Best Practices and Sizing Guidelines for Transaction Processing Applications with Microsoft SQL Server 2012 using EqualLogic PS Series Storage

A Dell EqualLogic Best Practices Technical White Paper

Dell Storage Engineering
March 2013

Revisions

Date	Description
March 13, 2013	Initial release
March 18, 2013	Corrected links

Table of contents

Acknowledgements

This best practice white paper was produced by the following members of the Dell Storage team:

Engineering: Lakshmi Devi Subramanian

Technical Marketing: Magi Kapoor

Editing: Camille Daily

Additional contributors:

Ananda Sankaran, Mike Kosacek, Darren Miller, Rob Young, and Maggie Smith

Feedback

We encourage readers of this publication to provide feedback on the quality and usefulness of this information by sending an email to SISfeedback@Dell.com.

SISfeedback@Dell.com

Executive summary

Online Transaction Processing (OLTP) applications such as enterprise resource planning (ERP), supply chain management (SCM), and web-based e-commerce systems can benefit from a Dell™ EqualLogic™ storage solution. With its unique peer storage architecture, the EqualLogic PS Series array delivers high performance and availability regardless of scale.

Systems such as large e-commerce websites that must respond to spikes in demand from a large number of users and a high volume of transactions need to be designed appropriately. Therefore, it is essential to configure a balanced end-to-end system to enable consistent user transactions without any delay or bottlenecks during the peak loads for SQL Server OLTP database environments. The storage related performance bottlenecks can only be prevented by properly sizing the storage for performance and capacity and regularly monitoring the resource utilization.

This paper includes the results of a series of storage I/O performance tests and provides capacity planning guidelines and best practices based on those results. These guidelines and best practices describe designing and deploying transaction processing applications with Microsoft SQL Server 2012 using the EqualLogic PS6100XV storage arrays.

Topics demonstrated in this paper are:

- EqualLogic PS Series arrays provided high levels of I/O performance for OLTP applications while still maintaining the Microsoft recommended latencies.
- RAID 10 performed better by offering higher IOPS compared to RAID 50 and 6 for OLTP workloads.
- Adding EqualLogic PS Series arrays scaled capacity as well as I/O performance. The scale-out architecture for all array resources, including controllers and NICs, scaled proportionately.
- Partitioning the largest and most accessed table offered better performance compared to just spreading the database data files into multiple volumes without a table partition.

Optimal operation of an OLTP application can be achieved when the applicable best practices laid out in this paper are adhered to. It must be ensured that the entire ecosystem including server, storage and networking resources are sized and configured appropriately to meet the workload performance requirements.

1 Introduction

This white paper presents the results of SQL Server I/O performance tests conducted on EqualLogic iSCSI SANs. It also provides sizing guidelines and best practices for running SQL OLTP workloads. The EqualLogic PS Series array builds on a unique peer-storage architecture that is designed to provide the ability to spread the load across multiple array members and provide a SAN solution that scales with customer needs. This pay as you grow model, allows customers to add arrays as their business demands increase the need for more storage or I/O capacity.

Careful planning prior to deployment is crucial for a successful SQL Server environment. Maximizing SQL Server performance and scalability is a complex engineering challenge as I/O characteristics vary considerably between applications depending on the nature of the access patterns. Several factors must be considered in gathering storage requirements before arriving at a conclusion. A key challenge for SQL Server database and SAN administrators is to effectively design and manage system storage, especially to accommodate performance, capacity and future growth requirements.

1.1 Objective

This paper identifies best practices and sizing guidelines for deploying SQL based OLTP applications with EqualLogic storage and also the scalability of EqualLogic PS Series arrays.

The following two major sections were analyzed during the test studies for this paper.

- I/O profiling tests using IOMeter were executed to establish baseline I/O performance characteristics of the test storage configuration when running OLTP-like I/O patterns before deploying databases.
- Performance characterization tests were executed using Benchmark Factory® for Databases to simulate SQL OLTP transactions by running a TPC-E type workload.

The test objectives determined:

- I/O performance of the storage using different RAID configurations with IOMeter generating the I/O workload.
- Scalability of the storage arrays I/O performance with an I/O workload simulated by IOMeter, as storage arrays were added.
- Scalability of the storage arrays I/O performance with an OLTP application simulation as storage arrays were added, while ensuring that the overall configuration was balanced with no resource bottlenecks on the server.

1.1.1 Audience

This white paper is primarily targeted to database administrators, storage administrators, VMware® ESXi administrators, and database managers who are interested in using Dell EqualLogic storage to design, properly size, and deploy Microsoft® SQL Server® 2012 running on the VMware vSphere™ virtualization

platform. It is assumed that the reader has an operational knowledge of Microsoft SQL Server configuration and management of EqualLogic SANs and iSCSI SAN network design, and familiarity with VMware ESXi Server environments.

1.2 Terminology

The following terms are used throughout this document.

Group: One or more EqualLogic PS Series arrays connected to an IP network that work together to provide SAN resources to host servers.

Hypervisor: The software layer in charge of managing the access to the hardware resources. It sits above the hardware and in between the operating systems running as guests.

Member: A single physical EqualLogic array.

OLTP I/O pattern: OLTP workloads tend to select a small number of rows at a time. These transfers happen anywhere in the data, and are each fairly small in size – typically between 8K and 64K. This causes the I/O pattern to be random in nature. The key metric in measuring performance of OLTP workloads is the number of I/Os per second (IOPS) that can be achieved while maintaining a healthy response time.

Perfmon: perfmon.exe is a process associated with the Microsoft® Windows® Operating System. Performance Monitor, or Perfmon, measures performance statistics on a regular interval, and saves those stats in a file. The database administrator picks the time interval, file format, and counter statistics to monitor.

Pool: A logical collection that each array is assigned to after being added to a group and contributes its storage space.

Primary data File (mdf): The primary data file contains the startup information for the database and points to the other files in the database. User data and objects can be stored in this file or in secondary data files. Every database has one primary data file. The recommended file name extension for primary data files is .mdf.

Primary Filegroup: The primary file and all system tables are allocated to the primary filegroup.

Range left partition function: The boundary value that specifies the upper bound of its partition. All values in partition 1 must to be less than or equal to the upper boundary of partition 1 and all values in partition 2 must be greater than partition 1's upper boundary.

Range right partition function: Here each boundary value specifies the lowest value of its partition. All values in partition 1 must be less than the lower boundary of partition 2 and all values in partition 2 must be greater than or equal to partition 2's lower boundary.

SAN HQ: SAN Headquarters (SAN HQ) monitors one or more PS Series groups. The tool is a client/server application that runs on a Microsoft Windows system and uses simple network management protocol (SNMP) to query the groups. Much like a flight data recorder on an aircraft, SAN HQ collects data over time

and stores it on the server for later retrieval and analysis. Client systems connect to the server to format and display the data in the SAN HQ GUI.

Secondary data file (ndf): Secondary data files are optional, are user-defined, and store user data. Secondary files can be used to spread data across multiple disks by putting each file on a different disk drive. Additionally, if a database exceeds the maximum size for a single Windows file, the secondary data files can be used so the database can continue to grow. The recommended file name extension for secondary data files is .ndf.

Transaction log file (ldf): The transaction log files hold the log information used to recover the database. There must be at least one log file for each database. The recommended file name extension for transaction logs is .ldf.

User-defined Filegroup: User-defined filegroups are specifically created by the user when the database is created or later modified. It can be created to group data files together for administrative, data allocation, and placement purposes.

Virtual Machine: An operating system implemented on a software representation of hardware resources (processor, memory, storage, network, etc.). Virtual machines are usually identified as guests in relation with the hypervisor that executes the processes to allow them to run directly on the hardware.

2 Product overview

2.1 Dell EqualLogic PS6100 Series

The EqualLogic PS6100 Series array serves as the storage foundation for the virtualized datacenter supporting critical applications such as databases, email, and virtual server workloads. With the same virtualized scale-out architecture as previous product generations, the PS6100 Series increases raw capacity, adds density, and boosts IOPS performance.

PS6100 Series arrays include 2.5 or 3.5 inch drives in 2U or 4U form factors while delivering an increase in drives per array of up to 50 percent over the previous generation. Options with a single EqualLogic PS6100 array are: all solid state drives (SSDs), a mix of SSDs and 10K serial attached SCSI (SAS) drives, all 10K SAS, or all 15K SAS drives. This provides flexibility in capacity and performance to best match various application needs. The EqualLogic PS6100XV storage arrays are optimized for critical data center applications with 14.4 TB in a high-performance SAS drive solution. Visit dell.com for feature and benefit details.

3 Database application workloads

Different types of database applications have varying needs and understanding the models for the most common database application workloads can be useful in predicting the possible application behavior. The most common database application workload models are Online Transaction Processing (OLTP) and Data warehouse (DW). This paper focuses on OLTP database workloads.

3.1 Online transaction processing

OLTP database applications are optimal for managing rapidly changing data. These applications typically have many users who are performing transactions while at the same time changing real-time data. Although individual data requests by users usually reference few records, many of these requests are being made at the same time. Examples of different types of OLTP systems include airline ticketing systems, banking/financial transaction systems, and web ordering systems.

Optimizing an OLTP database system running on a SQL Server allows the maximum number of transactions through the system in the least amount of time. The key metric in measuring performance of OLTP workloads is the number of I/Os per second (IOPS) that can be achieved while maintaining a healthy response time. For OLTP transactions to take place, SQL Server relies on an efficient I/O subsystem. According to a *Microsoft SQL Server best practices article* (http://technet.microsoft.com/en-us/library/cc966412.aspx), an OLTP transaction profile is composed of the following pattern:

- OLTP processing is generally random in nature for both reads and writes issued against data files.
- Read activity (in most cases) is constant in nature.
- Write activity to the data files occurs during checkpoint operations (frequency is determined by recovery interval settings).
- Log writes are sequential with a varying size, which is dependent on the nature of the workload.
- Log reads are also sequential in nature.

Note: It is common to perform queries and reports on OLTP databases that create a mixed OLTP and DW load on the storage. Running tests with this type of mixed loads is beyond the scope of this paper since the focus is only on OLTP workloads.

3.2 SQL Server I/O

It is essential to understand the read/write I/O block sizes and ratios directed by the application and the frequency of I/O characteristics in order to properly identify the I/O requirements for a storage system and to further develop future sizing guidelines with respect to performance or capacity. Buffer Cache size has significant impact on the change in I/O patterns in SQL Server. More details can be found in the white paper *OLTP I/O Profile Study with Microsoft SQL 2012 Using EqualLogic PS Series Storage* at http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/20308518/download.aspx

SQL Server reads: SQL Server performs two types of reads; logical and physical. Logical reads occur when the Database Engine requests a page from the buffer cache. Physical reads, on the other hand, occur when the page is not currently in the buffer cache and the database engine would retrieve the data from the I/O storage subsystem. These first copies are taken from the disk into the cache.

SQL Server writes: There are also logical and physical writes. Logical writes occur when data is modified in a page of the buffer cache and physical writes occur when the page is written to the disk from the buffer cache. Both reading from and writing to a page happen at the buffer cache. Each time a page is modified in the buffer cache, it is marked as dirty and is not immediately written back to the disk. A record of the changes is made in the log cache for every logical write. To avoid any loss of data, SQL Server makes sure that the log records are written first to a disk. The associated dirty page is removed from the buffer cache and written to a disk later. SQL Server uses a technique known as write-ahead logging that prevents writing a dirty page before the associated log record is written to a disk.

Transaction log: The SQL Server transaction log is a sequential write-intensive operation and is used to provide recoverability of data in case of database or instance failure.

Tempdb: Tempdb is a system database used by SQL Server as a temporary workspace. Access patterns for tempdb may vary but are generally more like OLTP data patterns.

The first step in being able to determine the requirements for a storage system is to understand the application I/O pattern. The frequency and size of reads and writes sent by the application are received and processed by the storage system. An understanding of their behavior and frequency is needed in order to properly understand the requirements of that system.

4 Test configuration

The SQL Server test system used to conduct testing for this paper is shown in Figure 1 and Figure 2.

4.1 Physical system configuration

The physical connectivity of the SQL Server that hosted the Databases used for testing is shown in Figure 1.

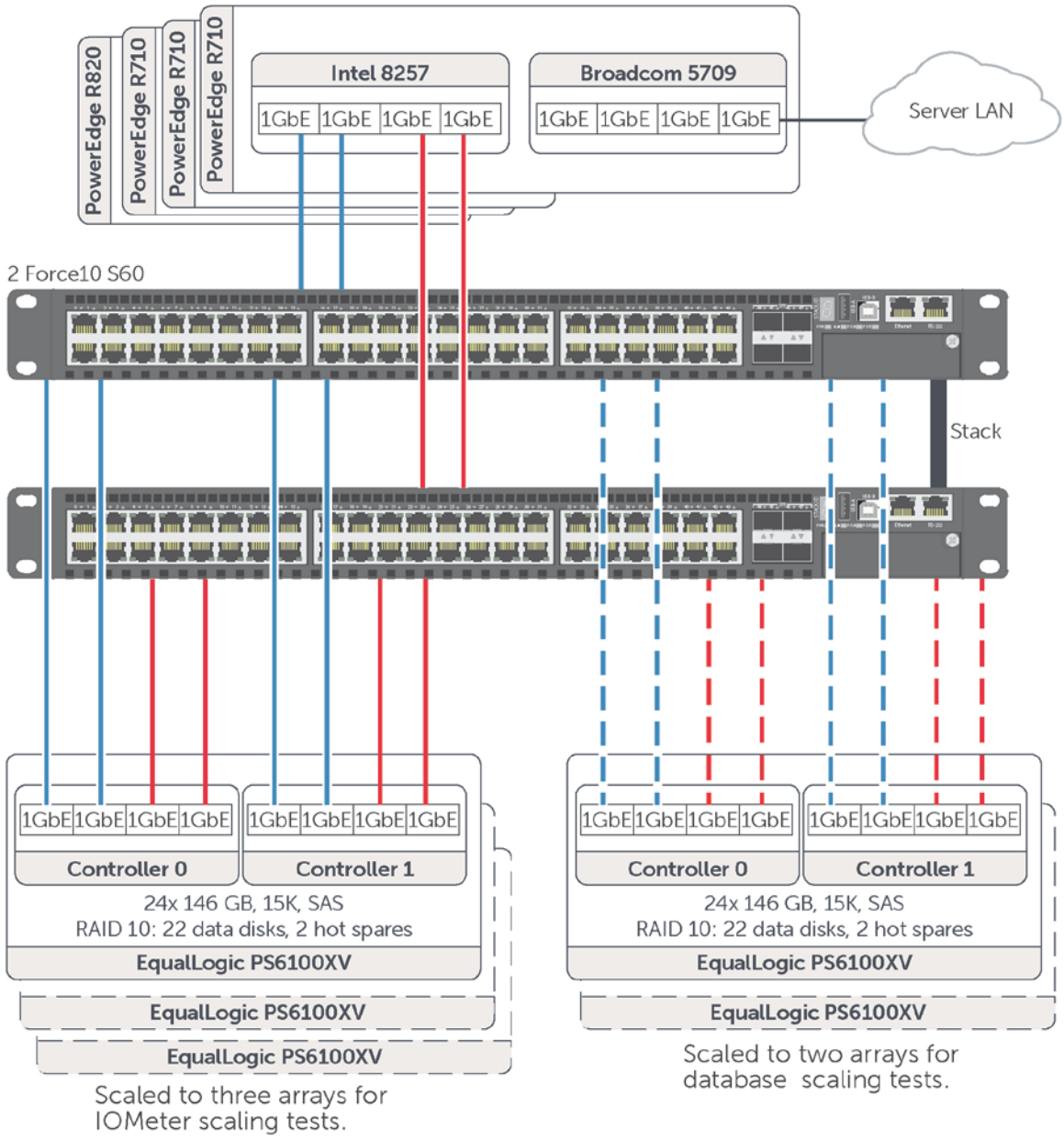


Figure 1 SQL Server LAN and iSCSI SAN connectivity

4.2 High-level system design

A high-level overview of the dell infrastructure components used for the test configuration is shown in Figure 2.

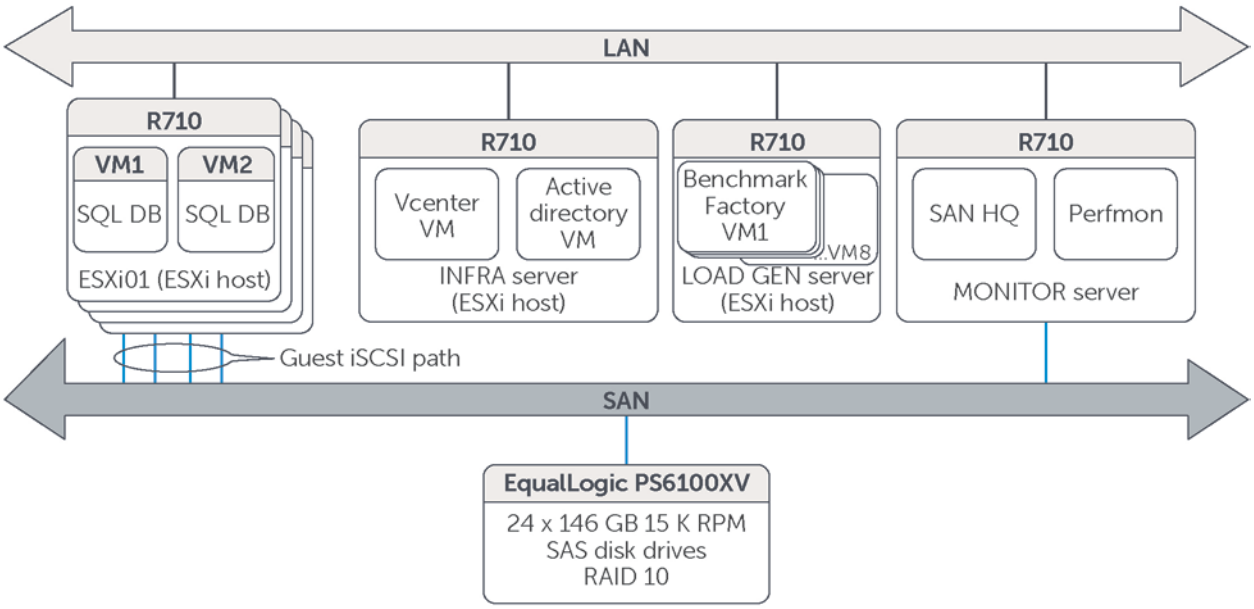


Figure 2 High-level overview of test configuration

Key design details of the test system configuration shown in Figure 2 include:

- Three R710s and one R820 Dell™ PowerEdge™ servers were used to host eight SQL Server VMs. Each of these VMs (SQL DB) had SQL Server 2012 Enterprise Edition installed on Windows Server 2008 R2 SP1 Enterprise Edition.
- Each virtual machine that hosted SQL Server 2012 was configured to use four virtual CPUs and 34 GB of reserved memory. 32 GB was allocated to the SQL Server by specifying the maximum server memory setting in SQL Server Management Studio.
- An example network configuration detail for ESXi01 Host with SQL Server is shown below. The numbered steps were followed to configure the LAN and SAN connectivity for other ESXi hosts (ESXi02, ESXi03 and ESXi04) that accommodated the SQL Server VMs.

1. The on board four port LOM (LAN on motherboard) Broadcom 5709 network controller was used for the Server LAN connection paths via a virtual switch, vSwitch0 created for LAN connectivity (refer to Figure 3).
2. An additional Intel Gigabit VT Quad Port network adapter was installed in the server and used for the connection paths between the database server (SQL DB) and the volumes on the PS6100XV array. As shown in Figure 4, these four NIC ports were assigned on the physical server to be used as uplinks to vSwitch1 for iSCSI SAN connectivity.

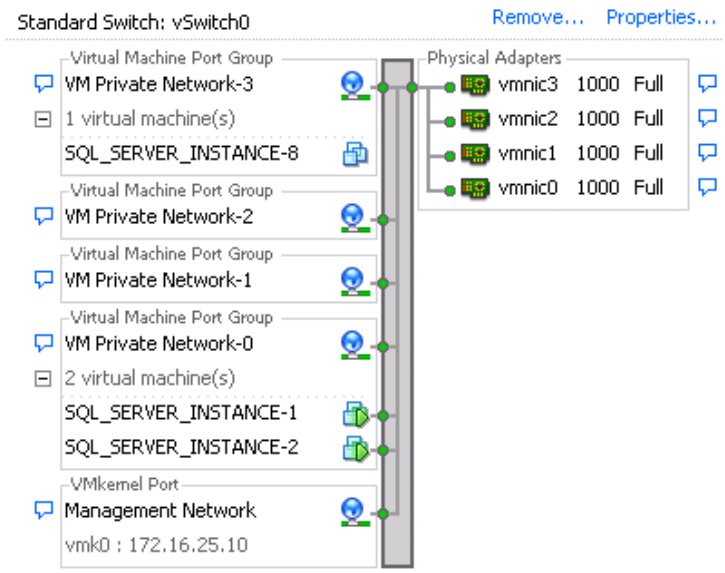


Figure 3 vSwitch0 LAN Configuration for ESXi 01 Host

3. A separate vSwitch (vSwitch1) was created for iSCSI SAN connectivity.
4. Virtual network adapters (type VMXNET 3) within the VM were created and were assigned to the vSwitch1 (refer to Figure 4) on the vSphere host. EQL MPIO DSM was used via Host Integration Tools (HIT) Kit to setup multiple paths from the guest VM to the storage volumes. These paths are labeled "Guest iSCSI Path" in Figure 2.

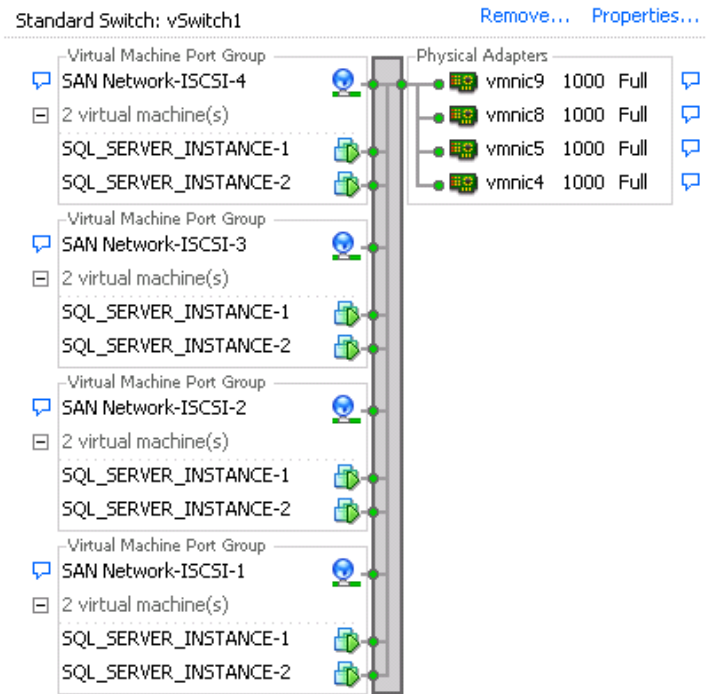


Figure 4 vSwitch2 SAN Configuration for ESXi01 Host

- ESXi 5 was installed on the servers that were used to deploy the SQL Server databases. The server disks were configured as RAID 5, and the guest virtual machine OS disk partitions were also hosted within the virtual machine file system on the disks.
- The INFRA server marked in Figure 2 had VMware ESXi 5 installed and hosted virtual machines for vCenter and Active Directory.
- The LOAD GEN server marked in Figure 2 had VMware ESXi 5 installed and was used to host eight Windows 2008 R2 workload simulation virtual machines with each running an instance of Benchmark Factory.
- The MONITOR server was a PowerEdge R710 running Windows 2008 R2 natively. It was used to host SAN HQ and Perfmon.
- The SAN switches consisted of two Dell™ Force10™ S60 switches configured as a single stack. Redundant connection paths were created from each array controller to each switch in the stack.
- Two EqualLogic PS6100XV arrays consisting of 24 x 146GB 15K RPM SAS disk drives in a RAID 10 configuration were used to host SQL Server database volumes.

5 I/O profiling using IOMeter

I/O profiling tests were executed using IOMeter, to establish baseline I/O performance characteristics of the test storage configuration before deploying any databases. Refer to Table 1 for the workload parameters used for this testing.

The baseline IOPS numbers were established by simulating:

- Different I/O block sizes
- Different RAID policies on the EqualLogic PS Series array
- Scaling the EqualLogic PS Series arrays from one to three

5.1 Block size and capacity utilization I/O studies

In this test, the IOPS were measured while evaluating the EqualLogic PS6100XV array running I/O patterns with different block sizes and read/write ratios. A series of IOMeter tests were executed with different I/O block sizes and read write percentages simulating database-like transactions. These tests were executed at increasing queue depths (number of outstanding I/Os) to determine the maximum IOPS the storage array would sustain within the 20 ms latency (read and write latencies measured separately).

In addition, three configurations were used to determine the IOPS from the PS6100XV array at different capacity utilization levels using varying volume counts and creating a baseline for comparison during the actual database deployment. Testing with these three array capacity utilizations simulated different customer environments with different array capacity utilization levels of 40%, 62%, and 85%. The configuration parameters for the test are shown in Table 1.

Table 1 Test parameters: I/O workload studies

Configuration Parameters	
EqualLogic SAN	One PS6100XV (2.5", 24 15 K SAS drives,146 GB)
Volume configuration #1	Six volumes, 100 GB each (40% of the PS6100XV capacity filled)
Volume configuration #2	Nine volumes , 100 GB each(62% of the PS6100XV capacity filled)
Volume configuration #3	12 volumes , 100 GB each(85% of the PS6100XV capacity filled)
RAID type	RAID 10
OLTP Workload Parameters	
I/O mix	**I/O block size(KB)**
100% read	8 K, 16 K, 32 K, 64 K
80% read/20% write	8 K, 16 K, 32 K, 64 K
70% read/30% write	8 K, 16 K, 32 K, 64 K
60% read/40% write	8 K, 16 K, 32 K, 64 K

The results collected from the tests are illustrated in Figures 5, 6, and 7. Even though 100% random read is not a typical OLTP database I/O pattern, this test was conducted to measure maximum small random read only IOPS achieved by the storage system. A mix of 70% read and 30% write random 8 K I/O represents a majority of OLTP database workloads. However, some OLTP workloads vary that use larger block sizes. Therefore, random I/O tests using 16 K, 32 K, and 64 K block sizes were executed to measure I/O performance under varying workloads. In addition to 70/30 read/write mix, 60/40 and 80/20 read/write mix were also simulated for obtaining baseline IOPS with those mixes.

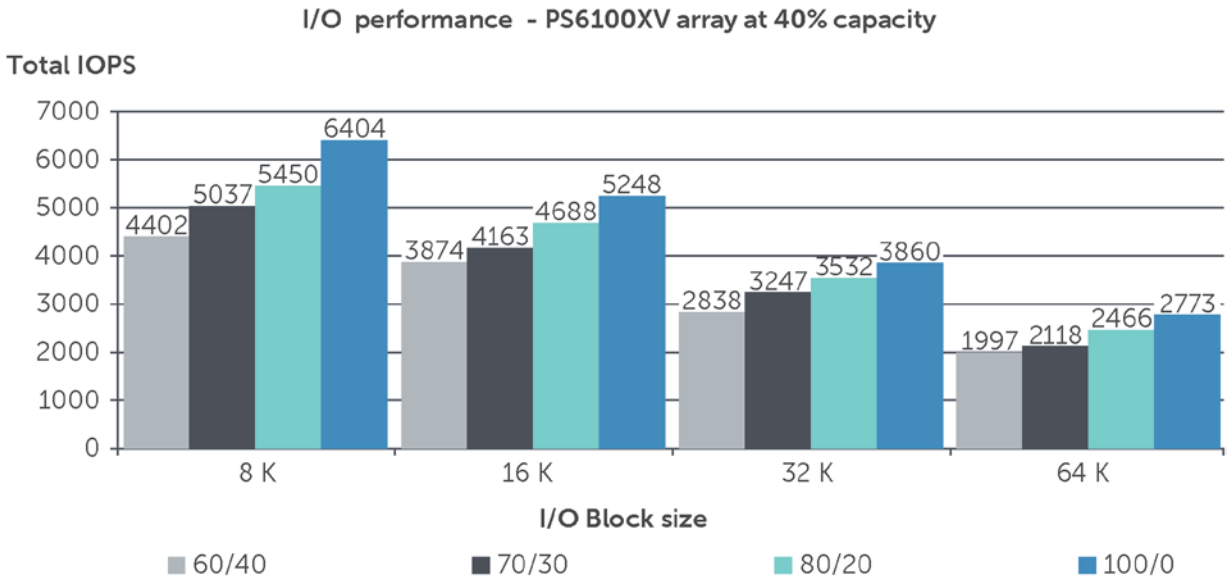


Figure 5 IOPS numbers for PS6100XV at 40% capacity utilization

Data in the PS6100XV array volumes with RAID 10 consumed almost 40% of the array's available capacity. The tests on that array produced approximately 5,037 IOPS for an 8 K block size with a 70/30 read/write mix workload while staying within the generally accepted disk latency limit of 20 ms (for both read and write IOPS measured separately). For block sizes as large as 64 K, a single 6100XV array was able to sustain approximately 2118 IOPS for the same 70/30 read/write mix workload. Figure 5 above illustrates that as the I/O block sizes increased, the IOPS reduced as expected.

All of the different read/write mixes performed in this test showed a decrease in IOPS as the I/O block size progressed from 8 K to 64 K as expected. Larger block sizes mean more data can be transferred with fewer transfers due to higher throughput.

The test results graphed in Figure 6 show the data in the volumes on a PS6100XV array with RAID 10 consumed almost 62% of the available capacity. These volumes also produced approximately 4,278 IOPS for a workload with 8 K block size and 70/30 read/write mix while staying within the generally accepted disk latency limit of 20 ms (for both read and write IOPS measured separately). For block sizes as large as 64 K, a single 6100XV array was able to sustain approximately 2,074 IOPS for the same workload of 70/30 read/write mix. Figure 6 also illustrates that as the I/O block sizes increased the IOPS reduced as expected.

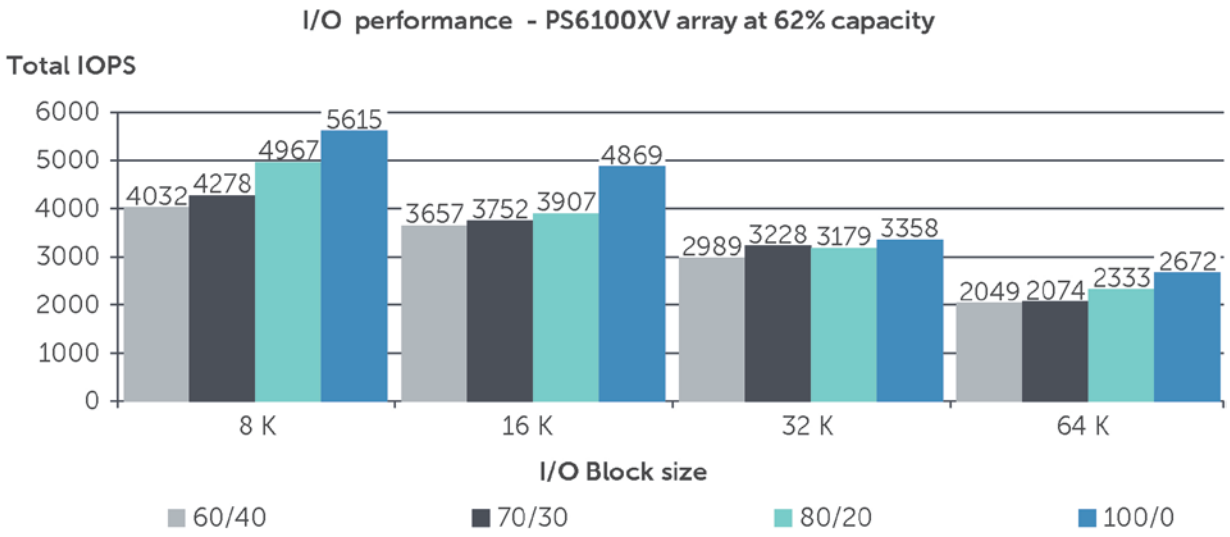


Figure 6 IOPS numbers for PS6100XV at 62% capacity utilization

The test results in Figure 7 show the data in the volumes on a PS6100XV array with RAID 10 consumed almost 85% capacity utilization. These volumes also produced approximately 4,175 IOPS for an 8 K block size for a workload of 70/30 read/write mix, while staying within the generally accepted disk latency limit of 20 ms (for both read and write IOPS). For block sizes as large as 64 K, a single 6100XV array is able to sustain approximately 1,858 IOPS for the same 70/30 read/write mix workload.

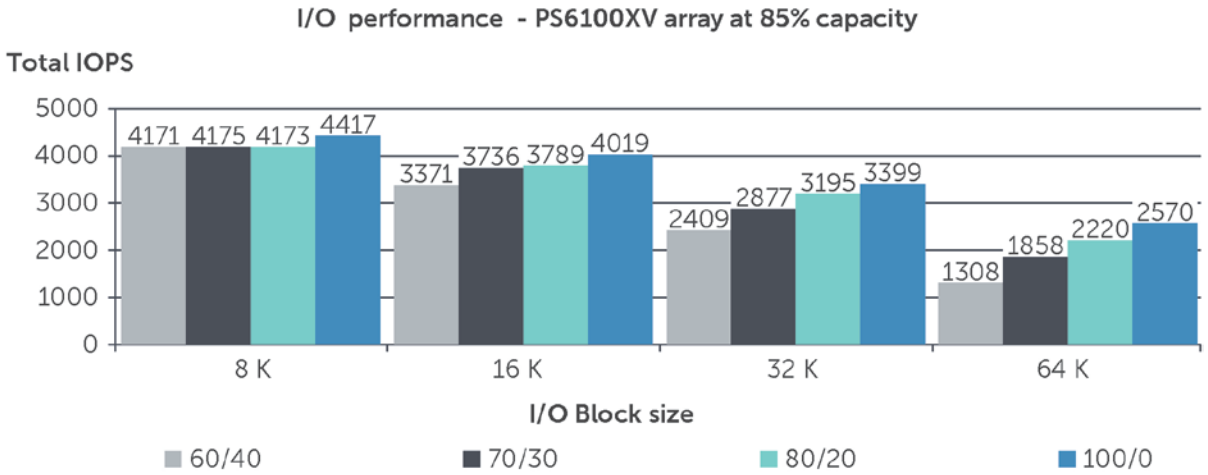


Figure 7 IOPS numbers for PS6100XV at 85% capacity utilization

Figure 8 below compares the I/O performance of a specific 70/30 read/write mix for an 8 K block size across different array capacity utilization levels.

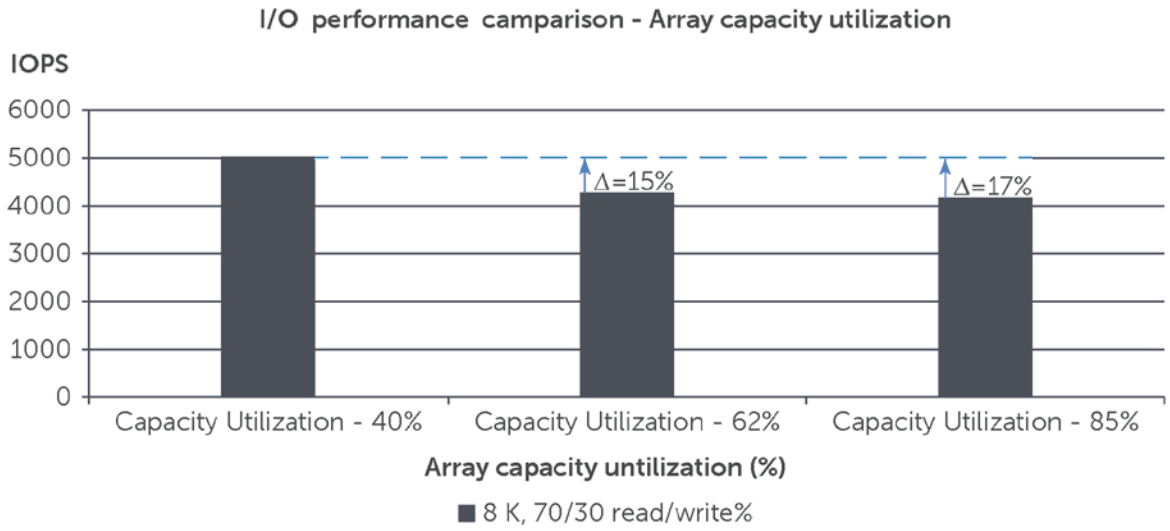


Figure 8 IOPs comparison for different array capacity utilizations

IOPS decrease when the PS6100XV array is almost 85% full, compared to 40% full because of data stored across the inner tracks/sectors of the disk drive as well as the outer sectors. The seek time, or I/O latency, for those blocks becomes higher resulting in latency exceeding 20 ms. With 40% capacity utilization, most of the data set sits on the outer sectors resulting in lower seek time or I/O latency leading to more IOPS under 20 ms. This effect is also due to the large dataset that is randomized and spread across more disk space and also the diminishing returns of the fixed size cache in the array as the data set size increases.

From the test results, when the array capacity utilization is above 85%, it is advisable to add another EqualLogic array to scale the capacity and performance. The 85% capacity utilization test was performed to show how the performance would look in a customer's setup that needed growth. At 40% array capacity utilization, the disks are partially filled and the array is not fully consumed. Data was stored across the outer tracks/sectors of the disk drives resulting in reduced seek time, or I/O latency (refer to Figure 8). Even though this yielded better performance, it is not recommended to use the 40% capacity utilization IOPS numbers to size the storage deployment. The 60-70% array capacity utilization would be the recommended utilization percentage to maintain good performance and also to plan storage sizing.

Note: In the case of a PS Series group, the recommended minimum free space in a storage pool is 5% of total capacity or 100 GB per member, whichever is less.

5.2 RAID studies

This test compared the IOPS values while implementing different RAID policies on the EqualLogic PS6100XV array by running a single workload. The configuration parameters for the test are shown in Table 2.

Table 2 Test parameters: RAID policies variation

Configuration Parameters	
EqualLogic SAN	One PS6100XV (2.5", 24 15K SAS drives,146 GB)
Volume configuration	12 volumes - 100 GB each
RAID policy	RAID 10/ RAID 50/ RAID 6
OLTP workload parameters	
I/O Mix (Read%/Write %)	70/30
I/O block size (KB)	8 K

The EqualLogic PS6100XV array provides a range of different RAID policies along with spare drives to protect data during failures. Each RAID level offers a distinct set of performance and availability characteristics dependent on the nature of the RAID policy and the workload applied. Refer to the white papers *EqualLogic PS Series Storage Arrays: Choosing a Member RAID Policy* at http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/19861480/download.aspx and the *EqualLogic Configuration Guide* at http://en.community.dell.com/techcenter/storage/w/wiki/2639.equallogic-configuration-guide.aspx

The IOMeter test results collected by running the workload against the different RAID levels are illustrated in Figure 9. The total IOPS achieved at each RAID level are also shown in the figure. When a workload of 8 K block size with 70/30 read/write mix was utilized, the disk latency stayed below the generally accepted limit of 20 ms for both read and write IOPS.

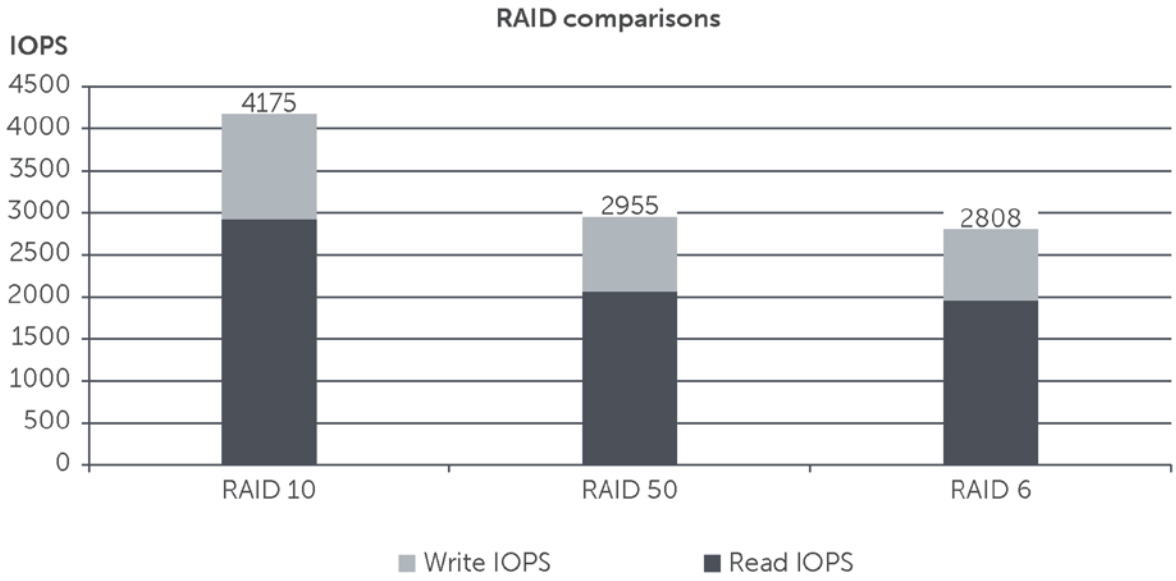


Figure 9 RAID comparisons

The results displayed in Figure 9 confirm a higher IOPS performance in RAID 10 compared to RAID 50 and 6 for a random workload. RAID 10 offered approximately 41% more IOPS compared to RAID 50 and 48% more IOPS compared to RAID 6 for a 70/30 read/write mix with 8 K block size. This is due to the performance cost of the write penalty posed by the distributed parity in RAID 50 and dual parity in RAID 6. For the actual database tests that follow these baseline tests, RAID 10 was used due to its higher performance for OLTP random workload.

5.3 SAN scaling studies

This test measured the baseline scalability of I/O performance as the number of EqualLogic PS6100XV arrays within a group increased from one to three before deploying the databases. The configuration parameters for the test are in Table 3.

Table 3 Test parameters: SAN scaling

Configuration Parameters	
EqualLogic SAN	Three PS6100XV (2.5", 24 15K SAS drives,146 GB)
Array configuration #1	One PS6100XV (in one EqualLogic storage Pool)
Array configuration #2	Two PS6100XV (in one EqualLogic storage Pool)
Array configuration #3	Three PS6100XV (in one EqualLogic storage Pool)
Volume Configuration	12 volumes - 100 GB each (85% array capacity utilization)
RAID Policy	RAID 10
OLTP Workload Parameters	
I/O Mix	**I/O block size (KB)**
70% Read/ 30%Write (random)	8 K

The results collected from the SAN scaling studies using IOMeter are reported in Figure 10. The number of outstanding I/Os per target was increased in IOMeter to push more load to the arrays, as the second and third PS6100XV arrays were added. The IOPS numbers maintained the generally accepted disk latency limit of 20 ms (for both read and write latencies measured separately) for random workload.

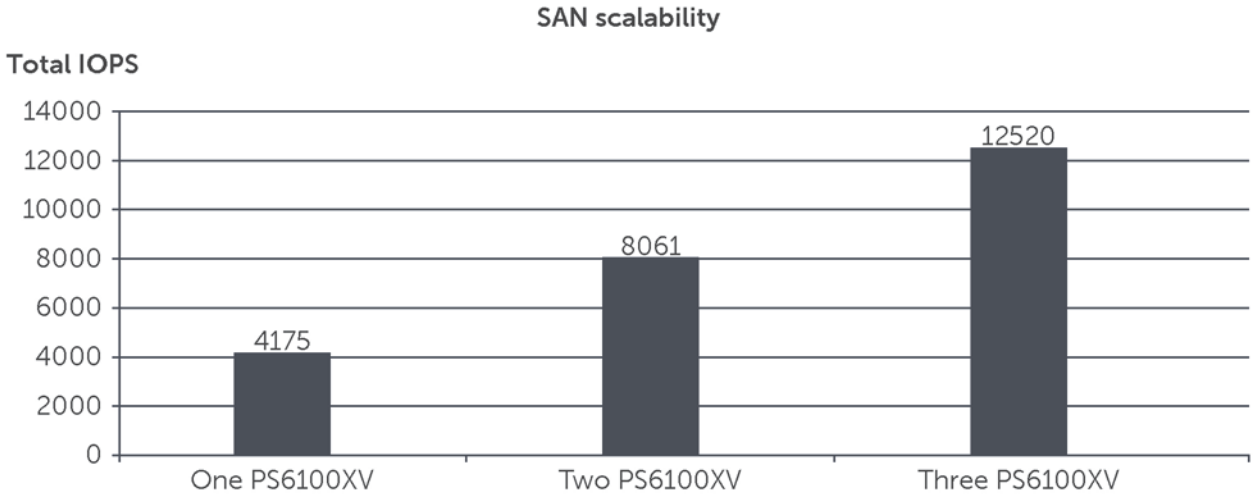


Figure 10 SAN scalability study

As expected, IOPS scaled linearly with the addition of more arrays. This is because the EqualLogic peer storage architecture scaled the available storage resources to provide the required IOPS. As arrays were added, the storage processing resources in the EqualLogic storage architecture also scaled to include the number of disk drives, storage process controllers, on-board cache, and number of storage network interfaces.

6 OLTP performance studies using TPC-E like workload

A series of OLTP application simulations were conducted using Benchmark Factory as the simulation tool running a TPC-E type workload. TPC-E is an industry standard benchmark for OLTP. OLTP like user transactions were run from Benchmark Factory to understand the I/O behavior at the storage arrays when the SQL Server database executed those queries. The test criteria used for the study was:

- Database access latencies (read and write) remain below 20 ms per volume
- Database log latency remain below five ms
- SQL server CPU utilization remain below an 80% average
- TCP retransmits on the storage network remain below 0.5%
- Application response times remain below two seconds

6.1 Database Files and volume layout studies

EqualLogic PS Series storage arrays are self-managing virtualized storage systems. EqualLogic storage simplifies the deployment and administration of consolidated storage environments by enabling perpetual self-optimization with automated load balancing across disks, RAID sets, connections, cache, and controllers. However, this test was performed to evaluate the differences between spreading the SQL Server database data files across multiple volumes versus just one volume.

In order to test this, a baseline test was first performed by having all the data in one huge file in a single volume. The log file and tempdb data files were located on separate volumes.

For the multiple database data files test, the SQL Server automatically distributed the data. The database data was split into multiple data files and each data file was placed in a volume. The database data was split across multiple files by:

- Creating a user-defined secondary file group and then three, four, five, and six .ndf files. Each file was placed on a separate volume.
- The primary file group had the .mdf file and the .ndf data files were all in the secondary file group which was set as the default file group.
- The database was created this way and populated with TPC-E like database data from Benchmark Factory.

A single database was used for this test and the configuration parameters were set as shown in Table 4.

Table 4 Test parameters: Database file and volume layout studies

Configuration Parameters	
EqualLogic SAN	One PS6100XV (2.5", 24 15K SAS drives,146 GB)
RAID Policy	RAID 10
Baseline SQL DB volume Configuration	
SQL DB Data	One 150 GB volume
SQL DB Log	One 52 GB volume
SQL Temp DB	One 20 GB volume
Multiple SQL DB data files and volume Configuration	
3 Files (1-mdf and 2-ndf files)	One 51 GB volume with one mdf file Two 75 GB volumes with one ndf file in each
4 Files (1-mdf and 3-ndf files)	One 51 GB volume with one mdf file Three 51GB volumes with one ndf file in each
5 Files (1-mdf and 4-ndf files)	One 51 GB volume with one mdf file Four 51 GB volumes with one ndf file in each
6 Files (1-mdf and 5-ndf files)	One 51 GB volume with one mdf file Five 51 GB volumes with one ndf file in each
SQL DB Log	One 51 GB volume
SQL Temp DB	One 20 GB volume
OLTP Workload Parameters	
User Transactions	TPC-E from Benchmark Factory for Databases
Number of users	Four
Test Duration	2.5 Hours
SQL Server VM Parameters	
Max SQL Server memory setting (GB)	32
VCPUs	Four
Database Size	~140 GB (including tables, indexes)

For the OLTP tests with TPC-E type workload, the impact of change in multiple data files on the following parameters was measured.

- IOPS
- User Transactions per second (TPS)
- Data/log volume storage latencies

6.1.1 IOPs, TPS and data/log volume latencies

Figure 11 shows the IOPs and user TPS for the multiple database data files test. The read/write mix was about 70/30% and a single database was deployed in an array. The TPS was collected at the Benchmark Factory application for multiple test runs with one (baseline), three, four, five, or six files. In addition to TPS, the data volumes read/write IOPS and log volume write IOPS were also measured using Perfmon for the volumes appearing as drive letters at the SQL Server. These two parameters were used to evaluate if the data spread across multiple volumes showed IOPS and TPS improvement for a constant user load compared to having all the data in a single file or volume.

Similar transaction rates, data read/write IOPS, and log IOPS were achieved for the tests where the database data was spread across three, four, five, and six files and volumes with a 1:1 ratio rather than having all the data in a single file and volume for a constant user load. By default, EqualLogic arrays enable perpetual self-optimization with automated load balancing across disks, RAID sets, connections, cache, and controllers.

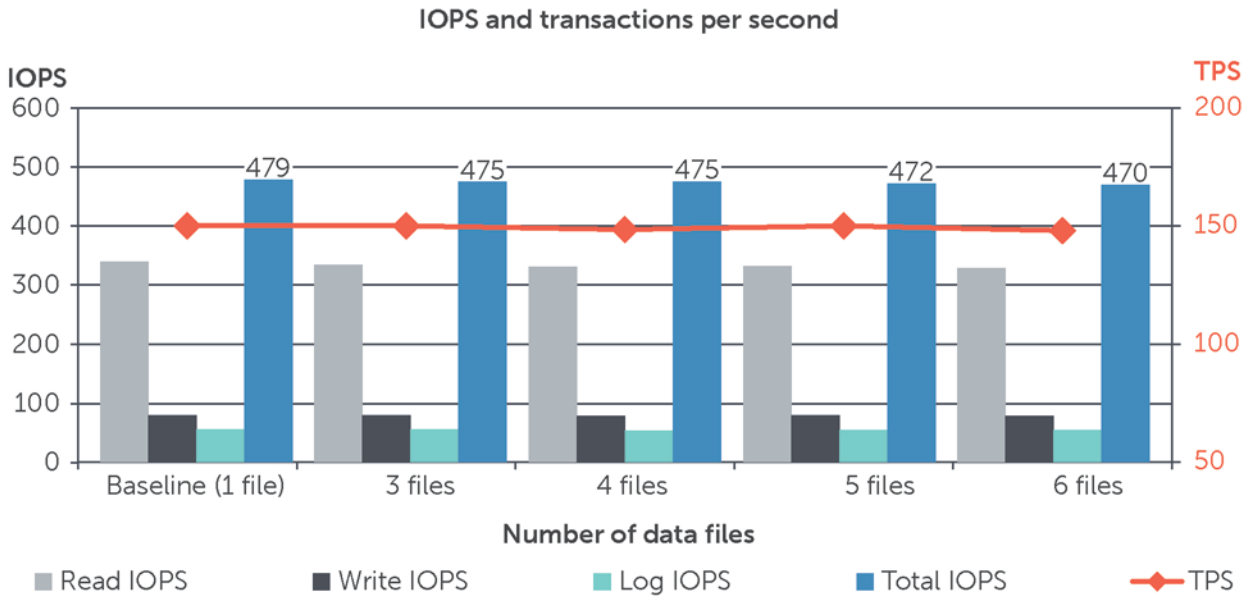


Figure 11 Multiple File/volume lay-out vs IOPS and Transactions per second

Figure 12 shows the data volume read/write latencies and log volume write latency. These were measured using Perfmon for the volumes appearing as drive letters at the SQL Server. As shown in Figure 12, similar data read/write latencies and log latencies were achieved for the tests where the database data was spread across three, four, five, and six files and volumes with a 1:1 ratio rather than having all the data in a single file and volume.

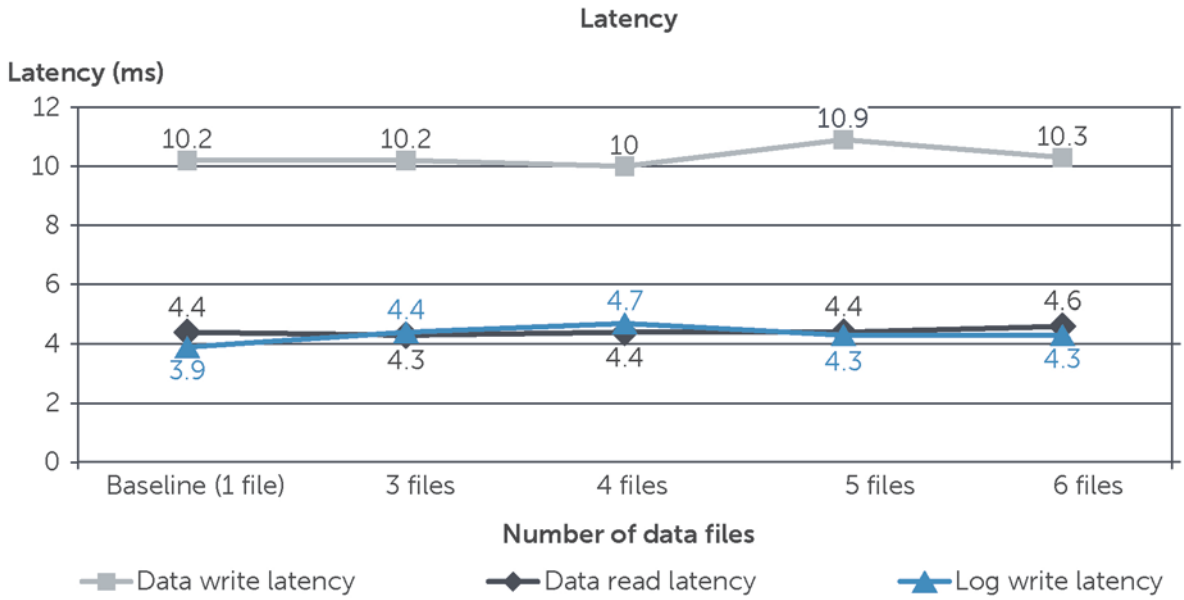


Figure 12 Multiple file/volume lay-out versus latencies

From the test results performed in section 6.1.1, where the SQL Server automatically split the data, the performance benefit achieved by spreading the database data across multiple files and volumes was insignificant. Reasons for this are:

- EqualLogic volumes are already virtualized and data distribution is handled automatically at the storage level.
- More data files require more requestors/threads. Traditional storage may benefit from this, but EqualLogic storage is already virtualized and the distribution is handled automatically. Therefore, further optimizing the database layout into multiple files and volumes may yield very little performance gains.

As seen in the above figures, even though splitting the database files over multiple volumes does not provide significant performance gains for this single OLTP like workload test, there are other advantages of distributing database files and logs across file groups and volumes. For example, splitting files can provide more flexibility when backing up and restoring data. Refer to the white paper *Microsoft SQL Server 2008 Backup and Restore using Dell EqualLogic* at http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/20063707/download.aspx. In addition, use of multiple volumes would be beneficial in a larger multi-application SAN for better load balancing of different application data.

6.2 Table partitioning studies

Partitioning breaks up database objects by allowing subsets of data to reside on separate file groups. This can be beneficial in several ways.

- Improved scalability and manageability - When tables and indexes become very large, partitioning can help by dividing the data into smaller, more manageable sections.
- Improved performance and availability – For large tables with multiple CPUs on the system, partitioning the table can lead to better performance through parallel operations.
- Reduced costs – Partitioning large tables can allow moving less valuable and accessed data lower-cost storage without impacting the entire table.

This test evaluated performance differences between row partitioning on the largest most accessed database table and spreading the database into multiple files without table partition.

The first step in table partitioning is identifying the best candidates for partition by:

1. Running a SQL Profiler trace and saving the trace into a trace file or as a database table. The trace file can be exported into a SQL Server table and SQL code can be used to check the most frequently executed T-SQL queries and stored procedures. Details are provided in Appendix B.
2. T-SQL the query that gives the statistics along with query execution plans for the top N most frequently executed queries. http://msdn.microsoft.com/en-us/library/windowsazure/hh977102.aspx
3. For the index and partition recommendations, SQL database tuning advisor can be used by providing the SQL profiler trace file as input.

Steps 1 and 2 were followed to identify the E_Trade table in the database as the largest table, generating the most logical reads (meaning it was the most accessed). The column ID used was T_ID. This table was chosen to perform range partition (For more information, refer to http://msdn.microsoft.com/en-us/library/ms345146%28v=sql.90%29.aspx) with the left boundary on the column ID T_ID which was also the primary key for the table. In order to check if partitioning, the table would yield better performance through parallel operations due to multiple VCPUs (four at the SQL Server). Tests were performed to compare the results of partitioning the table into four versus eight partitions.

The primary file group had the primary file (.mdf file) with startup information for the database and points to the other files in the database. All the tables, except the E_Trade table, were placed in the secondary file group (1) with a single .ndf data file. For the E_Trade table with four partitions, four new file groups were created and each partition was placed in each new file group. Similarly for the eight partition test, eight new file groups were created and each partition was placed in each new file group. The four and eight table partition data files and volume layout is shown in Figure 13.

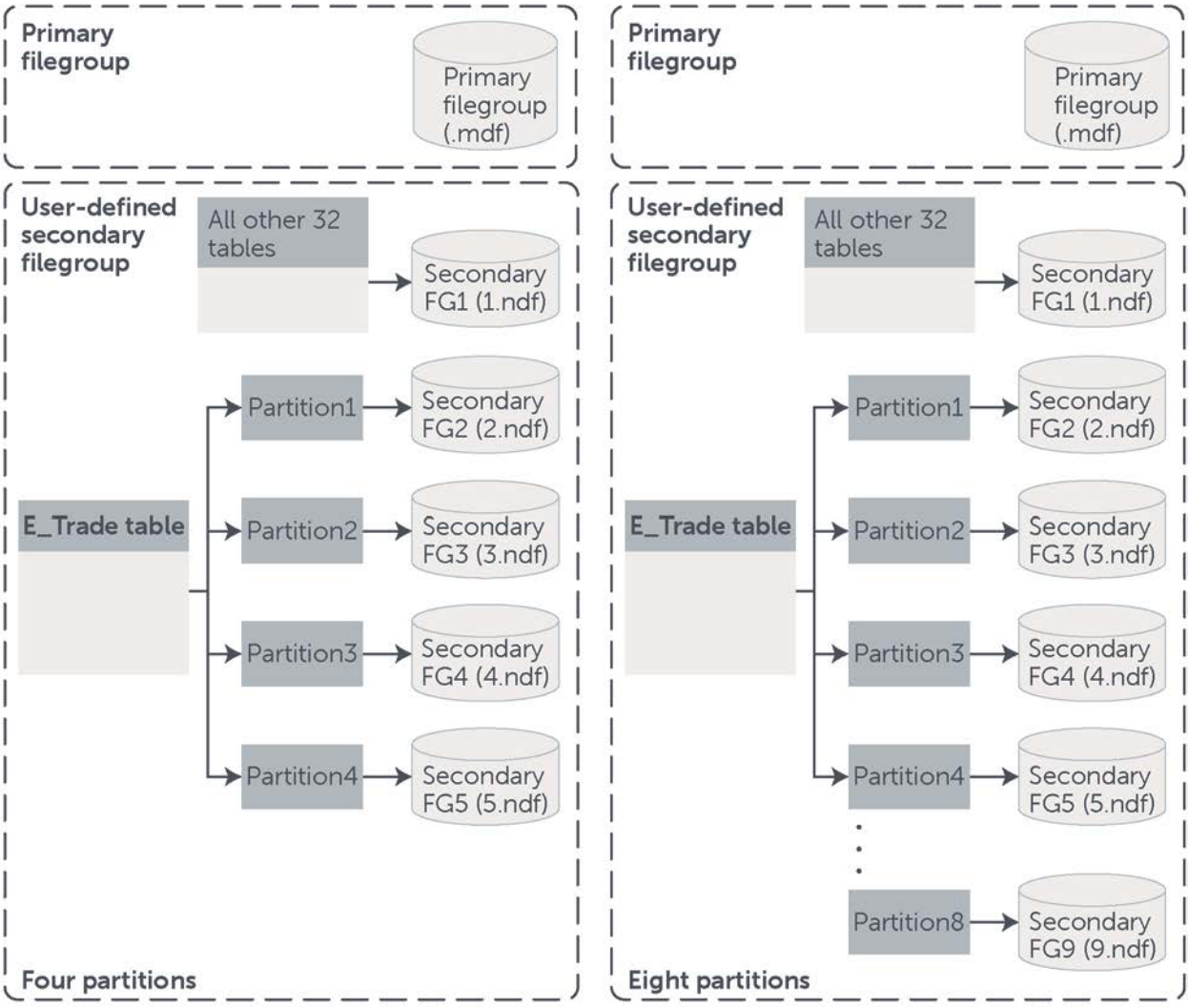


Figure 13 Four and Eight Table Partition data file layout

For this test the configuration parameters were set as shown in Table 5.

Table 5 Test parameters: Table partition studies

Configuration Parameters	
EqualLogic SAN	One PS6100XV (2.5", 24 15 K SAS drives,146 GB)
RAID Policy	RAID 10
Multiple SQL DB data files and volume Configuration	
Four partitions (one .mdf and five .ndf files)	• One 51 GB volume with one .mdf file (primary file group) • One 116 GB volume with one .ndf file (secondary file group 1 with all 32 tables and their indexes) • One 53 GB volume (secondary file group 2 with E_Trade table partition 1 and non-clustered indexes) • Three 6 GB volumes (secondary file groups 3, 4, & 5 with E_Trade table partitions 2, 3, & 4 in each)
Eight partitions (one .mdf and nine .ndf files)	• One 51 GB volume with one .mdf file (primary file group) • one 116 GB volume with one .ndf file (secondary file group 1 with all the 32 tables and their indexes) • One 53 GB volume (secondary file group 2 with E_Trade table partition 1 and non-clustered indexes) • Seven 6G B volumes (secondary file groups 3, 4, 5, 6, 7, 8, & 9 with E_Trade table partitions 2, 3, 4, 5, 6, 7, & 8 in each)
SQL DB Log	One 51 GB volume
SQL Temp DB	One 20 GB volume
OLTP Workload Parameters	
User transactions	TPC-E from Quest Benchmark Factory
Number of users	Four
Test duration	2.5 hours
SQL Server VM Parameters	
Max SQL Server memory setting (GB)	32
VCPUs	Four
Database Size	~140 GB (including tables and indexes)

6.2.1 Four Partitions

Figure 14 below shows the results of partitioning the E_Trade table into four partitions compared to spreading the data into multiple files without partitioning (as seen in section 6.1). There were six data files (one .mdf, and five .ndf files) in both the tests and each file was placed in its own volume. User transactions with constant user load of four concurrent users were run from Benchmark Factory to compare the results. The read and write latencies of data volumes and the write latency of the log volume were measured using Perfmon at the SQL Server where these volumes were exposed as NTFS volumes with drive letters.

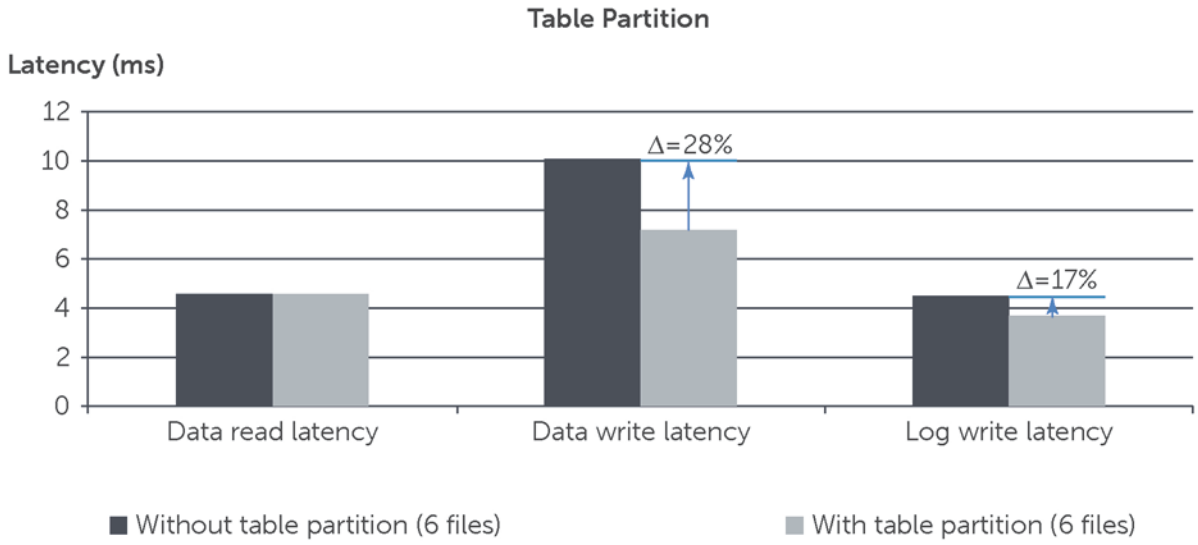


Figure 14 With and without Table partition comparison (4 partitions)

Figure 14 illustrates that partitioning the largest and most accessed table into smaller tables reduced the data volume write latency by 28% and log write latency by 17% compared to spreading the database data files into multiple volumes without partitioning. This is because of the benefit achieved by the parallel operations against individual smaller subsets due to table partition especially during the checkpoint write operations. This is understood from the average I/O write rate gathered for each data volume from SANHQ shown in Figure 15. The higher throughput seen on partition 1 in Figure 15 is due to the three non-clustered indexes of the partitioned table residing in that data file.

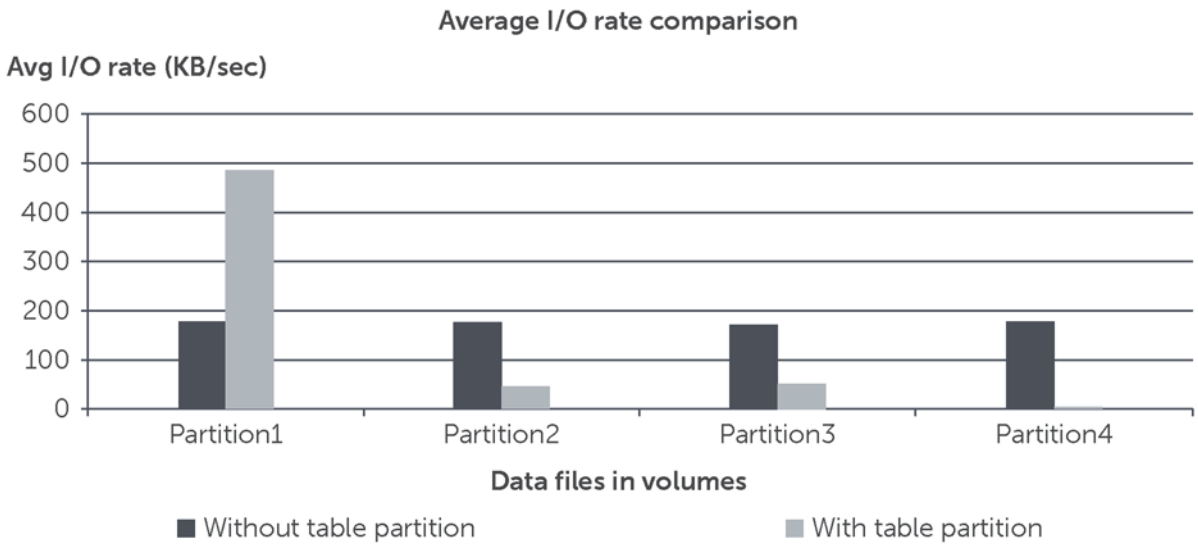
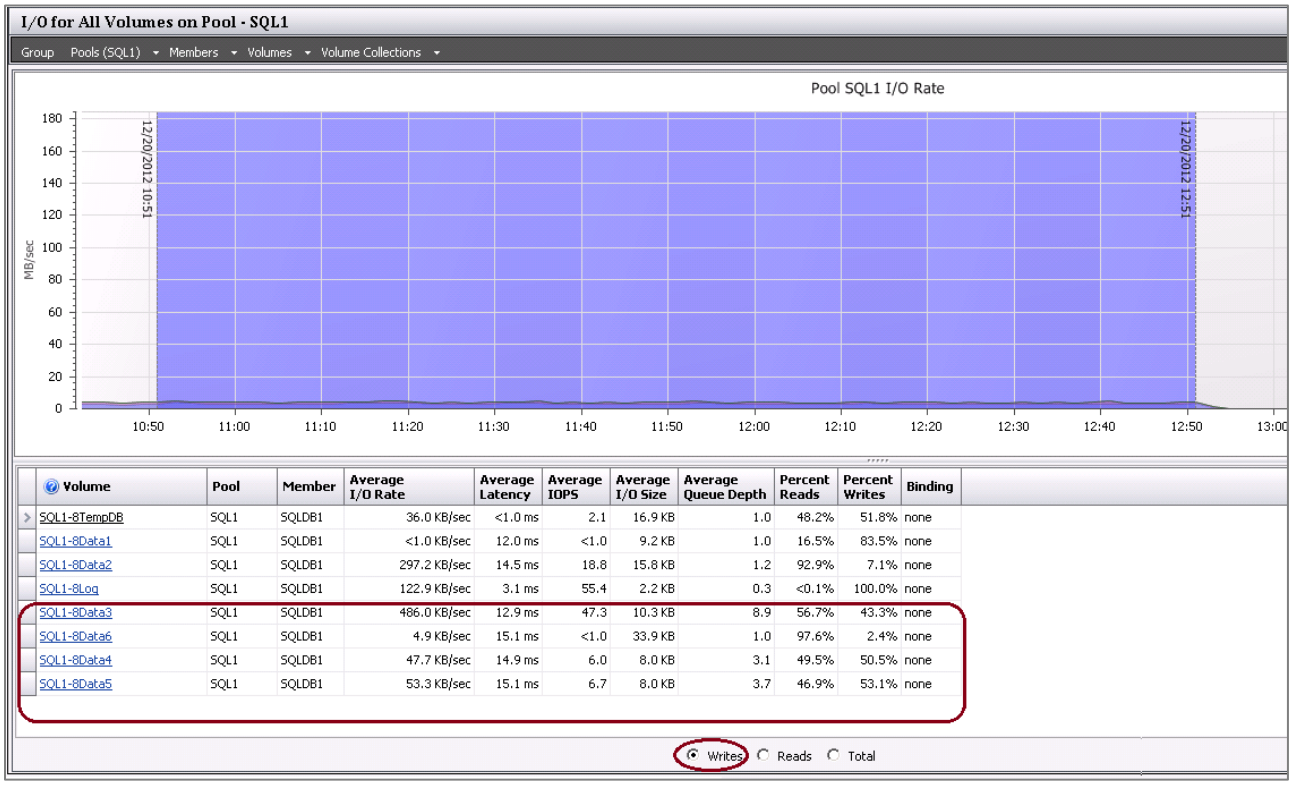


Figure 15 Average I/O rate on each data volume comparison



Volume	Pool	Member	Average I/O Rate	Average Latency	Average IOPS	Average I/O Size	Average Queue Depth	Percent Reads	Percent Writes	Binding
SQL1-8TempDB	SQL1	SQLDB1	36.0 KB/sec	<1.0 ms	2.1	16.9 KB	1.0	48.2%	51.8%	none
SQL1-8Data1	SQL1	SQLDB1	<1.0 KB/sec	12.0 ms	<1.0	9.2 KB	1.0	16.5%	83.5%	none
SQL1-8Data2	SQL1	SQLDB1	297.2 KB/sec	14.5 ms	18.8	15.8 KB	1.2	92.9%	7.1%	none
SQL1-8Log	SQL1	SQLDB1	122.9 KB/sec	3.1 ms	55.4	2.2 KB	0.3	<0.1%	100.0%	none
SQL1-8Data3	SQL1	SQLDB1	486.0 KB/sec	12.9 ms	47.3	10.3 KB	8.9	56.7%	43.3%	none
SQL1-8Data6	SQL1	SQLDB1	4.9 KB/sec	15.1 ms	<1.0	33.9 KB	1.0	97.6%	2.4%	none
SQL1-8Data4	SQL1	SQLDB1	47.7 KB/sec	14.9 ms	6.0	8.0 KB	3.1	49.5%	50.5%	none
SQL1-8Data5	SQL1	SQLDB1	53.3 KB/sec	15.1 ms	6.7	8.0 KB	3.7	46.9%	53.1%	none

Figure 16 Average I/O rate on each data volume -SANHQ

The SANHQ results for the I/O rate on each data volumes are shown in Figure 16.The highlighted volumes contain table partitions. Figure 15 shows the comparison of the average I/O write rate for the database with table partition and the database without table partition. For the database with table partition, the writes occur at a higher rate especially on volumes each containing E_Trade table partitions two, three, and four due to multiple writes occurring closer on the disks, which in turn improves the write throughput and latency. Partitioning improves write coalescence, which in turn reduces the latency incurred for writes. Table and index partitioning can allow improved manageability and performance, but it must be planned and implemented carefully based on every environment.

6.2.2 Eight Partitions

The E_Trade table was range partitioned with range left boundaries into eight partitions and each partition file was placed in a separate volume to check if having more partition files in volumes (files > 4 VCPUs) yields better performance compared to four partitions. There were six data files (one .mdf and five .ndf files) in four partition tests and ten data files (one .mdf and nine .ndf files) in eight partition tests. Each file was placed in its own volume; see Figure 13 for table partition data file layout. User transactions with constant user load of four concurrent users were run from Benchmark Factory in both tests to compare the results. The read and write latencies of data volumes and the write latency of the log volume were measured using Perfmon at the SQL Server where these volumes were exposed as file system drive letters.

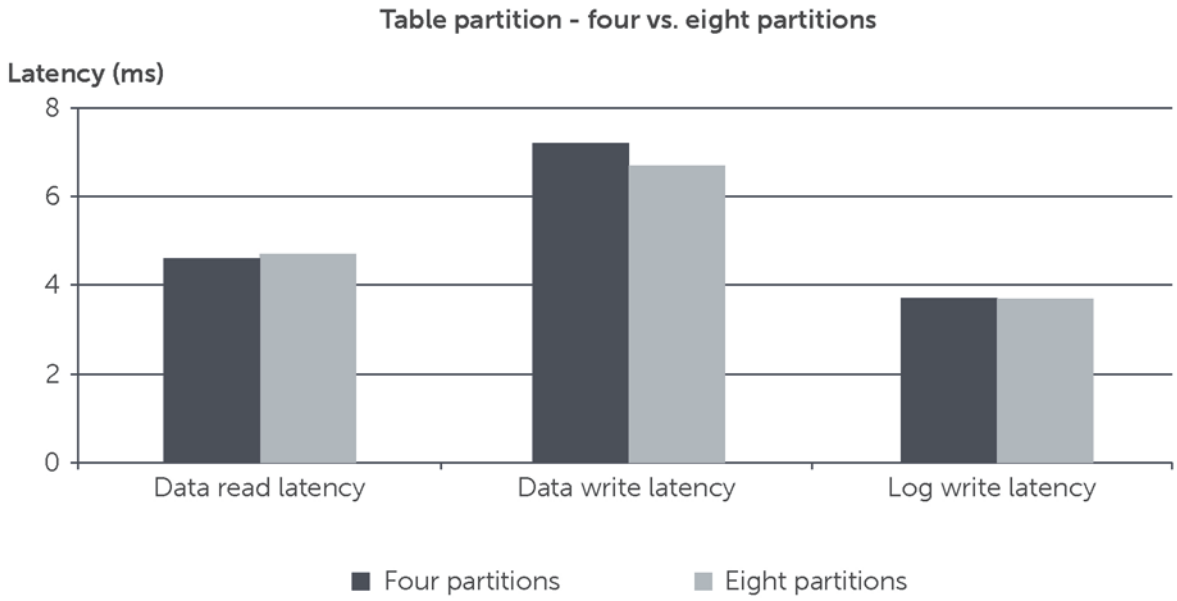


Figure 17 Table partition – four versus eight partitions

As illustrated in Figure 17 and described in section 6.1, there was very little or no performance improvement seen when eight partitions in separate volumes were used. This is because of the diminishing returns with more data partitions and volumes. This can be understood from Figure 18 and Figure 19 where it shows the average I/O write rate gathered from SANHQ for each data volume that contained partitions. The higher throughput seen on partition 1, for both the four and eight cases in Figure 18 is due to the three non-clustered indexes of the partitioned table.

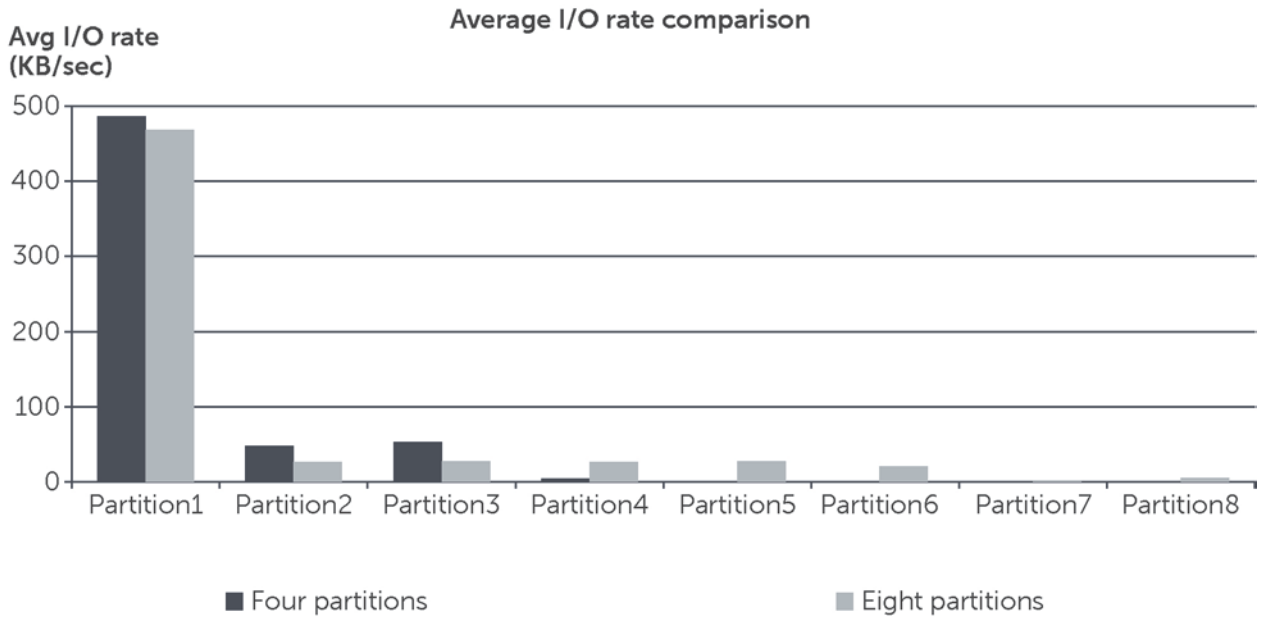
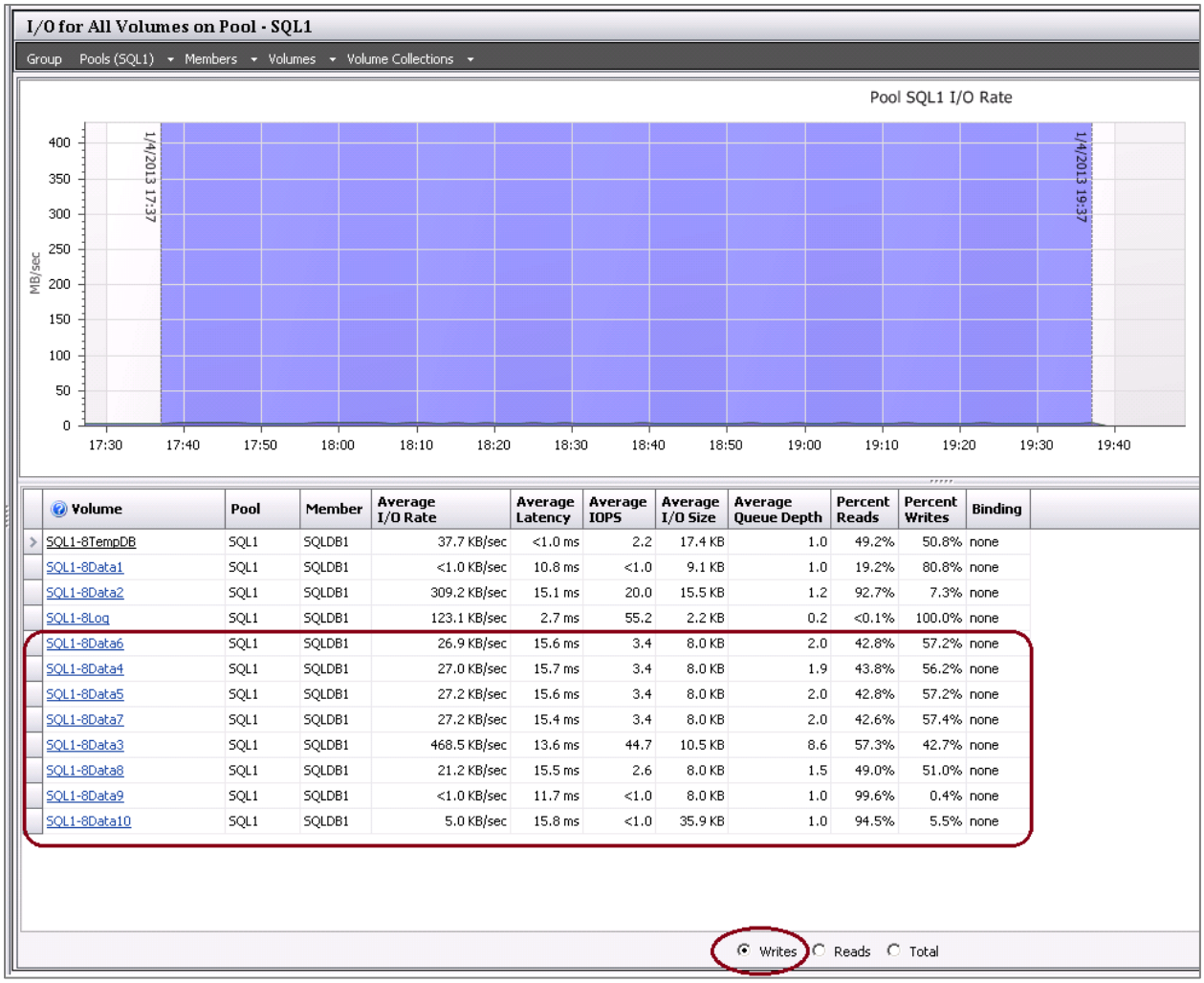


Figure 18 Average I/O rate comparison- four versus eight partitions



Volume	Pool	Member	Average I/O Rate	Average Latency	Average IOPS	Average I/O Size	Average Queue Depth	Percent Reads	Percent Writes	Binding
SQL1-8TempDB	SQL1	SQLDB1	37.7 KB/sec	<1.0 ms	2.2	17.4 KB	1.0	49.2%	50.8%	none
SQL1-8Data1	SQL1	SQLDB1	<1.0 KB/sec	10.8 ms	<1.0	9.1 KB	1.0	19.2%	80.8%	none
SQL1-8Data2	SQL1	SQLDB1	309.2 KB/sec	15.1 ms	20.0	15.5 KB	1.2	92.7%	7.3%	none
SQL1-8Log	SQL1	SQLDB1	123.1 KB/sec	2.7 ms	55.2	2.2 KB	0.2	<0.1%	100.0%	none
SQL1-8Data6	SQL1	SQLDB1	26.9 KB/sec	15.6 ms	3.4	8.0 KB	2.0	42.8%	57.2%	none
SQL1-8Data4	SQL1	SQLDB1	27.0 KB/sec	15.7 ms	3.4	8.0 KB	1.9	43.8%	56.2%	none
SQL1-8Data5	SQL1	SQLDB1	27.2 KB/sec	15.6 ms	3.4	8.0 KB	2.0	42.8%	57.2%	none
SQL1-8Data7	SQL1	SQLDB1	27.2 KB/sec	15.4 ms	3.4	8.0 KB	2.0	42.6%	57.4%	none
SQL1-8Data3	SQL1	SQLDB1	468.5 KB/sec	13.6 ms	44.7	10.5 KB	8.6	57.3%	42.7%	none
SQL1-8Data8	SQL1	SQLDB1	21.2 KB/sec	15.5 ms	2.6	8.0 KB	1.5	49.0%	51.0%	none
SQL1-8Data9	SQL1	SQLDB1	<1.0 KB/sec	11.7 ms	<1.0	8.0 KB	1.0	99.6%	0.4%	none
SQL1-8Data10	SQL1	SQLDB1	5.0 KB/sec	15.8 ms	<1.0	35.9 KB	1.0	94.5%	5.5%	none

Figure 19 Average I/O rate on each volume for eight partitions

With eight partitions, the write activity on these partitioned volumes (one-four) was much less compared to four partitions. Also the combination of the workload characteristics and the dataset used caused diminishing returns.

6.3 SAN scaling

The goal of the SAN scaling test was to measure how I/O performance scales as the number of EqualLogic PS Series storage arrays (members) were increased within a group. The configurations tested included one and two EqualLogic PS6100XV members with the group.

For this test, the four partition configuration shown in the table partition studies (section 6.2.1) was used for its performance improvement of data and log write latencies. For the one array scaling tests, four databases were deployed as shown in Test Configuration#1 (refer Figure 20). Scaling to two arrays doubled the number of databases to push more load on the two arrays. Eight databases were deployed as shown in Test Configuration#2 (refer to Figure 20).

The user load was run from each of the four Benchmark Factory consoles for one array scaling and eight Benchmark Factory consoles for the two array scaling. The volume layout used for the test configuration is shown in Figure 20 and the other configuration parameters for the test are listed in Table 6.

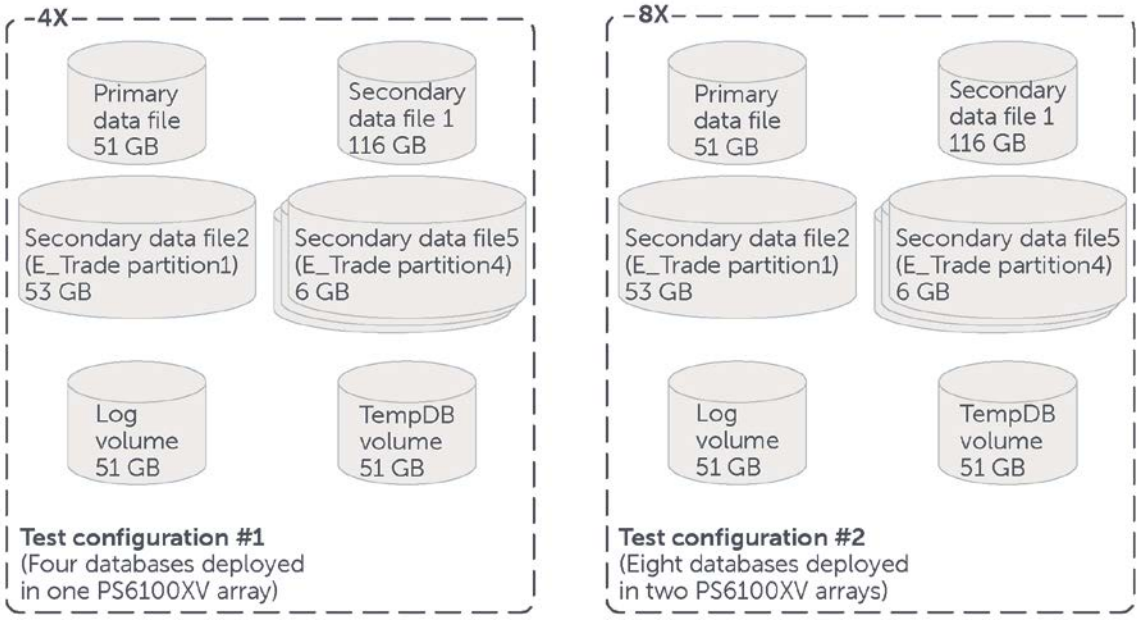


Figure 20 Volume layout for the SAN scaling studies

Table 6 Test parameters: SAN scaling

Configuration Parameters	
EqualLogic SAN	Two PS6100XV (2.5", 24 15 K SAS drives,146 GB)
Test configuration#1 (4XDatabases)	• One PS6100XV (in one EqualLogic storage pool) • Four databases • 80 users from Benchmark Factory • Database volumes: Refer Figure 20
Test configuration#2 (8xDatabases)	• Two PS6100XV (in one EqualLogic storage pool) • Eight databases • 160 users from Benchmark Factory • Database volumes: Refer Figure 20
RAID Policy	RAID 10
SQL Server VM Parameters	
Max SQL Server memory setting (GB)	32
VCPUs	Four

The results collected from the SAN scaling studies using Benchmark Factory are graphed in Figure 21. With one array the capacity utilization was at 85% and with two arrays, the capacity utilization was about 82%. For this test, the read/write ratio was about 70/30 and the IOPS numbers maintained the generally accepted disk latency limit of 20 ms (both read and write latencies measured separately) for random workload, and 5 ms for the sequential log write latency. Each SQL Server VM was allocated with 34 GB of memory with 32 GB allocated to the SQL Server application. The SQL Server memory allocation is established by the maximum server memory setting in the SQL Server Management Studio.

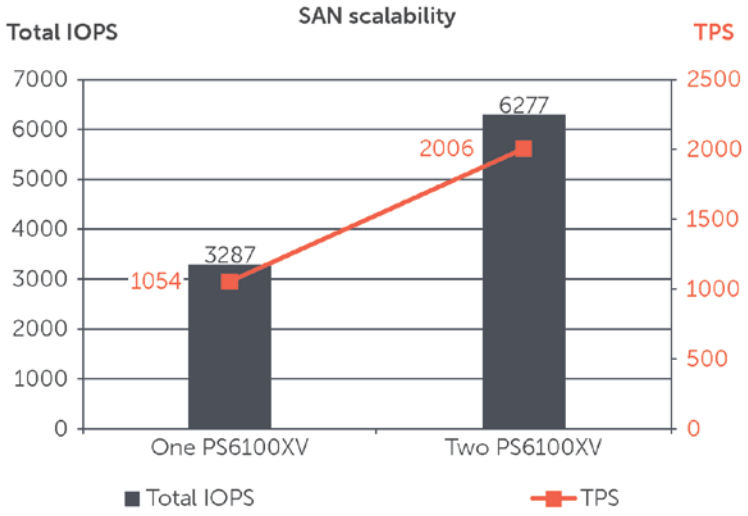


Figure 21 SAN scalability

The IOMeter scaling test results presented in section 5.3 show that a single PS6100XV (15K SAS) array can support up to 4,175 IOPS for 8 K small random I/O at a 70/30 read/write ratio (before exceeding the 20 ms disk latency threshold). Using the more realistic TPC-E Benchmark Factory test, the results presented in Figure 21 show that the same array was able to support 1,054 transactions per second corresponding to an IOPS level of 3,287. A difference in IOPS of around 22% was observed between IOMeter scaling and actual database scaling test. This is due to the fact that there is no application cache involved when using IOMeter, where the Benchmark Factory OLTP database test, the SQL Server buffer cache size plays a major role in optimizing the reads and thus resulting in the difference in IOPs.

As expected, IOPS and the user transactions per second scaled linearly with the addition of more arrays. The EqualLogic peer storage architecture scales the available storage resources linearly to provide the required IOPS. The storage processing resources used by EqualLogic storage architecture to accomplish linear scaling are the number of disk drives, the number storage processing controllers (including on-board cache), and the number of storage network interfaces.

Note: It was due to server memory limitation, that three array scaling was not performed using databases. However, linear scaling results would be achieved when the arrays are scaled to three with 12 databases deployed (Refer to section 5.3).

7 Best practice recommendations

7.1 Storage

Use the following best practices when configuring Dell EqualLogic storage arrays for a data protection solution.

- When possible, use high performance drives for the arrays that host the SQL Server database volumes. For the OLTP test configuration in this paper, 15 K RPM SAS drives were used on the PS6100XV arrays that hosted the database volumes.
- For OLTP applications, RAID 10 offers the best performance in addition to high availability. It also offers best performance when the RAID set is degraded compared to other RAID types.
- For large SQL implementations that require optimal performance, RAID 10 is a good choice for the group RAID level. When planning for I/O intensive workloads such as OLTP, it is also best to place database log files on RAID 10 volumes. The test configuration in this paper used RAID 10 for the PS6100XV array that hosted data and log volumes.

Note: RAID 6 is Dell's recommendation for EqualLogic arrays populated with SATA or NL SAS drives 3 TB or larger. RAID 6 minimizes the risk of data loss during a RAID reconstruction thereby maximizing protection of the RAID set. For more information on EqualLogic RAID policies, refer to the technical report *EqualLogic PS Series Storage Arrays: Choosing a Member RAID Policy* at http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/19861480/download.aspx

General recommendations for deploying SQL server in EqualLogic PS Series array is provided in the document titled *PS Series Grouped deploying Microsoft SQL Server in an ISCSI SAN* at http://www.equallogic.com/uploadedfiles/Resources/Tech_Reports/tr1013-sql-server.pdf

7.2 Network infrastructure

The following are network infrastructure design best practices:

- Since OLTP workloads tend to be random in nature with smaller block sizes and the key metric in measuring performance would be IOPs (rather than throughput for DSS), using a 1GbE connectivity between server and storage would be sufficient.
- Design separate network infrastructures to isolate the LAN traffic from the SAN traffic (iSCSI).
- Design redundant SAN component architectures. This includes the NICs on the servers and switches for the storage network (including server blade chassis switches and external switches).
- Make sure that the server NIC ports and storage NIC ports are connected so that any single component failure in the SAN will not disable access to any storage array volumes.
- Enable flow control on both the server NICs and switch ports connecting to the server and storage ports.
- Enable jumbo frames on the server ports and switch ports.

- On iSCSI SAN switches, spanning tree should be disabled on switch ports connecting to end devices like server and storage ports. The Portfast setting should be enabled in the switch configuration for these ports.

Note: General recommendations for EqualLogic PS Series array network configuration and performance is provided in the document titled *EqualLogic Configuration Guide* at http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/19852516/download.aspx.

7.3 VMware vSphere ESXi Server/VM

The lab test environment for this paper was comprised of a VMware ESXi server to host the SQL Server database virtual machines as well as the Benchmark Factory work load simulation, vCenter, and Active Directory virtual machines. The following best practices are applicable for running VMware ESXi based virtual machines in conjunction with EqualLogic storage and/or Microsoft SQL Server environments.

For these tests, iSCSI SAN storage access was setup for Windows based virtual machines to use a direct access path and the guest OS (Windows) iSCSI initiator

- Select performance optimized network adapters of VMXNET3 type for guest network adapters connected to the SAN network (VMware tools required in the guest operating system).
- Enable jumbo frames (large MTU) for the virtual switches assigned to SAN traffic (iSCSI).
- Enable TSO (TCP segmentation offload) and LRO (large receive offload) in the guest VM NICs for iSCSI traffic.
- When using guest iSCSI initiator, install the Dell EqualLogic Host Integration Tools (HIT) kit for Windows within the guest OS. This installs the EqualLogic DSM for the Windows Server MPIO framework. The DSM provides multi-path optimizations tailored to the EqualLogic storage arrays.

Note: The iSCSI volumes were natively formatted as NTFS and directly accessed within the Windows 2008 R2 Server VM.

When using the VMware ESXi software iSCSI initiator within vSphere host instead of within the virtual machine, install EqualLogic Multipathing Extension Module (MEM) for vSphere to manage iSCSI connection management and load balancing.

7.4 SQL Server best practices

The following best practices were implemented for the tests documented in this paper.

7.4.1 Database volume creation

- Use Basic disk type for all EqualLogic volumes.
- Use default disk alignment provided by Windows 2008 or greater.
- Use NTFS file system with 64 KB allocation unit for SQL database and log partitions.

7.4.2 Buffer cache size

SQL Server buffer cache highly optimizes the disk access. The more database pages found in the buffer cache, the more efficient SQL Server will be in responding to the queries which in turn improves application response times. The change in buffer cache sizes changes the logical I/O access pattern and in turn changes the physical I/O access pattern, and storage device's utilization level. The impact of buffer cache size on the read/write percentage, application response times, IOPS, I/O latency and TPS need to considered when planning for SQL Server storage sizing. Different customer environments will have different sizing requirements. The appropriate buffer cache size to be used for a specific database size can only be determined from experience with an existing monitored setup. Details on the impact of buffer cache size on SQL I/O pattern can be found in the white paper *OLTP I/O Profile Study with Microsoft SQL 2012 Using EqualLogic PS Series Storage* at http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/20308518/download.aspx.

7.4.3 Table Partition

The table partitioning study in section 6.2 proved that by partitioning the largest and most accessed table, SQL Server could most efficiently perform writes. This was seen in the reduction of the data and log write latencies. For the test, the E_Trade table was partitioned into four subsets. The subset data was spread across four different volumes residing in their own file groups. The rest of the database tables were hosted on the fifth volume. This partitioning scheme was based on a particular table column that contained the most common field in the user queries.

Table partitioning requires careful analysis on the existing data access patterns and current operational procedures. A successful execution of table partition requires:

- Planning the partition
  - Select a table or tables that can benefit from the increased manageability and availability of partitioning.
  - Select the column or column combination that will be the base of the partition.
  - Select the type of partitioning to be used.
- Specify the partition boundaries in a partition function.
- Create a partition storage plan for file groups using a partition scheme.

Table partitioning is a recommended best practice to improve query processing so that the SQL Server can efficiently fetch relevant data (see the following note). SQL Query efficiency and I/O throughput improvements depend on the partitioning scheme chosen, which in turn is heavily dependent on the data and workload characteristics. Typically, partitioning will offer benefits when implemented on frequently accessed tables.

However, partitioning may not be beneficial and yield performance degradation, if the partitioning scheme is not favoring the query behavior. So the query behavior has to be taken into account on designing the table partitions. In addition to that, the table index partition alignment also plays a major role in query performance improvement. If the indexes are not properly aligned with the table partitions, this may create a CPU overhead, which in turn results in increased query response time. So the table indexes also need to

be aligned with the partitions to yield the maximum benefits. It would be useful to monitor and understand the existing database setup and the query processing using a SQL Profiler, before planning on implementing the table partitions.

Note: Reference MSDN SQL Server Development Center, "Partitioned Table and Index Strategies Using SQL Server 2008": http://msdn.microsoft.com/en-us/library/dd578580.aspx

7.4.4 Files and file groups

A database file is a physical allocation of space and can be primary (.mdf), secondary (.ndf), or log (.ldf). Database objects can be grouped in file groups for allocation, performance, and administration purposes. User-defined filegroups and the primary filegroup are the two types of file groups. Either of them can be the default filegroup. The primary file is assigned to the primary filegroup. Secondary files can be assigned to user filegroups or the primary filegroup. Log files are never a part of a file group. Log space is managed separately from data space. Microsoft's recommendations are:

- If the primary filegroup is set as default, the size or the auto grow setting needs to be carefully planned to avoid running out of space.
- Microsoft recommends, with larger database deployments that are easily administrated and for performance reasons, to define a user-defined filegroup as the default. In addition, create all secondary database files in user-defined filegroups so that user objects do not compete with system objects for space in the primary filegroup.

7.4.5 Data file growth

If sufficient space is not initially assigned to a database, the database could grow continuously and performance would be affected. Performance is improved if the initial file size and the percent growth are set to a reasonable size to avoid the frequent activation of the auto grow feature. Microsoft's recommendations are:

- Leave the auto grow feature on at database creation time to avoid running out of space and also to let SQL Server automatically increase allocated resources when necessary without DBA intervention.
- Set the original size of the database to a reasonable size to avoid the premature activation of the auto grow feature.
- Set the auto grow increment to a reasonable size to avoid the frequent activation of the auto grow feature.

7.4.6 Transaction log file growth

The transaction log is a serial record of all modifications and their executions that occurred in the database. SQL Server uses the transaction log for each database to recover transactions. The log file size depends on the recovery models and the frequency of the log backups. If the transaction logs are frequently backed up, theoretically a larger transaction log might not be needed. However, remember that SQL Server can clear only the inactive portion of the log. If there are active transactions, they might not allow the log to clear immediately. This means that SQL Server will require a larger transaction log. The

most preferred recovery model is FULL to minimize downtime and data loss. Microsoft's recommendations for the transaction log are:

- Place the log and the data files into separate volumes to avoid both random and sequential I/O going to the same volume.
- Set the original size of the transaction log to a reasonable size to avoid constant activation of the auto grow feature, which creates new virtual files and stops logging activity as space is added.
- Set the auto grow percent to a reasonable but small enough size to avoid frequent activation of the auto grow feature and to prevent stopping the log activity for too long a duration
- Use manual shrinking rather than automatic shrinking.

7.4.7 Tempdb file growth

The tempdb database is a global resource that holds all the temporary tables and stored procedures for all users connected to the system. The tempdb database is recreated every time SQL Server starts so the system starts with a clean copy of the database. Determining the appropriate size for tempdb in a production environment depends on many factors like the existing workload and the SQL Server features that are used. Microsoft recommends the following:

- Set the recovery model of tempdb to SIMPLE. This model automatically reclaims log space to keep space requirements small.
- Pre-allocate space for all tempdb files by setting the file size to a value large enough to accommodate the typical workload in the environment. This prevents tempdb from expanding too frequently, which can affect performance. The tempdb database should be set to auto grow to increase disk space for unplanned exceptions.
- Set the file growth increment to a reasonable size to avoid the tempdb database files from growing by too small a value. Microsoft recommends the following general guidelines for setting the FILEGROWTH increment for tempdb files.

Tempdb file size	File growth increment
0 to 100 MB	10 MB
100 to 200 MB	20 MB
200 MB or more	10%

A Configuration details

This section contains an overview of the hardware configurations used throughout the testing described in this document.

Table 7 Test configuration hardware components

Test Configuration	Hardware Components
SQL Server® (ESXi01)	One PowerEdge R710 server running VMware ESXi v5, hosting 2 SQL Server database virtual machines BIOS Version: 6.1.0 2 x Quad Core Intel® Xeon® E5520 Processors 2.26 GHz 72 GB RAM, 4 x 146GB 15K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard), firmware version 6.4.5 Two Intel Quad Port VT network adapters (Intel 8257 1Gb) Firmware level 1.5-1
SQL Server® (ESXi02)	One PowerEdge R710 server running VMware ESXi v5, hosting 2 SQL Server database virtual machines BIOS Version: 6.1.0 2 x Quad Core Intel® Xeon® E5520 Processors 2.26 GHz 72 GB RAM, 4 x 146GB 15K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard), firmware version 6.4.5 Two Intel Quad Port VT network adapters (Intel 8257 1Gb) Firmware level 1.5-1
SQL Server® (ESXi03)	One PowerEdge R710 server running VMware ESXi v5, hosting 2 SQL Server database virtual machines BIOS Version: 6.1.0 2 x 6 Core Intel® Xeon®5690 Processors 3.45 GHz 98 GB RAM, 4 x 146GB 15K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard), firmware version 6.2.14 Two Intel Quad Port VT network adapters (Intel 8257 1Gb) Firmware level 1.5-1
SQL Server® (ESXi04)	One PowerEdge R820 server running VMware ESXi v5, hosting 2 SQL Server database virtual machines: BIOS Version: 1.2.6 4 x 8 Core Intel® Xeon® E5-4620 Processors 2.19 GHz 128 GB RAM, 4 x 146GB 15K SAS internal disk drives Broadcom 5720 1GbE quad-port NIC (LAN on motherboard), firmware version 7.2.14 Two Intel Quad Port VT network adapters (Intel 1350 1Gb) Firmware level 1.5-6

INFRA SERVER	One (1) Dell PowerEdge R710 Server running VMware ESXi v4.1, hosting a two (2) Windows Server 2008 R2 virtual machines for Active Directory and vCenter: BIOS Version: 6.1.0 Quad Core Intel® Xeon® X5570 Processor 2.26 GHz 48 GB RAM,2 x 146GB 15K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard) – firmware version 6.4.5
LOAD GEN SERVER	One (1) Dell PowerEdge R710 Server running VMware ESXi v4.1, hosting 1 Windows Server 2008 R2 virtual machine for Quest Bench Mark Factory: BIOS Version: 6.1.0 Quad Core Intel® Xeon® X5650 Processor 2.26 GHz 48 GB RAM, 2 x 146GB 15K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard) – firmware version 6.4.5
MONITOR SERVER	One (1) Dell PowerEdge R710 Server with Windows Server 2008 R2 for SANHQ: BIOS Version: 6.1.0 Intel Xeon X5650 Processor 2.26 GHz 48 GB RAM,2 x 146GB 15K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard) – firmware version 6.4.5
Network	2 x Force10 S60 1Gb Ethernet Switch ,Firmware: 8.3.3.7
Storage	2 x EqualLogic PS6100XV: 24 x 146GB 15K RPM SAS disk drives as RAID 10, with two hot spare disks Dual quad-port 1GbE controllers running firmware version 5.2.4

This section contains an overview of the software configurations used throughout the testing described in this document.

Table 8 Test configuration software components

Test Configuration	**Software Components**
Operating systems	Host: VMware vSphere ESXi Server v5 Guest: Microsoft Windows Server 2008 R2 Enterprise Edition (virtual machine): • MPIO enabled using EqualLogic DSM for Windows when using guest iSCSI initiator • EqualLogic Host Integration Toolkit(HIT) v4.0.0 installed
Applications	SQL Server 2012 Enterprise Edition
Monitoring Tools	EqualLogic SAN Headquarters version 2.2 Windows Perfmon
Simulation Tools	Benchmark Factory for databases version 6.8

B Table partition steps

The first step is identifying the best candidates for partitioning. SQL Profiler is a tool that helps gather the necessary details needed to select a table to partition and its partitioning column. This section provides examples of getting the SQL Profiler trace and analyzing the trace data. Below are some of the guidelines that were followed for the tests performed in identifying the table for partitioning.

- The first step in initiating a SQL Profiler trace is to define SQL Profiler trace properties, where the specific events (in order to minimize the load on the SQL server, a specific number of events are captured instead of selecting all) are selected to be logged.
- For the tests performed, the events tracked were, SP: StmtCompleted, SQL: BatchStarting, SQL: BatchCompleted and ShowPlan XML.
- The profiler trace was stored as a database table so it could be queried later on for the analysis once the profiler trace events were recorded and the profiler was started.
- User transactions were run from Benchmark Factory on the database, so the profiler could record the traces.
- After the completion of the user transactions, the profiler trace table was queried to get the top 10 SQL statements ordered by reads (because OLTP type workloads are mostly reads) in descending order. The common columns to query the trace are TextData, Duration, CPU, Reads, Writes, ApplicationName, StartTime, and EndTime.
- After repeating a couple of runs of the profiler trace, and getting the SQL query that had the most reads, the E_Trade table was found to be the largest and had the most access.

Microsoft also provides a T-SQL query that would list the top N frequently executed queries. http://msdn.microsoft.com/en-us/library/windowsazure/hh977102.aspx.

The query below was once again run to verify if the results match the SQL Profiler trace results.

```
SELECT TOP 20 SUBSTRING (qt.text, (qs.statement_start_offset/2)+1,((CASE
qs.statement_end_offset WHEN -1 THEN DATALENGTH(qt.text) ELSE
qs.statement_end_offset
END - qs.statement_start_offset)/2) +1), qs.execution_count,
qs.total_logical_reads, qs.last_logical_reads, qs.min_logical_reads,
qs.max_logical_reads, qs.total_elapsed_time,
qs.last_elapsed_time,qs.min_elapsed_time, qs.max_elapsed_time,
qs.last_execution_time, qp.query_plan,
FROM sys.dm_exec_query_stats qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) qt
CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) qp
WHERE qt.encrypted=0
ORDER BY qs.total_logical_reads DESC
```

The results of the query are shown below. The highlighted row is the one with maximum number of reads (which was the E_Trade table) and was selected as a good candidate to use for performing the partition studies.

	(No column name)	execution_count	total_logical_reads	last_logical_reads	min_logical_reads	max_logical_reads	total_elapsed_time	last_elapsed_time	min_elapsed_time	max_elapsed_time
1	Select Top 50 T_ID,T_DTS,ST_NAME,TT_NAME,T_S_SYMB,...	43549	548944775	107303	590	114913	1248836429	288016	1000	643036
2	FETCH API_CURSOR0000000000000003	906325	71491215	2	2	102	125979208	0	0	12000
3	Select WI_S_SYMB From E_WATCH_ITEM, E_WATCH_LIST ...	142082	28640092	253	105	306	68293911	0	0	105006
4	Select DM_CLOSE From E_DAILY_MARKET Where DM_S_SY...	8174851	24524553	3	3	3	107337138	0	0	9000
5	Select LT_PRICE From E_LAST_TRADE Where LT_S_SYMB = ...	8174851	16350184	2	2	4	147929464	0	0	172009
6	Select S_NUM_OUT From E_SECURITY Where S_SYMB = @P2;	8174851	16349702	2	2	2	83583774	0	0	3000
7	SELECT B_NAME, ISNULL(SUM(TR_QTY * TR_BID_PRICE),0)...	5258	4682500	931	839	956	182180419	39002	24001	70004
8	FETCH API_CURSOR000000000000298C	32330	2575492	2	2	122	5419313	0	0	1000
9	SELECT TOP 30 TH_DTS, T_QTY, T_S_SYMB, T_ID, ST_NA...	12883	1584972	111	96	144	31325790	0	0	87004
10	Select S_SYMB From E_INDUSTRY,E_COMPANY,E_SECURIT...	7819	1541468	212	127	306	3355194	1000	0	5000
11	Select T_BID_PRICE,T_EXEC_NAME,T_IS_CASH,TT_IS_MRK...	104483	626898	6	6	6	4082233	0	0	9000
12	Insert Into E_TRADE(T_ID,T_DTS,T_ST_ID,T_TT_ID,T_IS_CA...	21574	578304	24	24	67	165875484	8000	0	473027
13	SELECT CA_ID, CA_BAL, ISNULL(SUM(HS_QTY * LT_PRICE),...	8472	437157	26	2	115	1707099	0	0	2000
14	Select Top 3 TH_DTS,TH_ST_ID From E_TRADE_HISTO...	104483	419268	4	4	5	434653885	6000	0	166009
15	Select SE_AMT,SE_CASH_DUE_DATE,SE_CASH_TYPE ...	104483	417932	4	4	4	434561852	3000	0	226012
16	Select CT_AMT, CT_DTS,CT_NAME From E_CASH_TRAN...	95860	383440	4	4	4	403699074	3000	0	224012
17	Select C_L_NAME,C_F_NAME,B_NAME From E_CUSTO...	43549	287991	7	3	7	1654092	0	0	1000
18	SELECT [S_EX_ID],[S_NAME] FROM [E_SECURITY] WHERE [...	132391	264782	2	2	2	2297133	0	0	9000
19	SELECT [LT_PRICE] FROM [E_LAST_TRADE] WHERE [LT_S...	129252	258512	2	2	4	2569146	0	0	8000
20	SELECT [CH_CHRG] FROM [E_CHARGE] WHERE [CH_C_TIE...	129252	258504	2	2	2	1661094	0	0	1000

Figure 22 Results showing the most frequently executed queries

Additional resources

Support.dell.com is focused on meeting your needs with proven services and support.

DellTechCenter.com is an IT Community where you can connect with Dell Customers and Dell employees for the purpose of sharing knowledge, best practices, and information about Dell products and your installations.

Referenced or recommended Dell publications:

- Dell EqualLogic Configuration Guide:
  http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/19852516/download.aspx
- PS Series Groups Deploying Microsoft® SQL Server in an ISCSI SAN
  http://www.equallogic.com/uploadedfiles/Resources/Tech_Reports/tr1013-sql-server.pdf

Referenced or recommended Microsoft publications:

- Partitioned Table and Index Strategies Using SQL Server 2008":
  http://msdn.microsoft.com/en-us/library/dd578580.aspx
- SQL Server 2012 What's New, July 2011
  http://www.microsoft.com/sqlserver/en/us/learning-center/resources.aspx
- Microsoft SQL Server 7.0 Storage Engine Capacity Planning Tips
  http://msdn.microsoft.com/en-us/library/aa226175%28v=sql.70%29.aspx
- Optimizing tempdb Performance
  http://msdn.microsoft.com/en-us/library/ms175527%28v=sql.105%29.aspx
- Analyzing I/O Characteristics and Sizing Storage Systems for SQL Server database applications, April 2010
  http://msdn.microsoft.com/en-us/library/ee410782(v=sql.100).aspx
- Top SQL Server 2005 Performance Issues for OLTP Applications
  http://technet.microsoft.com/en-us/library/cc966401.aspx
- SQL Server 2000 I/O Basics, January 2005
  http://technet.microsoft.com/library/Cc966500

For EqualLogic best practices white papers, reference architectures, and sizing guidelines for enterprise applications and SANs, refer to Storage Infrastructure and Solutions Team Publications at:

- http://dell.to/sM4hJT