storage devices directly attached to servers, this segregation usually means different physical devices. For example, two devices can be mirror-protected for log files and another four devices can be protected with RAID 5 for data files. However, each storage media can only provide a certain amount of I/O at low latency and as more I/O lands, the latency eventually increases. More dedicated devices may be either oversized when the system is free or become a bottleneck in the event of a large amount of burst I/O.

Wide-striping on multiple storage devices is a common technique to provide more concurrent I/O with low latency without hotspots on particular devices. However, because the I/O pattern of SQL Server data files and log files are different, a simple solution like disk wide-striping cannot meet the SQL Server I/O requirement. Intelligent persistent cache in an HPE Storage array absorbs the small block size of write I/O with lowest latency on SQL Server log files before coalescing them into large block I/O and flushing them to a wide-striped back-end disk. HPE Storage arrays predict the data access pattern in order to prefetch a large amount of data from disk.

Because business environments are usually dynamic, often other workloads like reporting services, extract, transform, load (ETL) processes, OLAP, data warehouse, and data backup and data restore can potentially be running on the same SQL database at different times of day. Those workloads bring an even larger amount of I/O to the data files and usually stress not only the storage media (HDD or SSD) but all the components along the data paths between the storage and the CPU and DRAM.

TABLE 1. An illustration of typical I/O profiles for various SQL Server I/O files

	Typical OLTP	Typical OLAP	HPE recommendations
SQL data files (mdf/ndf)	Majority is random reads with small blocks sizing from 8 KB to 256 KB, mixed with some sequential read and write I/O	Majority of I/O is large block random/sequential read I/O, for example, 256 KB or more	Low read I/O latency, high concurrency, high availability
SQL log files (ldf)	Majority is sequential writes. Small block sizes up to 60 KB. Sequential reads in some scenarios as well (such as backup). Fewer writes in OLAP workload.		Low write I/O latency, high availability

NOTE
As technology has evolved, scenarios have developed where SQL Server does not adopt this 8 KB page data structure and the I/O characteristic could be quite different, such as SQL Server In-Memory OLTP.

The HPE Nimble InfoSight team conducted research on I/O block sizes of different applications in 2016[2]. Some of the results of this research related to SQL Server I/O block size information include the following points:

- A month's worth of user I/O from more than 7,500 HPE Nimble Storage customers was sorted and counted by block size.
- SQL Server performs more than 50% of its operations in either the 8 KB or 64 KB bins showing strong bimodality.
- SQL Server deployments cluster around 8 KB/64 KB.
- SQL Server seems to frequently exhibit transaction-optimized write activity.

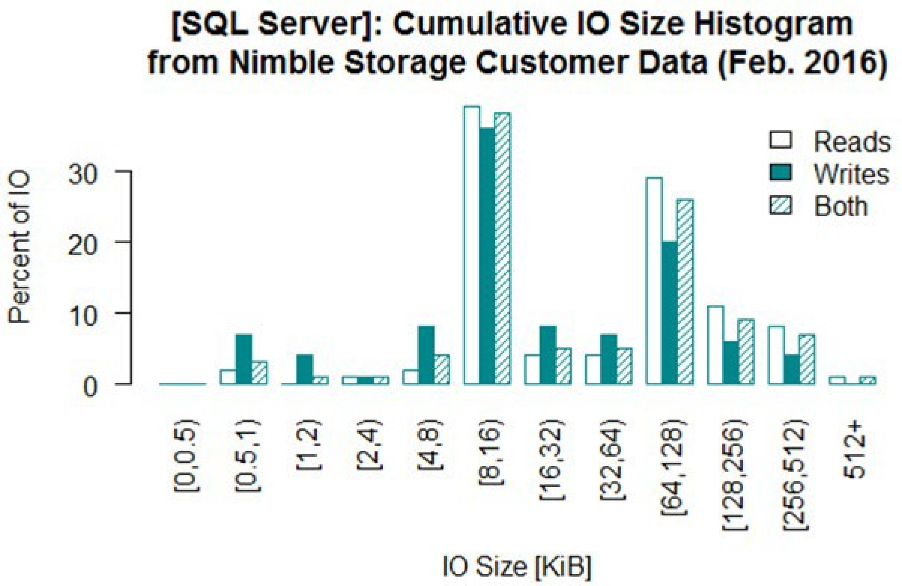


FIGURE 1. Cumulative I/O size histograms for SQL Server databases

SQL Server high availability and disaster recovery

There are two approaches for SQL Server customers to build a high availability/disaster recovery solution:

- **Always-On Failover Cluster Instance (FCI)**—With FCI, a SQL Server instance with related resources is registered in a Windows Server Failover Cluster (WSFC) as a cluster-wide role. FCI is implemented at the instance level with one active instance and one standby instance. A key requirement for FCI is shared storage between multiple SQL Servers, which is usually an external storage array such as an HPE Nimble Storage or HPE Primera array. Software-defined storage (SDS) such as Azure Stack HCI on HPE Apollo 4200 Gen10 with internal drives is also fully supported. WSFC features such as cluster manager and quorum are leveraged for the instance role to fail over and fail back. With the data replication capabilities in HPE arrays, FCI can be extended to be not only a high availability solution but also a disaster recovery solution. FCI is included in both SQL Server Standard and Enterprise editions.

[2] https://regmedia.co.uk/2016/08/31/nimble-labs-research-report_mapping-real-world-apps.pdf

- **Always-On availability group (AG)**—AG is a SQL Server database-level high availability/disaster recovery solution that has been available since SQL Server 2012. On Windows platforms, AG also leverages WSFC features such as Failover Cluster Manager and quorum service for high availability. SQL Server also includes a technology to replicate databases between a primary replica and secondary replicas over an IP network, either in synchronous mode or in asynchronous mode. As a result, there is no requirement for shared storage between different servers. This makes AG more flexible for deployment in heterogenous and nonconventional environments such as containers and hybrid clouds. Although a basic version of AG is included in SQL Server Standard edition, SQL Server Enterprise edition is necessary to take advantage of most AG features.

SQL Server FCI and AG are not mutually exclusive. They are commonly implemented at the same time to protect mission-critical SQL Server databases. A common combination is to leverage FCI for high availability/disaster recovery in same data center or across two adjacent data centers and to use AG to replicate SQL Server to a disaster recovery/dev/testing environment.

SQL Server data compression

As of SQL Server 2019, several data compression options are available:

- **Row and page compression**—This option is designed for OLTP databases. It includes several technologies to minimize the data footprint. Row compression has a minimal impact on CPU/DRAM by converting most fixed-length data such as bigint, which usually allocate 8 bytes, to variable length it actually occupies, which could be 2 bytes. Page compression saves more space by enabling prefix and dictionary compression in addition to row compression, but more CPU resources are required.

  There are some limitations on SQL Server compressions because some SQL Server features are not compatible with SQL Server compression. SQL Server row and page compression have no impact on log files.
- **Columnstore and columnstore archival compression**—All columnstore tables and columnstore indexes are always stored with columnstore compression by default. Optionally, columnstore archival compression could be activated to provide even more space savings.
- **Backup compression**—This option is applicable only for SQL Server backups. After this option is enabled, SQL Server data files are read from disk and compressed before they are written to a backup device to reduce the backup file footprint. For databases that already have page compression enabled, little space would be saved from backup compression.

Virtualized SQL Server

Although the deployment of SQL Server on physical servers is still common in mission-critical environments (Tier 0) to realize benefits such as fault isolation and resource guarantees, an increasing number of SQL Server instances are being deployed inside VMs as part of data center modernization. Virtualized SQL Server became attractive for business-critical environments (Tier 1) because it provides the following benefits:

- Improved resource utilization and reduced cost of infrastructure
- Dramatically improved hardware resource (CPU/memory/network) utilization
- Increased rack space density and power and cooling consumption reduced to a fraction of previous usage
- Lower management costs
- Fewer physical servers to manage, patch, upgrade, and maintain
- Completely online, easier migration between physical servers with a few mouse clicks
- Reduced cost of software licenses because many software licenses are based on the number of physical CPUs on a physical server regardless of the number of VMs sharing them
- Flexible scalability into large data centers

Not only can VMs be easily provisioned, migrated, and relocated based on policy, but also the physical server de-coupled from VMs can be maintained, upgraded, retired, or repurposed without impacting the applications running in the VMs.

NOTE

For more details, read the HPE white papers Deploy Microsoft SQL Server 2019 Big Data Clusters on Kubernetes and HPE Nimble Storage and HPE Reference Architecture for delivering insight across all your data with Microsoft SQL Server 2019 Big Data Clusters.

GENERAL BEST PRACTICES FOR DEPLOYING SQL SERVER ON HPE STORAGE ARRAYS

There are a few general best practices to bear in mind when working with SQL Server on HPE Storage products.

NOTE
Because this paper focuses primarily on the design and implementation of SQL Server and HPE Storage solutions, best practices for servers and networks are not covered. Many reference architectures, best practices, and white papers are available from the Hewlett Packard Enterprise Information Library.

Choose the right storage subsystem for SQL Server workload

SQL Server with an OLTP workload relies on disk I/O to keep transactions consistent. SQL Server also leverages prefetch and lazy write operations to move data between the buffer pool (in DRAM) and disks to reduce I/O. However, storage subsystem performance is one of the key factors impacting SQL Server and the entire OLTP performance. In addition to performance metrics, consider other factors including reliability, manageability, upgradability, and serviceability when choosing a storage subsystem for a SQL Server workload.

Faster storage media is always welcome for SQL Server OLTP workloads. Flash media like SSD is a vast improvement over traditional magnetic HDDs and NVMe-based flash devices. Flash media provides even better latency and throughput compared to SAS-based flash devices. Other technologies such as PCIe-based Storage Class Memory (SCM) can drive storage I/O latency even lower.

Of all the brands and models of flash media on the market, only a handful of vendors manufacture NAND modules and a few other brands are available for SSD disks. Some SSD disks can be installed inside an x86 server (called “direct-attached storage”) or in an external storage array. A theoretical comparison of the I/O latency on these disks shows that direct-attached disks might seem to have advantages because I/O does not need to travel outside of the server through the SAN, and then through another storage controller similar to another x86 server, and back. However, the value of the storage array is in the purpose-built software that can leverage RAM-based cache to stage write I/O, prefetch read I/O, optimize data transfers for back-end disks, wide-stripe data, and many other advanced functions. Consequently, you should not evaluate a storage subsystem solely by storage media type.

A widely adopted operation model of some SQL Servers carries out OLTP workloads during the day while other servers carry out ETL/DSS/backup operations during off hours. A centralized storage subsystem can allocate the necessary resources for all servers and all workloads. In the future, a centralized storage subsystem could be upgraded easily with more flash models to improve back-end data transfer rates. The HPE Timeless Storage program is designed to provide customers a nondisruptive controller refresh every three years with a seamless upgrade to new platforms.

NOTE
For more details, visit the HPE Timeless Storage webpage

With these approaches, a centralized storage system is easier to scale up and serve host I/O with consistent low latency as the data size and transaction numbers grow.

Prepare Windows Server with multipath path management

Most HPE Storage products are connected to Windows Server over Fibre Channel or iSCSI protocols. Usually there is more than one path available between the server and the array. It is critical to ensure that adequate multipath policies are in place to provide availability with accurate path failover/failback settings, as well as data traffic optimization for lower I/O latency and higher data throughput.

Some general modules and patches are required or recommended before connecting an HPE array to a Windows server. HPE advises visiting the Hewlett Packard Enterprise Information Library and following the appropriate Windows Server deployment/implementation guide for each HPE array.

Keep HPE Storage firmware and software up to date

Each HPE Storage product runs firmware to orchestrate the internal operation of the product, as well as other software installed outside of the array that orchestrates with the array. The HPE Nimble Storage Windows Toolkit is an example; it is installed on Windows Servers connected to an HPE Nimble Storage array.

HPE highly recommends maintaining all firmware and software at the HPE recommended version. The recommended version might contain fixes to improve stability, optimized algorithms to allocate resources to boost performance, or better manageability for monitoring and tuning system performance.

Plan SQL Server database files placement

HPE strongly suggests separating SQL Server data files and log files into different logical unit numbers (LUNs)[3] in an HPE array.

The separation of SQL data files and log files into individual LUNs should not be misinterpreted as separation of these files into different physical devices (HDD/SSD). All HPE arrays have built-in technology to wide-stripe any front-end LUN to back-end physical devices to avoid potential performance bottlenecks and for easy management.

Best practice to separate these files into different front-end LUNs is for the array cache management algorithm, which is usually associated with front-end LUNs, to detect the I/O patterns of each LUN and to optimize the cache resource allocation accordingly. This practice also makes it easier for performance monitoring and tuning from the array.

Microsoft recommends putting data and log files on separate disks. Although the default option during SQL Server installation is to place the data and transaction logs on the same drive and path, this choice is not optimal for production environments.

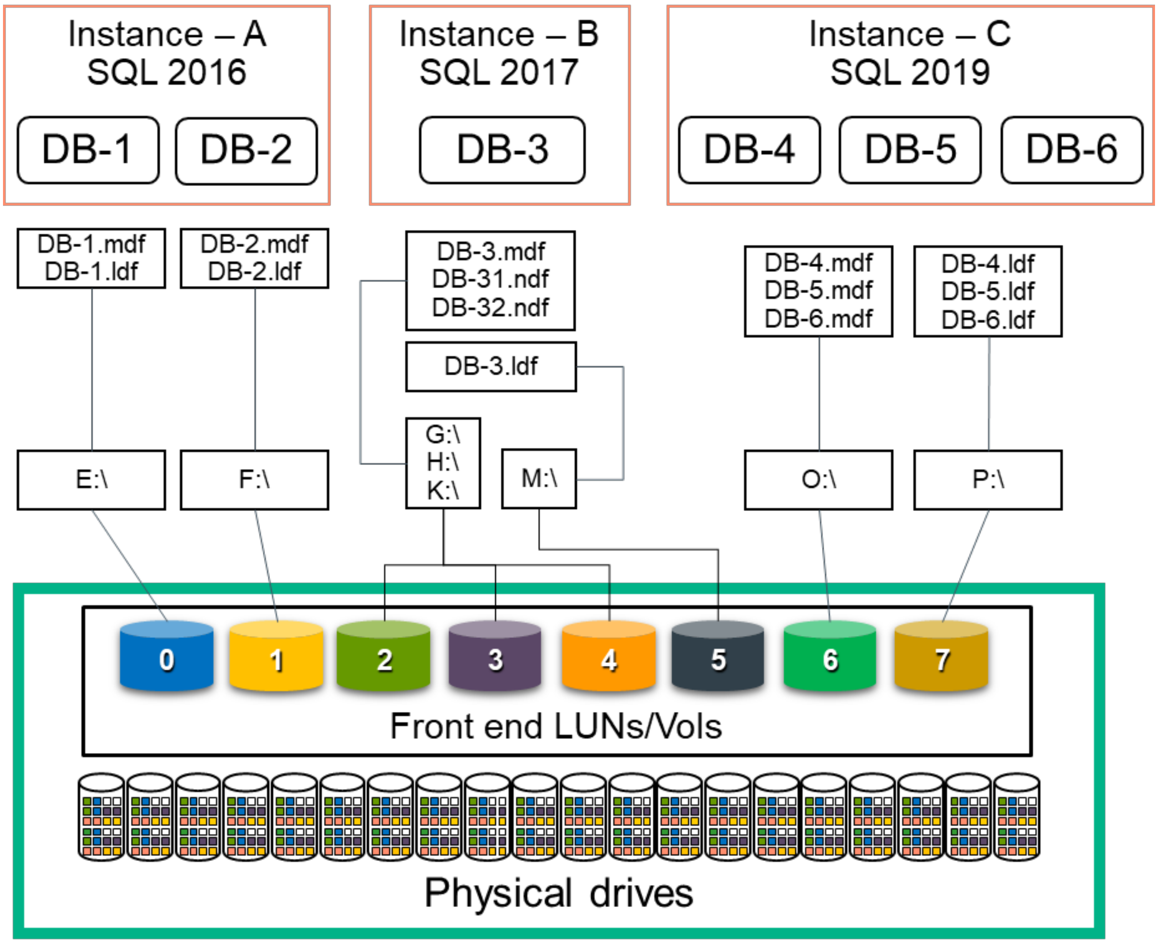


FIGURE 2. Three strategies for placing SQL Server data files in a practical environment with multiple databases on multiple instances on same server

Data file placement as DB-1 and DB-2 is generally not recommended.

Data file placement as DB-3 is recommended for SQL Server running on physical servers or in VMs for a mission-critical (Tier 0) environment. This configuration provides the best performance isolation from data files and log files. If a single *.mdf file cannot meet customer requirements and multiple *.ndf files are created, these *.ndf files can be stored on a single LUN or spread across multiple LUNs.

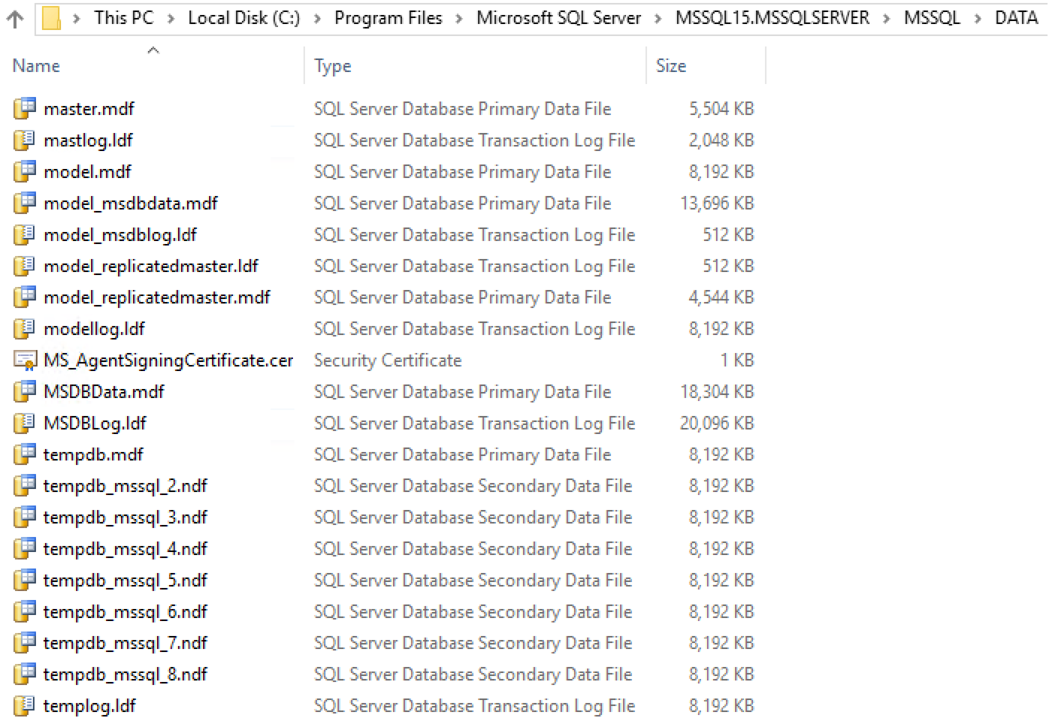
A placement policy such as DB-4, DB-5, and DB-6 is also a good practice. Compared to the design for DB-3, this design maintains the workload isolation and avoids the need to use a mount point folder rather than drive letters. However, it also means the databases sharing

[3] HPE Storage uses different names for front-end LUNs such as volumes, vols, vLUNs, and disks. This paper uses "LUN" to describe a device addressed by the SCSI or SAN protocols that encapsulate SCSI, such as Fibre Channel or iSCSI. "LUN" is most often used to refer to a logical disk created on a SAN. (https://en.wikipedia.org/wiki/Logical_unit_number)

the same LUN can compete with each other for I/O. It would be impossible to monitor and adjust the service level agreement (SLA) for each database from the external storage array. This design might be a good fit in a business-critical environment.

These best practices apply to customer database files. Although mdf/ldf files are present, they should be treated differently from customer databases.

Master, model, and msdb contain small amounts of configuration data and less I/O activity, so they could be placed in a separate directory sharing the same disk as SQL Server data files.



This PC > Local Disk (C:) > Program Files > Microsoft SQL Server > MSSQL15.MSSQLSERVER > MSSQL > DATA

Name	Type	Size
master.mdf	SQL Server Database Primary Data File	5,504 KB
mastlog.ldf	SQL Server Database Transaction Log File	2,048 KB
model.mdf	SQL Server Database Primary Data File	8,192 KB
model_msdbdata.mdf	SQL Server Database Primary Data File	13,696 KB
model_msdblog.ldf	SQL Server Database Transaction Log File	512 KB
model_replicatedmaster.ldf	SQL Server Database Transaction Log File	512 KB
model_replicatedmaster.mdf	SQL Server Database Primary Data File	4,544 KB
modellog.ldf	SQL Server Database Transaction Log File	8,192 KB
MS_AgentSigningCertificate.cer	Security Certificate	1 KB
MSDBData.mdf	SQL Server Database Primary Data File	18,304 KB
MSDBLog.ldf	SQL Server Database Transaction Log File	20,096 KB
tempdb.mdf	SQL Server Database Primary Data File	8,192 KB
tempdb_mssql_2.ndf	SQL Server Database Secondary Data File	8,192 KB
tempdb_mssql_3.ndf	SQL Server Database Secondary Data File	8,192 KB
tempdb_mssql_4.ndf	SQL Server Database Secondary Data File	8,192 KB
tempdb_mssql_5.ndf	SQL Server Database Secondary Data File	8,192 KB
tempdb_mssql_6.ndf	SQL Server Database Secondary Data File	8,192 KB
tempdb_mssql_7.ndf	SQL Server Database Secondary Data File	8,192 KB
tempdb_mssql_8.ndf	SQL Server Database Secondary Data File	8,192 KB
templog.ldf	SQL Server Database Transaction Log File	8,192 KB

FIGURE 3. Illustration of system database files with default installation options

TempDB is used by SQL Server during various operations as an intermediate stage and therefore requires high performance I/O. HPE recommends specifying the path for TempDB to a dedicated high-performing storage device, such as a dedicated LUN from the HPE array, instead of the default C drive.

NOTE
SQL Server 2019 introduced memory-optimized TempDB metadata to improve TempDB performance for some use cases.

The TempDB LUNs should be excluded from array snapshots and remote replications because the data on the LUNs can be disregarded without affecting SQL Server database consistency. TempDB might contain large data change rates, which is costly to disk array snapshots and remote replications.

NOTE
To use Always-On FCI for instance failover, all databases in the SQL Server instance, including the master, model, and msdb, must reside on shared disks such as a cluster shared volume (CSV) or a Server Message Block (SMB) disk. TempDB, on the other hand, could be hosted on a local high-performance disk drive.

HPE recommends changing the data root directory and TempDB data directories during SQL Server installation (as shown in Figure 4). Changing them after installation and relocating them to a new directory requires extra effort and an offline window.

Microsoft recommends formatting the disk drive for SQL Server with a 64 KB allocation unit size.

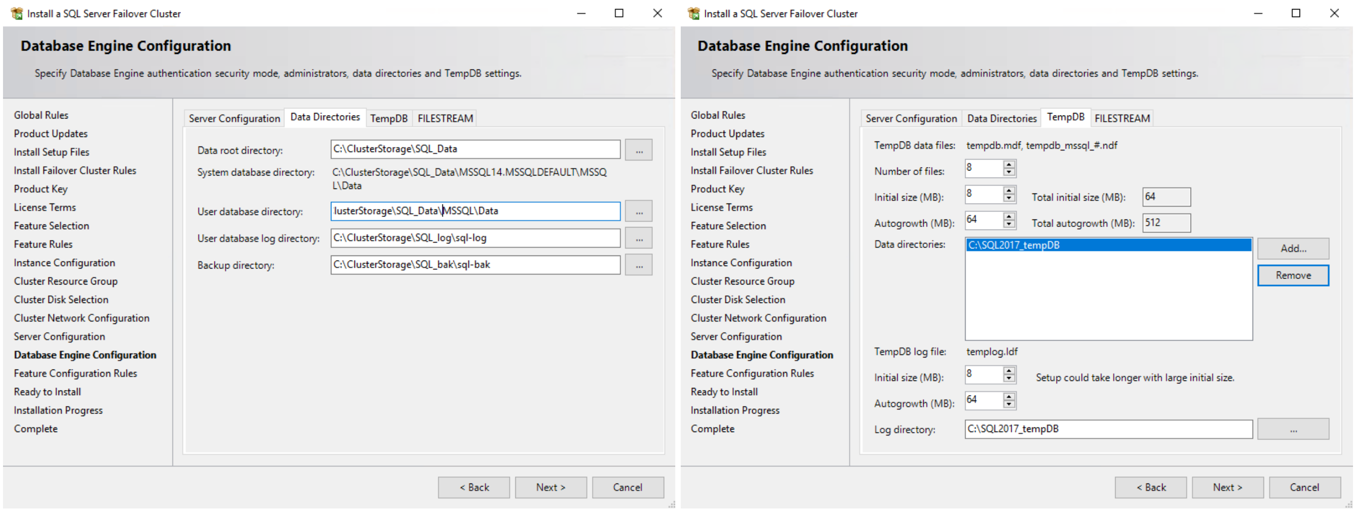


FIGURE 4. SQL Server installation wizard screen showing the data files placement options

Considerations for virtualized SQL Server

For a virtualized SQL Server running inside a VMware® VM, it is more challenging to strictly following the data files separation recommendations.

If all VMs are configured with different virtual disk formats (known as "VMDK" in VMware vSphere® and "VHDX" in Microsoft Hyper-V) for different types of files, such as operating system files, SQL data files, and SQL log files, these VMDK or VHDX files should not be placed in same datastore. If they are in the same datastore, all these files would be hosted by the same LUN, which defeats SQL Server file placement best practices. Even if these different VMDK files are placed in different datastores and carefully monitored with tools such as VMware Distributed Resource Scheduler (DRS), it is possible that all VMs sharing the same datastore could generate too much I/O at one time and impede the performance of important SQL Server VMs (noisy-neighbor problems). These issues are challenges not only for SQL Server administrators but can be problematic for any app/database administrator in a highly virtualized world.

Due to these concerns, for SQL Servers running mission-critical apps that require absolute guaranteed performance, particularly an OLTP operation such as online ticket booking, HPE recommends designing dedicated storage LUNs presented through raw device mappings (RDMs), pass-through disks, or in-guest iSCSI to avoid I/O contention. It is not uncommon for some customers to have dedicated physical servers running VMware ESXi™ or Hyper-V virtualization with only a few VMs exclusively for a suite of apps and databases. In this way, customers can maximize the benefits of virtualization such as standardized infrastructure management tools, advanced technologies including VMware vMotion®, as well as guaranteed performance and fault isolation.

For other SQL Servers running applications that are not expecting much I/O pressure, such as a website hosting an employees' wellness program that tracks steps per day through an app that uses Microsoft Internet Information Services (IIS) and SQL Server, storage administrators could choose to focus on management and resource utilization. The admins could pool SQL Server data in one datastore mixed with VMDK files from other VMs if they are hosted by a high-performing disk such as an HPE Nimble Storage All Flash array. Techniques available to fine-tune the I/O performance even in such a mixed environment include leveraging VMware Storage I/O Control (SIOC) and scheduling OLTP/ETL/OLAP/backup operations for different time periods in a day or a week.

Another highly recommended solution for virtualized SQL Server is VMware vSphere Virtual Volumes™ (vVols). Implementing vVols not only minimizes the management cost but also allows for better performance by separating SQL Server files into different storage LUNs. With vVols, SQL Server administrators and VM administrators can create multiple vdisks for each SQL Server VM (for example, the C drive for operating system files, the D drive for SQL Server data files, the E drive for SQL Server log files, and the F drive for TempDB). VMware vCenter® on the HPE array will automatically generate LUN-like disk objects inside the array for each vdisk. Separated vVols are similar to separated LUNs for each SQL Server data file, which yields best performance. At the same time, managing vVol is similar to managing a VMware environment with a VMware vSphere Virtual Machine File System (VMFS) datastore, completely managed and monitored within VMware vCenter.

Enable instant file initialization

Without instant file initialization (IFI) enabled, SQL Server performs data file initialization by sending actual data (mostly zeros) to a newly created file. For example, when a 100 GB SQL data file is created, SQL Server writes 100 GB of data (mostly zeros) to the file on disk. Although most HPE arrays can eliminate the space allocation for those all-zero blocks, it takes a long time for a server to write the data blocks from server memory through a Fibre Channel or iSCSI adapter, through the network, and all the way to the storage controller.

With IFI enabled, SQL Server data file creation is instantaneous, which saves a lot of time during creation or growth of the data files. Figure 5 shows a screenshot captured from a Windows Resource Monitor showing SQL Server actively writing data into a *.mdf file hosted by an HPE Nimble Storage array during creation of a data file without IFI. IFI has no effect on log files.

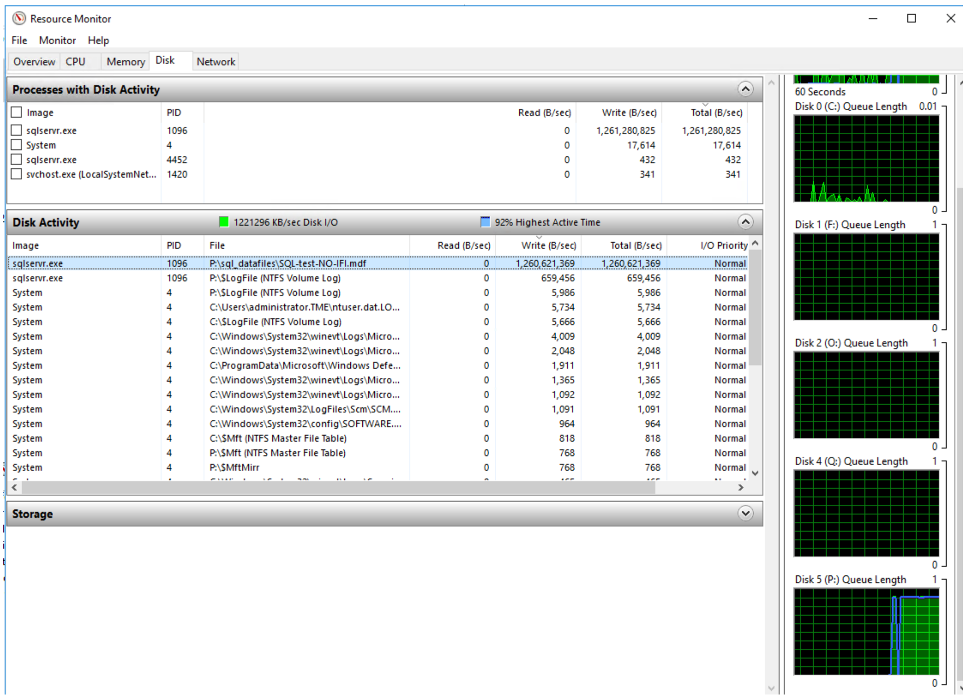


FIGURE 5. Windows Resource Monitor: SQL Server is actively writing data into a *.mdf file hosted by an HPE Nimble Storage array during creation of a data file without IFI

NOTE

For more information about IFI, refer to the Microsoft Database Instant File Initialization document.

SQL Server high availability and copy data management

For a high-availability design, HPE recommends using Always-On FCI deployment with HPE storage for a seamless automatic failover of an entire SQL Server instance including all internal databases. For a disaster recovery design, HPE recommends leveraging array-based data replication technology, if possible. This technology has been proven to be reliable and robust over decades. In scenarios such as a heterogeneous storage subsystem between different data centers and generating reports from secondary database copies, Always-On AG is a complementary technology.

Database maintenance operations require a convenient and efficient way to generate point-in-time copies for databases with minimal impact on production systems. These copies are usually called “snapshots.” They can be used for fast backup/recovery, test/development, verification, or replication to another site as a disaster recovery copy[4].

The snapshot feature is one of the most popular benefits of HPE Storage products. All HPE Storage products provide array-based snapshot and replication capabilities. SQL Server provides a snapshot feature as well, but many customers still use snapshots on HPE Storage arrays because these two technologies are not mutually exclusive. They are based on different technologies and are designed for different use cases although they share similar names. SQL Server snapshots are the better choice for DBA or app administrators who need to create a

[4] Although some products might have different names and the implemented technologies might be slightly different, ”snapshot” here refers to a point-in-time copy of data within same storage devices under names like snapshot, virtual copy, physical copy, instant clone, and volume copy. Similarly, “replication” here refers to the technology to replicate data between different storage devices under names like remote copy, remote replication, and remote snap.

logical view of the databases for read-only access. Array-based snapshots usually require collaboration between DBA and infrastructure administrators.

TABLE 2. Summary of differences between SQL Server and array-based snapshot capabilities

Features	SQL Server snapshot	Array-based snapshot
Read/write	Read only	Read and write
Export snapshot to another host/instance for read/write	No	Yes
Extra license	No	Maybe
Used for reporting purpose	Yes, easy	Yes, need extra steps
Back up from snapshot	No	Yes
Use snapshot as fast backup/restore	Partially	Yes
Data consistency	Transactionally consistent	Transactionally consistent with Microsoft Volume Shadow Copy Service (VSS) integration

Replication from HPE Storage products has played a key role in SQL Server high availability/disaster recovery solutions due to the reliability, agnostic data format, and minimal impact to the system of the replication technology. Figure 6 illustrates a typical high availability/disaster recovery design based on HPE Peer Persistence technology with data replication running between HPE Storage arrays in two data centers. All SQL Servers in Datacenter-A read and write data into Array-A. If Array-A is inaccessible, SQL Servers in Datacenter-A can immediately access Array-B in Datacenter-B. The entire failover is automatically managed by disk arrays; the Windows operating system does not sense or report any errors.

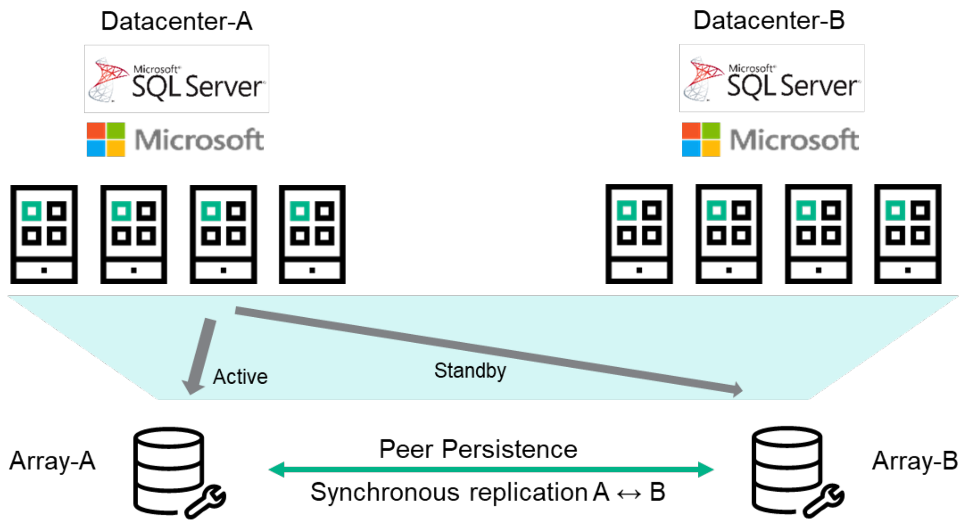


FIGURE 6. HPE Storage Peer Persistence technology for SQL Server high availability

Replication requires homogeneous operating systems and storage hardware. In scenarios where these requirements cannot be satisfied, log shipping and database mirroring could be leveraged. Since SQL Server 2012, Always-On AG has been another option to implement SQL Server high-availability/disaster recovery. Table 3 shows a comparison between array-based data replication technology and Always-On AG.

TABLE 3. Comparison between array-based data replication technology and SQL Server AG

Features	SQL Server AG	Array-based replication
Synchronous copy/auto failover/asynchronous copy	Yes	Yes
1 (source) to N (multi-target) fan-out copy	Yes	Yes
Replication link	Ethernet	Ethernet or Fibre Channel
License	Basic AG (SQL Server Standard edition) Advanced AG (SQL Server Enterprise edition)	Depends on storage array
Granularity of replication	Per database	Per storage LUN (contains one or more databases)

Optimize data traffic between SQL Servers and storage arrays

There are many choices for storage subsystem connections. Fibre Channel is the most mature and mainstream protocol and is evolving into 32/64 Gb offerings. iSCSI has been embraced by some customers. On the other hand, NVMe is emerging, with at least three different variations: NVMe over Fabrics (NVMe-oF), NVMe over FC (FC-NVMe), and NVMe over TCP (NVMe/TCP)[5]. In addition to these options, many SDS solutions leverage Ethernet/IP networks for communication between servers and nodes, including Azure Stack HCI.

No matter which protocol is used to carry data traffic between server and storage (conventional) or between servers and servers (SDS), there are a few practices in common for SQL Server OLTP workloads:

- Design a dedicated and physically separated storage network for storage traffic to provide the best I/O latency, data throughput, and system stability. This is particularly critical for protocols based on Ethernet/IP, including iSCSI, FCoE, and NVMe-oF. A dedicated storage network not only yields better security because of the physical isolation from enterprise network and better performance due to the minimization of firewalls, routings, tunnels, but also better manageability because the storage network management can be integrated inside an Infrastructure management software such as HPE OneView.
- Build a storage network with purpose-designed hardware. Compared to enterprise network products that include more features such as routing, tunneling, firewall, and multiprotocol label switching (MPLS), storage networks, including Fibre Channel and Ethernet protocols, require more focus on low latency and high throughput as the nature of the storage traffic dictates. HPE has a dedicated storage networking portfolio for handling storage traffic.
- Optimize the storage network settings according to best practices. Unique optimization techniques are available, depending on various protocol and product combinations. The deployment guides and best practice guides published in the Hewlett Packard Enterprise Information Library is a good place to start.

Space-saving data compaction features and encryption

Many HPE Storage arrays have data compaction features built-in. Data compaction includes primarily three different technologies: thin provisioning, data deduplication, and data compression. With thin provisioning, the storage array only allocates disk capacity to each LUN for any non-zero data blocks. With data deduplication, the storage array can compare data blocks from same LUN or different LUNs and replace the duplicated blocks with just a pointer. Data compression can further compress each unique data block and save the data footprint, after data deduplication.

Because SQL Server always stores data as files in a filesystem and initializes log files when they are created, thin provisioning from the array should always be enabled to provision just enough space for SQL Server disks.

For data compression, array-based compression can gain substantial space savings in most OLTP SQL Server environments, even with SQL Server row compression or page compression enabled. If a SQL Server OLAP feature such as columnstore is enabled, the savings could be diminishing due to the default compression algorithm in columnstore. HPE recommends monitoring the changes of space savings from the HPE array GUI.

Data deduplication does not save space within a single SQL database or multiple independent databases, no matter how big the databases are because each 8 KB SQL page has a unique page ID. In a customer environment where a production database will be copied multiple times for analytics, reporting, or testing, data deduplication could work well.

If SQL Server Transparent Data Encryption (TDE) is enabled, all this space saving technology from the disk array would be neutralized, which is by design.

TABLE 4. Suggested data compaction options in HPE Storage for SQL Server

HPE Storage array features	Database with TDE	SQL Server OLTP databases	SQL Server OLAP columnstore
Thin-provisioning	No savings	Yes, substantial savings	Yes, substantial savings
Deduplication	No savings	Savings among multiple copies of the same database. No savings within one database or between different databases.	Potentially no savings because columnstores are seldom copied multiple times
Compression	No savings	Yes, substantial savings	Less savings because a columnstore has implemented compression

[5] For more information about NVMe-oF, FC-NVMe, and NVMe/TCP, read the HPE technical white paper NVMe over Fabrics.

Integrated monitoring and management

There is no shortage of tools available on the market for SQL Server monitoring. Many tools provide comprehensive configuration and performance metrics of SQL Server running status, including latches, query time, and trace flags, not only for monitoring but also for further analysis and tuning.

NOTE

Microsoft has a list of Performance Monitoring and Tuning Tools that can be used, depending on the activity to be monitored or tuned.

Most SQL Server administrators and storage administrators have their trusted tools and favorite metrics to help them evaluate the running status of SQL Server. However, a common challenge for monitoring or troubleshooting a storage-related problem is to obtain a holistic full-stack view of the SQL Server databases, instances, Windows operating system, HBAs, SANs, and storage arrays. Another challenge is to quickly filter and identify relevant components of the root cause of the issue.

Several tools and utilities are available for customers with HPE arrays that enable them to have a unified view from SQL Server and Windows to the SAN:

- For customers with SQL Server running on an HPE Nimble Storage array, Nimble Windows Toolkit (NWT) provides Microsoft PowerShell cmdlets for SQL/Windows administrators to identify the HPE Nimble Storage array and associated LUNs associated with each SQL Server database.
- For customers that already rely on HPE OneView software to manage their composable infrastructure, HPE OneView can monitor and manage the entire infrastructure layer from the operation system to the array and LUN.
- For customers using Microsoft System Center Operation Manager (SCOM) or Microsoft System Center Virtual Machine Manager (SCVMM), the HPE Storage Management Pack for System Center plug-in creates system topology and manages events and configurations.
- For customers running SQL Server inside a VM in a VMware environment, the HPE Storage Management Pack for VMware vCenter helps administrators quickly find the mapping information from VM to storage LUN including all relevant components.
- HPE 3PAR Host Explorer is another tool for HPE Primera and HPE 3PAR arrays to set up and align communication between Windows servers and the storage arrays. It is a server-based utility that collects information and sends it to the HPE Primera and HPE 3PAR Storage system. With Host Explorer, storage administrators can easily identify the host connection path.
- For customers already deploying Windows Admin Center (WAC), a public preview of the HPE Storage extension for WAC is available.
- HPE InfoSight is another cloud-based and AI-powered tool to manage infrastructure and cross-stack analysis. HPE InfoSight stores not only the historical configuration and performance information for operating system, server, and storage, but also automatically identifies and tracks potential issues and alerts operators. Furthermore, HPE InfoSight can be used as a modeling and sizing tool with cross-stack data analysis, for example from a SQL VM, ESXi host, or HBA card to a storage controller and a storage LUN. It is important to have every HPE array connected to HPE InfoSight, if possible.

NOTE

For more information, read the HPE white paper Using HPE InfoSight to optimize Microsoft SQL Server 2017 on VMware with HPE 3PAR and HPE Nimble Storage.

SQL Server was designed to leverage server memory (DRAM) as data buffers pool to eliminate disk read/write I/O as much as possible. Before you start to monitor and tune the performance on storage subsystem, it is always a good idea to first verify that the server memory is working as expected, for example, with a sufficient buffer cache hit ratio.

HPE DPP II: SQL Server workload and performance analysis tool

Windows Performance Monitor (perfmon) is probably the most widely used tool for collecting and evaluating Windows server and storage metrics. It is included as part of a Windows operating system and is very powerful. However, although it works well when collecting metrics, it is difficult to generate a perfmon report aligning with SQL Server configurations.

HPE Database Performance Profiler (DPP) II is a database workload and performance analysis tool. It leverages Windows perfmon to collect selected performance counters for SQL Server and CPU/disks, as well as T-SQL commands to collect SQL Server configurations in order to generate an easy-to-consume report based on specific SQL Server configurations.

DPP II greatly eliminates potential concerns about data security from even the most demanding customers because:

- The data collection step includes standard *.cmd, *.ps1 and *.sql scripts so a security administrator can inspect the script before execution.
- The data collected are *.csv files as perfmon standard output and *.txt files as SQL Server configuration output. System administrators can search and replace any database name, host name, and disk name before sending the data back to HPE for report generation.

The DPP II tool is not tied to any HPE hardware and it does not incur any cost to HPE customers. It requires an HPE employee with a valid HPE email address to download the script from an HPE internal website and generate the report. Any HPE customer can contact an HPE representative for more details about DPP II, including a sample report, instructions, and related scripts.

Integrated data protection

Because every enterprise data center faces many challenges in addition to hardware failures and data center outages, having only a high availability and disaster recovery system is not enough protection for a production system. Viruses, ransomware, and human errors are just a few examples of problems that require a data backup and recovery system in addition to high-availability/disaster recovery solutions. SQL Server has data backup functions that could be accessed via T-SQL or SQL Server Management Studio (SSMS), but these tools do not meet all customer requirements. Traditionally, customers would deploy additional third-party backup software for a more comprehensive SQL Server data backup solution.

HPE Storage systems further the HPE data protection vision by providing a complimentary copy of the HPE Recovery Manager Central (RMC) data management solution. RMC has several modules for different applications; one is for SQL Server. With RMC for SQL Server, HPE customers can experience faster backup/restore performance, access the integrated management GUI, and achieve the lowest TCO.

NOTE
For more detailed information about RMC for SQL Server, watch the HPE Recovery Manager Central for Microsoft SQL demo video and read the HPE Recovery Manager Central for Microsoft SQL Server Configuration and best practices guide. The collateral at the HPE Recovery Manager Central website is also helpful.

HPE PRIMERA AND HPE 3PAR PRODUCT FAMILIES

Powered by AI, HPE Primera storage redefines mission-critical storage for Tier 0 applications. Designed for NVMe and SCM, HPE Primera delivers remarkable simplicity and application-aware resiliency for mission-critical workloads through intelligent autonomous storage that anticipates and prevents issues across the infrastructure stack.

HPE Primera delivers on the promise of intelligent storage advanced data services and simplicity for mission-critical applications with a services-centric operating system that sets up in minutes and upgrades seamlessly to minimize risk and be transparent to applications. All these capabilities add up to enable HPE Primera to provide 100% availability guaranteed.

Best practices for connecting HPE Primera and HPE 3PAR to SQL Server

HPE Primera and HPE 3PAR arrays usually target mission-critical environments with 5 nines, 6 nines, or 100% data availability and the highest performance demands.

HPE recommends the following design and implementation principles when connecting HPE Primera or HPE 3PAR arrays to a physical SQL Server or a virtualized SQL Server:

- HPE recommends a four-node (or more) configuration because it can maintain both write cache performance and data availability even in the event of the unlikely failure of one storage controller. SQL Server log file performance relies completely on write cache.
- HPE recommends a single common provisioning group (CPG) with all physical disks using recommended RAID protection together with a separate LUN (VV) hosting SQL Server data files, log files, and TempDB.
- Use the App Volume Sets option to group volumes hosting SQL Server data files, log files, and TempDB into one set and export the set to SQL Server hosts. During snapshot and remote copy configuration, LUNs hosting the same SQL Server data files and log files or LUNs hosting files from business-related databases must be kept in one set or group. However, LUNs dedicated to hosting TempDB files should be excluded from the set.

- HPE 3PAR Host Explorer collects the following information and sends it to the HPE Primera and HPE 3PAR arrays. Support engineers can leverage this convenient information for problem troubleshooting.
  - Host operating system and version
  - Fibre Channel and iSCSI HBA details
  - Multipath driver and current multipath configuration
  - Cluster configuration information
  - Device and path Information
- HPE LUNInfo is a server-based command line utility that provides valuable information on the volume mapping between the host and the HPE Primera and HPE 3PAR Storage. HPE recommends installing HPE LUNInfo[6], which provides the following details:
  - LUNs that are exported to the host
  - LUN to device file mapping information
  - List of LUNs along with additional details
- Although no special multipath software is required for Windows Server 2016/2019 connected to HPE Primera or HPE 3PAR arrays because the required device-specific modules (DSMs) are built-in, HPE recommends confirming that all exported LUNs from the arrays are under the Round Robin multipath policy, except the LUNs with Peer Persistence, which should be under the Round Robin with Subset policy. To verify the policy, run the `mpclaim` command at the Windows command prompt or check the MPIO tab of each disk from the Disk Management settings. If HPE Host Explorer is installed, issue the `tpdhostagent list` command to list all exported LUNs from the arrays and the MPIO settings as well.

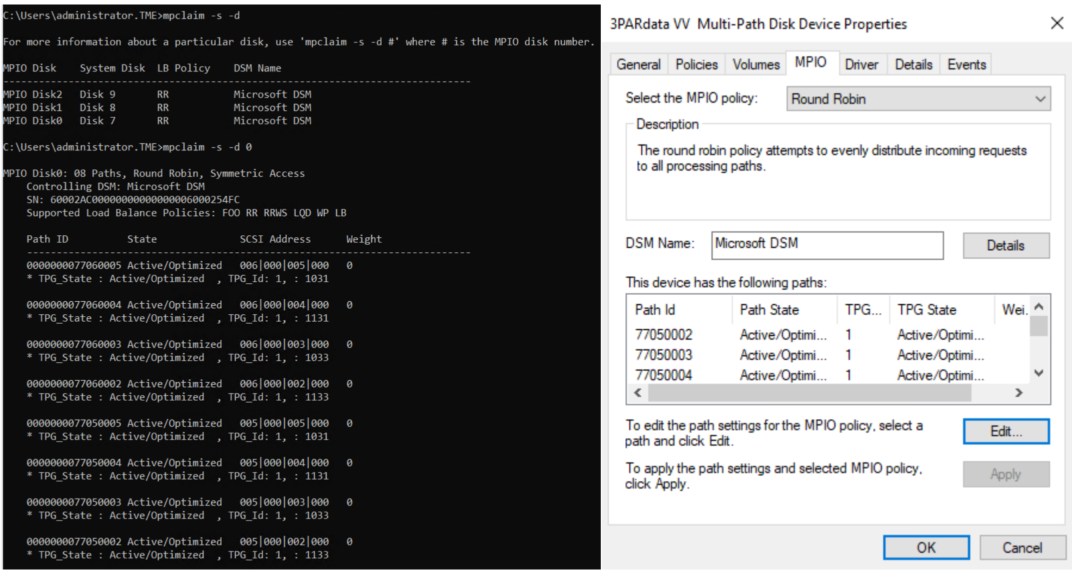


FIGURE 7. Multipath setting for HPE 3PAR/Primera LUNs on the Windows MPIO tab

With virtualized SQL Server within VMware, similar settings in ESXi through vCenter should be applied, although the GUI and the CLI are different.

- HPE Peer Persistence technology can guarantee the highest RPO/RTO while greatly simplifying recovery procedures during a failure of a single storage array, a rack, or a single site. HPE highly recommends this solution for customers that value extreme high-availability. A configuration of Peer Persistence known as three data center Peer Persistence (3DC PP) can include a third site for data replication to meet a customer's most demanding high-availability requirement.
- Thin provisioning should be enabled by default. For OLTP workloads, compression normally is enabled because substantial space savings are witnessed from HPE InfoSight data. Deduplication only works when multiple SQL Server databases with the same data blocks (8 KB pages) are stored in the same array, for example, a production database and its database clones for the test/dev environment.

[6] You can download and install HPE LUNInfo from the HPE Software Depot: https://myenterpriselicense.hpe.com/cwp-ui/free-software/3PARinfo

TABLE 5. Suggested data compaction settings in HPE Primera arrays for LUNs hosting SQL Server files

Desired outcome	DB data volume data reduction	TempDB volume data reduction	DB log volume data reduction
Max potential space savings and least management effort	Yes: Deduplication Yes: Compression	Yes: Deduplication Yes: Compression	Yes: Deduplication Yes: Compression
More granular tuning	No: Deduplication Yes: Compression	No: Deduplication No: Compression	No: Deduplication Yes: Compression
Extreme performance	No: Deduplication No: Compression	No: Deduplication No: Compression	No: Deduplication No: Compression

- If SQL Server TDE is enabled, neither compression nor deduplication will work. Therefore, HPE advises using either HPE Primera/3PAR Storage array-based data-at-rest encryption, or disable array-based deduplication and compression when the application, operating system, or host has implemented data-at-rest encryption.
- Additional SCM module could be added to select models of HPE 3PAR arrays to further reduce the I/O latency of an all-flash storage system. Based on HPE lab tests, SCM can reduce read I/O latency up to 52% in a SQL Server environment, which benefits SQL Server prefetch activity in OLTP workloads as well as other OLAP/reporting workloads.

NOTE
For more details, refer to the HPE white paper Optimizing Microsoft SQL Server 2017 with HPE 3PAR storage and HPE Memory-Driven Flash.

- The PowerShell Toolkit for HPE Primera and HPE 3PAR is a set of cmdlets based on the HPE Primera and HPE 3PAR REST API. The toolkit is tightly integrated with Windows PowerShell, enabling Windows administrators and storage administrators to automate a wide variety of administrative tasks. The PowerShell Toolkit works well for infrastructure and storage admins but it can also be very powerful when combined with SQL Server cmdlets.
- For backup and restore of SQL Server, HPE recommends having an integrated backup/restore solution that leverages HPE Primera and HPE 3PAR array-level snapshot and replication technologies to minimize the I/O impact to SQL Server. Industry partners including Veritas, Commvault, Micro Focus, and Veeam offer a wide selection of such integrated solutions. RMC also provides an integrated backup/restore solution.

HPE NIMBLE STORAGE PRODUCT FAMILY

HPE Nimble Storage All Flash arrays combine a flash-efficient architecture with HPE InfoSight predictive analytics to achieve fast, reliable access to data and 99.9999% guaranteed availability. Radically simple to deploy and use, the arrays are cloud-ready, providing data mobility to the cloud through HPE Cloud Volumes. The storage investment made today can be supported well into the future.

HPE Nimble Storage All Flash arrays include all-inclusive licensing, easy upgrades, and flexible payment options. They are future-proofed for new technologies such as NVMe and SCM.

Best practices for connecting HPE Nimble Storage arrays to SQL Server

For HPE Nimble Storage connected to SQL Server running on a physical server running Windows, consider the following best practices:

- HPE Nimble Windows toolkit (NWT) contains three components critical to SQL Server customers:
  - The Nimble Connection Manager automatically adjusts multipath settings for both Fibre Channel and iSCSI connections.
  - NWT installs a cmdlet that amins can use to identify the mapping information between each SQL Server database to HPE Nimble Storage volume and run HPE Nimble Storage array-level snapshot and restore operations.
  - NWT includes a VSS hardware provider from HPE Nimble Storage. With this VSS hardware provider, SQL and storage admins can schedule application-consistent snapshot from either PowerShell on the host or the HPE Nimble Storage array GUI.
- Assign the correct Nimble performance policy to each volume hosting SQL Server files. HPE Nimble Storage volumes hosting SQL Server data files and volumes hosting SQL Server TempDB files should be assigned the SQL Server performance policy. Volumes hosting SQL Server log files should be assigned the SQL Server Logs policy.

- Create folders and put all relevant SQL Server volumes into one folder. Although this does not impact performance or data consistency, it greatly simplifies management.
- The HPE Nimble Storage AF Turbo feature leverages SCM to further reduce I/O latency on select all-flash models. Based on HPE lab tests, SCM can reduce read I/O latency up to 33% in SQL Server environments, which benefits SQL Server prefetch activity in OLTP workloads as well as other OLAP/reporting workloads both in physical servers and virtualized deployments.

NOTE
For more details, read the HPE white paper Optimizing Microsoft SQL Server 2019 with HPE Nimble Storage and Storage Class Memory.

- Peer Persistence technology can guarantee the highest RPO/RTO while greatly simplifying recovery procedures during a failure of a single storage array, a rack, or a single site. HPE highly recommends this solution for customers that value extreme high-availability.
- Thin provisioning should be enabled by default. For OLTP workloads, compression normally could be enabled as substantial space savings had been witnessed from HPE InfoSight data. Deduplication only works when there are multiple SQL Server databases with same data blocks (8 KB pages) are stored in same array, for example, production databases and its database clones for a test/dev environment.

TABLE 6. Suggested data compaction settings in HPE Nimble Storage arrays for LUNs hosting SQL Server files

Desired outcome	DB data volume data reduction	TempDB volume data reduction	DB log volume data reduction
Max potential space savings and least management effort	Yes: Deduplication Yes: Compression	Yes: Deduplication Yes: Compression	Yes: Deduplication Yes: Compression
More granular tuning	Yes: Deduplication Yes: Compression	No: Deduplication Yes: Compression	No: Deduplication Yes: Compression
Extreme performance	No: Deduplication No: Compression	No: Deduplication Yes: Compression	No: Deduplication Yes: Compression

- If SQL Server TDE is enabled, neither compression nor deduplication will work. HPE recommends using HPE Nimble Storage array-based data-at-rest encryption, or disable array-based deduplication and compression in cases where the application, operating system, or host implemented data-at-rest encryption.
- The PowerShell Toolkit for HPE Nimble Storage is a set of cmdlets based on the HPE Nimble Storage REST API. The toolkit is tightly integrated with Windows PowerShell, enabling Windows and HPE Nimble Storage administrators to automate a wide variety of tasks. The PowerShell Toolkit works well for infrastructure and storage admins but it can also be powerful when combined with SQL Server cmdlets.
- For backup and restore of SQL Server, HPE highly recommends having integrated backup/restore solution leveraging HPE Nimble Storage array-level snapshot and replication to minimize the I/O impact to SQL Server. RMC is such an integrated solution; industry partners such as Veritas, Commvault, Micro Focus, and Veeam also provide integrated backup/restore solutions supported by HPE Nimble Storage.

HPE NIMBLE STORAGE DHCI

The HPE Nimble Storage dHCI solution is a disaggregated hyperconverged infrastructure (HCI) platform that delivers the flexibility of converged infrastructure and the simplicity of HCI. This scalable solution is designed, tested, and documented to address the business requirements, workloads, and applications of HPE customers. The solution incorporates a wide range of products into a portfolio of repeatable, scalable, and composable technologies that are supported by HPE.

Best practices for connecting HPE Nimble Storage dHCI to SQL Server

HPE Nimble Storage can be connected to SQL Server running as a virtual machine in a VMware environment. The setup could be part of a HPE Nimble Storage dHCI solution or part of a more common combination of an x86 server with VMware and HPE Nimble Storage.

Additional best practices for connecting HPE Nimble Storage dHCI to SQL Server include:

- For SQL Server VMs running in a virtualized environment, including HPE Nimble Storage dHCI, options are available to provision storage, including VMFS datastores, physical disks, HBA pass-throughs, pRDM, vRDM, in-VM iSCSI, and vVols. In general, the implementation of storage for a SQL Server VM could be traditionally categorized into two designs: shared or dedicated. A shared design means SQL Server VMs share the same storage LUN with other VMs running different applications with different I/O profiles. A dedicated design means only SQL Server VMs access a storage LUN.
- A VMFS datastore is usually implemented as a shared design although it also could be a dedicated design. RDM and vVol are the most common dedicated design options because they meet the requirements of SQL Server FCI/Windows Server Cluster.
- Virtualization is excellent for improving overall resource utilization yet challenging for resource isolation and performance management. This is particularly true for a shared design model with a mix of applications accessing a LUN at the same time as a SQL Server workload.
- For a mission-critical environment that requires the highest performance, reliability, and serviceability, a dedicated design of storage provisioning is more common. The best practice for this type of environment is similar to a physical SQL Server environment, such as when several RDM LUNs from HPE Nimble Storage to SQL Server VMs with system db/user-db/user-log/tempdb/backup files are deployed on different LUNs.
- VMware vVol is another choice of storage provisioning for SQL Server VMs because it provides I/O performance and management cost benefits. VMware vVol is supported with the HPE Nimble Storage plug-in for vCenter, both for dHCI deployment and standard HPE Nimble Storage deployment.
- Shared design is more common among SQL Server VMs because it provides extensive savings in management costs. Multiple VMs running SQL Server or other applications share the same storage infrastructures, including the same storage array, HBA, and LUN.
- An important best practice to shared design, which could potentially be called "cluster design," provides better performance segregation. The principle is to place vdisks hosting SQL Server files of similar I/O patterns from different VMs into one VMFS datastore in order to create different I/O patterns in different SQL Server files. For example, one datastore for VM vdisks could host SQL Server system database files, another datastore for VM vdisks could host SQL Server log files, and so on. Combined with HPE InfoSight Cross-Stack Analytics, this solution provides better manageability with fewer VMFS datastores, meets I/O separation requirements, and makes it easier for system administrators to identify noisy neighbor problems.

  Figure 8 illustrates these designs. Instance A storage implementation is a shared design. Instance B storage implementation is a dedicated design and Instance C storage implementation is a cluster design.

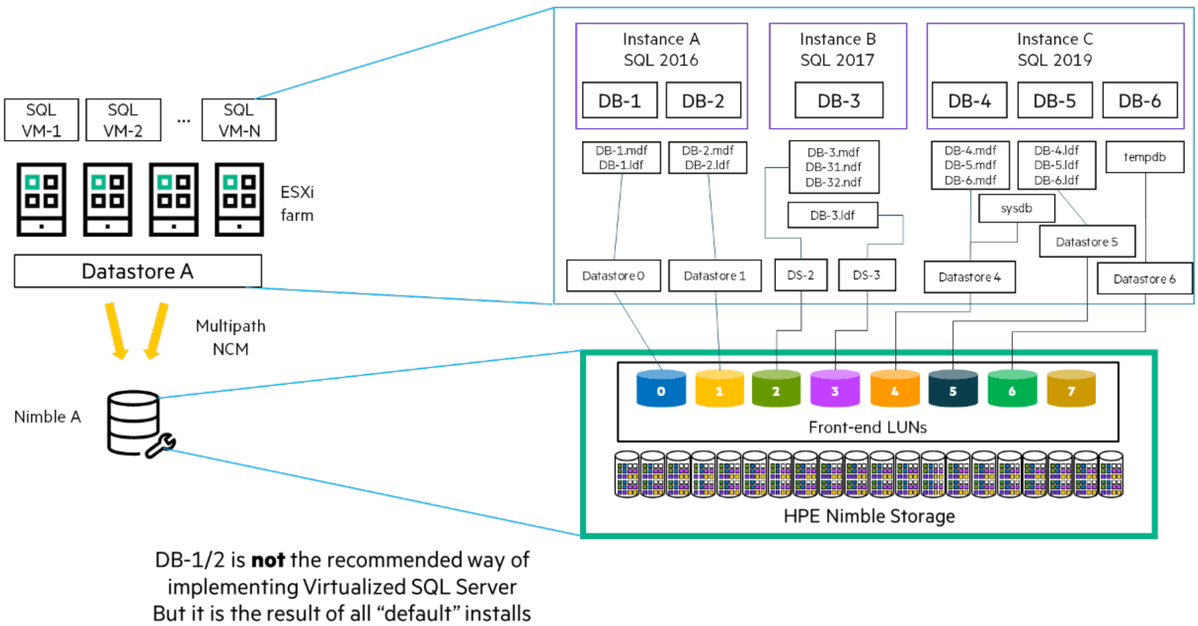


FIGURE 8. HPE Nimble dHCI—SQL Server VM disks and VMFS datastore

- Create folders and put all relevant VMware volumes from the same vCenter, the same data centers, and the same ESXi clusters into one folder. This greatly simplifies management tasks. A folder is optional in normal storage LUN provisioning with HPE Nimble Storage but mandatory for VMware vVol deployment.
- HPE Nimble Storage can leverage SCM to further reduce I/O latency on select all-flash models. Based on HPE lab tests, SCM can reduce read I/O latency up to 50% in SQL Server environments, which benefits SQL Server prefetch activity in OLTP workloads as well as other OLAP/reporting workloads both in physical servers and virtualized deployments. For more details, read the HPE white paper Optimizing Microsoft SQL Server 2019 with HPE Nimble Storage and Storage Class Memory.
- Jumbo frames can boost iSCSI performance. Enable jumbo frames on HPE Nimble Storage dHCI systems, vSwitches, vNICs, pNICs, and physical switches. Enable jumbo frames from the HPE Nimble Storage management GUI by clicking **Subnet** under **Network**.
- For HPE Nimble Storage LUNs consumed by SQL Server VMs via dedicated models such as RDM, DirectPath IO, physical disks, or in-guest iSCSI, the performance policy should be SQL Server for data files and TempDB files and SQL Server Logs for log files, similar to physical servers.
- For HPE Nimble Storage LUNs exported as VMFS datastores and shared by both SQL Server VMs and other VMs, the performance policy should be ESXi.
- Peer Persistence technology can guarantee the highest RPO/RTO while greatly simplifying recovery procedures during a failure of a single storage array, a rack, or a single site (for example, a power outage). HPE highly recommends this solution for customers that demand extreme high-availability.
- Thin provisioning should be enabled by default. For OLTP workloads, compression normally is enabled because substantial space savings are witnessed from HPE InfoSight data. Deduplication only works when multiple SQL Server databases with the same data blocks (8 KB pages) are stored in same array, for example, production database and its DB clones for test/dev environment.

TABLE 7. Suggested data compaction settings in HPE Nimble DHCI arrays for LUNs hosting SQL Server files

Desired outcome	DB data volume data reduction	TempDB volume data reduction	DB log volume data reduction
Max potential space savings and least management effort	Yes: Deduplication Yes: Compression	Yes: Deduplication Yes: Compression	Yes: Deduplication Yes: Compression
More granular tuning	Yes: Deduplication Yes: Compression	No: Deduplication Yes: Compression	No: Deduplication Yes: Compression
Extreme performance	No: Deduplication No: Compression	No: Deduplication Yes: Compression	No: Deduplication Yes: Compression

HPE MSA PRODUCT FAMILY

HPE MSA Gen6 is the next generation of HPE entry-level storage arrays. New high-performance models match today's workloads, automating and simplifying IT processes, and protecting business data anywhere. These affordable storage arrays include a choice of HDD and SSD media.

A leading entry-point solution for HPE ProLiant servers for more than a decade, HPE MSA Storage includes three Gen6 models that deliver simple, high-performance, flash-ready storage for SMBs:

- **HPE MSA 1060** is designed for users with fixed capacity and performance needs supporting smaller IT workloads
- **HPE MSA 2060** offers the flexibility to scale both performance and capacity to meet the most demanding IT workloads
- **HPE MSA 2062** includes an HPE MSA 2060 bundled with two 1.92 TB SSD drives and an Advanced Data Service license, providing a substantial savings (up to 31% off list price) and the ability to start from Day One with a high-performance, affordable, hybrid flash storage solution.

All HPE MSA Gen6 arrays include a standard set of features that are ideal for SMB customers looking to accelerate to a hybrid flash SAN infrastructure. The Gen6 architecture simplifies interacting with the array. The HPE MSA Storage Management Utility (SMU) v4 provides intuitive, step-by-step guided workflows that configure the array straight out of the box, eliminating common errors, and dramatically improving the user experience, especially for tasks users seldom perform. Guided workflows simplify these tasks and practically eliminate the need for a reference manual.

Best practices for connecting an HPE MSA array to SQL Server

In addition to general best practices of deploying SQL Server, additional considerations for HPE MSA storage array include:

- No additional multipath software installation is required for a Windows Server 2016/2019 connected to HPE MSA storage array because the required DSMs are built-in. HPE recommends confirming that all LUNs exported from the MSA array are configured with multiple paths in Windows Server. You can verify this configuration from the MSA Control Panel by selecting the MPIO check box. The optimal number of MPIO paths to an MSA LUN is no more than eight, including four active/optimized and four active/unoptimized paths. In some cases, a manual disk/path claim and a reboot may be required.
- Change the MPIO timers on Windows Server hosts when connecting to large numbers of LUNs from the system default of 20 seconds to 90 seconds for Fibre Channel LUNs and 300 seconds for iSCSI LUNs.
- MSA Gen6 arrays support various configuration options to meet customers' demands, including abilities to leverage SSD drives for lower I/O latency and higher I/O throughput. SSD drives could be used as a read cache of SAS drives to improve read I/O latency. Another option is to configure SSD drives as a performance tier in a storage pool. Within a pool, MSA Automated Tiering software can allocate each 4 MB page between different tiers dynamically based on I/O profiles. MSA Gen6 arrays support all-flash configuration to meet low latency demand.
- Because OLTP SQL Server I/O profiles are sensitive to I/O latency with a large amount of random reads, HPE highly recommends including SSD drives in an HPE MSA array. Configure read cache, automated tiering, or all-flash, based on performance requirements, budget, and the application I/O profile.

TABLE 8. Comparison of SSD uses for SQL Server files

	SQL Server data files	SQL Server log files	SQL Server TempDB files	Cost
Single SSD tier (all-flash)	✔ ✔ ✔	✔ ✔ ✔	✔ ✔ ✔	$$$
SSD in hybrid pool(tiering)	✔ ✔	✔ ✔	✔ ✔	$$
SSD as read cache	✔		✔	$

- Design a storage pool and a disk group to stripe SQL Server data to as many disks as possible. Recommended configurations include one storage pool with balanced performance and management cost, which fits most customer scenarios, or a dual-pool design for very large HPE MSA Gen6 configurations to unleash the full potential performance/capacity.
- HPE recommends that SSD drives in HPE MSA Gen6 arrays be protected by using RAID 1/10/5 as a storage tier in a storage pool or as read cache. All mechanical disks should be protected with either MSA-DP+ or RAID 6. Read cache can improve random read I/O generated on SQL Server data files but does not benefit write I/O on SQL Server log files.

  Best practices for HPE MSA Gen6 for SQL Server include:

  – HPE MSA storage arrays leverage thin provisioning for all LUNs, which is ideal for SQL Server deployments.

  – HPE MSA storage arrays are supported by HPE Storage Management pack for VMware vCenter and HPE Storage Management Pack for System Center. HPE recommends using these integrated tools for managing and monitoring HPE MSA arrays in physical and virtualized environments.

  – HPE MSA Gen6 storage arrays provide a VSS hardware provider to be integrated with VSS. It is possible for ISVs and Windows/SQL admins to create application-consistent snapshots in HPE MSA arrays as an instant backup/restore for SQL Server.

NOTE

For more details and updated information, read HPE MSA 1060/2060/2062 Storage Arrays Best Practices and HPE MSA GEN6 Virtual Storage Technical Reference Guide. You can find these documents under the MSA Storage tab in the Hewlett Packard Enterprise Information Library.

HPE APOLLO 4200 PRODUCT FAMILY AND AZURE STACK HCI

The HPE Apollo 4200 Gen10 server offers an architecture optimized for Big Data analytics, SDS, backup and archive, and other data storage intensive workloads. Its unique, easily serviceable 2U design saves data center space with up to 28 LFF or 54 SFF hot-plug drives. It delivers accelerated performance with a superior bandwidth and balanced architecture, Intel® Xeon® processors, and NVMe connected SSDs. The focus on security extends from FIPS 140-2 Level 1 validated storage controllers to the system silicon level, taking full advantage of HPE innovations in firmware protection, malware detection, and recovery.

Microsoft Azure Stack HCI (ASHCI) is a hyperconverged infrastructure operating system delivered as an Azure service. It provides the latest security, performance, and feature updates. Deploy and run Windows and Linux virtual machines in the data center or at the edge leveraging existing tools, processes, and skillsets. Extend the data center to the cloud with capabilities such as Azure Backup, Azure Monitor, and Azure Security Center.

Best practices for connecting HPE Apollo 4200 Gen10 and ASHCI to SQL Server

In addition to general best practices for deploying SQL Server, additional best practices to consider when deploying SQL Server on HPE Apollo 4200 Gen10 with Azure Stack HCI include:

- HPE Apollo A4200 is a capacity/density-optimized server that supports up to two CPU sockets, 2 TB DRAM, six NVME devices, and 24 LFF devices or 48 SFF devices in a 2U rack space. This makes it a perfect choice for SDS for SQL Server workloads that require large volumes of data, high data throughput, and low cost per TB. Examples include data warehouses, decision support system, reporting, historical data, and archives.
- ASHCI supports a maximum of 4 PB per storage pool. A single A4200 could hold up to 737 TB SSDs (48 15.36 TB SAS SSD with six NVMe cache drives) or 384 TB HDDs (24 16 TB SAS HDDs with six NVMe cache drives).
- HPE recommends using the protection method of three-way mirroring for critical data when deploying SQL Server on ASHCI disks or in Hyper-V VMs on ASHCI disks.
- HPE Persistent Memory featuring Intel® Optane™ DC persistent memory (PMEM) can benefit SQL Server workloads by reducing I/O latency. HPE Apollo A4200 supports PMEM with a maximum capacity of 512 GB per CPU socket. HPE recommends configuring PMEM as a SQL Server hybrid buffer pool, which can potentially boost I/O operations not only because the data is on PMEM, which is much faster than any other storage type, but also because the normal file system I/O stack is omitted because Windows, NTFS, and SQL Server support DAX mode.
- Because all log write I/O, including commits and checkpoints, must be copied to persistent storage in a remote server before acknowledgement of "I/O complete", the network latency for ASHCI data traffic is critical for SQL Server OLTP performance. All network best practices should be followed and RDMA is particularly important in this case.

NOTE
For more detailed information about deploying Microsoft Azure Stack HCI on HPE Apollo 4200 Gen10 server, refer to the Windows Azure Stack HCI with HPE Apollo 4200 Gen10 Server Deployment Guide.

SUMMARY

Based on each customer's unique business requirement and IT environment, the storage platform for Microsoft SQL Server could be dramatically different for each customer. HPE, with more than 30 years of partnership with Microsoft and hundreds of thousands of HPE storage platforms connected to SQL Server, understands that there is no "one product fits all" solution.

There is a variety of storage products and solutions from HPE Storage, with integration and optimization built-in for business applications such as SQL Server. A thorough and elegant architectural design and deployment is critical for a successful project delivery. HPE recommends that customers contact HPE Pointnext Services, HPE partners, and HPE resellers to get the most professional service.

Resources, contacts, or additional links

HPE and Microsoft Alliance website
hpe.com/us/en/alliance/microsoft.html

HPE Microsoft Storage Solutions home
hpe.com/storage/microsoft

HPE Servers
hpe.com/servers

HPE Storage
hpe.com/storage

HPE Apollo Systems
hpe.com/apollo

LEARN MORE AT

hpe.com/pointnext

a00108570ENW, December 2020

Hewlett Packard Enterprise