Technical Report

Best Practice Guide for Microsoft SQL Server with NetApp EF-Series

Mitch Blackburn, Pat Sinthusan, NetApp
July 2015 | TR-4259

Abstract

This best practice guide is intended to help storage administrators and database administrators successfully deploy Microsoft SQL Server on NetApp® EF-Series storage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

In most online transaction processing (OLTP) systems, the processor, memory, and I/O subsystem in a server are well balanced and are not considered performance bottlenecks. The major source of performance issues in OLTP environments typically relates to the storage I/O activity. The speed of traditional hard disk drive (HDD)–based storage systems does not match the processing capabilities of the servers.

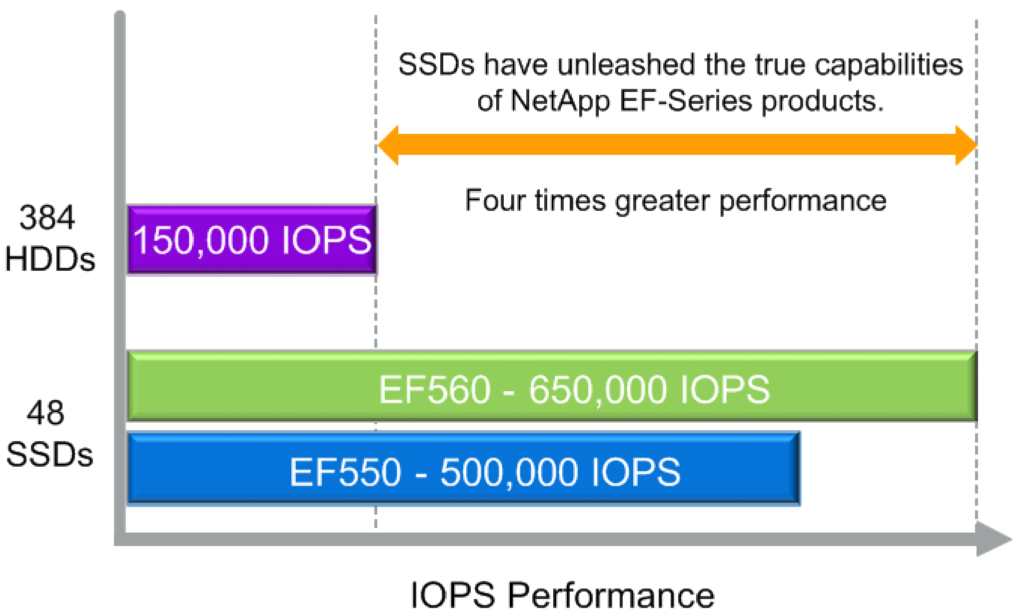
As a result, a situation often occurs in which a powerful processor sits idle, waiting for the storage I/O requests to complete, negatively affecting user and business productivity. The negative effect on productivity extends the time to return on investment (ROI) and increases overall total cost of ownership (TCO). Therefore, storage input/output operations per second (IOPS) performance and latency become strategic considerations for business. It is critical to make sure that the response time goals are met and that performance optimization is realized for other system resources (processor and memory).

The NetApp EF-Series flash array provides a robust platform for delivering exceptional performance of up to 650,000 IOPS for the EF560 flash array, with submillisecond response times to mission-critical applications. The EF-Series flash array leverages the latest technology in solid-state drives (SSDs). It also has a strong heritage of providing the capability to handle diverse workloads to provide superior business value through the acceleration of latency-sensitive and high-I/O environments.

The NetApp EF-Series flash array leads the market in delivering performance and latency. The importance of latency varies depending on business needs. For example, imagine a global investment banking firm that calculated that every millisecond of latency represents $100M per year in lost opportunity. That equals $100,000 per microsecond. An EF array configured in a single Dynamic Disk Pool (DDP) running a typical 80/20 read/write database workload can deliver over 300K IOPS at 0.6ms. Other all-flash arrays deliver the 300K IOPS with about 1.2ms latency, which equals $60M per year in additional opportunity. That's the value of latency.

As shown in Figure 1, SSDs unleash the true capabilities of NetApp EF-Series products. The EF-Series flash array allows database administrators to increase the performance of existing applications while lowering the cost of the IOPS ratio without rearchitecting the application. The E-Series with HDDs is only disk bound, not CPU bound. SSDs enable the utilization of all of the performance capabilities of the multicore EF-Series.

Figure 1) Performance comparison between HDD and SSD.



The EF560 can deliver this performance in one-seventh of the rack space required for HDD drives. The EF-Series, available with up to 192TB of raw SSD capacity, provides capacity and bullet-proof reliability to meet the requirements of the most demanding organizations. This technical report provides an overview of best practices for Microsoft SQL Server 2014 with NetApp EF-Series flash array.

1.1 Intended Audience

This technical report is intended for NetApp customers, partners, employees, and field personnel who are responsible for deploying a SQL Server database solution in a customer environment. NetApp assumes that the reader is familiar with the various components of the solution.

1.2 Caveats

This document assumes that the database is either being relocated to an EF-Series storage system or being created on an EF-Series storage system to achieve high performance.

This document also assumes that the user wants to improve performance of an OLTP application.

2 Introduction to EF-Series

2.1 EF-Series Hardware Overview

The EF-Series flash array continues NetApp’s long-standing heritage of delivering powerful solutions to meet unique business needs. With high IOPS and submillisecond response times, the EF-Series enables business-critical applications to deliver faster results and improve the customer experience. This combination of high IOPS and ultra-low latency makes an EF-Series flash array a great choice for database-driven applications requiring a dedicated extreme performance solution.

Table 1 provides an overview of EF560 flash array performance.

Table 1) EF-Series EF560 performance.

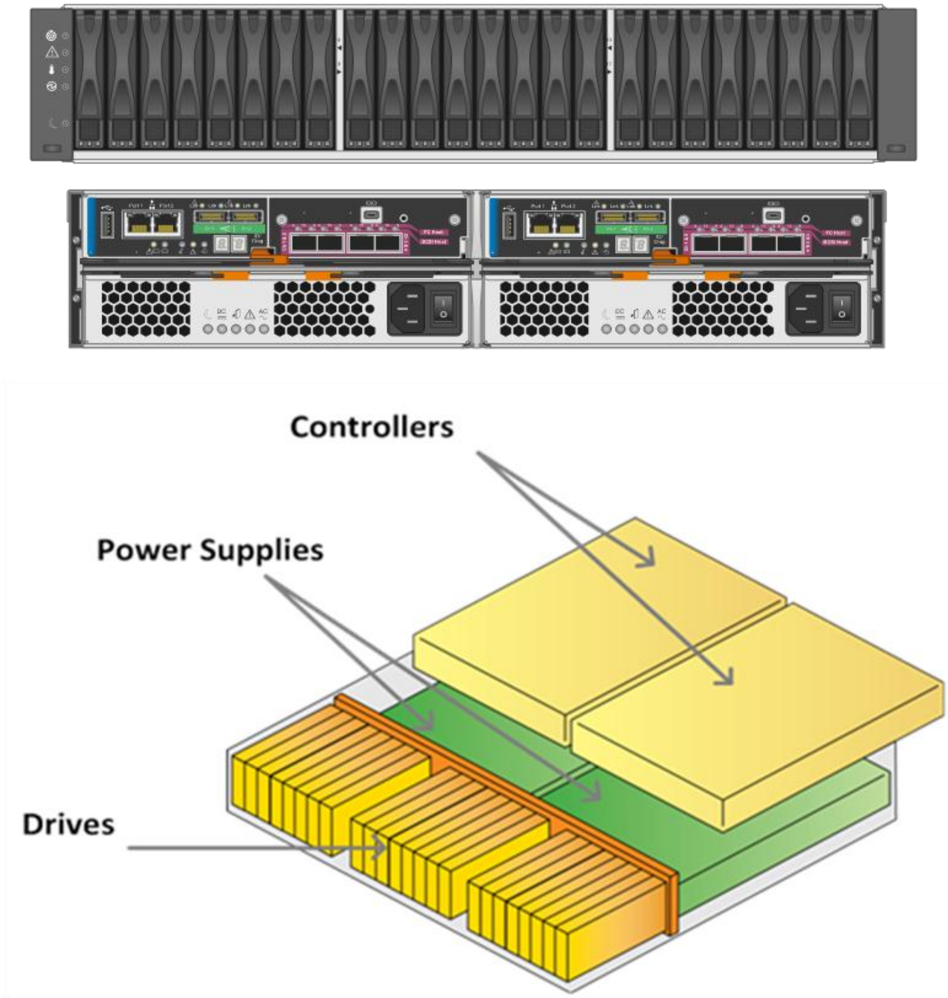
Performance Comparison	EF560
Burst I/O rate: cache reads (512KB)	900,000 IOPS
Sustained I/O rate: random disk reads (4KB) —DDP	650,000 IOPS @ 0.78ms 300,000 IOPS @ 0.21ms
Sustained I/O rate: random disk writes (4KB) —DDP	100,000 IOPS @ 2.5ms 68,000 IOPS @ 0.46ms
Sustained throughput: sequential disk reads (512KB)	12Gb/sec
Sustained throughput: sequential disk write (cache mirror disabled 512KB)	9Gb/sec
Sustained throughput: sequential disk write (cache mirror enabled 512KB)	6.5Gb/sec

The EF-Series flash array is a fully redundant 2U/24 system that supports up to four additional 2U trays of SSDs for up to 192TB of raw capacity.

The EF-Series has dual active controllers, four 6Gb SAS disk expansion ports, and 24GB of battery-backed controller cache. It runs on the performance-optimized and enterprise-proven NetApp SANtricity® software.

As illustrated in Figure 2, the EF-Series supports dual controller canisters, power supplies, and fan units for hardware redundancy. The shelves are sized to support up to 24 2.5" disk drives in 2U of rack space.

Figure 2) EF-Series EF560 with 16GB FC host interface option.



Along with performance, the key to maximizing value is to maximize efficiency. Historically, companies have sacrificed efficiency to achieve extreme performance levels by overprovisioning their storage. But that is changing. The all-flash EF-Series helps customers balance performance and efficiency by eliminating overprovisioning, thereby dramatically reducing costs.

With the performance of more than 1,000 traditional drives, a single EF-Series can meet extreme requirements with 95% less rack space, power, and cooling. This is a significant benefit to customers who are used to deploying partially filled disks to improve application performance.

In addition to cost efficiency, the EF-Series flash array provides application efficiency. By completing a higher volume of application operations, customers can become efficient and drive better results.

The EF-Series has a fully redundant I/O path that provides automated failover. Surprisingly this is not a given in many flash products available today, but it is an absolute requirement for enterprise accounts looking to implement this type of technology.

All management tasks are performed while the EF-Series remains online with complete read/write data access. This feature allows storage administrators to make configuration changes and conduct maintenance without disrupting application I/O.

The EF-Series flash array also offers advanced data protection common to enterprise storage. This benefit protects against data loss and downtime events, both locally with NetApp Snapshot® technology and remotely with synchronous and asynchronous replication.

2.2 SANtricity

NetApp SANtricity is the GUI management interface for the EF-Series that is based on the Java framework. SANtricity can be installed on Windows or Linux operating systems. The management application must be installed on a management node that does not participate in production data delivery. The software is available in 32-bit and 64-bit versions, and the installation process detects whether installation of the package was performed on the incorrect OS version.

SANtricity client software can be installed on Windows or Linux operating systems for out-of-band management of the storage array. In this configuration, the host agent functionality for in-band management does not function, and the number of client connections is limited to eight. To manage the storage arrays by using in-band connections, the management client must be run on a server OS and have Fibre Channel (FC) connectivity to all arrays. The eight-connection limit for out-of-band management client connections does not apply to in-band management.

To create volume groups on flash arrays, the first step while configuring SANtricity is to assign a redundant array of inexpensive disks (RAID level). This assignment is then applied to the disks selected to form the volume group.

Flash arrays support RAID levels 0, 1, 3, 5, 6, and 10 or DDP.

To simplify the storage provisioning, NetApp recommends using the SANtricity automatic configuration feature. The configuration wizard analyzes the available disk capacity on the array. It then selects disks that maximize array performance and fault tolerance while meeting capacity requirements, hot spares, and any other criteria specified in the wizard.

2.3 EF-Series Volume Group

The volume group is the top-level unit of storage in the EF-Series. When a storage array is deployed, the first step in presenting the available disk capacity to various hosts is to create:

- The number of disks required to meet performance requirements
- The desired level of RAID protection to meet specific business requirements
- Volume groups with sufficient capacity

Capacity planning depends on detailed customer input and discovery; however, protection and performance planning is a standardized implementation practice.

For example, EF-Series flash arrays support multiple RAID levels, and each RAID level provides standardized functionality with the associated best practices. One of these best practices includes disk selection criteria to achieve disk-level, drawer-level, and shelf-level protection from common disk failure scenarios.

When selecting disks to create a volume group, administrators follow a standard pattern that uses both controller channels and spreads the disks across shelves and drawers in the configuration. This method does not offer a level of data protection for RAID 0 volume groups. However, it does establish the disk selection pattern of allocating disks for RAID levels that offers protection against single- and double-disk fault scenarios.

Size the volume groups to meet business requirements. However, large volume groups that use RAID levels 1, 3, 5, 6, or 10 require a significant number of hours to complete the reconstruction process from a failed disk. The larger the volume group, the longer the reconstruction time.

SANtricity E-Series DDPs simplify RAID management, improve data protection, and maintain predictable performance under all conditions. They do this using next-generation technology to minimize the performance effect of drive failures and return the system to optimal conditions up to eight times faster than traditional RAID technology. With shorter rebuild times and patented prioritization reconstruction technology, DDP significantly reduces exposure to numerous cascading disk failures. It also offers a level of data protection that cannot be achieved with traditional RAID. Flash arrays support T10PI data assurance technology. During the volume group creation process, the arrays also use the immediate availability format (IAF) to initialize disks. When the T10PI technology is integrated with the IAF disk initialization method, initialization cycles might last longer than 24 hours.

To create a volume group, storage administrators should consider the following:

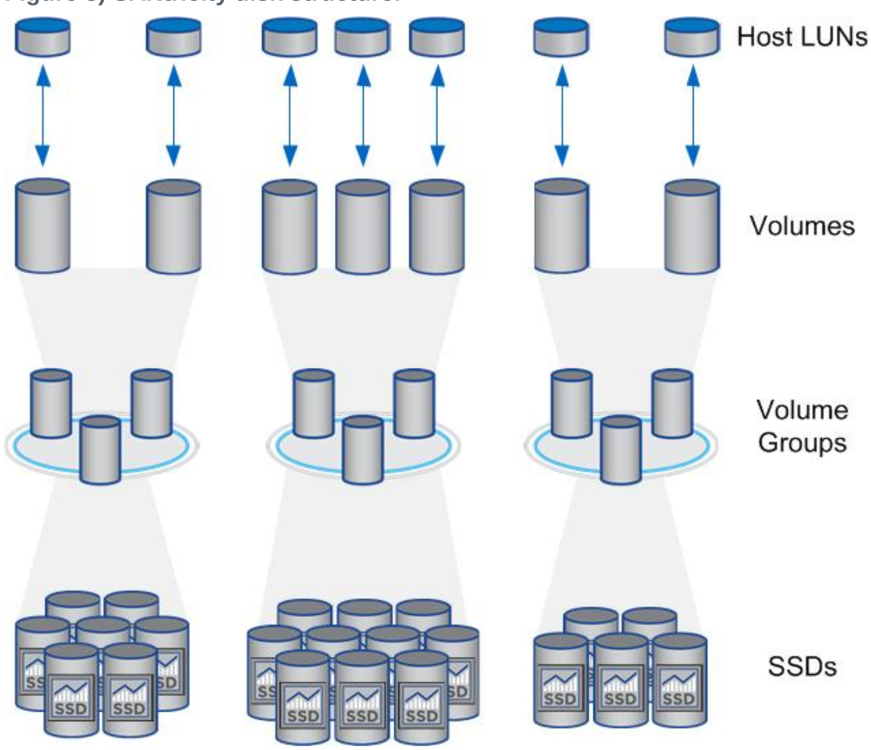
- The reconstruction time, especially for business-critical and high-availability (HA) applications
- The availability of hot-spare disks that meet the following requirements:
  - The disk type must match the disks being protected.
  - Full disk encryption (FDE)–enabled disks can protect non–FDE enabled disks; however, the reverse is not true.
  - Disk capacity must exceed the used capacity of the protected disks.
  - Hot spares are global; protection is extended to all assigned disks in the array regardless of volume group assignments.
  - There must be a sufficient quantity of spare disks to protect multiple volume groups based on the business-critical nature of the groups.

For more information about setting up volume groups, refer to the SANtricity online help documentation.

2.4 EF-Series Volume

As shown in Figure 3, a volume is the logical storage entity created for a host to access disks on the storage array. A volume is created from the capacity available on a volume group. Although a volume might include more than one drive, a volume appears as one logical entity to the host. The volume is presented to the host as disk capacity in the form of a LUN.

Figure 3) SANtricity disk structure.



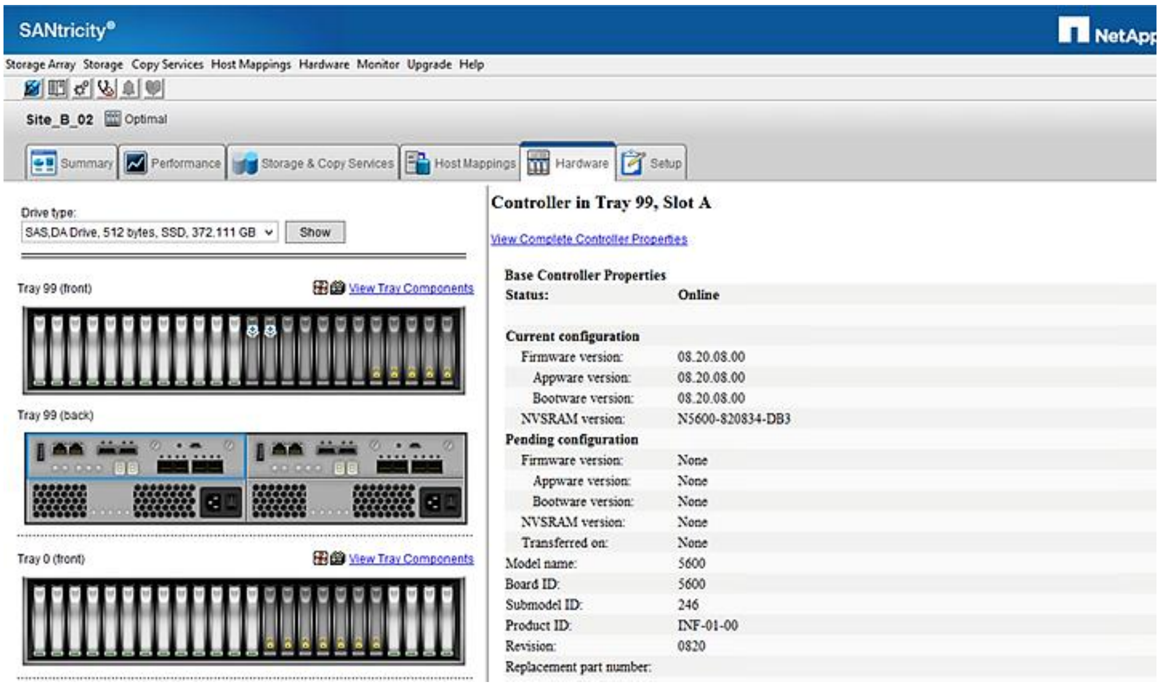
3 Provisioning the EF-Series

3.1 Provision EF-Series Storage Using SANtricity Storage Manager GUI

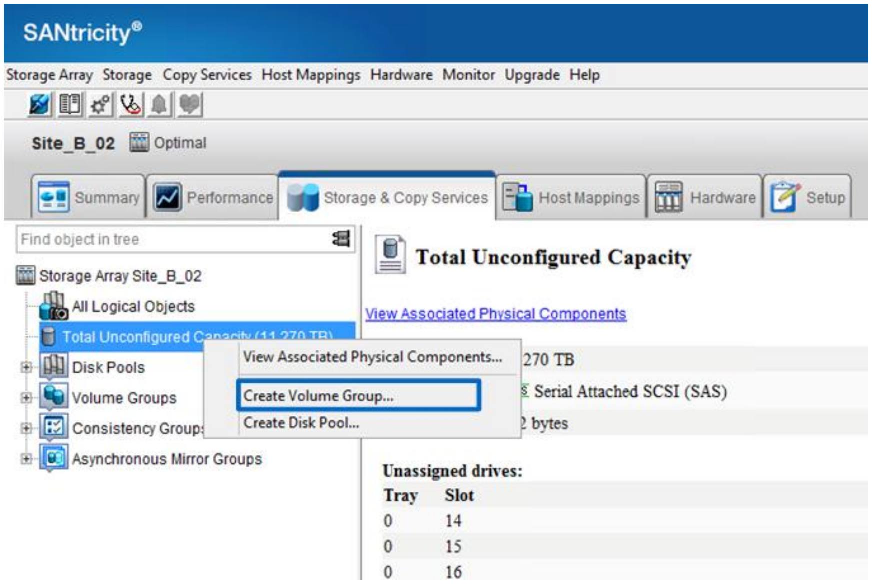
To create a volume group using SANtricity from unconfigured capacity in the storage system, complete the following steps:

1. In the Array Management Window (AMW), click the Hardware tab and verify that the required number of hot spare drives has been allocated. In the example screenshot, two are shown.

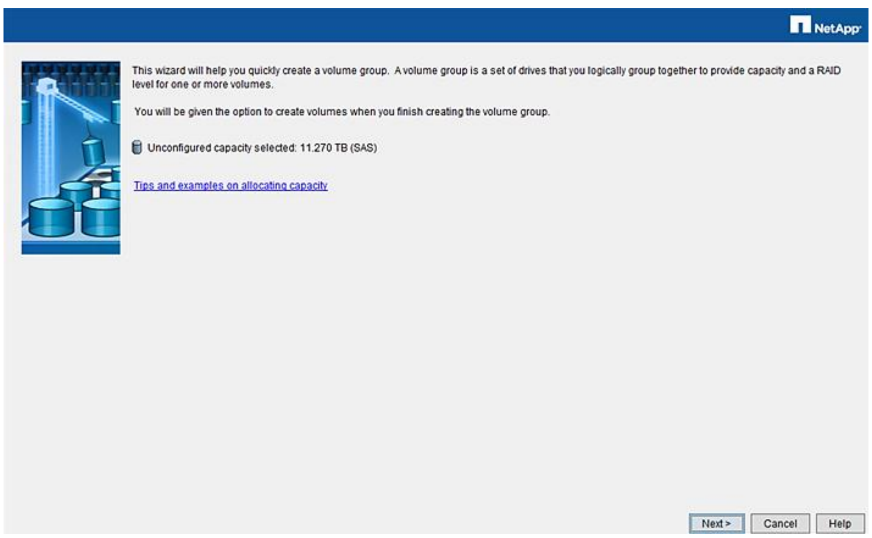
   Note: For more information about configuring hot spare drives, refer to the SANtricity online help documentation, “Using Hot Spare Drives.”



2. From the AMW, click the Storage & Copy Services tab, right-click Total Unconfigured Capacity, and select Create Volume Group.



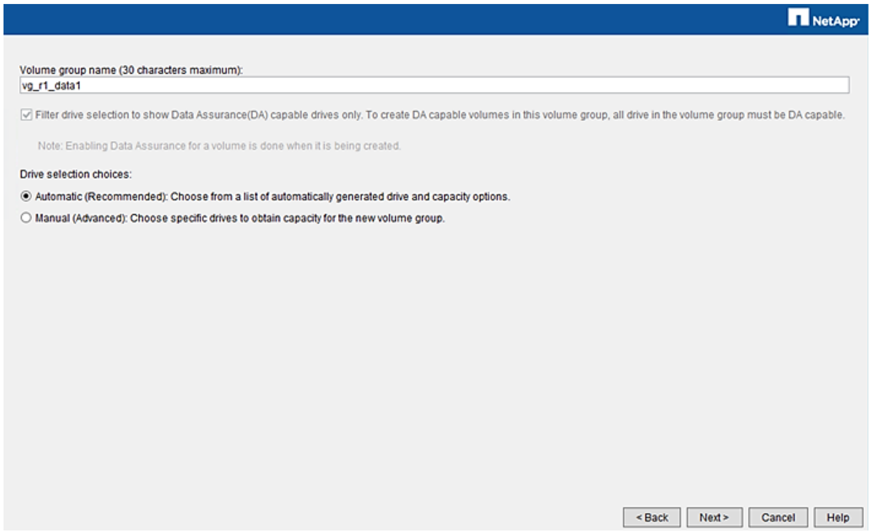
3. On the Welcome page, click Next.



4. Enter a volume group name that will aid in managing the environment over time.

   Note: Volume group names must not exceed 30 characters and cannot contain spaces. The name may contain letters, numbers, underscores (_), dashes (-), and pound signs (#).

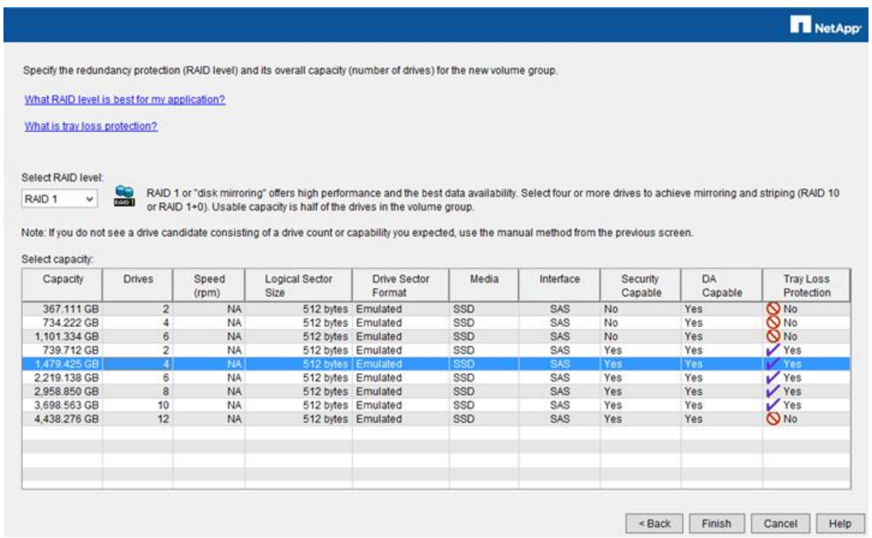
5. To create a volume group automatically, select Automatic (Recommended) and click Next.



6. Select the desired RAID level from the drop-down list. For database and tempdb files, NetApp recommends using RAID 10.

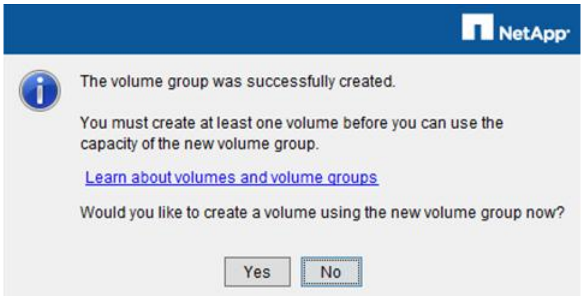
   Note: RAID 1 or “disk mirroring” offers high performance and the best data availability. Select four or more drives to achieve mirroring and striping (RAID 10 or RAID 1+0). The usable capacity is half of the drives in the volume group.

7. Select the desired volume group configuration from the list of available configurations and click Finish.



8. The volume group wizard prompts you to create a new volume. To create a volume immediately, click Yes to continue with the volume creation wizard.

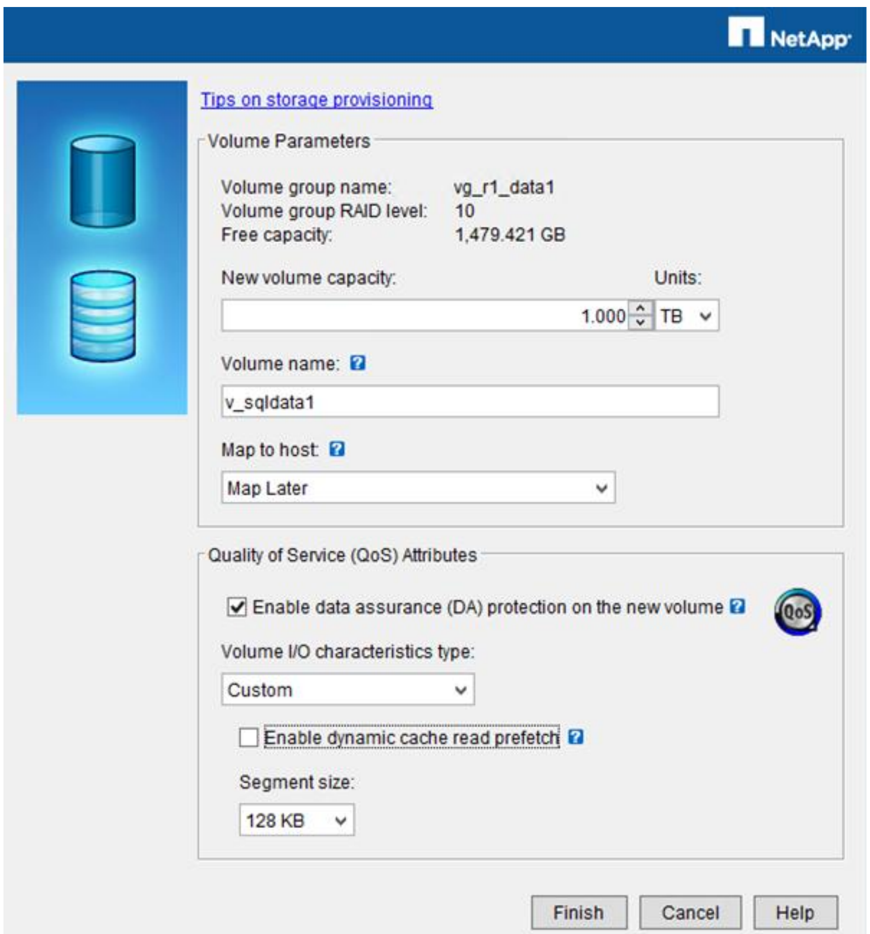
   Note: At least one volume must be created before the storage resource can be mapped to a host.



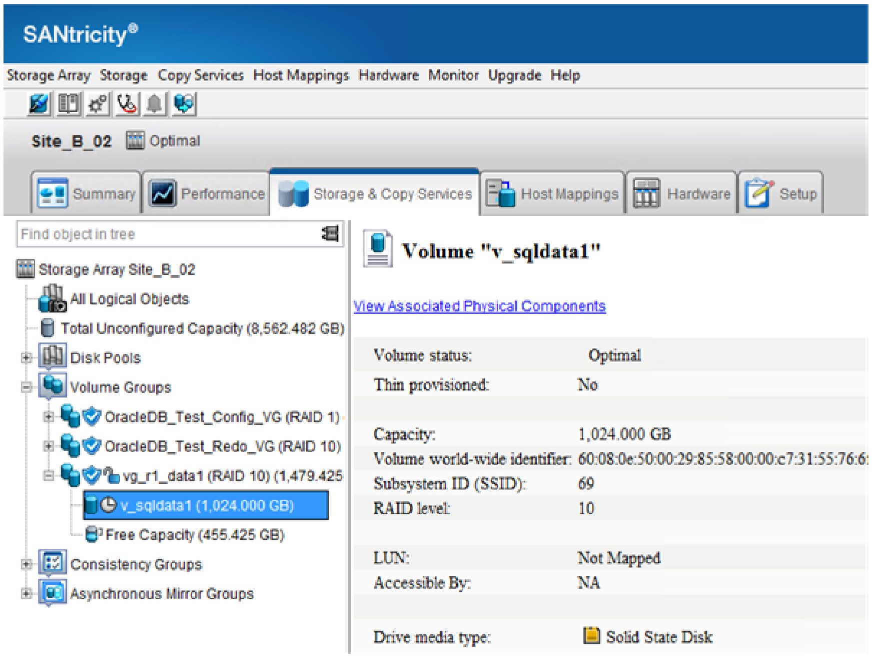
9. To create the volume, complete the following steps:
   a. Enter the new volume capacity from the available capacity in the new volume group.
   b. Enter a new volume name.

   Note: Volume names must not exceed 30 characters and cannot contain spaces. Names may contain letters, numbers, underscores (_), dashes (-), and pound signs (#).

   c. From the Map to Host drop-down list, select either Map Later or a predefined host group or host.
   d. For databases using SSD drives, select Custom from the Volume I/O Characteristics Type drop-down list.
   e. Deselect the Enable Dynamic Cache Read Prefetch checkbox for databases using SSD drives.
   f. For OLTP databases, select 128KB for the segment size. For DSS databases, select 256KB.
   g. Click Finish to create the new volume.



10. From the Storage & Copy Services tab, confirm that the new volume group is displayed in the storage system tree and that the new volume is branching from the new volume group.

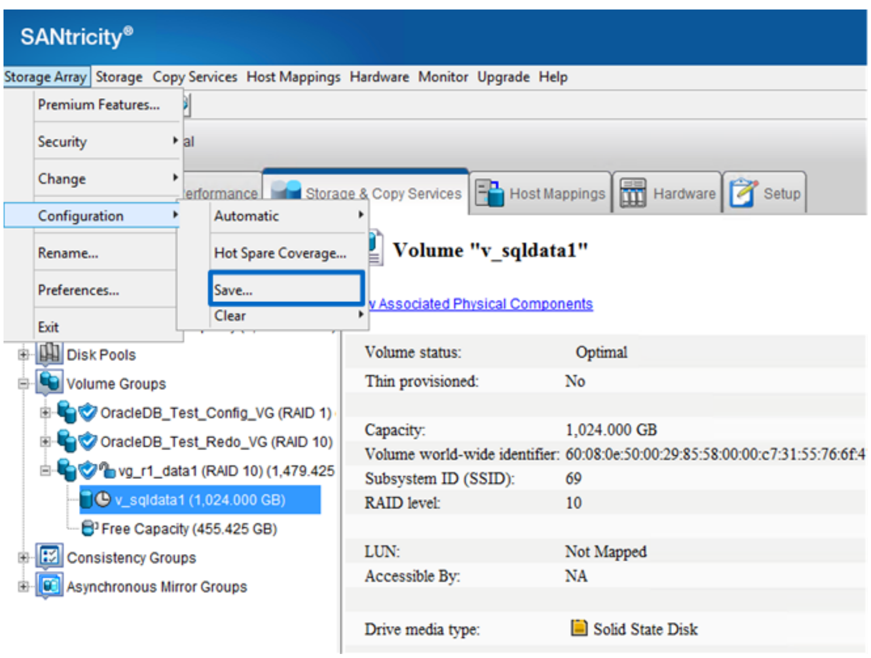


3.2 Provision EF-Series Storage Using SANtricity Storage Manager CLI

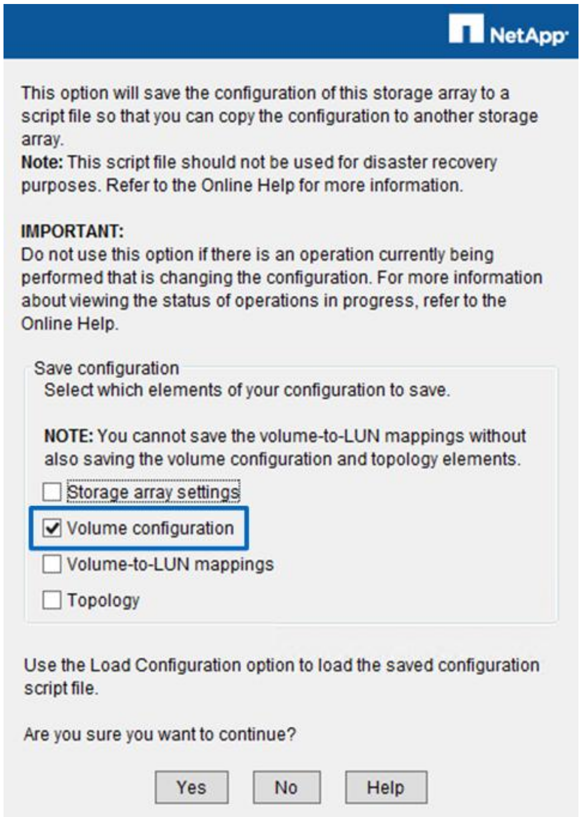
The creation of the volume group and volume shown in section 3.1, "Provision EF-Series Storage Using SANtricity Storage Manager GUI," can also be scripted and run from SANtricity.

To create a script of the storage objects using SANtricity, complete the following steps:

1. From the AMW, select the Storage Array menu and click Configuration > Save.



2. Select the volume configuration to save and click Yes.



3. Follow the Windows prompts to save the configuration file to a location of your choice.
4. After the storage configuration is saved, edit it to recreate the same configuration on multiple storage arrays as desired.

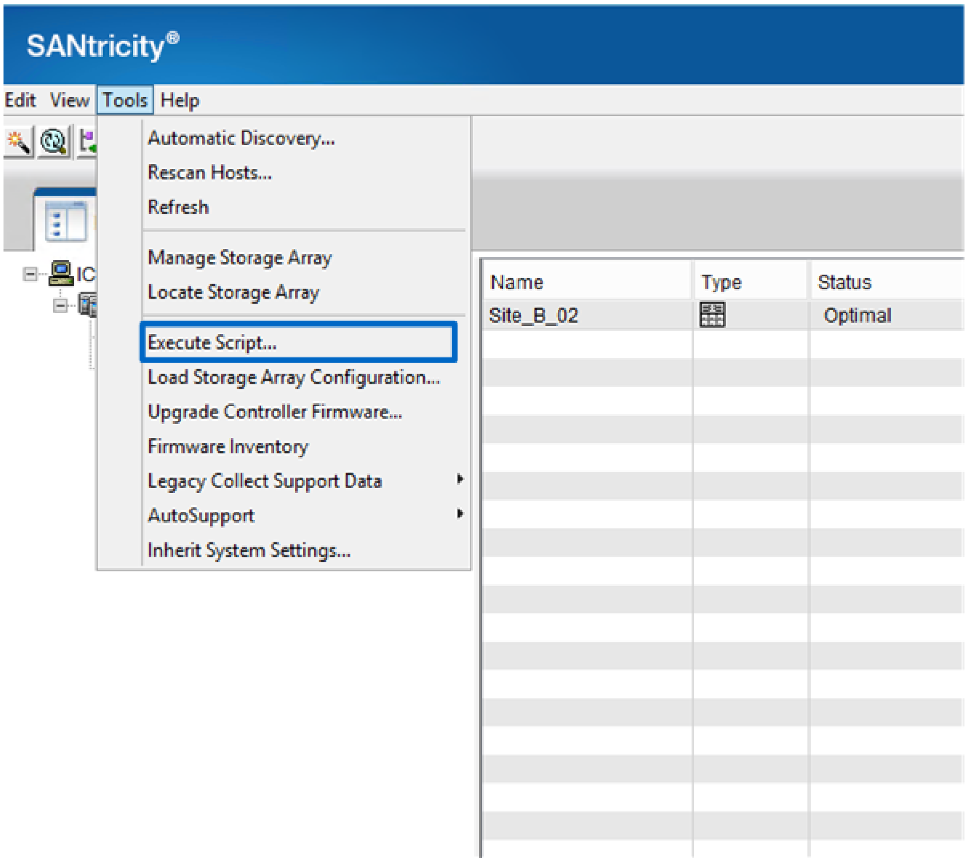
   Note: Verify that the volume group drives are the same as in the original array or modify them as needed.

5. The following script was generated using the functionality in SANtricity to save a storage configuration in a text file.

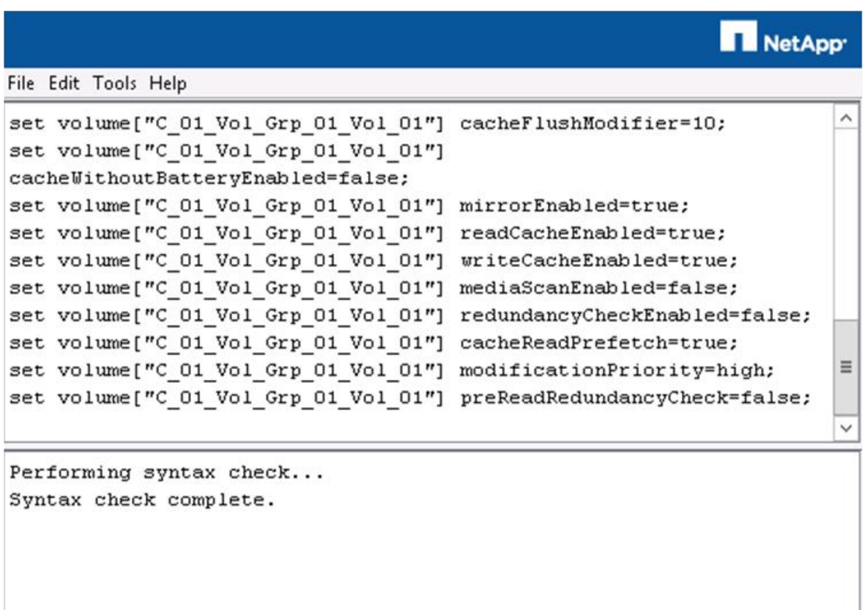
```
// Logical configuration information from Storage Array Site_C_01.
// Saved on December 9, 2013
// Firmware package version for Storage Array Site_C_01 = 98.10.02.01

show "Creating Volume Group C_01_Vol_Grp_01, RAID 5.";
//This command creates volume group <C_01_Vol_Grp_01>.
create volumeGroup drives=(99,2,5 99,3,6 99,4,5 99,5,6 99,2,6 99,3,7 99,4,6 99,5,7 99,2,7)
RAIDLevel=5 userLabel="C_01_Vol_Grp_01" securityType=capable  dataAssurance=none;

show "Creating volume C_01_Vol_Grp_01_Vol_01 on volume group C_01_Vol_Grp_01.";
//This command creates volume <C_01_Vol_Grp_01_Vol_01> on volume group <C_01_Vol_Grp_01>.
create volume volumeGroup="C_01_Vol_Grp_01" userLabel="C_01_Vol_Grp_01_Vol_01" owner=A
capacity=107374182400 Bytes segmentSize=512 dssPreAllocate=true dataAssurance=enabled
mapping=none;
show "Setting additional attributes for volume C_01_Vol_Grp_01_Vol_01.";
// Configuration settings that can not be set during Volume creation.
set volume["C_01_Vol_Grp_01_Vol_01"] cacheFlushModifier=10;
set volume["C_01_Vol_Grp_01_Vol_01"] cacheWithoutBatteryEnabled=false;
set volume["C_01_Vol_Grp_01_Vol_01"] mirrorEnabled=true;
set volume["C_01_Vol_Grp_01_Vol_01"] readCacheEnabled=true;
set volume["C_01_Vol_Grp_01_Vol_01"] writeCacheEnabled=true;
set volume["C_01_Vol_Grp_01_Vol_01"] mediaScanEnabled=false;
set volume["C_01_Vol_Grp_01_Vol_01"] redundancyCheckEnabled=false;
set volume["C_01_Vol_Grp_01_Vol_01"] cacheReadPrefetch=true;
```

```
set volume["C_01_Vol_Grp_01_Vol_01"] modificationPriority=high;
set volume["C_01_Vol_Grp_01_Vol_01"] preReadRedundancyCheck=false;
```

6. From the Enterprise Management window, click Tools > Execute Script.



7. Paste the script shown in step 5 into the Script Editor and verify the syntax before executing it.



8. As shown in section 3.1, from the Storage & Copy Services tab, confirm that the new volume group is displayed in the storage system tree. Also confirm that the new volume branches from the new volume group.

3.3 Windows Volume Mount Points

NetApp storage solutions and Microsoft SQL Server 2005 and later support mount points. Mount points are directories on a volume that can be used to mount a different volume. Mounted volumes can be accessed by referencing the path of the mount point. Mount points eliminate the Windows 26-drive-letter limit and offer greater application transparency when moving data between LUNs, moving LUNs between hosts, and unmounting and mounting LUNs on the same host. They can do this because the underlying volumes can be moved around without changing the mount point path name.

NetApp recommends using NTFS mount points instead of drive letters to surpass the 26-drive-letter limitation in Windows. When using volume mount points, the name given to the volume label and mount point must be the same.

4 Provisioning SQL Server 2014

4.1 SQL Server Database Files

Provisioning database files on the EF-Series flash array can be done in two ways:

- To create a database that has database files residing on an EF-Series LUN, use the following T-SQL script during database creation:

```
-- Assuming C:\MSSQL\Data and C:\MSSQL\Log is the mount points of EF-Series LUNs

USE master;
GO
CREATE DATABASE Sales
ON
( NAME = Sales_dat,
 FILENAME = 'C:\MSSQL\Data\saledat.mdf',
 SIZE = 10,
 MAXSIZE = 50,
 FILEGROWTH = 5 )
LOG ON
( NAME = Sales_log,
 FILENAME = 'C:\MSSQL\Log\salelog.ldf',
 SIZE = 5MB,
 MAXSIZE = 25MB,
 FILEGROWTH = 5MB ) ;
GO
```

- To move the database files from non EF-Series LUNs to EF-Series LUNs, stop the SQL Server service and detach the database files. After you detach the databases, copy the files to the path or mount points that reside in the EF-Series LUNs. After the files are copied, attach the database from the new location.

The common best practice is to separate data, transaction logs, and tempdb files in separate logical LUNs. The origin of this recommendation lies with the separation of the types of workloads between different physical storages. This is still a valid recommendation for environments in which you can guarantee that separation. However, it is common to see customers deploy SQL Server in a shared storage environment in which physical separation is much harder to achieve and usually not necessary for performance reasons.

It is still a good idea to maintain separation to help with manageability so that potential problems are easier to isolate. For example, separating tempdb onto its own logical disk means that you can presize it to fill the disk without worrying about space requirements for other files. The more separation you implement, the easier it is to correlate logical disk performance to specific database files.

4.2 Database Files for tempdb

The tempdb system database is a global resource that is available to all users connected to the SQL Server instance, and it is used to hold the following:

- Temporary user objects that are explicitly created, such as global or local temporary tables, temporary stored procedures, table variables, or cursors
- Internal objects that are created by the SQL Server Database Engine, such as work tables to store intermediate results for spools or sorting
- Row versions that are generated by data modification transactions in a database that uses read-committed row versioning isolation or Snapshot isolation transactions
- Row versions that are generated by data modification transactions for features, such as online index operations, multiple active result sets, and AFTER triggers

Operations within tempdb are minimally logged, enabling transactions to be rolled back. Tempdb is recreated every time SQL Server is started so that the system starts with a clean copy of the tempdb database. Temporary tables and stored procedures are dropped automatically on disconnect, and no connections are active when the system is shut down. Therefore, there is nothing in tempdb to be saved from one session of SQL Server to another. Backup and restore operations are not allowed on tempdb.

Every SQL Server instance has a shared database named tempdb that is used by temporary objects and there is only one tempdb per instance. Therefore it often causes a bottleneck for systems that use it heavily. Typically, this happens because of PAGELATCH, an in-memory latch contention on the allocation bitmap pages inside the data files.

It is possible to reduce the contention on the in-memory pages by adding additional data files to tempdb with the same initial size and autogrowth configuration. This works because SQL Server uses a round-robin, proportional-fill algorithm to stripe the writes across the data files. When multiple data files exist for a database, all of the writes to the files are striped to those files. The writes to any particular file are based on the proportion of the file's free space to the total free space across all of the files. This means that writes are proportionally distributed to the files according to their free space, regardless of their size, so that they fill at the same time.

Microsoft recommends up to a one-to-one mapping between the number of files and logical CPUs. During testing of massive workloads, Microsoft has seen performance benefits, even with hundreds of data files.

A more pragmatic approach, however, is to have a one-to-one mapping between files and logical CPUs up to eight. Then add files if you continue to see allocation contention or if you must push the I/O subsystem harder.

Because the NetApp EF-Series flash array provides a robust platform for delivering exceptional performance, you can create multiple tempdb files and place them on the EF-Series flash array. You can accomplish this by using the following Transact SQL script.

```
select  *
from    sys.database_files

use master
go
-- Change logical tempdb file name first since SQL Server shipped with logical file name called
tempdev
alter database tempdb modify file (name = 'tempdev', newname = 'tempdev01');

-- Change location of tempdev01 and log file to C:\MSSQL\Tempdb path
alter database tempdb modify file (name = 'tempdev01', filename =
'C:\MSSQL\Tempdb\tempdev01.mdf');
alter database tempdb modify file (name = 'templog', filename = 'C:\MSSQL\Tempdb\templog.ldf');

-- Assign proper size for tempdev01
ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'tempdev01', SIZE = 2GB, FILEGROWTH = 100 MB );
ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'templog', SIZE = 4GB, FILEGROWTH = 100 MB );
```



```
-- Add more tempdb files
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev02', FILENAME =
N'C:\MSSQL\Tempdb\tempdev02.ndf' , SIZE = 2GB , FILEGROWTH = 100 MB);
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev03', FILENAME =
N'C:\MSSQL\Tempdb\tempdev03.ndf' , SIZE = 2GB , FILEGROWTH = 100 MB);
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev04', FILENAME =
N'C:\MSSQL\Tempdb\tempdev04.ndf' , SIZE = 2GB , FILEGROWTH = 100 MB);
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev05', FILENAME =
N'C:\MSSQL\Tempdb\tempdev05.ndf' , SIZE = 2GB , FILEGROWTH = 100 MB);
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev06', FILENAME =
N'C:\MSSQL\Tempdb\tempdev06.ndf' , SIZE = 2GB , FILEGROWTH = 100 MB);
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev07', FILENAME =
N'C:\MSSQL\Tempdb\tempdev07.ndf' , SIZE = 2GB , FILEGROWTH = 100 MB);
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev08', FILENAME =
N'C:\MSSQL\Tempdb\tempdev08.ndf' , SIZE = 2GB , FILEGROWTH = 100 MB);
```

5 SQL Server Performance on EF-Series

The EF560 flash array and SQL Server 2014 were tested in an OLTP database configuration using the open source database load testing tool HammerDB. Table 2 lists the components used in the test.

Table 2) Configuration components.

Component	Details
Storage	• Controller: 2 x EF560, 12GB cache each • Firmware: SANtricity 8.20.08 • 24 x 400GB SSD drives • RAID 10/DDP: Each RAID type used all 24 drives • For each database server: 10 x 100GB volumes created (8x data, 1x log, and 1x temp database)
FC switch	Brocade 6505
SQL Server database servers	• 4x Dell PowerEdge R720 • Memory on server: 80GB • Processors/cores: 2 processors with 6 cores each • 2x dual-port QLogic FC host bus adapter • Microsoft Windows Server 2012 R2 Datacenter • Microsoft SQL Server 2014 (64-bit) • HammerDB, TPC-C OLTP Workload Generator

5.1 Database Performance

To fully load the EF560, four Dell R720 servers were used, each running an OLTP database created by HammerDB and the Hammer DB client. All performance data was captured at the host by using the Microsoft Performance Monitoring tool. The results from each host were then averaged. Testing with HammerDB provided a database with an 8K block size and approximately an 80/20 read/write ratio.

For a typical SQL Server OLTP database running on an EF560, the RAID level chosen for the volume group has a significant effect on performance. Figure 4 and Figure 5 show performance for RAID 10 and DDP testing, respectively.

For the highest performance, choose RAID 10. As shown in Figure 5, we sustained latencies below 0.5ms while delivering approximately 350,000 IOPS. Further, while still delivering submillisecond latency, a workload of 4,800 users, with each user running 1,000,000 transactions, exceeded 400,000 IOPS.

The results for DDP are also impressive on the EF560, as shown in Figure 5. Over a large workload range, the latency stays extremely consistent between 0.6ms and 0.7ms while delivering over 330,000 IOPS at the highest workloads.

Figure 4) RAID 10 OLTP test results.

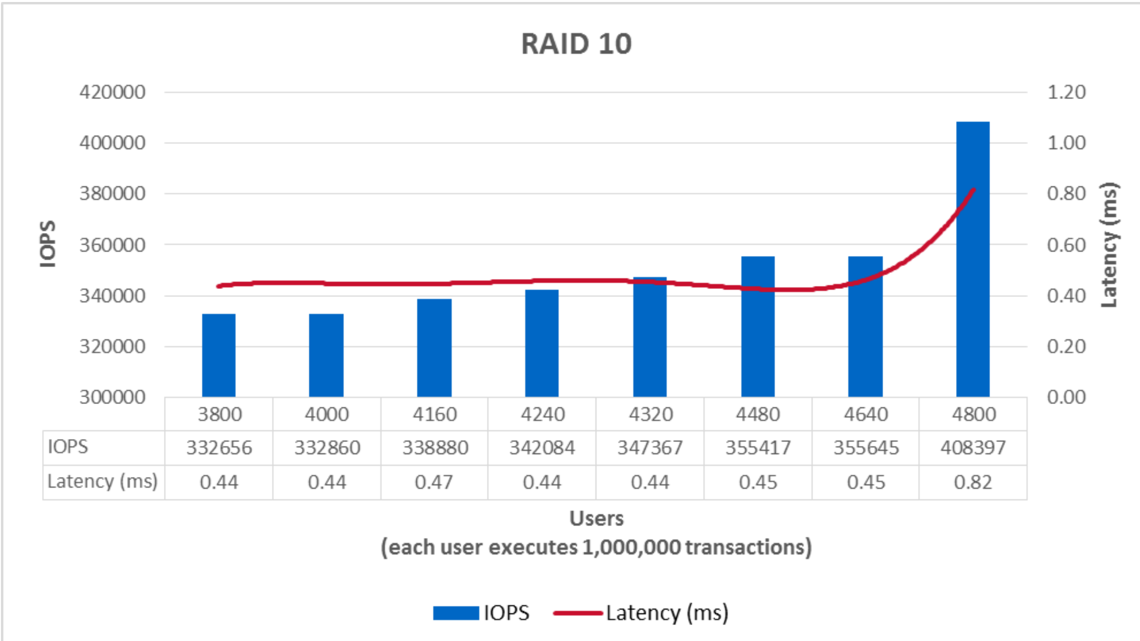
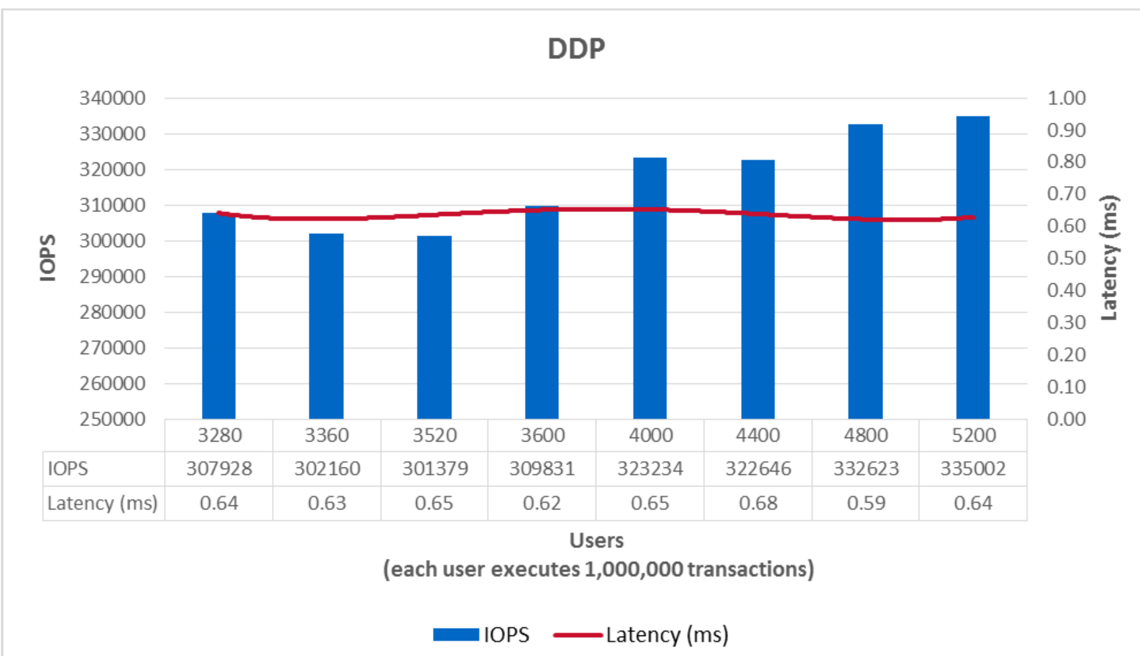


Figure 5) DDP OLTP test results.



To achieve maximum performance, NetApp recommends using RAID 10 for all volume groups. If space is limited, store application data files and tempdb data files on RAID 10 and store volume groups and database log files on a DDP.

5.2 EF-Series Replication Performance

The E-Series and EF-Series synchronous mirroring and asynchronous mirroring features are for online, real-time replication of data between two storage arrays in separate locations. With synchronous mirroring or asynchronous mirroring, mirrored pairs are created to copy data from one storage array to another storage array. This action protects against local or wide-area disasters or centrally backs up your data at a remote location.

The mirrored volume pair is created from two volumes that are logical structures created on a storage array for data storage. The pair consists of a primary volume at a local storage array and a secondary volume at a remote storage array.

If a disaster occurs, or if there is a catastrophic failure in the local storage array, the secondary volume can be promoted in the remote storage array to the role of primary volume. This action enables it to take over maintenance of computer operations. For more information, refer to the NetApp E-Series Storage Systems Mirroring Feature Guide.

Two EF560 arrays were used for replication testing. Each array was configured with 12 drives in a DDP to provide both replication performance data and a base with no replication. One Dell R720 server with one SQL Server database configured as previously described was deployed for each array. All database data files and the log file were replicated. The tempdb was not replicated. Both synchronous and asynchronous replication performance was tested. All performance was measured at the array using SANtricity Storage Management 11.20. HammerDB was employed to provide the effects of an OLTP database. SQL Server memory on the primary server was constrained to 1024GB to force I/O to the array.

Synchronous Replication

Synchronous mirroring is a continuous remote replication method for mirrored volumes. Any host write request is written to the primary volume and then copied to the secondary volume. The controller sends an I/O completion indication to the host system after the copy has been successfully completed.

Table 3, Figure 6, and Figure 7 show the performance effects of the tested EF560 when using synchronous mirroring. Multiple tests were carried out to provide a range for the performance impact.

Table 3) Synchronous replication effect on performance (measured at the array).

Performance	Primary Node Effect*
IOPS	Decrease by 2%–8%
Latency	Increase by 33%–54%
Throughput	Decrease by 2%–19%

*As compared to non EF-Series synchronous replication.

Figure 6) Latency compared with synchronous replication on and off.

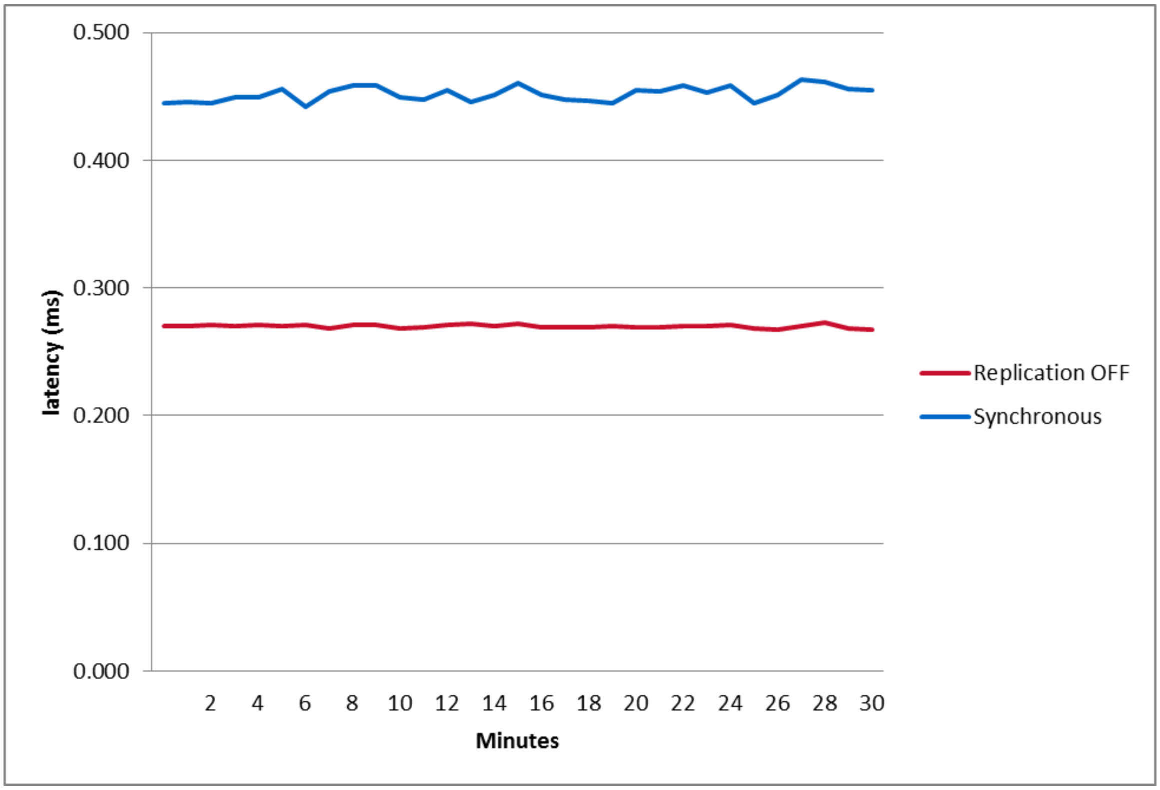
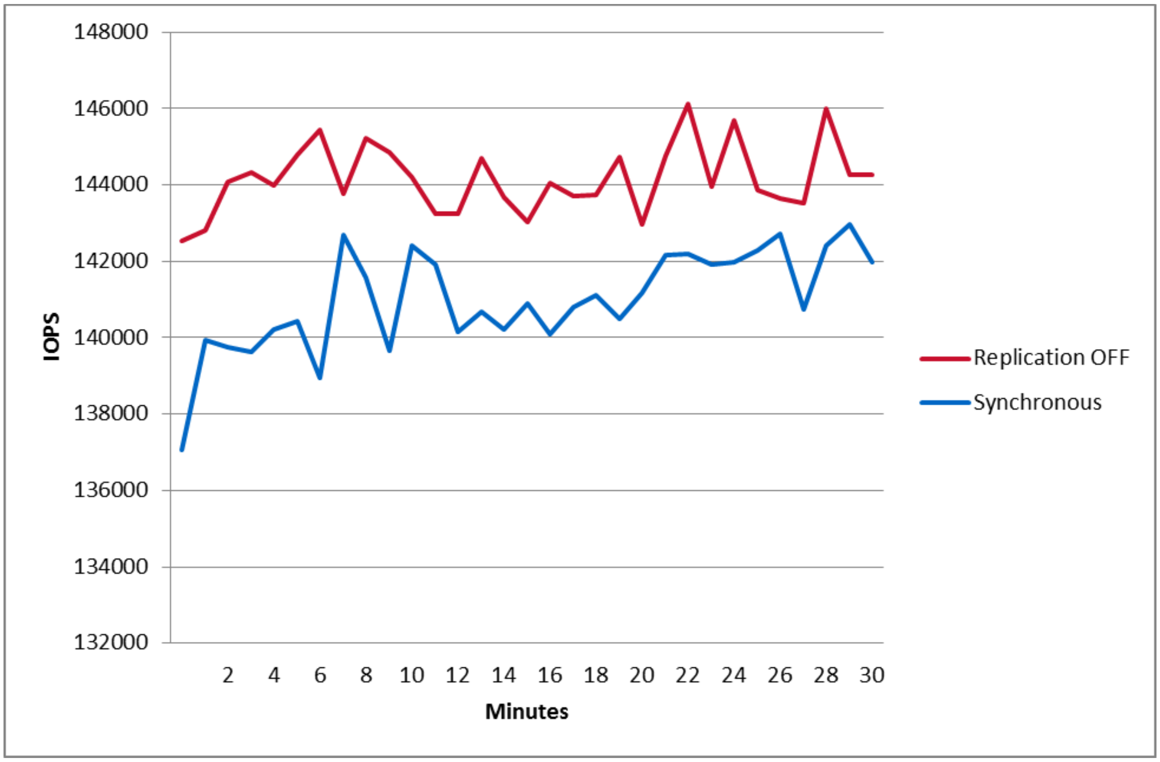


Figure 7) IOPS compared with synchronous replication on and off.



Asynchronous Replication

Asynchronous mirroring is a remote point-in-time replication method that captures the differences between the current point-in-time image and the previous one and copies the differences to the remote storage array. These differences are written to a repository in between regular synchronizations.

An asynchronous mirror group is a container that can house several mirrored pairs that are synchronized as a coordinated dataset to provide a consistent dataset at the remote site. The asynchronous mirror group is associated with a local storage array and a remote storage array that are both used for mirroring.

Because changed data is written to a repository until replication occurs at a defined interval, there is an ongoing effect on performance, as shown in Table 4. Multiple tests were carried out to provide a range for the performance effects.

Note: As with synchronous replication, the greatest effect is on latency.

Synchronization occurs at regular intervals every 10 minutes at a minimum. Latency spikes and then returns to its previous level, as shown in Figure 8. IOPS, as shown in Figure 9, and throughput, which is not shown, also experience brief drops in performance.

Note: The time it takes to synchronize data in any environment depends on the amount of data.

Table 4) Asynchronous replication effect on performance (measured at the array) when not synchronizing.

Performance	Primary Node Effect*
IOPS	Decrease by 0%–2%
Latency	Increase by 15%–44%
Throughput	Decrease by 0%–2%

*As compared to non EF-Series asynchronous replication.

Figure 8) Latency compared with asynchronous replication on and off.

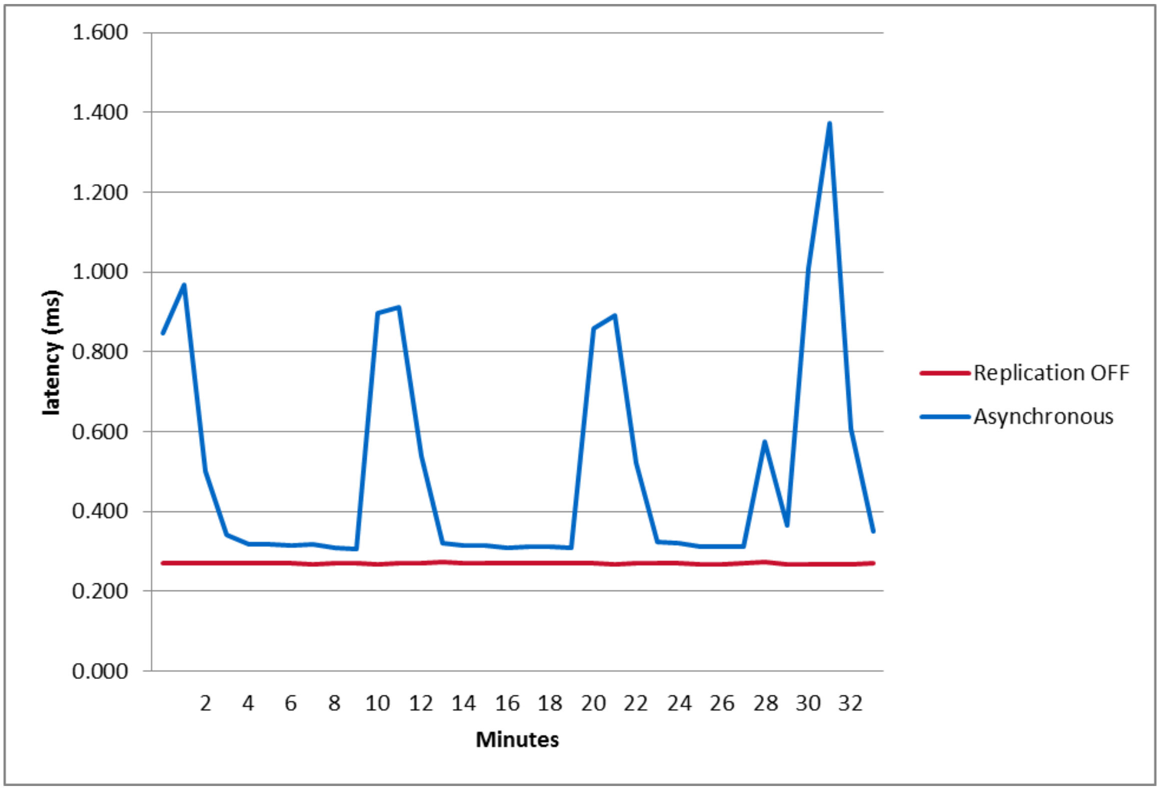
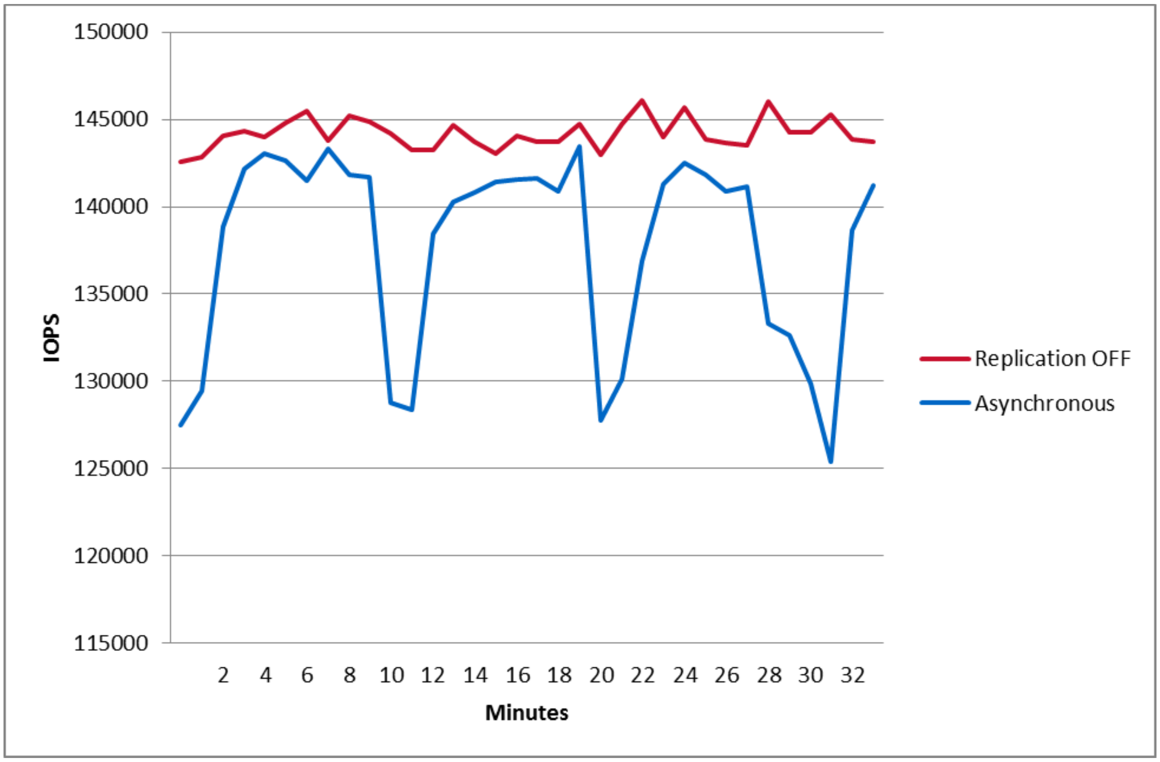


Figure 9) IOPS compared with asynchronous replication on and off.



6 High Availability

6.1 EF-Series and SANtricity Storage Manager

The NetApp EF-Series storage system has been architected for high reliability and HA with features such as:

- Dual active controller with automated I/O path failover
- RAID levels 0, 1, 5, 6, and 10 or DDP
- Redundant, hot-swappable controllers, disk drives, power supplies, and fans
- Automatic drive failover detection and rebuild using global hot spares
- Mirrored data cache with battery backup and destage to memory
- Nondisruptive controller firmware upgrades
- Proactive drive health monitoring
- Background media scan with autoparity check and correction

All components are fully redundant and may be swapped without powering off the system or even halting operation. This includes controllers, disk drives, power supplies, and fans. The EF-Series power supplies offer an 80-plus efficiency rating. The EF-Series flash array features several functions designed to protect data in every circumstance. Multiple RAID levels are available for use with varying levels of redundancy. Failover from one path to another in the case of a lost connection is also automatically included with the system. Within the shelf, each drive has a connection to each controller so that even internal connection issues can be quickly overcome. Volumes on the system are available for host I/O from the moment they are created and can even have significant properties altered without stopping I/O.

Other features of the EF-Series flash array that protect data include mirroring and backing up controller cache. If power is lost to the system during operation, on-board batteries destage the data from cache memory to internal controller flash so that it is available when power is restored. The RAID algorithms allow the system to recreate any lost data in the rare case of drive failure. You also have the option to confirm data with RAID parity at all times and even continue a rebuild if you hit an unreadable sector.

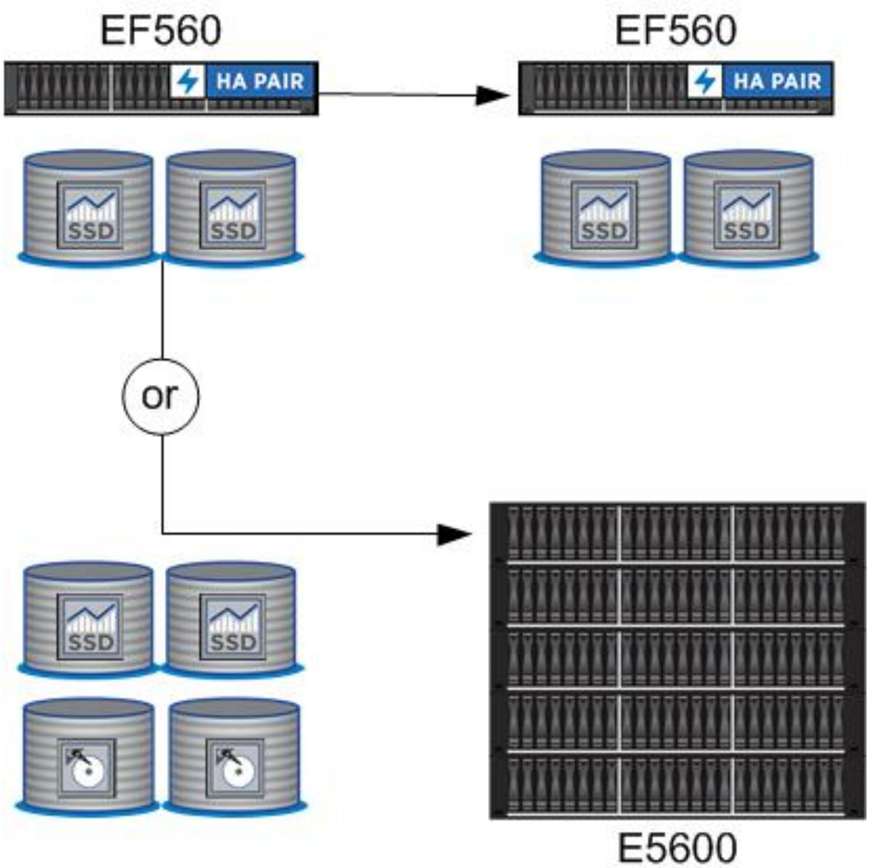
Behind the scenes, the system performs other tasks that protect data at all times. The optional media scan feature looks for inconsistencies even on sectors not currently accessed by any host. The EF-Series proactively tracks SSD wear and flags drives that are approaching the end of their expected life. All types of diagnostic data are constantly collected for use later by support, if necessary.

Not only does the EF-Series offer the detailed reliability and availability features already described, but by using the SANtricity software, it is also possible to maximize availability. For example, SANtricity software:

- Enables high-speed, high-efficiency Snapshot technology
- Protects data in seconds
- Reduces flash consumption by storing only changed blocks
- Provides robust disaster recovery (DR) protection
- Supports synchronous mirroring for no-data-loss protection of content
- Supports asynchronous mirroring for long-distance protection and compliance
- Maximizes ROI with flexible protection
- Supports flash, NL-SAS, or a mix of recovery targets based on cost and performance needs
- Delivers speed without breaking budgets

Figure 10 shows HA possibilities using Snapshot technology and mirroring. For more information, refer to the Documentation library on the NetApp Support site and SANtricity online help.

Figure 10) EF-Series HA using Snapshot technology and mirroring.



Using E-/EF-Series Mirroring and Snapshot Copies

It is possible to use either synchronous or asynchronous replication to a secondary E-/EF-Series system for disaster recovery. The database on the secondary site will not be available for direct use while the replication mirror is in place. After the mirror is broken, the secondary database volumes can be mounted and the database brought online.

To use the secondary site, create a point-in-time Snapshot image of the secondary volumes and then create Snapshot volumes from this image. These Snapshot volumes can then be mounted for read/write access so the database can be accessed. A repository is required when a Snapshot volume is mounted read/write, because EF-Series Snapshot copies use a copy-on-write technology. For more information, refer to the Documentation library on the NetApp Support site and SANtricity online help.

If desired, a clone of the database can be created from the Snapshot volume by using the SANtricity volume copy feature. This feature creates a byte-by-byte copy of the available volumes for purposes such as those related to a development environment.

6.2 SQL Server HA Options

The high availability of SQL Server solutions masks the effects of hardware or software failures and maintains the availability of applications to minimize the perceived downtime for users. SQL Server provides several HA solutions.

Log Shipping

Log shipping operates at the database level. It can maintain one or more warm standby databases (referred to as secondary databases) for a single production database that is referred to as the primary database. For more information about log shipping, refer to About Log Shipping (SQL Server).

Database Mirroring

Database mirroring increases database availability by supporting almost instantaneous failover. Database mirroring can be used to maintain a single standby database or mirror database for a corresponding production database that is referred to as the principal database. For more information, refer to Database Mirroring (SQL Server).

AlwaysOn Failover Cluster Instances

AlwaysOn Failover Cluster Instances leverage Windows Server Failover Clustering (WSFC) functionality to provide local HA through redundancy at the server instance level: a failover cluster instance (FCI). An FCI is a single SQL Server instance that is installed across WSFC nodes and possibly across multiple subnets. On the network, an FCI appears to be an SQL Server instance running on a single computer. However, the FCI provides failover from one WSFC node to another if the current node becomes unavailable. For more information, refer to AlwaysOn Failover Cluster Instances (SQL Server).

AlwaysOn Availability Groups

AlwaysOn Availability Groups are an enterprise-level high-availability and DR solution introduced in SQL Server 2012 to enable you to maximize availability for one or more user databases. AlwaysOn Availability Groups require that SQL Server instances reside on WSFC nodes. For more information, refer to AlwaysOn Availability Groups (SQL Server).

AlwaysOn Availability Groups support two availability modes: asynchronous-commit mode and synchronous-commit mode:

- Asynchronous-commit mode is a DR solution that works well when the availability replicas are distributed over considerable distances. For more information, refer to Asynchronous-Commit Availability Mode.
- Synchronous-commit mode emphasizes HA over performance, at the cost of increased transaction latency. In synchronous-commit mode, transactions wait to send the transaction confirmation to the client until the secondary replica has hardened the log to disk. For more information, refer to Synchronous-Commit Availability Mode.

  With synchronous-commit mode, all transactions in the primary node must wait until the fail over partner replica commits its transaction. AlwaysOn Availability Groups performance impact testing has been conducted. Table 5 summarizes the effects of AlwaysOn synchronous-commit mode configured between two physical servers with databases residing on two EF560 flash arrays.

Table 5) AlwaysOn synchronous-commit mode effect on performance.

Performance	Primary Node Effect*
IOPS	Average decrease by 30%
Latency	Average increase by 10%
Throughput	Average decrease by 23%

*As compared to non AlwaysOn Availability Groups.

The test indicated that AlwaysOn Availability Groups with near-site synchronous commit had some effect on SQL Server performance.

Automatic failover provides high availability by making sure that the database is quickly made available again after the loss of the primary replica. To configure an availability group for automatic failover, you must set both the current primary replica and one secondary replica to synchronous-commit mode with automatic failover.

AlwaysOn Availability Groups Using EF-Series and FAS Storage

NetApp FAS storage systems allow you to create Snapshot copies and easily clone databases by using NetApp SnapDrive® for Windows and NetApp SnapManager® for SQL Server (SMSQL). Figure 11 shows a sample architecture design that uses AlwaysOn Availability Groups to synchronize database transactions between EF-Series and NetApp FAS storage. Table 6 lists the level of transaction latency based on the used storage.

Figure 11) Sample architecture.

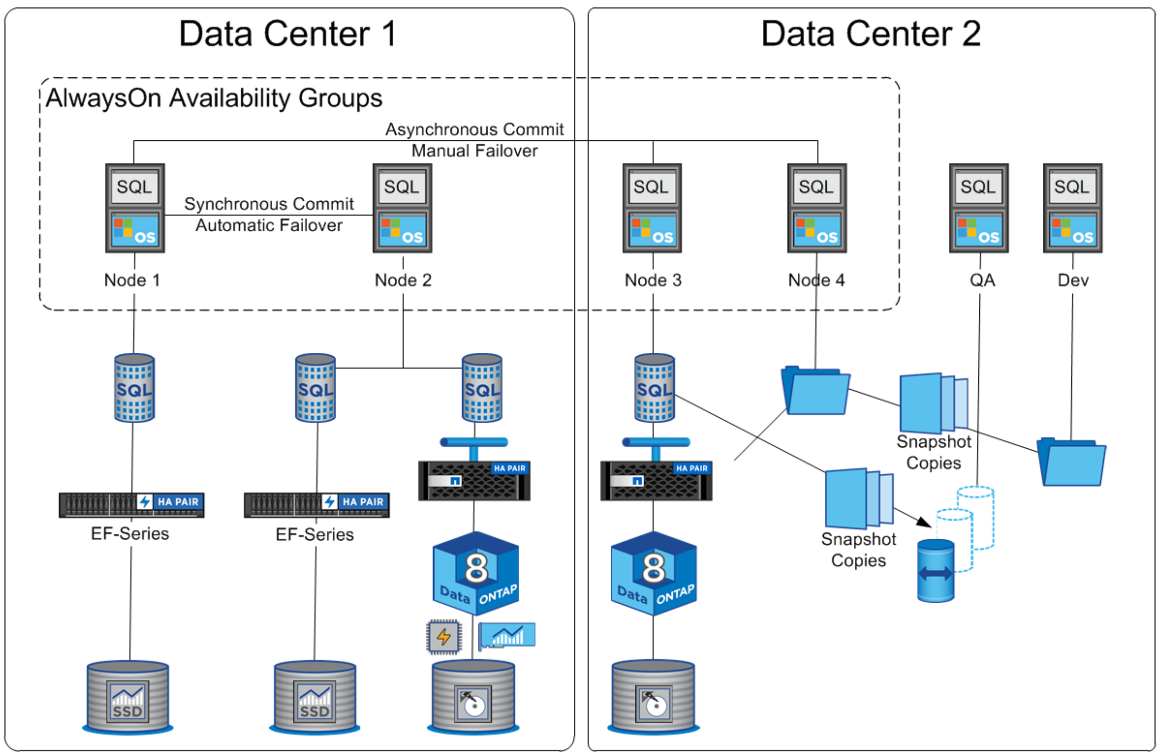


Table 6) Transaction latency versus used storage.

Relative Transaction Latency	Storage Architecture
Low	EF-Series flash array to EF-Series flash array or EF-Series flash array to FAS with NetApp Flash Cache™ or NetApp Flash Pool™ intelligent caching
Medium	EF-Series flash array to FAS SAS drives
High	EF-Series flash array to FAS SATA drives

The following requirements must be met to set up hybrid storage:

- Install SnapDrive for Windows 7.1 or later on all Windows hosts.
- Place other replicas on FAS storage and install SMSQL for backup. Clone and utilize SMSQL clone lifecycle.

- Install the NetApp clustered Data ONTAP® 8.2.1 or later operating system to use SMSQL with SMB 3.0 to enhance ease of use for developers and test engineers.

6.3 HA Comparison

Both NetApp EF-Series and Microsoft SQL Server provide multiple HA options. The EF-Series provides options at the storage level while SQL Server works at the application level. Table 7 compares the key advantages and disadvantages of each method.

Table 7) HA options compared.

HA Options	Advantages	Disadvantages
SANtricity replication	• Can be configured as synchronous or asynchronous replication • No additional cost • Available with any SQL Server edition • Does not require database full recovery mode • Asynchronous replication interval is configurable	• Medium to high effects on primary node performance • Database is not available for use without also using SANtricity Snapshot technology • Does not provide automatic failover
SQL Server AlwaysOn Availability Groups	• Can be configured as synchronous or asynchronous commit • Can be configured for automatic failover • Low effect on primary database performance • Readable secondary replica • Database can be backed up from replica node • Replica can reside on FAS storage to take advantage of clustered Data ONTAP capabilities	• Requires additional cost to purchase SQL Server Enterprise Edition • Requires database to be in full recovery mode • Can only replicate up to four nodes

In general, using HA features affects the performance of SQL Server databases. Performance of SANtricity storage-based replication and SQL Server 2014 AlwaysOn Synchronous Commit was tested on the EF560. Table 8 compares the test results.

Table 8) Comparison of performance effects on an OLTP database.

Performance	AlwaysOn Availability Groups: Synchronous Commit Primary Node Effect	EF-Series Synchronous Replication Primary Node Effect	EF-Series Asynchronous Replication Primary Node Effect
IOPS	Average decrease by 30%	Decrease by 2%–8%	Decrease by 0%–2%
Latency	Average increase by 10%	Increase by 33%–54%	Increase by 15%–44%

Performance	AlwaysOn Availability Groups: Synchronous Commit Primary Node Effect	EF-Series Synchronous Replication Primary Node Effect	EF-Series Asynchronous Replication Primary Node Effect
Throughput	Average decrease by 23%	Decrease by 2%–19%	Decrease by 0%–2%

Note: AlwaysOn Availability Groups asynchronous-commit mode was not tested. Because of the AlwaysOn architecture, we assumed the effects would be negligible.

SQL Server AlwaysOn provides better latency performance than SANtricity because it sends Data Manipulation Language or Data Definition Language to the secondary replica. However, SANtricity replicates all block changes to the secondary replica. With AlwaysOn Synchronous Commit, all transactions must be acknowledged by the secondary replica, which decreases IOPS and throughput.

7 Sizing

SQL Server performance has generally been centered on I/O. Traditionally, this performance was improved by either increasing the number of spindles or making the spindles go more quickly. With the advent of the EF-Series flash array, performance improvement can be achieved by using SSDs.

7.1 EF-Series I/O Overview

Several factors can affect the overall performance of an EF-Series storage system, including physical components, such as networking infrastructure, and the configuration of the underlying storage itself. Generically, storage system performance tuning can be defined as following a 40/30/30 rule. This rule states that 40% of tuning and configuration is at the storage system level, 30% is at the file system level, and the final 30% is at the application level. The following sections describe the 40% related to storage system specifics. At a high level, some of the considerations at the file system and application level include:

- **I/O size.** EF-Series storage systems are largely responsive systems that require a host to request an I/O operation to complete that operation. The I/O size of the individual requests from the host can have a significant effect on either the number of IOPS or throughput (generally described in terms of megabytes per second [MB/sec] or gigabytes per second [GB/sec]). Larger I/Os typically lead to lower numbers of IOPS and larger MB/sec, and the opposite is true as well. This relationship is defined with the equation Throughput = IOPS × I/O size.
- **Read versus write requests.** In addition to the I/O size, the percentage of read versus write I/O requests processed at the storage system level also has a potential effect on the storage system. This potential should be considered when designing a solution.
- **Sequentiality or randomness of the data stream.** The sequentiality (or lack thereof) of the host requests to the underlying disk media logical block addresses has a significant effect on performance at the storage system level. This effect is in terms of the physical media's capability to respond effectively to the request with minimal latency as well as the effectiveness of the storage system's caching algorithms. An exception to increased latency of random requests is for SSDs, which do not have mechanically invoked latency.
- **Number of concurrent I/O operations.** The number of outstanding I/O operations applied to a given volume can vary based on several factors, including whether the file system uses raw, buffered, or direct I/O. Generally, most volumes in an EF-Series storage system are striped across several disk drives. Providing a minimal amount of outstanding I/O to each individual disk can cause underutilization of the resources in the storage system. This can result in less than desired performance characteristics.

For those SQL Server customers new to the NetApp EF-Series, it might be helpful to review the differences between RAID 10, RAID 5, and DDP technology. For more information, refer to the E-Series Performance Sizing Guide found on the NetApp Support site. Table 9 compares the usable capacity for differing RAID levels. For completeness, all RAID levels supported by the NetApp EF-Series are shown.

Table 9) Comparison of usable capacity for differing RAID levels.

Desired Feature	RAID 0	RAID 10	RAID 5	RAID 6	DDP
Usable capacity	100%	50%	(N-1) ÷ N where N is the selected drive count in the volume group	(N-2) ÷ N where N is the selected drive count in the volume group	80% minus selected preservation capacity

7.2 SQL Server I/O Overview

SQL Server is sensitive to I/O latency issues because of the concurrent transactional nature of the SQL Server engine. SQL Server is built on a complicated system of row, page, extent, and table locks that provides transactional consistency throughout the SQL Server system. A poor I/O structure (for example, when I/O takes too long to respond) causes resources to be held longer than necessary, resulting in blocking within the system. When this occurs, it is typically not obvious that the I/O subsystem is the root cause.

- **SQL Server reads.** When reading data from SQL Server, the client first goes to the buffer cache. If the data is not in the buffer cache, SQL Server goes to the I/O subsystem to retrieve the data. The statement does not complete until 100% of the data is read; the user connection or process remains in an I/O wait state until completion.
- **SQL Server writes.** The user writes to the transaction log and the buffer cache. If the data to be modified is not already in the buffer cache, then it must be read into the buffer cache from the I/O subsystem. The buffer manager enables the transaction log to be written to first, before changes are written to the database. This process is known as write-ahead logging. When the user makes the change and the commit is executed, a log write is displayed about the change that took place, allowing the commit to complete. After the commit is complete, the user process can continue to the next stage or command without waiting for the changes to be written to the disk. Rollback transaction follows the same process as that of the commit, but in reverse. The buffer manager moves the data from the cache to the disk. It keeps track of log sequence numbers for each log record.
- **Transaction log.** The SQL Server transaction log is a write-intensive operation that is sequential in nature. The transaction log is used to provide recoverability of data in the case of database or instance failure.

The OLTP database system within the SQL Server environment depends the most on getting the greatest number of transactions through the system in the least amount of time. Examples of different types of OLTP systems include web order systems and manufacturing tracking systems. OLTP systems can have large volumes of transactions per second, and for the OLTP system it is all about throughput. For these transactions to take place, SQL Server relies on an efficient I/O subsystem. Based on a Microsoft SQL Server best practices article, an OLTP transaction profile has the following attributes:

- OLTP processing is generally random in nature for both reads and writes issued against data files.
- I/O activity is approximately 80% read and 20% write.
- In most cases, read activity is consistent and uses point queries; it does not consist of large time-consuming queries.
- Write activity to the data files occurs during checkpoint operations (frequency is determined by recovery interval settings).

- Log writes are sequential in nature with a varying size that depends on the nature of the workload (sector aligned up to 60KB).
- Log reads are sequential in nature (sector aligned up to 120KB).

7.3 Estimating I/O

Estimating the number of I/O operations required for a system is crucial when sizing a database. This exercise helps the administrator understand how to keep the database instance performing within acceptable limits. You must estimate I/O when you are unable to get the actual physical I/O numbers for the system. This is typically the case in new systems that are in the process of being constructed. The following sections provide formulas for estimating I/O.

New OLTP Database System

To estimate I/O for a new database system without access to the system, complete the following steps:

1. Estimate the number of transactions for a given time period.
2. Multiply the number of transactions by the 0.85 saturation rate, and then divide that by the number of seconds in a day. The seconds in a day are determined by the hours of operation for the database. If the database operates in a 24-hour environment, the number is 86,400.

   The formula for estimating the number of I/O operations is:

   (Estimated number of transactions × 0.85) ÷ seconds in a day = Total I/O

   For example, if there are 40,000 transactions on a system that operates 24 hours per day, the formula is:

   (40,000 × 0.85) ÷ 86,400 = 0.3935 IOPS
3. After determining the I/O required, determine the read and write I/O by multiplying the number of I/O operations by the percentage of reads or writes. The I/O activity will be approximately 67% read and 33% write for an OLTP system.

   The formula for I/O in megabytes is:

   (Number of transactions × 0.85) ÷ seconds in a day × type % = I/O megabytes

   For example, to determine the reads and writes for an OLTP system, the formula is as follows:

   ((40,000 × 0.85)/86,400) × 0.67 = 0.3148MB reads

   ((40,000 × 0.85)/86,400) × 0.33 = 0.0787MB writes

Existing OLTP Database System

When sizing for an existing database environment, understanding the type of workload and interpreting the statistical data are helpful. It is important to gather statistics during periods of peak stress on the system. PerfMon allows you to see the high-water marks for the time frame in which you monitor the system.

After either IOPS or throughput (MB/sec) of the system is captured, it can be entered into the E-Series Performance Sizing tool. Figure 12 and Figure 13 show the input fields for the sizing tool. Figure 14 lists the output.

Figure 12) E-Series performance sizing tool.

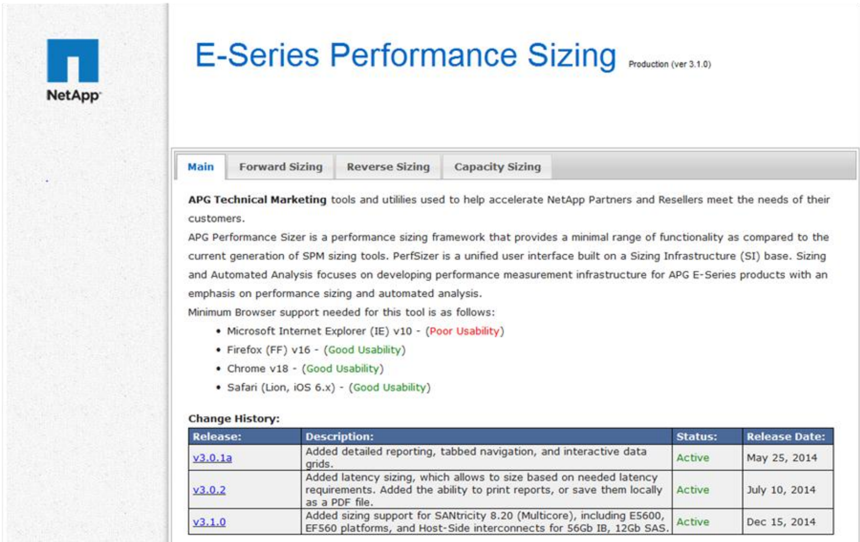


Figure 13) Input fields for E-Series performance sizing.

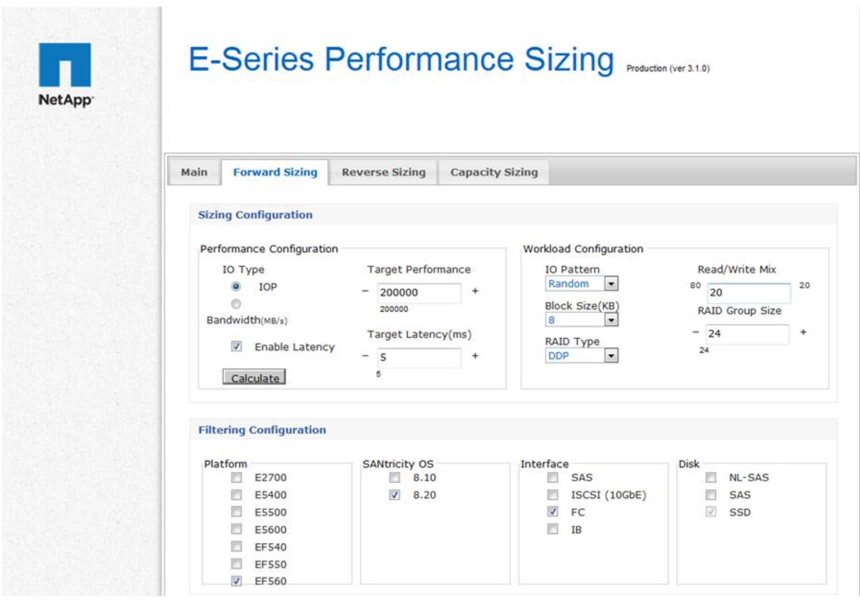
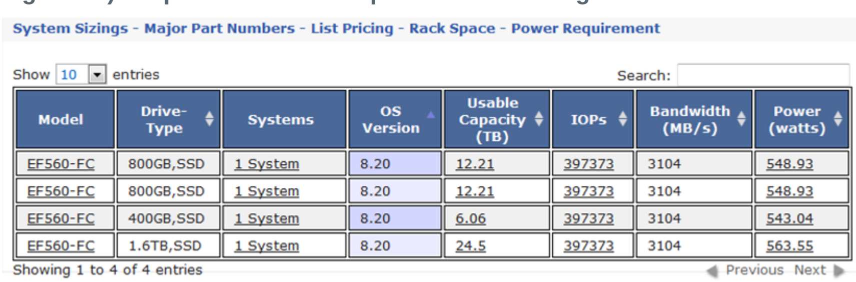


Figure 14) Output from E-Series performance sizing tool.



Model	Drive-Type	Systems	OS Version	Usable Capacity (TB)	IOPs	Bandwidth (MB/s)	Power (watts)
EF560-FC	800GB,SSD	1 System	8.20	12.21	397373	3104	548.93
EF560-FC	800GB,SSD	1 System	8.20	12.21	397373	3104	548.93
EF560-FC	400GB,SSD	1 System	8.20	6.06	397373	3104	543.04
EF560-FC	1.6TB,SSD	1 System	8.20	24.5	397373	3104	563.55

8 Monitoring

8.1 Performance Monitoring Using SANtricity Add-in for SQL Server Management Studio

The SQL Server Management Studio (SSMS) is a tool included with Microsoft SQL Server for configuring, managing, and administering all components in SQL Server. This tool includes both script editors and graphical tools that work with objects and features of the server.

A central feature of SSMS is the object explorer, which allows you to browse, select, and act upon any of the objects in the server. SSMS is the principal database administration portal for SQL Server databases, and many database administrators spend a large percentage of time using the tool to perform their job responsibilities.

The NetApp SSMS Storage Explorer add-in extends the out-of-box SSMS functions to give the database administrator insight into the NetApp EF-Series storage subsystem. The add-in also provides feedback about the proper functioning of the storage.

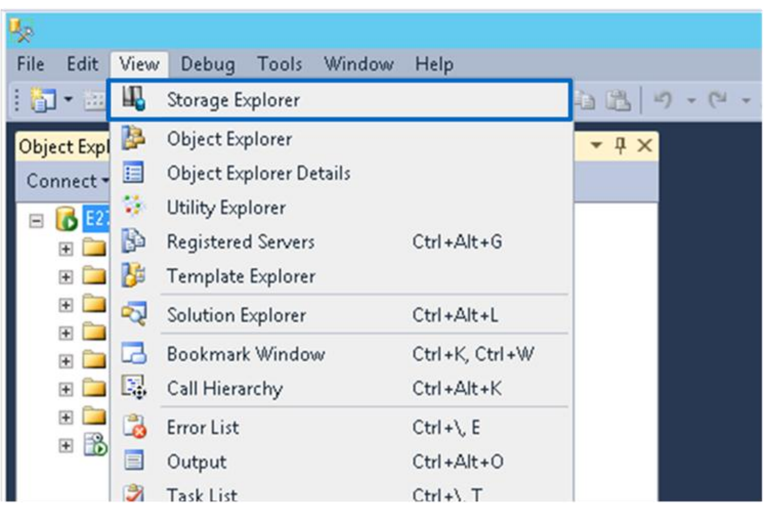
Storage Explorer is integrated with SSMS as a client-side extension that does the following:

- Displays attached storage properties (logical and physical)
- Generates and displays performance reports
- Provides storage alerts

Use SSMS Storage Explorer

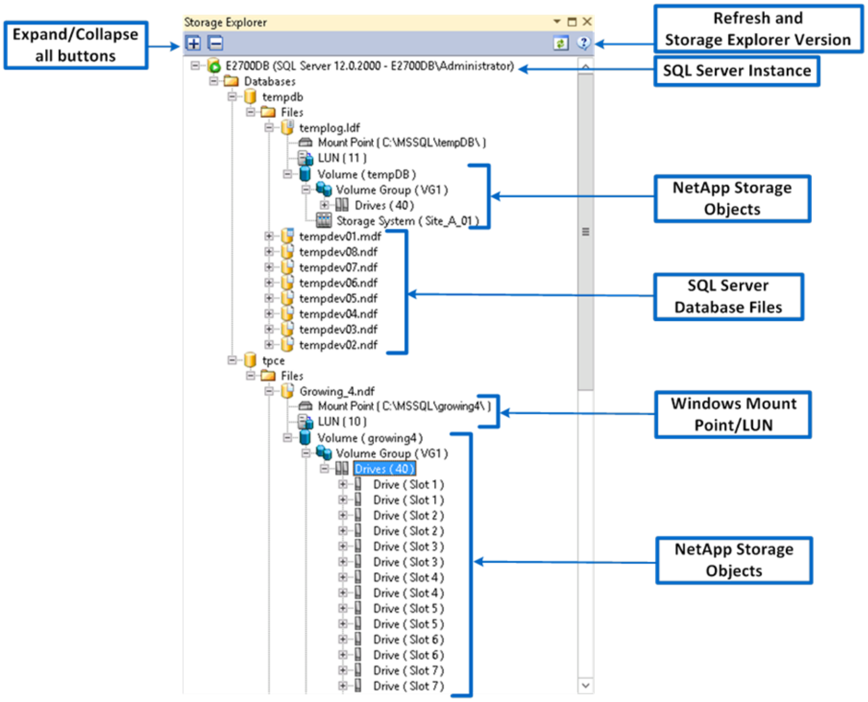
To use the SSMS Storage Explorer, complete the following steps:

1. Start the SSMS.
2. Using Object Explorer, select an instance on EF-Series storage and click View > Storage Explorer.



3. After Storage Explorer starts, expand the databases object to see all databases with files stored on NetApp EF-Series storage. Scroll through the list to see all objects.
4. To change the instance being viewed, select another instance and click View > Storage Explorer.

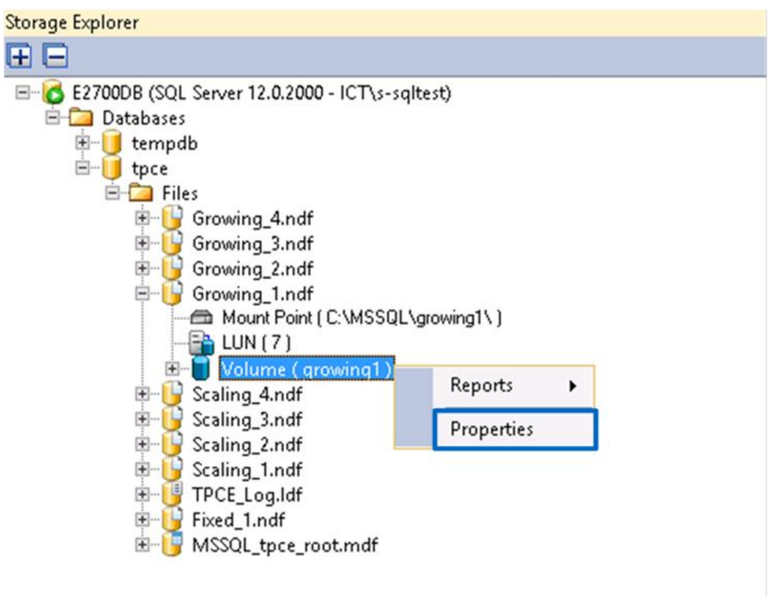
   The following screenshot shows the expanded Storage Explorer tree.



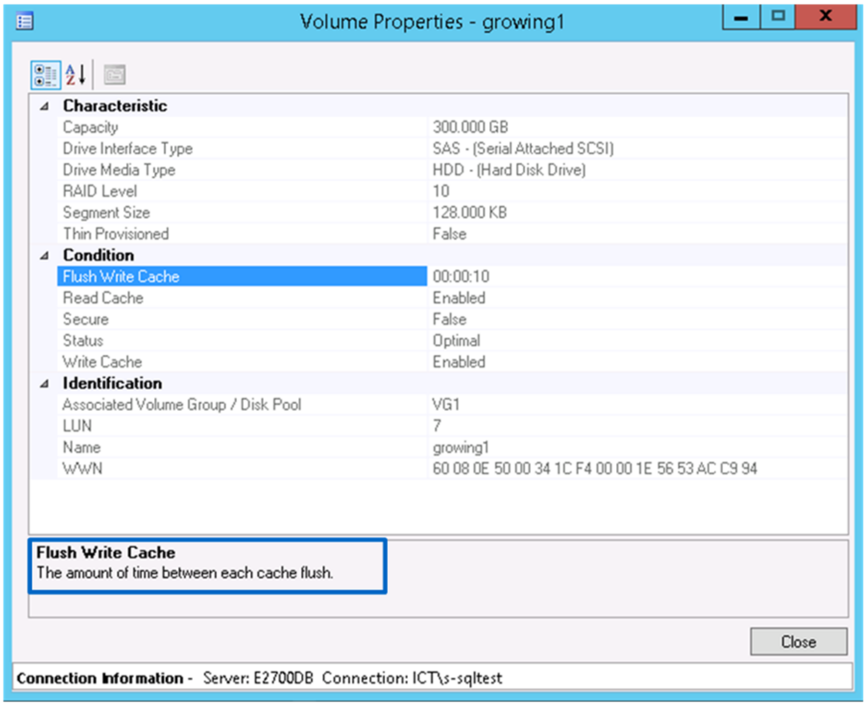
View Properties

To view the properties of a NetApp storage object, complete the following steps:

1. Right-click an object. The objects that display properties are volume, thin-provisioned volume (TPV), volume group, DDP, drive, tray, and storage system.



2. Select a property in the list to view a brief description of the property.



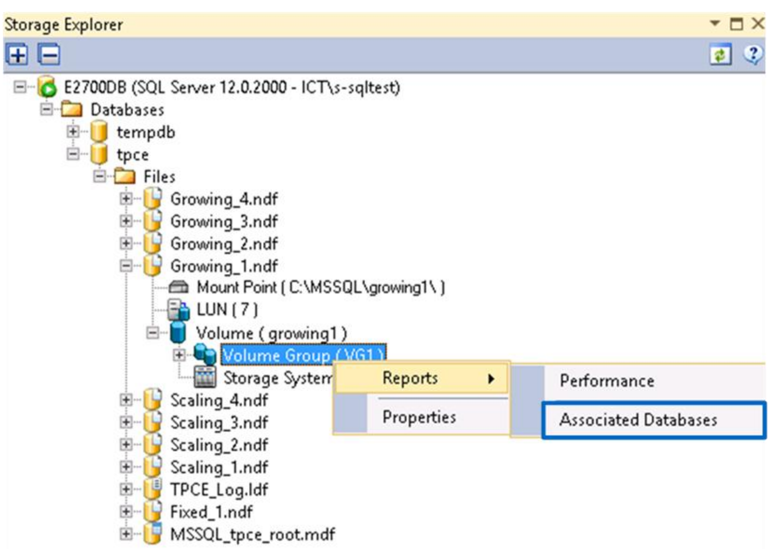
View Reports

Storage Explorer generates two types of reports: associated database reports and performance reports.

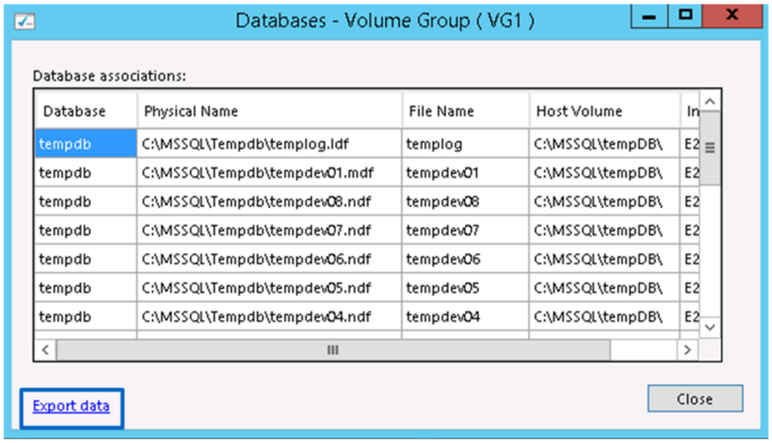
To access the two types of reports that are provided by Storage Explorer, right-click an object and choose the type of report.

Associated Databases Report

Associated databases reports are available for volumes, volume groups, TPV, and DDP.

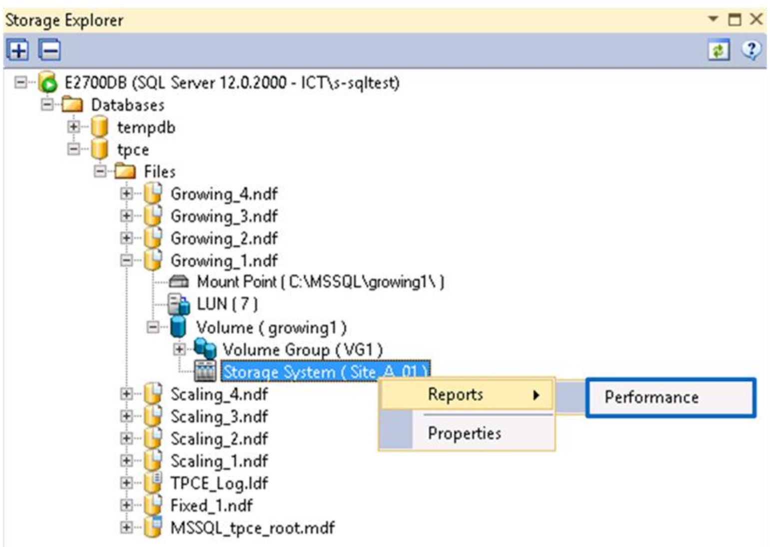


An associated databases report lists all database files stored on the selected object.

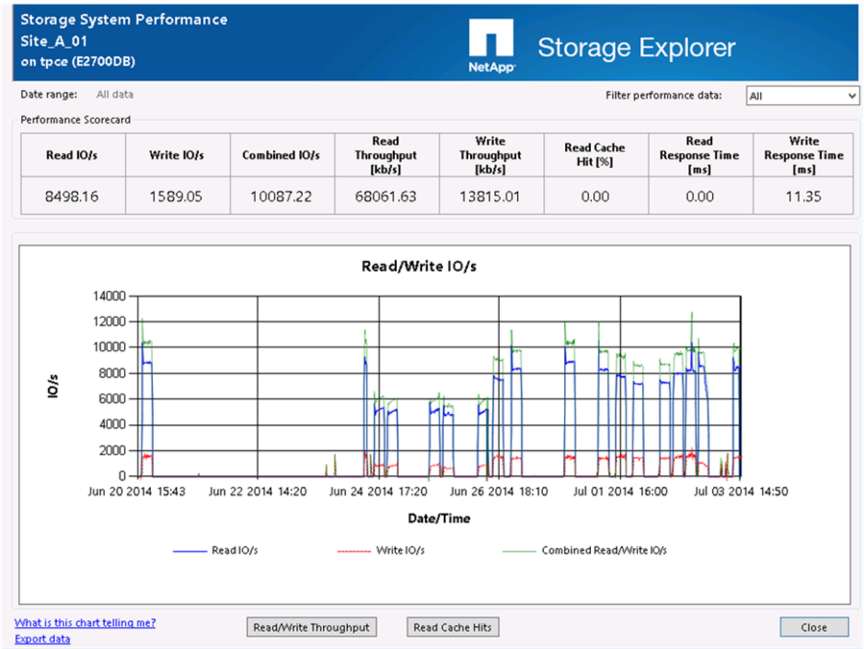


Performance Report

Performance reports are available for volumes, volume groups, TPV, DDP, and storage systems.



A performance report shows all activity occurring on the specific object. In the following example, on the storage system performance report, the I/O activity for all databases on the storage system, not just for one database or data file, is shown.



This information allows the database administrator to begin at a volume level and verify which database files are on the volume using the associated databases report. The administrator can then look at the performance reports at this level before proceeding to the volume group level. The same type of activity is then performed at the volume group level and finally for the entire storage array.

You can download the user's guide with complete information on installation and operation of the NetApp EF-Series add-in for SSMS from NetApp Support with the add-in software.

8.2 EF-Series Performance Monitoring Using SANtricity Storage Manager

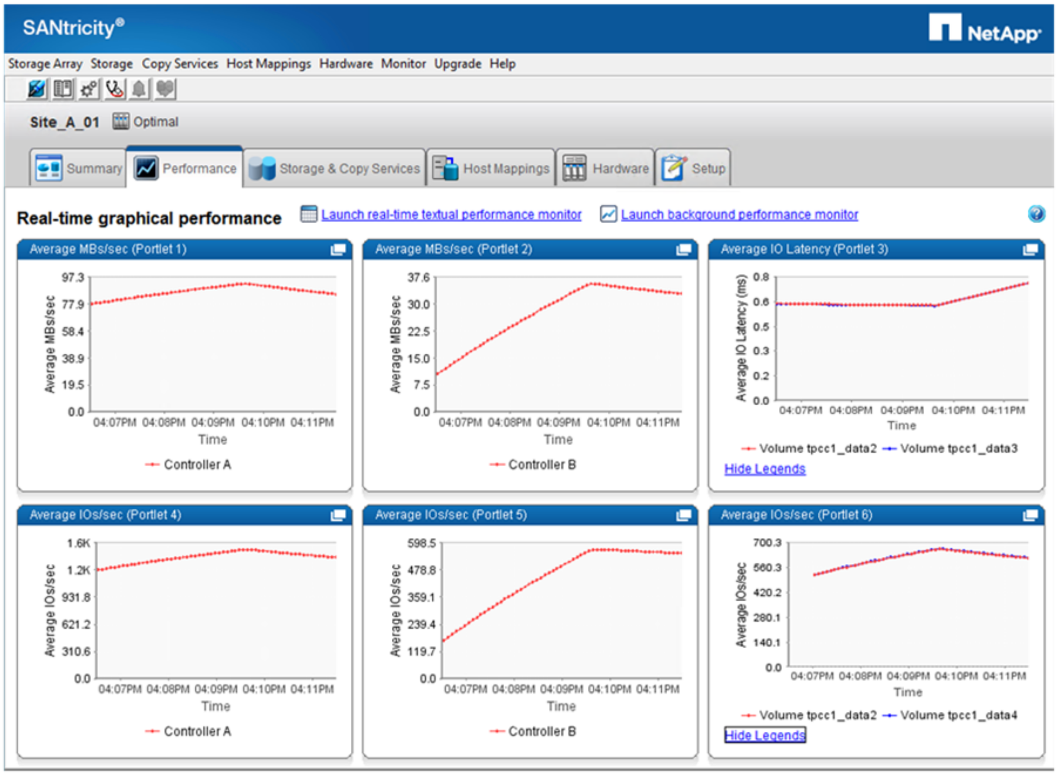
During storage system operations, it can be useful to monitor storage system performance. Using SANtricity, you can view EF-Series performance data in both textual and graphical dashboard formats. Additional details of the monitor feature are located in "Concepts for SANtricity Storage Manager," available on the NetApp Support site Documentation library and SANtricity online help.

The performance monitor provides visibility into performance activity across your monitored storage devices. You can use the performance monitor dashboard to perform these tasks:

- View different performance metrics on six graphs in real time for up to five monitored devices per graph.
- Collect performance metrics including:
  - I/O latency for drives and volumes
  - Current or maximum I/O per second
  - Throughput for the entire storage array and volumes
  - Cache hit percentage
  - Total I/Os
- Monitor textual performance or background performance by using convenient links on the dashboard.

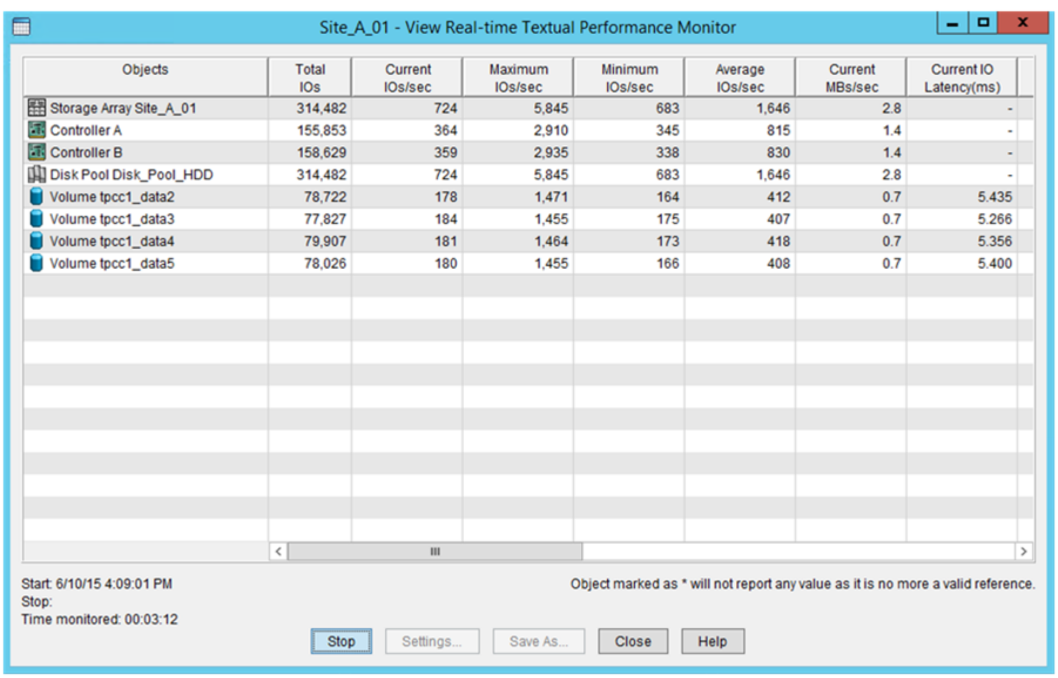
Figure 15 shows the performance data that can be collected and viewed through the SANtricity performance monitor dashboard.

Figure 15) Performance monitor dashboard.



Real-time performance data (actual values of the collected metrics) can also be monitored in tabular format and saved to a file for later analysis.

Figure 16) Real-time performance data in tabular format.



Site_A_01 - View Real-time Textual Performance Monitor

Objects	Total IOs	Current IOs/sec	Maximum IOs/sec	Minimum IOs/sec	Average IOs/sec	Current MBs/sec	Current IO Latency(ms)
Storage Array Site_A_01	314,482	724	5,845	683	1,646	2.8	-
Controller A	155,853	364	2,910	345	815	1.4	-
Controller B	158,629	359	2,935	338	830	1.4	-
Disk Pool Disk_Pool_HDD	314,482	724	5,845	683	1,646	2.8	-
Volume tpcc1_data2	78,722	178	1,471	164	412	0.7	5.435
Volume tpcc1_data3	77,827	184	1,455	175	407	0.7	5.266
Volume tpcc1_data4	79,907	181	1,464	173	418	0.7	5.356
Volume tpcc1_data5	78,026	180	1,455	166	408	0.7	5.400

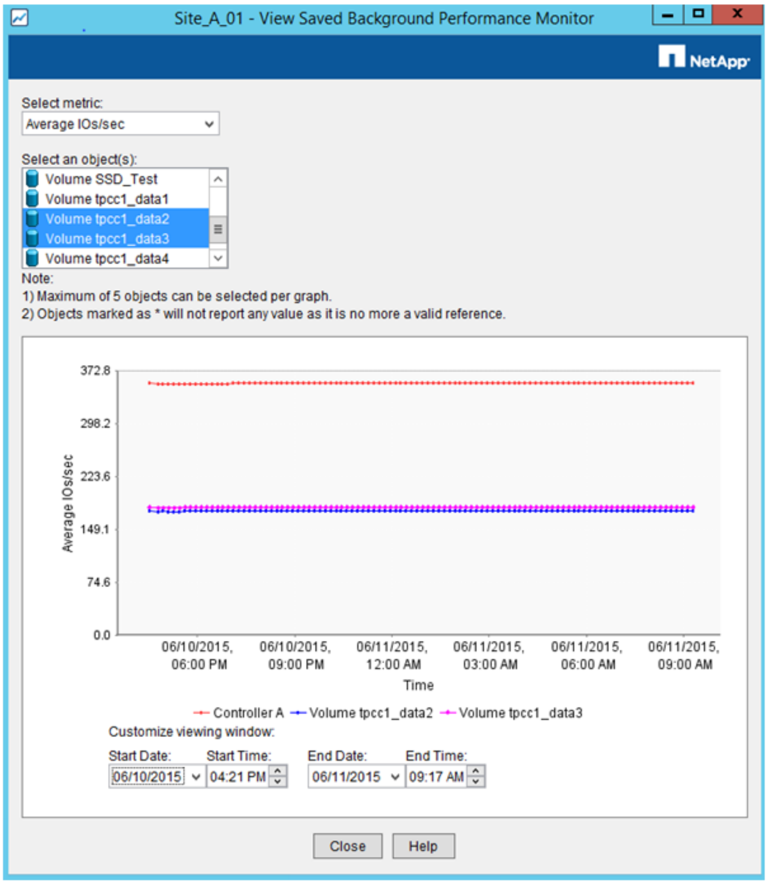
Start: 6/10/15 4:09:01 PM
Stop:
Time monitored: 00:03:12

Object marked as * will not report any value as it is no more a valid reference.

Stop | Settings... | Save As... | Close | Help

SANtricity also enables background performance monitoring. Various reporting attributes, such as time increments and filtering criteria, can be specified to examine performance trends and to pinpoint the cause of availability and performance issues.

Figure 17) Background performance monitoring.



9 Summary

The NetApp EF-Series flash array is a market leader in delivering high performance and consistent low latency. The importance of this fact will vary depending on business needs.

In addition to meeting extreme latency requirements, the EF-Series also resolves SQL Server database challenges in the following ways:

- Dramatically boosts the performance of existing applications and lowers the cost-per-IOPS ratio without the need to rearchitect the application.
- Increases SQL Server performance with RAID 10 and DDPs.
- Increases user productivity with better response times, improving business efficiency.
- Provides HA and DR capabilities with NetApp Snapshot and replication technology.
- SQL Server AlwaysOn Availability Groups provide an alternate solution for HA and DR.

Though the EF-Series array can be configured in many different ways, the best practice is to configure a single DDP with all the SSDs participating in the pool. Volumes are carved out of this pool regardless of the purpose of the volume (such as a log or a database). This action results in over 300K IOPS and approximately a 0.6ms response time for an 80/20 read/write 8K workload. Only in extreme cases does the EF-Series flash array need to be configured in a RAID 10 configuration in which IOPS will increase and latency decrease. The objective should be to keep the solution as simple and fast as possible.

References

This report references the following documents and resources:

- NetApp Support site documentation library
  http://support.netapp.com/portal/documentation
- NetApp Support
  http://support.netapp.com/NOW/cgi-bin/software

Version History

Version	Date	Document Version History
Version 2.0	July 2015	Updated for EF560 and SQL Server 2014.
Version 1.0	January 2014	Initial release.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp®

www.netapp.com