Dell EMC PowerVault ME4 Series and Microsoft SQL Server

Abstract

This document provides best practices for deploying Microsoft® SQL Server® with Dell EMC™ PowerVault™ ME4 Series arrays, including recommendations and considerations for performance, availability, and scalability.

October 2018

Revisions

Date	Description
September 2018	Initial release
October 2018	Added performance section

Acknowledgements

Author: Doug Bernhardt

Table of contents

Executive summary

This paper provides guidance for using Dell EMC™ PowerVault™ ME4 Series storage systems in a Microsoft® SQL Server® environment. SQL Server is a robust product that can be used in a variety of solutions, allowing you to prioritize performance, manageability, and flexibility depending on your environment. This paper provides important considerations and recommendations to help meet your design goals, and builds upon the best practices in the ME4 Series *Administrator's Guide* on Dell.com/support.

This document was developed using the PowerVault ME4024 array, but is also applicable to ME4012 and ME4084 arrays.

Note: While following the best practices in this document is strongly recommended by Dell EMC, some recommendations may not apply to all environments. For questions about the applicability of these guidelines in your environment, contact your Dell EMC representative.

Audience

This document is intended for ME4 Series administrators, database administrators, architects, partners, and anyone responsible for configuring ME4 Series storage systems. Some familiarity with Dell EMC storage systems is assumed.

We welcome your feedback along with any recommendations for improving this document. Send comments to StorageSolutionsFeedback@dell.com.

1 Introduction

The PowerVault ME4 Series is next-generation, entry-level storage that is purpose-built and optimized for SAN and DAS virtualized workloads. Available in 2U or dense 5U base systems, the low-cost ME4 Series simplifies the challenges of server capacity expansion and small-scale SAN consolidation with up to 336 drives or 4PB capacity. It also comes with all-inclusive software, incredible performance, and built-in simplicity with a new web-based HTML5 management GUI, ME Storage Manager. Connecting ME4 Series storage to a PowerEdge server or to a SAN ensures that business applications will get high-speed and reliable access to their data — without compromise.

Product features include the following:

Simplicity: ME4 Series storage includes a web-based management GUI (HTML5), installs in 15 minutes, configures in 15 minutes, and easily deploys in 2U or 5U systems.

Performance: Compared to the predecessor MD3 Series, the ME4 Series packs a lot of power and scale with the Intel® Xeon® processor D-1500 product family. The ME4 Series processing power delivers incredible performance gains over the MD3 Series, as well as increased capacity, bandwidth, and drive count.

Connectivity: ME4 Series storage goes to the next level with robust and flexible connectivity starting with a 12Gb SAS back-end interface, and a front-end interface options including four 16Gb FC ports per controller, four 10Gb iSCSI ports per controller (SFP+ or BaseT), or four 12Gb SAS ports per controller.

Scalability: Both 2U and 5U base systems are available, with the 2U system supporting either 12 or 24 drives and the 5U system supporting 84 drives. Each of the 2U (ME4012 and ME4024) and 5U (ME4084) base systems supports optional expansion enclosures of 12, 24, and 84 drives, allowing you to use up to 336 drives. Drive mixing is also allowed.

All-inclusive software: ME4 Series software provides volume copy, snapshots, IP/FC replication, VMware® VCenter Server® and VMware Site Recovery Manager™ integration, SSD read cache, thin provisioning, three-level tiering, ADAPT (distributed RAID), and controller-based encryption (SEDs) with internal key management.

Management: An integrated HTML5 web-based management interface (ME Storage Manager) is included.

For more information, see the ME4 Series product page.

2 Best practices overview

Use the following general steps to set up and configure an ME4 Series system for SQL Server:

1. Capture the storage I/O performance characteristics and capacity requirements of your SQL Server workload.
2. Review the remaining sections of this document and apply the best practices that are applicable to your workload and environment. Since SQL Server workloads can vary, not all recommendations may apply.
3. Follow the deployment instructions for setting up an ME4 Series system found in the ME4 Series *Deployment Guide* on Dell.com/support.
4. Configure the ME4 Series system using the ME4 Series *Administrator's Guide,* applying best practices for Microsoft Windows and SQL Server as recommended by Dell EMC.

3 SQL Server design considerations

The I/O storage system is a critical component of any SQL Server environment. Sizing and configuring a storage system without understanding the I/O requirements can have disastrous consequences. Analyzing performance in an existing environment using a tool like Live Optics can help define the I/O requirements. Your Dell EMC representative can assist with Live Optics data collection and analysis. For best results, capture performance statistics for a period of at least 24 hours that includes the system peak workload.

3.1 OLTP workloads

While every environment is unique, an online transaction processing (OLTP) workload typically consists of small, random reads and writes. A storage system for OLTP workloads is primarily sized based on capacity and the number of IOPS required.

3.2 OLAP/DSS workloads

An online analytic processing (OLAP) or decision support system (DSS) workload is typically dominated by large, sequential reads. A storage system for OLAP/DSS workloads is primarily sized based on throughput. When designing for throughput, the performance of the entire path between the server and the drives in the ME4 Series array needs to be considered. For best throughput, consider using 16 Gb Fibre Channel (FC) or 10 Gbps iSCSI connectivity to the array. To meet high-throughput requirements, multiple physical paths may be required.

3.3 Mixed workloads

The most common scenario for a SQL Server environment is a mixed workload. Typically, SQL Server I/O patterns do not strictly fall into an OLTP or OLAP pattern. This is what can make SQL Server workloads challenging because no two workloads behave the same. In addition, the same SQL Server host or instance may be servicing multiple applications or transaction workloads.

A mixed workload can also imply that multiple applications (in addition to SQL Server) are residing on the same host or accessing the same storage. The combined workload of these applications invalidates any typical application I/O usage pattern. For these reasons, it is important to gather actual performance metrics for best sizing results.

3.4 ME4 Series configuration

3.4.1 Balanced configuration

Creating a balanced storage configuration is important because SQL Server workloads can vary greatly and I/O patterns can often fluctuate due to changes in the database environment, evolving data-access patterns, or data growth. For most SQL Server workloads, it is recommended to configure the ME4 Series array using the virtual storage type and ADAPT as the RAID level.

For best performance, a minimum of 24 drives should be used when using the ADAPT RAID level because this is the minimum number required to create two virtual storage pools, one per controller in a dual-controller system. Start with 24 SSD drives and add additional drives as needed to achieve performance and capacity requirements. An ME4 Series array with SSDs spread evenly across 2 virtual storage pools (one per controller) configured with the ADAPT RAID level provides the best overall balance of performance, flexibility, capacity, and data protection.

There is a variety of other RAID levels and storage configurations available for very specific workloads. Make sure the design tradeoffs are completely understood when choosing custom configurations and settings. In many cases, modifying the storage configuration for existing volumes will involve halting I/O on those volumes, resulting in a SQL Server outage. For detailed information on all available choices, including ADAPT, consult the ME4 Series *Administrator's Guide*.

3.4.2 RAID levels and performance

The performance characteristics of the ME4 Series are largely dependent on the RAID level chosen for the storage configuration. While the ADAPT RAID level provides the most balanced configuration, other RAID levels are available for maximum performance or maximum capacity. The two additional levels discussed in this section are RAID 5 and RAID 10. For all available RAID levels, consult the ME4 Series *Administrator's Guide.* The performance numbers presented are maximum values on an ME4 Series array with a minimum of 24 drives and two storage pools, one per controller.

Note that performance among RAID levels only has significant variation in workloads such as OLTP where frequent writes occur. For read-only workloads such as OLAP, there is practically no performance benefit to select RAID levels based on performance.

Use care when configuring multiple RAID levels on the same ME4 Series array. Ensure that the number of drives in the storage pool is sufficient to provide the performance desired. Mixing RAID levels within a storage pool is not recommended.

Remember that when selecting RAID levels, there are design considerations other than performance. RAID levels impact the data protection, capacity, and flexibility of the overall storage design. Therefore, finding the best balance of performance, data protection, capacity, and flexibility is the goal.

3.4.2.1 ADAPT

When using the ADAPT RAID level, the ME4 Series can perform up to 99,000 IOPS using an OLTP workload[1] and up to 7 GB/sec for OLAP workloads[2]. The solid OLTP performance and outstanding read performance, as well as the balance of performance, capacity, and data protection, make ADAPT a good choice for mixed-database workloads.

3.4.2.2 RAID 5

When additional performance and capacity is required, RAID 5 may be used. In a RAID 5 configuration, some disk resources that were reserved for data protection are utilized for additional capacity and performance. Therefore, RAID 5 offers less data protection than ADAPT. However, it can deliver up to 115,000 IOPS for OLTP workloads[1].

3.4.2.3 RAID 10

For maximum OLTP performance, RAID 10 can deliver up to 192,000 IOPS for OLTP workloads[1]. In a RAID 10 configuration, the usable capacity is reduced to 50% of raw capacity. Therefore, a RAID 10 solution requires almost double the number of drives compared to more space-efficient RAID types.

[1] An OLTP workload is defined as having an 8k block size and a 70/30 read/write mix.
[2] An OLAP workload is defined as having a 128k+ block size with 100% reads.

Figure 1 compares the OLTP performance for each RAID type.

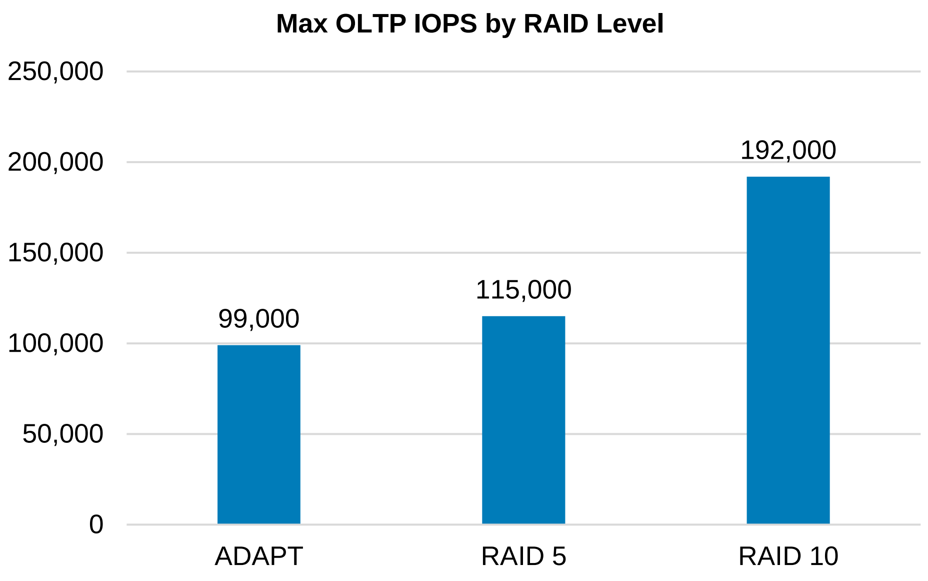

Figure 1 Maximum OLTP IOPS by RAID level[3]

3.5 Validating the storage design

Once the I/O requirements have been defined, it is easy to determine whether the hardware can provide the desired performance by running some simple tests. Diskspd is a free Microsoft utility that can simulate I/O patterns generated by SQL Server. There are several other utilities available as well. When selecting a utility to simulate I/O, verify that it meets the following requirements:

- Ability to configure block size
- Ability to specify number of outstanding requests
- Ability to configure test file size
- Ability to configure number of threads
- Support for multiple test files
- Does not write blocks of zeros during tests

3.5.1 Validating the I/O path

The first thing to test on a new configuration is the path between the server and the array. Running a large block sequential read test using small files should saturate the path between the server and the array. This test verifies that all paths are fully functional and can be used for I/O traffic. Run this test on a dedicated server and array; a live system could cause significant performance issues.

[3] An OLTP workload is defined as having an 8k block size and a 70/30 read/write mix.

To validate the I/O path, run a large block sequential read test using the following guidelines:

- Create one LUN per storage processor.
- Format the volumes using a 64 KB allocation unit.
- Use a block size of 512 KB for the test.
- Configure the test for 32 outstanding I/Os.
- Use multiple threads. Eight is the recommended starting point.

If the displayed throughput matches the expected throughput for the number of HBA ports in the server, the paths between the server and ME4 Series array are set up correctly.

3.5.2 Validating the drives

Once the I/O path has been validated, the next step is to test the drives. For best results when testing drives on an ME4 Series array, use the following guidelines when configuring the test:

- In a dual-controller system, use at least one volume per pool with each pool on a separate controller. This ensures that I/O will be distributed across both controllers. Using both controllers provides a more accurate simulation of real-world activity. For best results, use the same number of volumes on each controller.
- When performing I/O tests on any storage platform, it is important to use files that are larger than the controller cache. For more accurate results, use a file size that matches the amount of data being stored. In an environment where that is not practical due to a large data set, use a file size of at least 100 GB.
- Avoid using test utilities to generate files full of zeros for drive validation. Some I/O test tools, including Diskspd, SQLIO and IOMeter, can be used to write zeroes for drive validation, which causes inaccurate results when testing with files containing only zeros. The contents of the test file can be verified by viewing the test file with a hex editor after different stages of a test. For example, create a small test file and view it after the initial creation, as well as after the test has run for a few seconds. If the file is filled with zeros, select another utility. Diskspd and IOMeter initially create test files filled with zeros, and then write random characters when performing write tests. To properly initialize a Diskspd or IOMeter test file, run a sequential write test until the entire file has been overwritten with non-zero data. Unfortunately, SQLIO writes zeros during write tests and therefore is not recommended for drive validation.

The purpose of this drive testing is to validate that the storage design will provide the required throughput and IOPS with acceptable latency. It is important that the test does not exceed the designed capacity of the array. For example, an array designed for a workload of 5,000 IOPS is likely to perform poorly with a workload of 10,000 IOPS. If a test is generating a workload higher than the designed capacity, adjust the workload being generated by reducing the number of threads or outstanding I/Os.

The results of the Live Optics analysis provide an I/O target to simulate using these tests. To estimate the performance capabilities of the array, run I/O tests with a range of I/O sizes commonly seen with SQL Server. When testing random I/O, test with an I/O size of 8 KB and 64 KB. When testing sequential I/O, start with I/O sizes of 8 KB and 64 KB. Since processes like read-ahead scans and backups can issue much larger sequential I/O, it is a good idea to also test block sizes up to 1024 KB.

4 Deploying SQL Server on ME4 Series storage

Proper architecture and configuration of the SQL Server environment is critical to optimize performance and manageability of the ME4 Series array and SQL Server environment. Apply the following best practices when designing, configuring, and managing SQL Server databases on ME4 Series storage.

4.1 Volume configuration

4.1.1 Creating volumes

There are many types of files that are part of a SQL Server instance. Those types of data often have different performance requirements. For performance-sensitive applications, Dell EMC recommends creating at least five volumes for an instance of SQL Server as shown in Table 1.

Table 1 Volume-provisioning recommendations

File type	Number of volumes	Typical performance requirements
User DB data	At least 1 per instance	Lower performance may be acceptable
User DB transaction log	At least 1 per instance	High performance required
Data root directory (includes system DBs)	1 per instance	Lower performance may be acceptable
Tempdb data and transaction log	1 per instance	High performance may be required
Native SQL Server backup	1 per instance	Lower performance may be acceptable
Memory-Optimized Filegroup (if used)	At least 1 per instance	High performance required

4.1.2 Performance considerations

When there is one group of databases that require high performance and another group that does not, consider creating a set of volumes for each group of databases. This strategy will make it easier to adjust the storage configuration in the future. It also makes it easier to distribute the I/O load evenly across both controllers. Databases that have very high performance requirements can be spread across two or more data files on separate volumes to leverage resources on both controllers.

4.1.3 Flexibility and manageability

For ultimate flexibility, create a volume for each user database file. This provides the ability to independently optimize the storage for each individual database. With thin provisioning, there is no space penalty for creating numerous volumes. However, a large number of volumes can be difficult to manage, especially in virtualized environments. It is up to the DBA or storage administrator to find the right balance between flexibility and manageability when determining the number of volumes to create. Virtualized SQL Server environments are a good example where placing multiple file types on a single volume can make sense. Understanding the database I/O patterns is critical to making the best decisions.

4.1.4 Windows setup and configuration

4.1.4.1 Allocation unit size

Use a 64 KB allocation unit size when formatting volumes that will contain database files (transaction log and data) or database backups.

4.1.4.2 MPIO

ME4 Series arrays support Asymmetric Logical Unit Access (ALUA), and when MPIO is configured, the default MPIO policy is round robin with subset. This is the recommended setting for all database volumes. This setting works best for most environments because it is easy to manage and performs very well. Use other MPIO policies with caution and remember to review custom MPIO policies when adding or removing volumes from the host.

4.2 SQL Server I/O reduction

4.2.1 Memory

Unnecessary I/O can be avoided and performance can be increased by allocating the proper amount of memory to SQL Server. SQL Server performs all I/O through the buffer pool (cache) and therefore uses a large portion of its memory allocation for the buffer pool. Ideally, when SQL Server performs I/O, the data is already in the buffer pool and it does not need to go to disk. This type of I/O is referred to as logical I/O and is the most desirable because it results in the best performance. If the SQL Server data does not need to reside in the buffer pool, it will need to access disks, resulting in physical I/O.

Proper memory allocation is critical to SQL Server performance and can improve storage performance as well. In many cases, SQL Server and storage performance can be further improved by adding memory. Adding memory generally improves performance, but there is a point of diminishing returns that is unique to each environment.

4.2.2 Buffer pool extension

With SQL Server 2014, the buffer pool can be extended to a file on the file system to provide additional space to cache data or index pages. Using this feature can provide significant performance benefits without adding memory to the database server in some cases. By caching more pages on the server, the I/O load on the array is reduced.

When placing the buffer pool extension on the array, create a separate volume for the buffer pool extension and do not take snapshots of the buffer pool extension volume. The buffer pool data is repopulated by SQL Server when the instance is restarted, therefore data recovery does not apply.

4.2.3 Database compression

The overall I/O workload can be reduced by enabling database compression in SQL Server. While there is a tradeoff in terms of CPU utilization on the database server, compression is still a viable option to consider and test in any environment. Database compression reduces I/O by reducing the amount of data that needs to be stored. The SQL Server data pages are compressed in memory before being written to disk, resulting in fewer pages needed to store the same number of rows and therefore less I/O.

4.2.4 Instant file initialization

By default, SQL Server writes zeros to the data file during the allocation process. The process of zeroing out the data files consumes I/O and acquires locks as the SQL Server data pages are written. This activity can occur for minutes or even hours depending on the file size. While this may seem minor, writing zeros to these files can occur at critical periods when time and performance are critical such as database auto growth, expanding a full data file, replication, or restoring a database as part of a disaster-recovery event.

When Instant File Initialization is enabled, SQL Server will skip the process of zeroing out its data files when allocating space. Dell EMC recommends enabling Instant File Initialization.

4.2.5 Resource Governor

The Resource Governor was added in SQL Server 2008 to allow database administrators to limit the CPU and memory resources that a query is able to consume. This feature was enhanced in SQL Server 2014 to allow I/O resources to be limited as well. For example, the Resource Governor can be used to reduce the impact of a user running an I/O-intensive report by limiting the maximum number of IOPS that user can perform. While a query throttled by the Resource Governor will take more time to complete, overall database performance will be better.

4.2.6 Database design considerations

Reducing SQL Server I/O requires a holistic approach. Many of the items in this section will require involvement from the whole team responsible for the SQL Server applications including the business owner, architect, developer, database administrator, and system administrator. Decisions at the design level have a multiplied impact downstream because data is written and read multiple times and duplicated in various types of database copies including databases copied for other uses such as testing and reporting, replicated databases, replicated storage, and backups.

One of the most challenging aspects of SQL Server is that the I/O pattern and the amount of I/O that is generated can vary greatly depending on the application, even if those applications have databases of the same size. This is because the design of both the database and the data-access code control the SQL Server I/O.

Database tuning can be one of the most cost-effective ways to reduce I/O and improve scalability. At a high level, consider the tips in the following subsections when tuning a database to reduce I/O.

4.2.6.1 Database design

The foundation of the entire database and the schema for how data will be stored and ultimately accessed is determined by the database design. The database design should support both usability and efficient data access. This includes efficient table design and data types as well as indexes, partitioning, and other features that can improve efficiency. It is common for database design to only be focused on usability while performance and scale are overlooked.

4.2.6.2 Query design

How a query is written can greatly affect the amount of I/O SQL Server needs to perform when executing the query. Queries should return only the required amount of data in the most efficient manner possible. Tune the queries responsible for consuming the most resources for best performance and scale.

4.2.6.3 Application design

Consider how applications are using the data and how the data is requested. Sometimes code and component reuse can result in the same data being unnecessarily retrieved repeatedly. All data access should be purposeful.

4.2.6.4 Maintenance

SQL Server uses a cost-based optimizer to generate query plans for data access. These plans are based on the statistics regarding how data is distributed in the tables. If the statistics are inaccurate, bad query plans may result and unnecessary I/O will be performed. Proper database maintenance includes ensuring that statistics are up to date.

Frequent data modifications can also lead to fragmentation within SQL Server data files, producing unnecessary I/O. Fragmentation can be addressed through index reorganization or rebuilds as part of regular database maintenance.

The database maintenance process itself can also have a large I/O impact. Typically, every table and index does not need to be rebuilt or reorganized every time maintenance is run. In addition, table partitioning strategies can also be leveraged to make the maintenance process more selective. Consider implementing intelligent maintenance scripts that utilize usage statistics to perform maintenance on an as-needed basis.

For mission-critical databases, maintenance activities need to be considered as part of the overall design. If maintenance is not considered as part of the overall process, issues can arise such as unmanageable sizes and feature incompatibilities that limit available options and strategies.

A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage Solutions Technical Documents provide expertise that helps to ensure customer success on Dell EMC storage platforms.

A.1 Related resources

The following ME4 Series publications and additional resources are available at Dell.com/support.

- Administrator's Guide
- Deployment Guide
- CLI Guide
- Owner's Manual
- Support Matrix

For additional SQL Server information, see the following resources:

- SQL Server I/O Basics
- SQL Server I/O Basics, Chapter 2
- Pre-deployment I/O Best Practices
- Analyzing I/O Characteristics and Sizing Storage Systems for SQL Server Database Applications
- Dell SQL Server Solutions
- Microsoft SQL Server homepage
- SQL Server Customer Advisory Team
- Microsoft online SQL Server forum
- Professional Association for SQL Server
- Other SQL Server resources:
  - http://www.sqlservercentral.com
  - http://www.sqlteam.com
  - http://www.sql-server-performance.com