Dell EqualLogic Best Practices Series

Microsoft SQL Server 2008 Backup and Restore using Dell EqualLogic

A Dell Technical Whitepaper

Storage Infrastructure and Solutions Engineering
Dell Product Group
March, 2012

Table of Contents

Acknowledgements

This whitepaper was produced by the PG Storage Infrastructure and Solutions of Dell Inc.

The team that created this whitepaper:

Lakshmi Devi Subramanian, Puneet Dhawan, and **Camille Daily**

We would like to thank the following Dell team members for providing significant support during development and review:

Ananda Sankaran , **Darren Miller,** and **Bob Ganley**

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

Creating an optimized Microsoft® SQL Server® database backup is the foundation for every DBA's data protection plan. This paper studies the impact on SQL Server host during backup and provides best practices and reference architecture for backup and restores operations. The tests were performed on virtualized Microsoft SQL Server 2008 R2 Databases running on Dell™ EqualLogic™ SAN. Methodologies to improve backup throughput and duration while still maintaining data integrity are also covered.

For tests performed in this paper, the database data was spread across multiple source volumes and backup performance was compared while running backups on multiple target files hosted on a single or multiple target volumes.

From the tests and data analysis, we conclude in this paper that:

- Compared to the backup performed on multiple target files in a single target volume, backing up to multiple target files within multiple volumes:
    - Increased the backup throughput by 35%,
    - Reduced the backup duration by 26%, and
    - Reduced the restore duration by 24%.
- Backups using SQL Server Management Studio had a significant impact on production SQL server resources. Running full backups:
    - Increased the average CPU utilization percentage from 1.7% to 13% at production SQL Server, and
    - Increased the application query response time by 77%.
- Enabling backup compression at production SQL Server:
    - Increased the average CPU utilization percentage from 1.7% to 61% during full backup, and
    - Reduced the backup file size by 38% during full backup.

1 Introduction

SQL Server databases form the backbone for many businesses, housing data for critical applications. Because access to this data is critical for day to day business operations, Service Level Agreements (SLAs) demand that databases remain online and available—there is little tolerance for downtime.

Database protection and disaster recovery are among the top concerns for database administrators. Many types of failure scenarios can impact SQL server database application availability. Requirements for reducing database backup windows, restore times, data integrity, and impact on application performance are some of the key considerations while devising a robust data protection strategy, that would ensure critical information remains available and well protected.

Hardware redundancy is one way to protect databases from hardware problems. Some of the software based data protection schemes include data backup, snapshots, and replication based solutions. The EqualLogic PS Series arrays offer the availability features necessary for a holistic plan that is designed to protect SQL server data. These arrays feature redundant components, and provide the ability to create transactionally consistent snapshots, clones, and replicas of database and log volumes. The EqualLogic Auto-Snapshot Manager/Microsoft Edition application provides SQL server awareness, via the Microsoft VSS interface when creating and restoring from such protection sets. An in-depth discussion on SQL Server data protection using EqualLogic smart copy snapshots can be found at http://en.community.dell.com/techcenter/storage/w/wiki/enhancing-sql-server-protection-using-equallogic-smart-copy-snapshots.aspx.

The goal of this paper is to describe best practices and reference architecture for performing traditional data backup operations of virtualized Microsoft SQL Server 2008 R2 Databases running on Dell EqualLogic SAN using SQL Server Management Studio. This paper begins with basic considerations for data backup and the various options for data backup and recovery available in Microsoft SQL Server. The test cases performed in this paper were specifically designed to find:

- Best practices for backup target file layout - Impact of different backup target file layouts on backup performance.
- Best Practices for backup target volume layout –Impact of different backup target volume layouts on backup performance.
- The impact of performing full, differential, and transaction log backups on SQL server.
- The impact of performing full backup with compression, on SQL server.

1.1 Audience

This white paper is primarily targeted to database administrators, storage administrators, ESXi/VMware administrators, and database managers who are interested in using Dell EqualLogic storage to optimize their backup and restore strategies in Microsoft SQL Server environments. It is assumed that the reader has an operational knowledge of SQL Server backup and restore strategies, configuration and management of EqualLogic SANs and iSCSI SAN network design, and familiarity with VMware® ESXi Server environments.

2 Considerations for selecting a data backup solution

Business environments differ widely, from small businesses running just a few servers to large data centers managing complex server and storage architectures. Each organization must choose a data protection solution that fits its unique business environment. Four questions to consider when choosing a backup solution are:

What are the Recovery Point Objective (RPO) and Recovery Time Objective (RTO)?

RPO and RTO provide a baseline for how much data loss and how much downtime can be tolerated when recovering from an outage.

What are the backup retention policies?

If the backed up data exceeds the retention policies, it is possible to delete that data and reuse the media. Alternatively, data may be archived to different media and kept for long term storage.

What are the overall IT organizational goals?

Performance, capacity, portability, and cost factors will determine the different storage media (such as disk, tape, and optical drives) included in a data backup and recovery solution.

Will the solution use data protection software?

If a solution includes data protection software, it is important to understand the value it adds. For example, does the software schedule regular backup and restore jobs? Does it require little to no monitoring? And, most important, how easy is the software to manage?

3 Test System Configuration

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

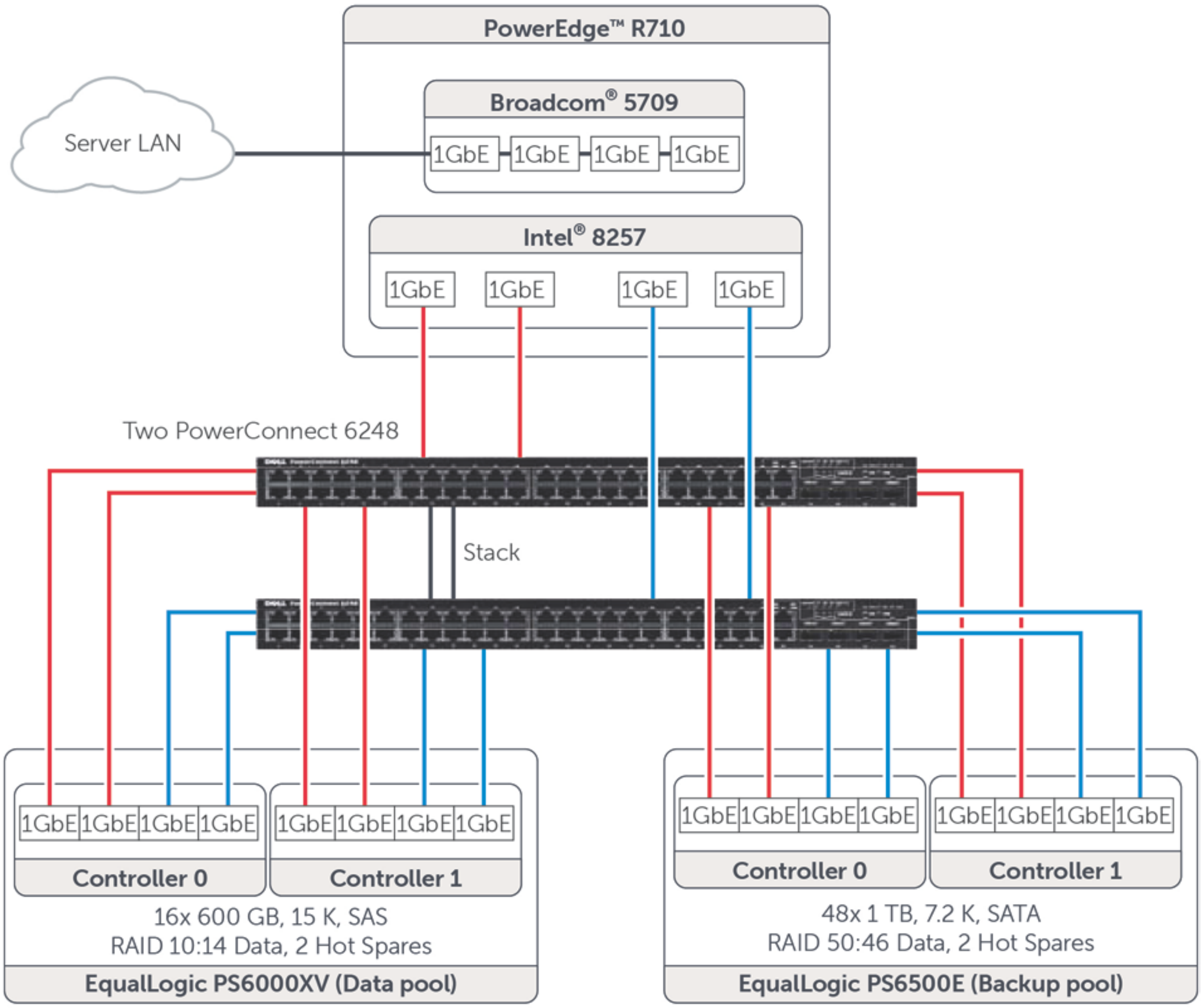

Figure 1 Native SQL Server LAN and iSCSI SAN connectivity

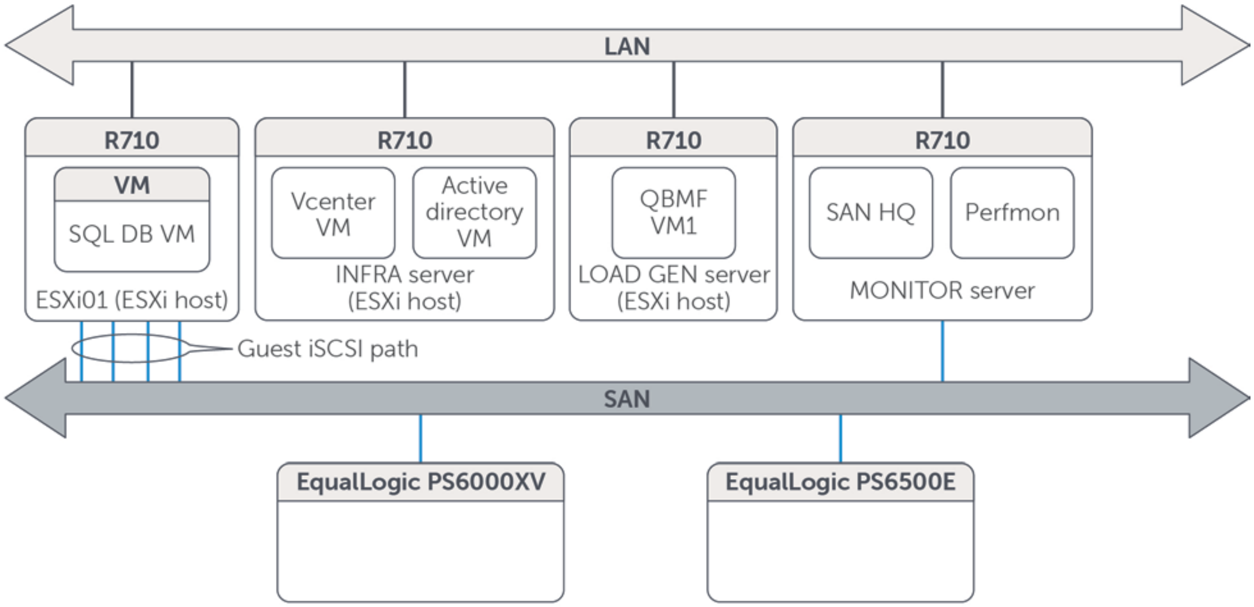

Figure 2 High level view of test system components

Key design details in the test system configuration include:

- Installed and configured SQL Server 2008 R2 Enterprise Edition in a Windows Server 2008 R2 Enterprise Edition virtual machine (SQL DB VM) hosted on the VMware vSphere ESXi4.1 server on a Dell PowerEdge R710 server.
- The virtual machine that hosted the SQL Server was configured to use eight virtual CPUs and 16GB of reserved memory.

  Network configuration details for ESXi01 (Host):

  - The on board four port LOM (LAN on motherboard) Broadcom 5709 network controller was used for the Server LAN connection paths via virtual switch, vSwitch0 created for LAN connectivity(refer to Figure 3).
  - 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 VM) and the volumes in the data pool on the PS6000XV array and backup pool on the PS6500E 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.

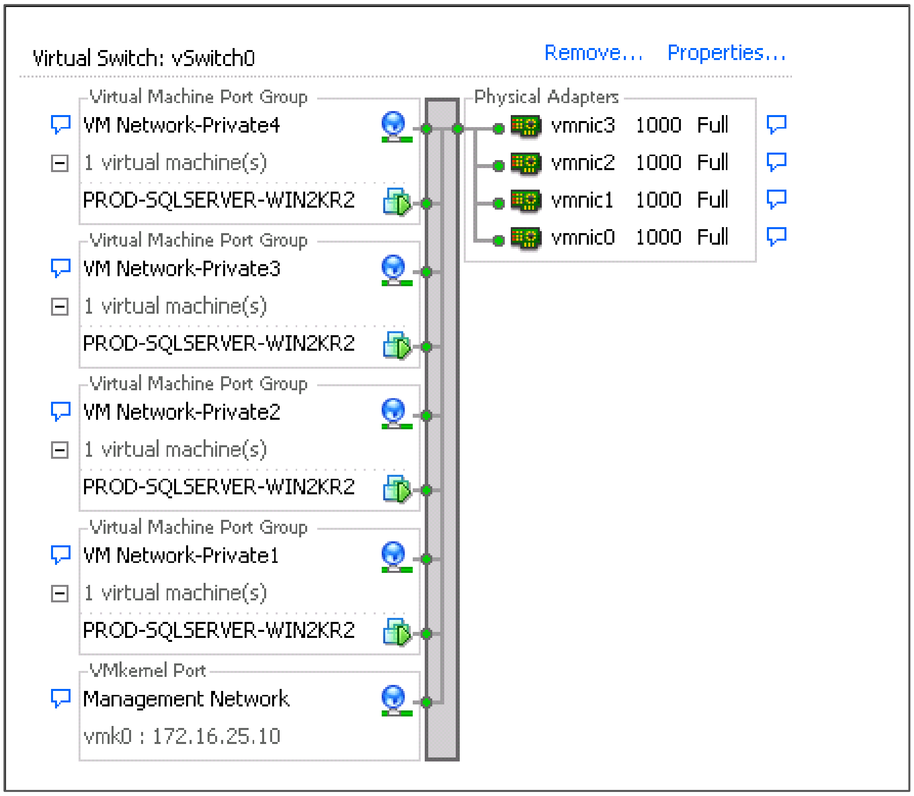

Figure 3 vSwitch0 configuration

- o A separate vSwitch (vSwitch1) for iSCSI SAN connectivity.
- o Created virtual network adapters (type VMXNET 3) within the VM and assigned them to the vSwitch1 (refer to Figure 4) on the vSphere host. Used EQL MPIO DSM via Host Integration Tools (HIT) Kit to setup multiple paths from the guest VM to the storage volumes. These paths are labeled as “Guest iSCSI Path” in Figure 2.

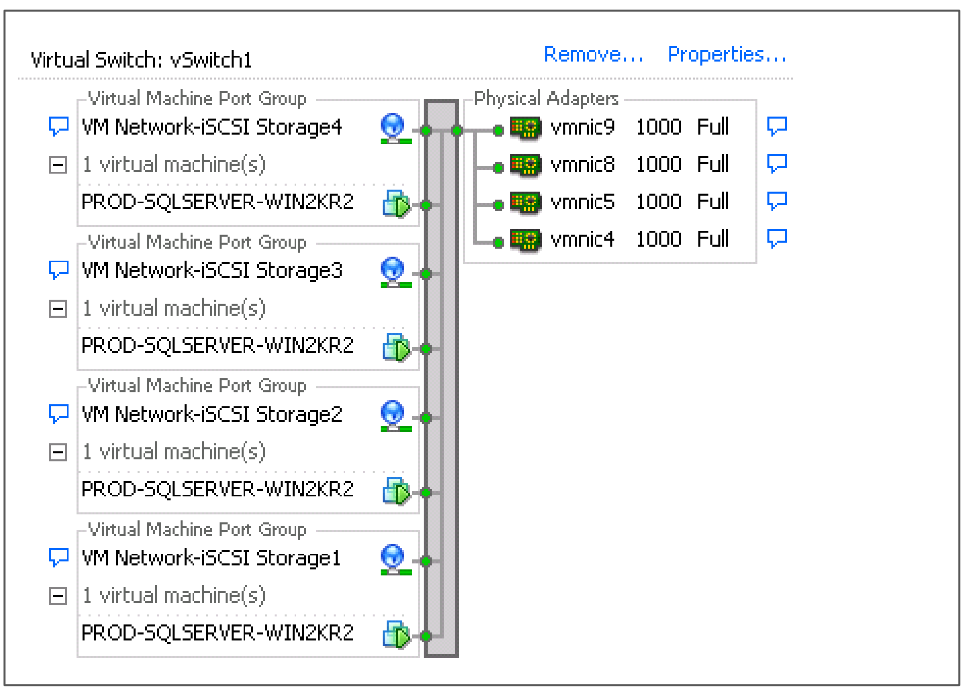

Figure 4 vSwitch1 configuration

- The two local disks installed in the R710 server were configured as a RAID 1 set. ESXi 4.1 was installed on these disks, and the guest virtual machine OS disk partitions were also hosted within the virtual machine file system on these disks.
- A second VMware ESXi 4.1 server (INFRA) was used to host virtual machines for vCenter™ and Active Directory.
- A third VMware ESXi 4.1 server (LOAD GEN) was used to host a Windows 2008 R2 workload simulation virtual machine, running an instance of Benchmark Factory® by Quest Software® (QBMF).
- The MONITOR server was a Dell PowerEdge R710 server running Windows 2008 R2 natively. It was used to host the following monitoring tools: EqualLogic SAN Headquarters (SANHQ), and Perfmon.
- The SAN switches consisted of two Dell PowerConnect 6248 switches, configured as a single stack. Redundant connection paths from each array controller to each switch in the stack were created.
- One EqualLogic PS6000XV consisting of 16 x 600GB 15K RPM SAS disk drives in a RAID 10 configuration was used to host SQL Server volumes and snapshots (Data pool).
- One EqualLogic PS6500E consisting of 48 x 1TB SATA drives in a RAID 50 configuration was used to host the backup data volume(Backup pool) and other SQL Server components.
- Detailed configuration specifications for each test system component are provided in Appendix A.

4 Test studies

To support development of this paper, a full-scale virtualized Microsoft SQL server configuration running within a VMware vSphere v4.1 virtual machine environment was deployed and tested.

4.1 Backup target volume and file layout baseline study

The goal of these tests was to measure the variation in I/O throughput of database backup and restore operations when using:

- Multiple target backup files within single target volume during full backup and restore
- Multiple target backup files within multiple target volumes during full backup and restore

Initially a series of tests was conducted to establish a configuration that best utilizes the system resources and produce a baseline configuration for database backup and restore operations. QBMF was used to create and load a SQL Server Database. QBMF is a load generator that can simulate Online Transaction Processing (OLTP) workloads on a database at various load settings. A single EqualLogic PS6000XV (Data pool) array configured with RAID 10, hosted both the database and log volumes. Another single EqualLogic PS6500E (Backup pool) array configured with RAID 50 was used as a backup target for the database backup files. The volume layout in each pool is detailed in Table 1. The baseline tests performed and the volume configurations are shown in Table 2.

Table 1 – Storage volumes layout

Volume	Size	Purpose
Data Pool (PS6000XV RAID 10)		
SQL-SERVER-DB-DATA-VOL1	500GB	Database files
SQL-SERVER-DB-DATA-VOL2	500GB	Database files
SQL-SERVER-DB-DATA-VOL3	500GB	Database files
SQL-SERVER-DB-DATA-VOL4	500GB	Database files
SQL-SERVER-DB-LOG	600GB	Transaction Log files
SQL-SERVER-DB-TEMPDB	100GB	TempDB Files
Backup Pool (PS6500E RAID 50)		
DB-BACKUP1	1TB	SQL Server DB Backup Files
DB-BACKUP2	1TB	SQL Server DB Backup Files
DB-BACKUP3	1TB	SQL Server DB Backup Files

Table 2 Baseline test parameters

SQL Server database details	
Database Size (Tables+ Indexes +Log)	~1 TB
Baseline Test#1 Refer to Figure 5	1 Target Volume with 1 backup files. 1 Target Volume with 3 backup files. 1 Target Volume with 5 backup files.
Baseline Test#2 Refer to Figure 7	1 target volume with 2 backup files. 2 target volumes with 2 backup files. 3 target volumes with 2 backup files.

4.1.1 Baseline Test #1: Impact of multiple backup target files on backup throughput

Baseline test #1, studies the behavior of increasing the backup process speed by creating multiple writer threads via configuring multiple target files within a single target volume. SQL server uses one backup writer thread per target backup file configured.

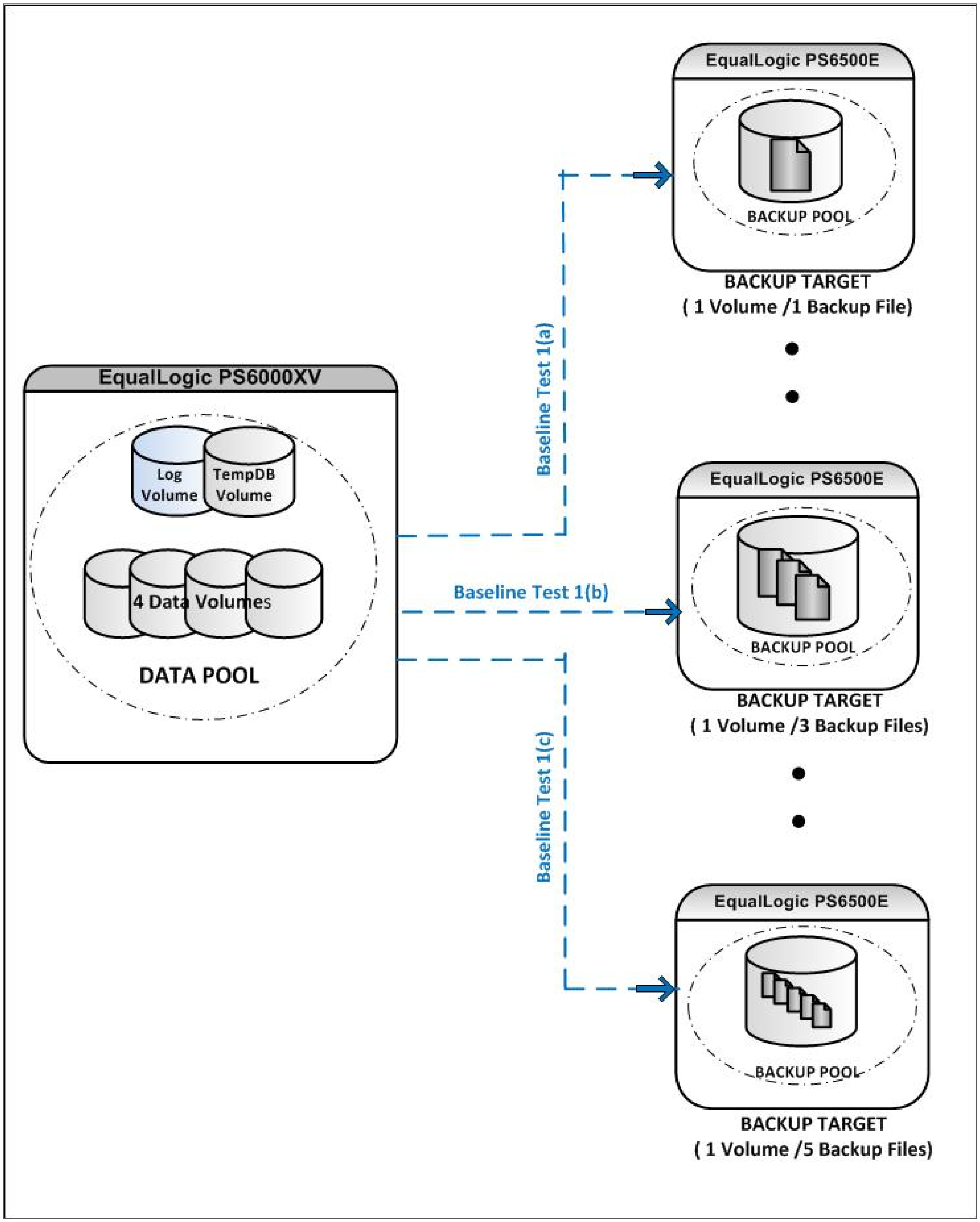

Figure 5 Logical representation of baseline test #1

The read/write I/O throughput is the main performance measure which impacts database backup duration. The database backup is mostly a sequential operation with a block size of approximately 256KB or larger. The configuration with five backup files in a single volume (Baseline Test 1(c)) as shown in baseline test#1, (refer to Figure 5), was able to achieve maximum throughput compared to configurations with fewer backup files in a single volume. This throughput was achieved because the number of writer threads increased with the number of target files resulting in the ability to concurrently write to more files at the same time.

Figure 6 below illustrates the read/write throughput achieved in each of the configuration shown in Figure 5. The read throughput initiates from the source volumes and write throughput is directed to the target volume and files. Note the highlighted data with maximum write throughput of 227MB/sec for the configuration with five backup files in a volume. An increase in throughout was not observed when more than five target files were used because the single target array was being saturated to its maximum achievable throughput for this test case.

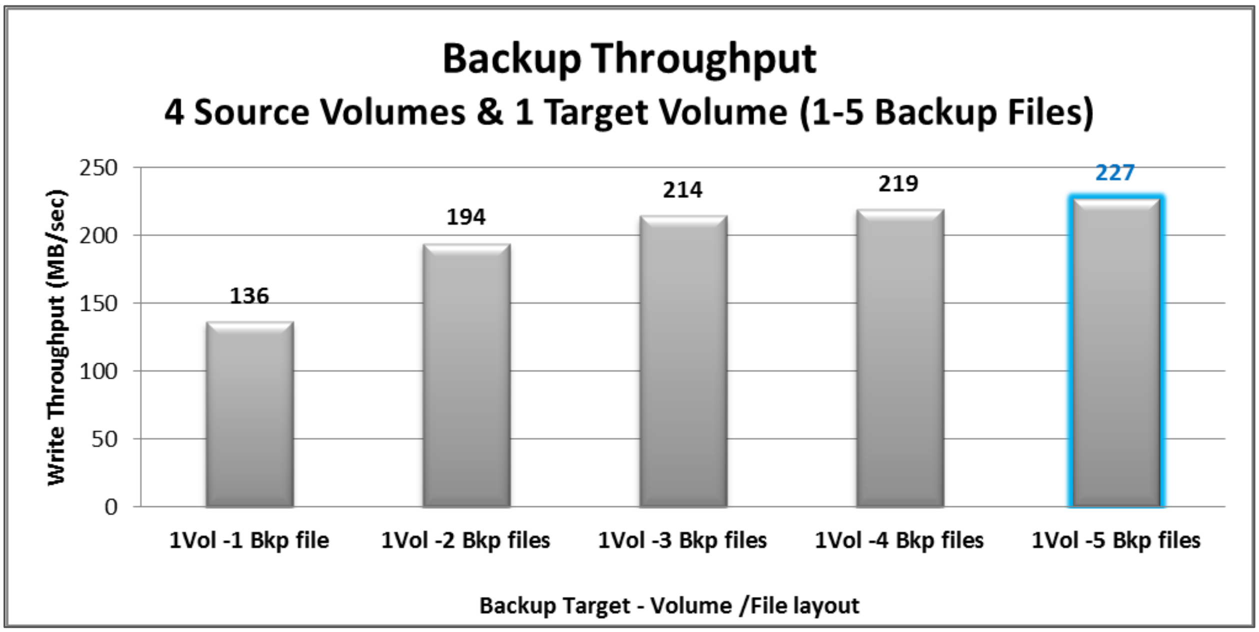

Figure 6 Throughput of single target volume with one or more backup files

Throughput increased because multiple writer threads were used by the native SQL backup process when configured with multiple target files for backup (http://msdn.microsoft.com/en-us/library/ms190954.aspx). The multiple writer threads improve backup throughput by working concurrently and utilizing the buffer cache resources at the SQL server efficiently for backup write operations. Backup data is concurrently written across the multiple files by these writers thus improving overall backup throughput.

4.1.2 Baseline Test #2: Impact of multiple target volumes/files on backup throughput

Baseline test #2, studies the impact of increasing storage volumes for hosting the backup target files. As described in figure 7, three scenarios with one to three backup target volumes were simulated with two backup files per volume.

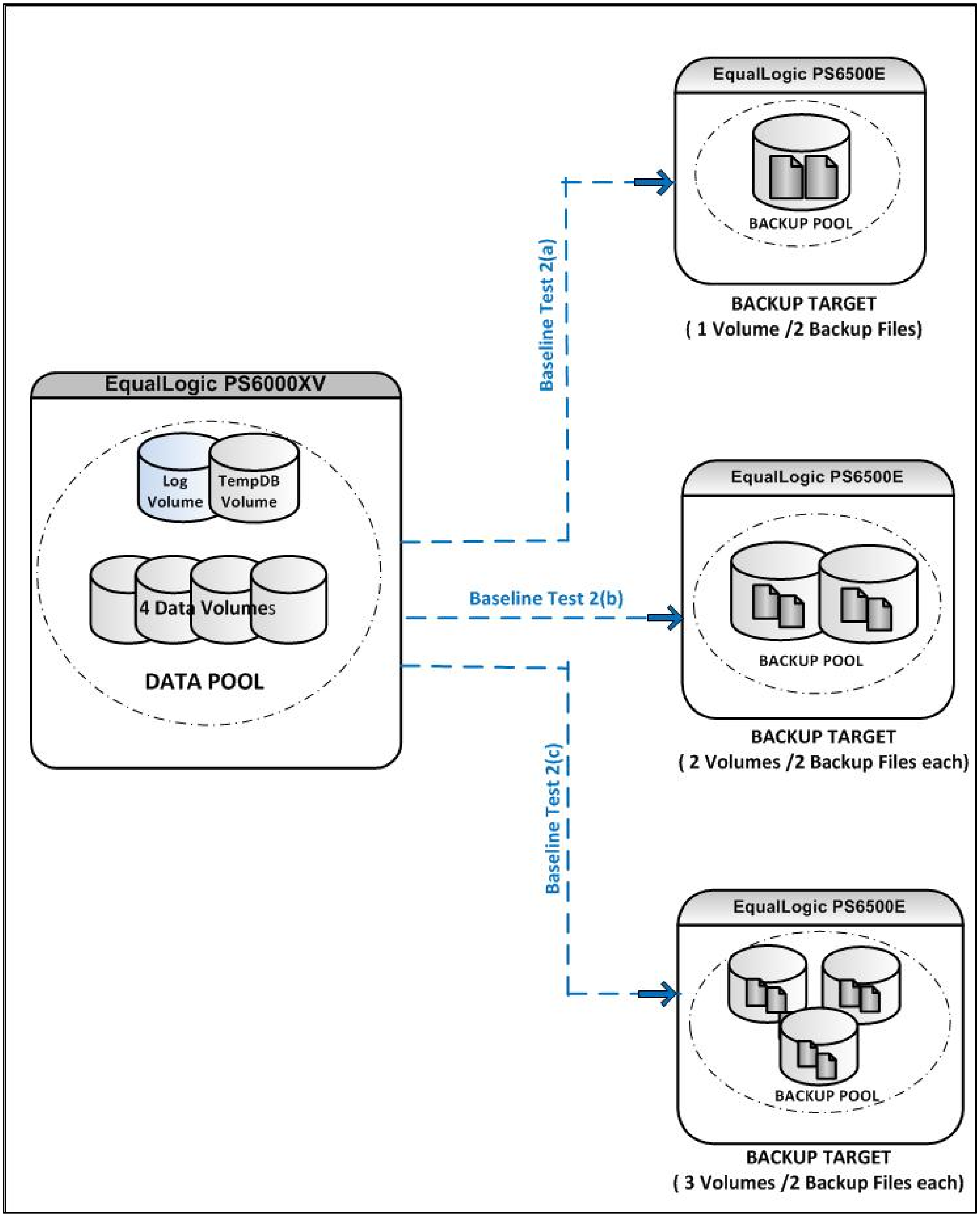

Figure 7 Logical representation of baseline test #2

Figure 8 illustrates the write I/O throughput achieved in each of the configurations shown in baseline test#2 (refer to Figure 7). Note the highlighted data with maximum throughput of 307MB/sec for the configuration with two backup files each in three backup volumes.

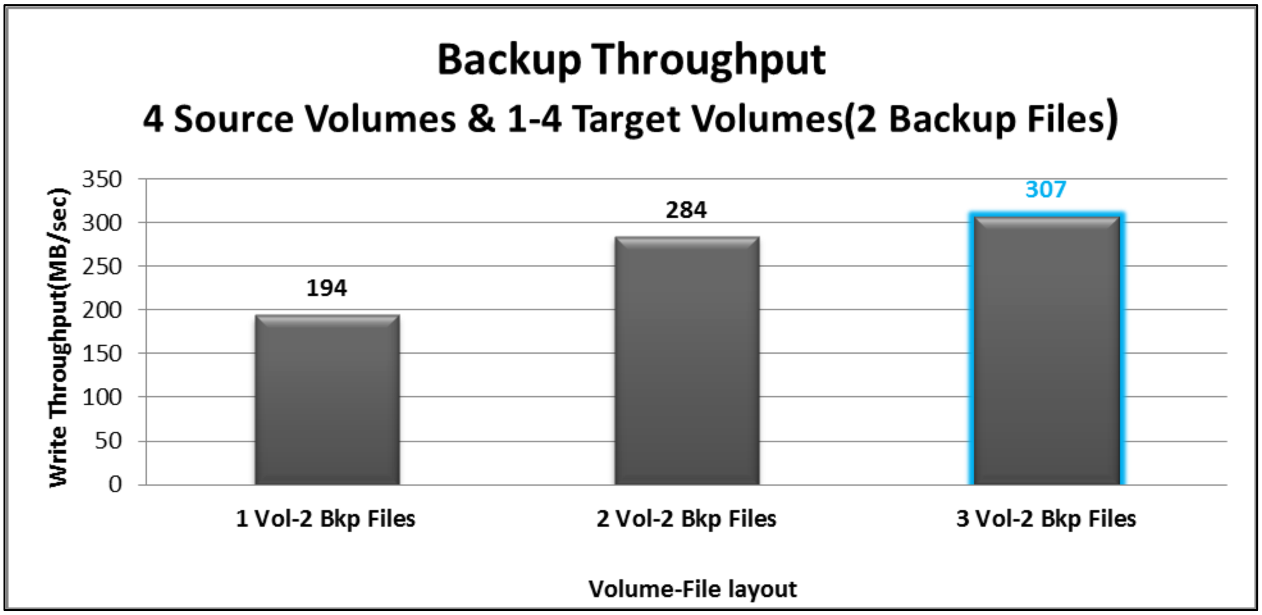

Figure 8 Throughput of multiple target volumes with 2 backup files each

Using three volumes hosting a total of six target files proved to be an efficient use of storage and provided the maximum throughput. An increase in throughout was not observed when four volumes and a total of eight target files were used because the single array was running at maximum achievable data backup rate for this test case and more volumes/files caused further parallelization of I/O and contention of resources. Performance would continue to increase if more storage arrays were added to the pool providing more storage processing resources such as number of storage processing controllers including on-board cache, and the number of storage network interfaces.

4.1.3 Analysis and Conclusion

Restores were done from the full backups that were performed in the best configurations chosen from baseline test#1 (i.e. Test 1(c)-one target volume with five backup files) and baseline test#2 (i.e. Test 2(c) - three target volumes with two backup files each). Figure 9, illustrates the recorded backup and restore times in both configurations. Configuration with multiple volumes and multiple backup files perform better (Refer figure 10), because it enables SQL Server to use parallel I/O to increase the speed of backup and restore operations because each backup file in each volume can be written(during backup) to and read from(during restore)at the same time. The backup duration depends on I/O throughput and decreases with increased throughput. From the tests and data analysis, when compared to the backup performed to multiple target files in a single target volume; backing up to multiple target files within multiple volumes,

- Reduced the backup duration by 26%
- Reduced the restore duration by 24%
- Increased the backup throughput by 35%

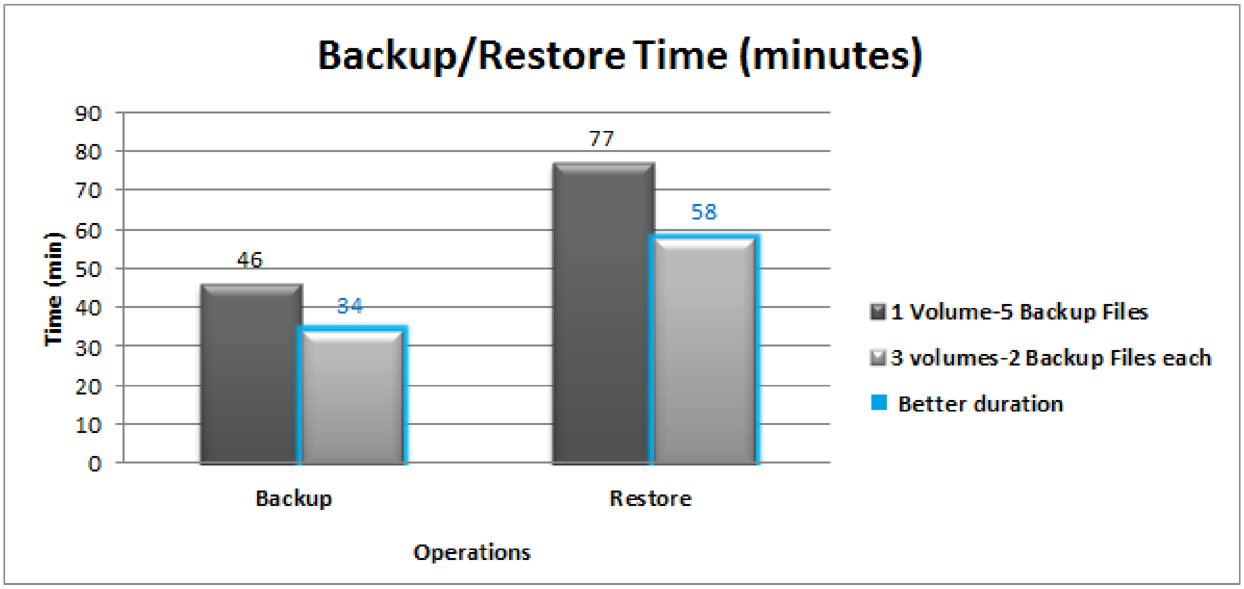

Figure 9 Backup/restore times in both the configurations

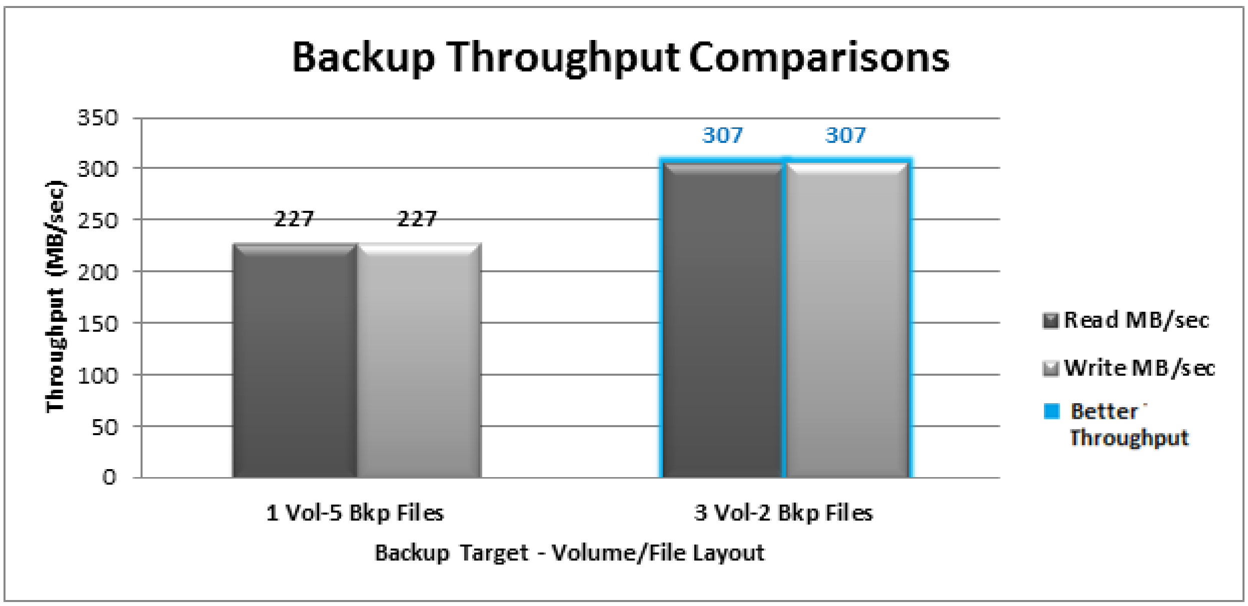

Figure 10 Backup read/write Throughput comparisons

Writing to multiple files across different disk volumes provides better data backup rates and reduced backup windows as shown in the tests above. Specifically, for our storage configuration, three target volumes with two backup target files each yielded the best throughput. All our following tests would use this configuration (i.e., four source data volumes and three backup volumes with two backup files each).

Note: These results are specific to our test environment and would vary depending upon the number of storage arrays, type of the array and RAID configurations used. However the analysis presented provides insight on the impact of number of target volumes and backup files can have on backup throughput.

4.2 Impact of backup and restore processes on SQL host using native SQL server tool

The goal of this test scenario was to study the performance impact on the production server when running the SQL server native backup utility. This is done by simulating database backup and restore scenarios across the SAN fabric to and from the volumes hosted on a second storage array. In this scenario, the backup is initiated at the SQL server, where the server reads the source data files for backup and writes them to the target volumes. Since the data moves through the SQL Server during the backup, there is an impact on SQL server's resources during the backup process. Figure 11 illustrates the data flow across the SAN during the database backup operations.

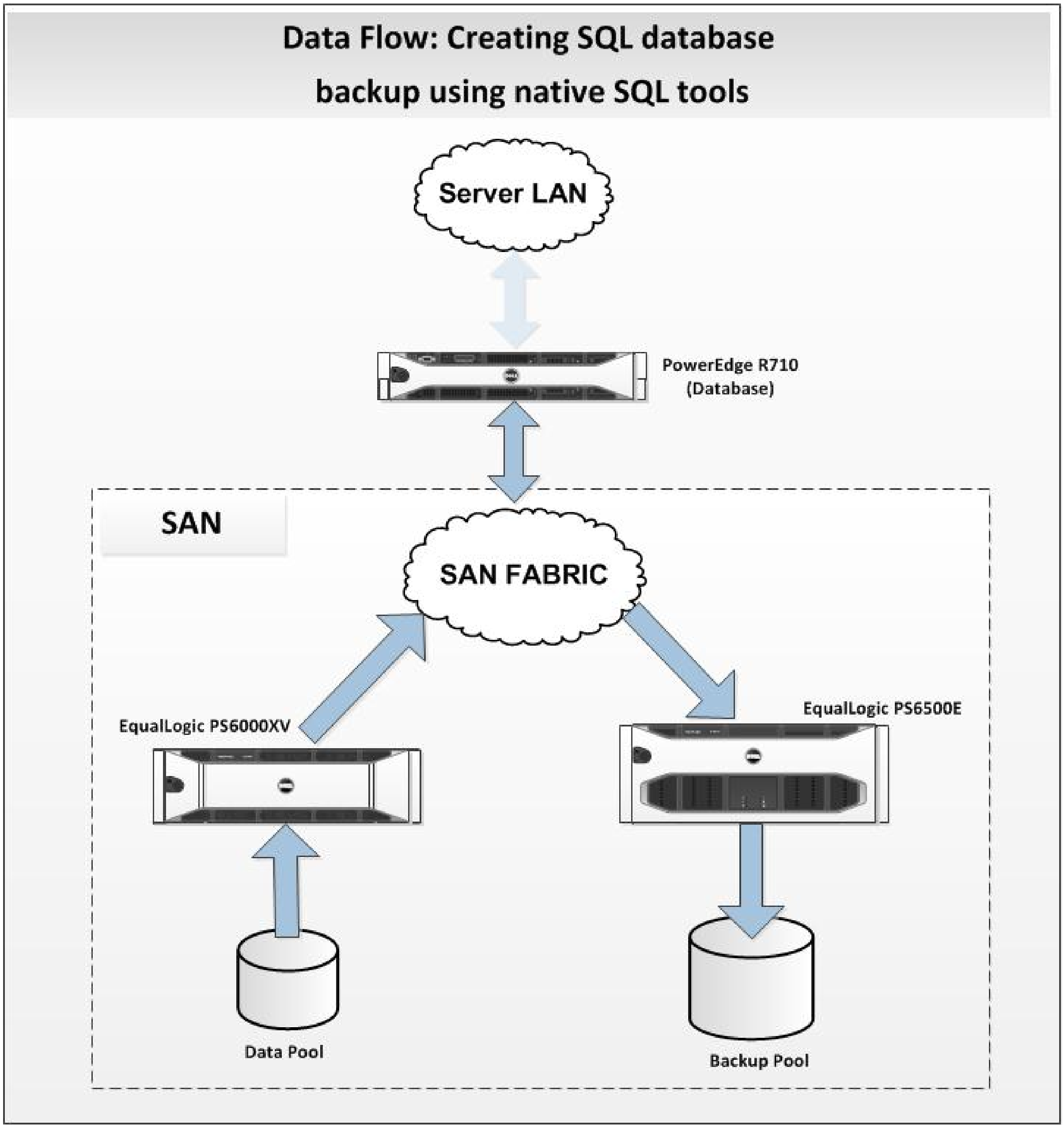

Figure 11 Database backup dataflow

Test details:

- Total database size at the full backup point: ~1TB
- Database workload: During the test timeline Quest Benchmark Factory was used to simulate an OLTP style workload. A constant 500 user workload was simulated, generating an average I/O load of 25 transactions per second (tps) against the database.
- We simulated three types of backup scenarios, gathered the results, and analyzed each case. We used the SQL server Management Studio to perform the following:
    - Performance impact studies during full and transaction log backups
    - Performance impact studies during full, transaction log, and differential backups
    - Performance impact studies during full backup with compression

4.2.1 Test #1: Performance impact studies during full and transaction log backups

This test simulates an environment where the typical production duration is an eight hour workday and full backups are taken daily and transaction log backups scheduled every 30 minutes during the eight hour workday.

- To simulate this test, the user load from Quest Benchmark Factory was run for ten hours;
    - A full backup was scheduled after one hour of user load run
    - After hour and half after full backup and verify, transaction logs were scheduled for every 30 minutes for seven and half hours with a total of 16 log backups.
- The backups were verified for integrity and checksums by enabling *Verify backup when finished* and *Perform Checksum before writing to the media* in the native SQL Server backup options screen.
- Restore to a new database was performed from the full and transaction log backups. The restored database was verified using *DBCC CHECKDB* T-SQL statement procedure part of SQL Server.

For this test we performed full and log backups as shown in the Figure 12. The performance counters measured for this test were CPU utilization, application response times, IO operations at the storage array, backup/restore times from the SQL server and disk read latency at the storage array. The sections below highlight the performance impact at each step (i.e. full backup, transaction log backup, and restore).

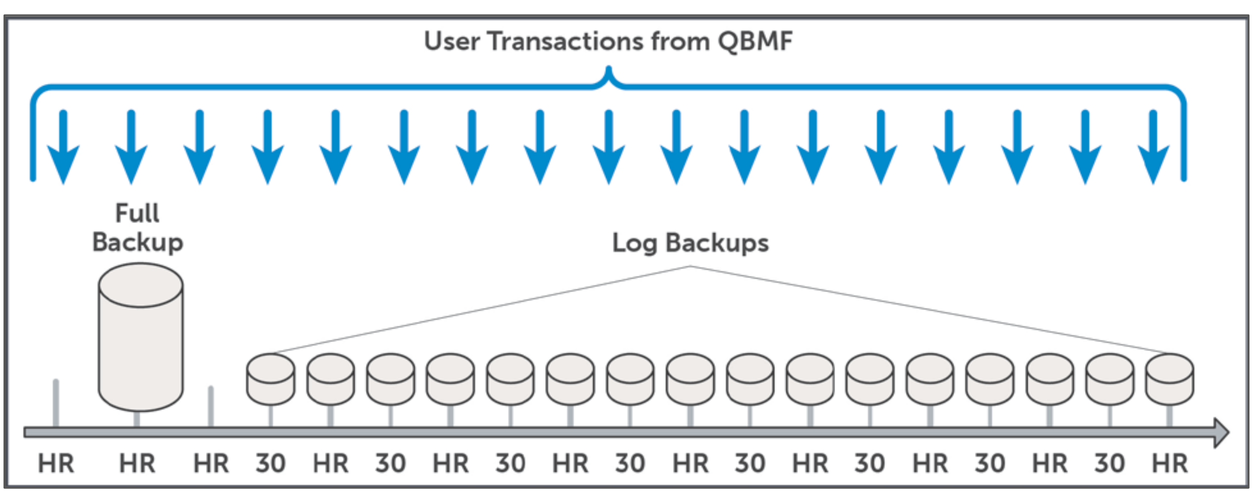

Figure 12 Backup operations in test #1

Impact of full backup & verify

Figure 13 shows an increase in CPU utilization at the SQL production server during full backup and verify. The average CPU utilization percentage increased from 1.7% to 13% during backup and 1.7% to 10% during verify. As expected, we saw a significant increase in CPU load on the system during the time that the backup utility was creating the backup.

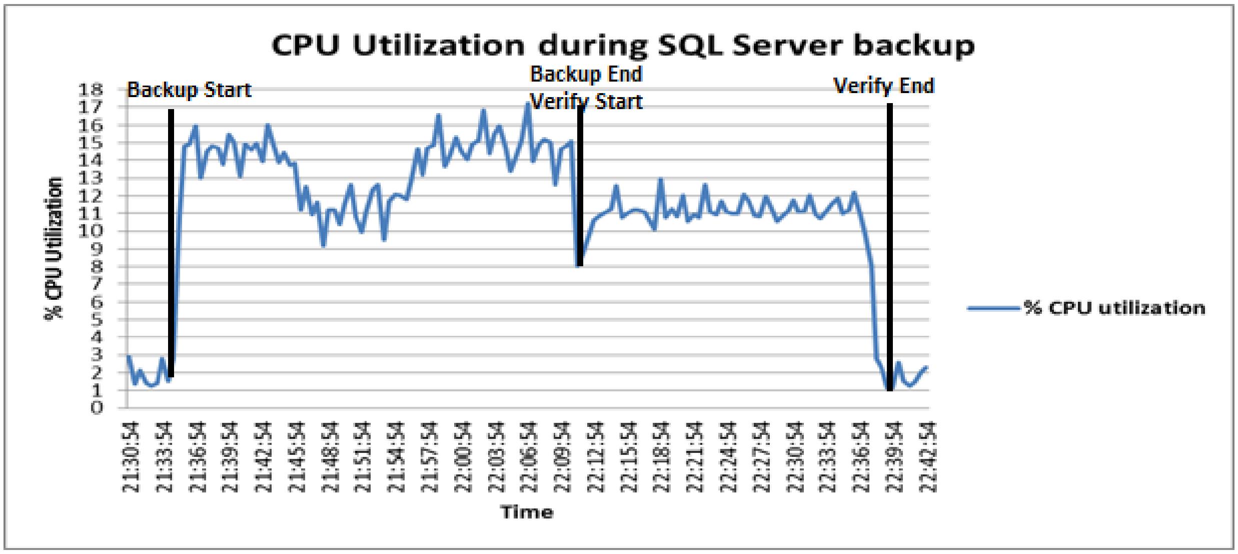

Figure 13 Impact on CPU utilization during full backup and verify

Figure 14 highlights the increase in application response time (measured at QBMF) during the backup and verify period. As shown, we noticed a 77% increase in the application response time. This is because the backup was taken while user transactions were running on the same database. This is a measure of the average response times incurred by SQL transactions submitted by the simulation tool (QBMF). Because of the smaller user load used in the test configuration, the response time changes reported here are in few milliseconds. However, the application response times may change drastically during full backup while running heavy user load along with heavy use of production SQL server's resources.

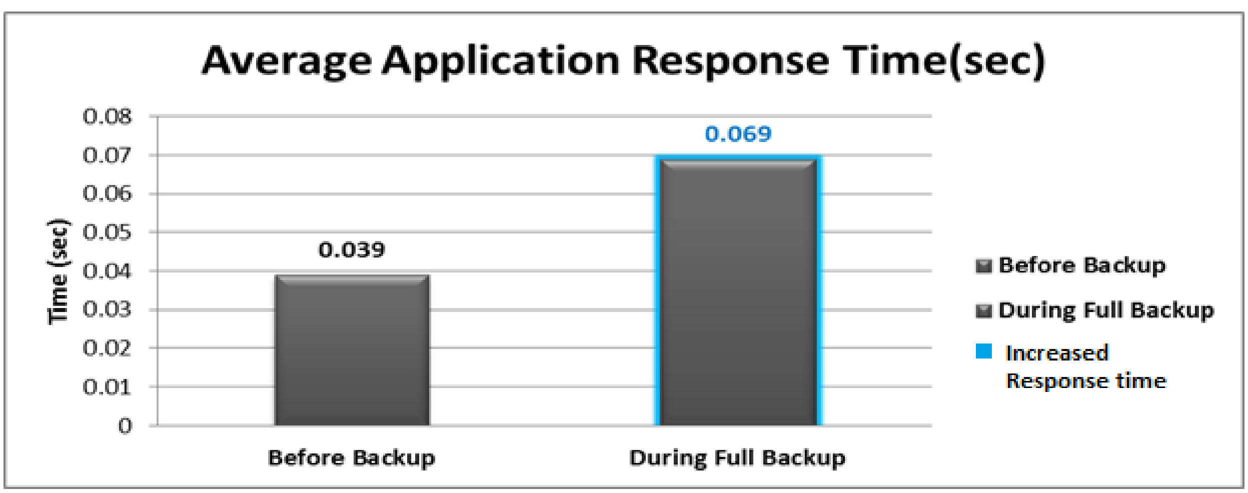

Figure 14 Impact on application response time during full backup and verify

Figure 15 shows the IO operations at the storage arrays, recorded using SANHQ. It shows the IOPS before full backup and during full backup. The user transactions have been running throughout before, during and after full backup. As expected the IOPS increased during full backup due to extra read write operations involved during backup process plus IO operations from the user transactions. The figure shows the combined IOPS at both data and backup pools.

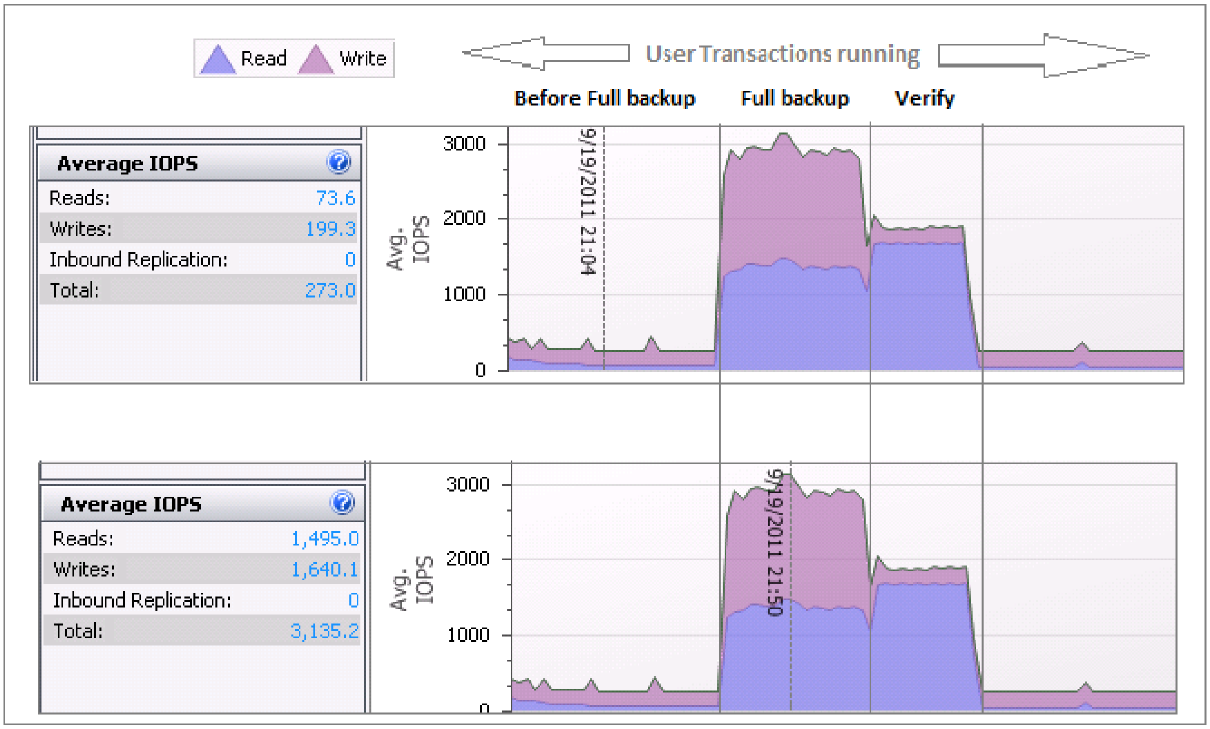

Figure 15 IOPS before and during full backup and verify

Figure 16 shows increase in disk read latency (measured using Perfmon on the SQL Server host) from the EqualLogic array that is part of the data pool during the backup and verify period. Read happens across four volumes (source database data spread across four volumes) shown as four drives in Figure 16.

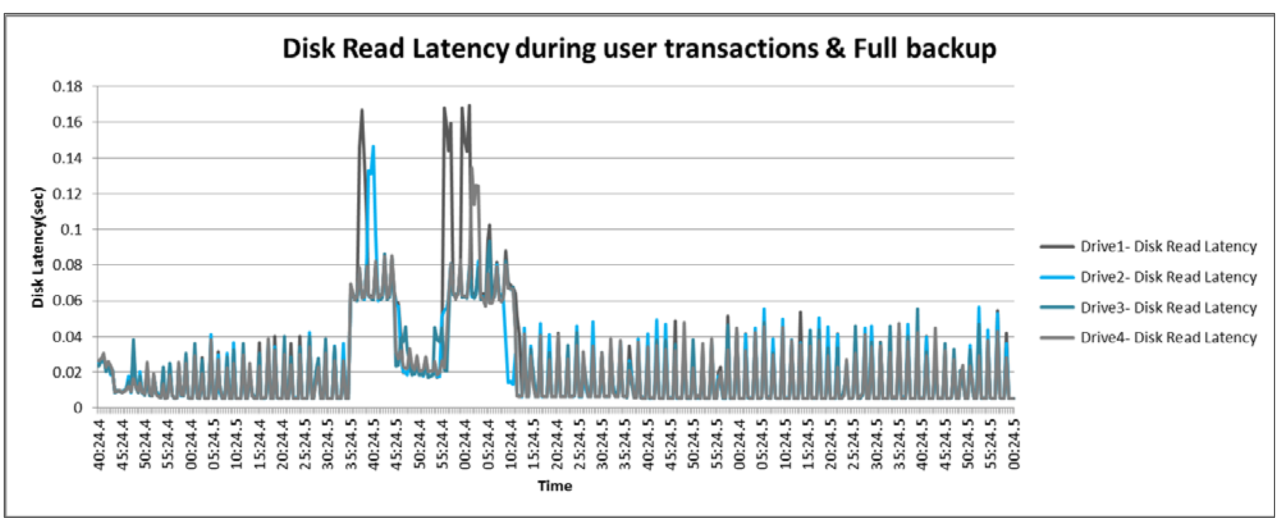

Figure 16 Impact on disk read latency during backup and verify

Impact of transaction logs backup

Figure 17 illustrates the increase in CPU utilization during every transaction log backup. This increase appears as a spike in utilization for each of the transaction log backups. The impact to application and disk response times were minimal compared to the impact during full backup and are not shown.

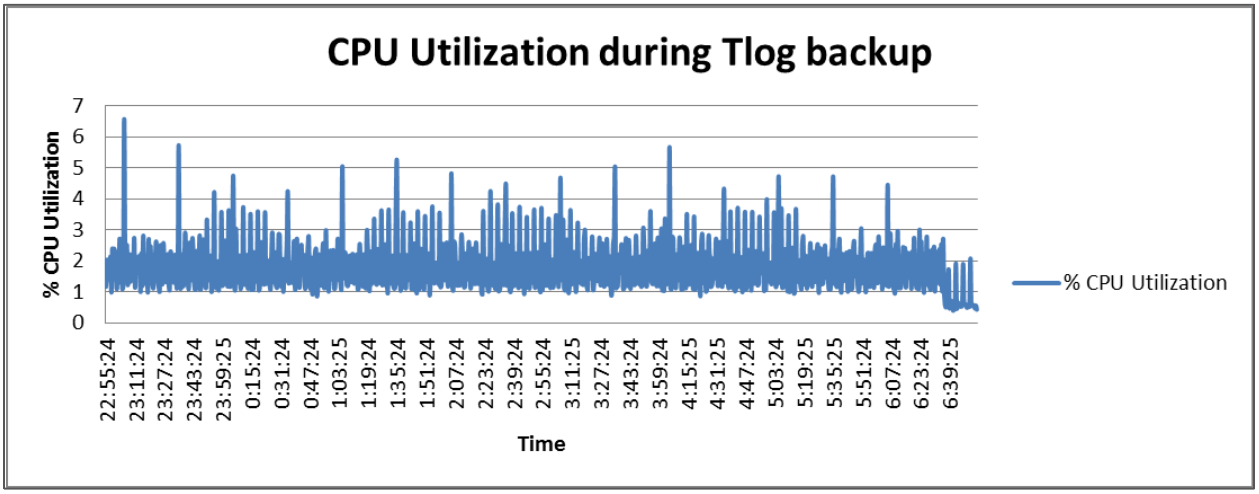

Figure 17 Impact on CPU utilization during transaction log backup

Backup duration and Throughput

The total time required to complete full database and transactional log backups was measured along with throughput during each operation. The baseline full backup in Figure 18 is from the baseline test configuration 2c (i.e. three target volumes with two backup files each) from Figure 7. Figure 18 shows a slight increase in backup time for the backups taken during the user load versus the backup taken with no user load. It also shows the decrease in write throughput of full backup taken with user load compared to the baseline full backup with no user load.

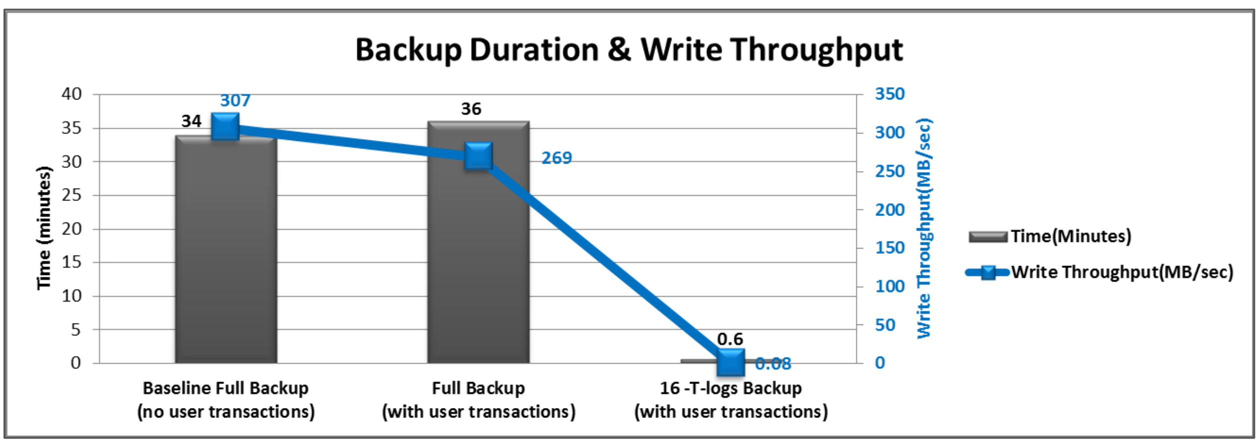

Figure 18 Backup duration and throughput comparisons

As seen in Figure 18, user transactions caused a decrease in write throughput and increase in backup time slightly. However, as shown in previous sections, the impact on server CPU utilization, transaction response times, and transaction I/O latency were much higher during the full backup.

Restore time and throughput

The total time required to complete full database and transactional log restore were measured along with throughput during each operation. Table 3 shows the restore operations performed and their descriptions.

Table 3 Test Case#1 restore operations

Restore Operations	Description
Baseline - Full Restore	Restore of a baseline full backup (Full backup taken when no user transactions running on the Database)
Test Case#1(TC#1)-Full Restore	Restore of a full DB backup (Full Backup set taken while user transactions running on the Database)
Test Case#1(TC#1)-TLog Restore	Restore of 16 TLog backups (Tlog Backups taken while user transactions running on the Database)

To restore a database to a point in time, the last full backup needs to be restored first followed by the transaction log backups until the required point in time. For this test, the full backup was restored initially followed by 16 transaction log backup restores. The "Baseline-Full Restore" results listed in Figure 19 and 20 are from the baseline test configuration 2c (i.e. three target volumes with two backup files each) from Figure 7. Figure 19 shows the restore times in each case. The additional time taken for Test Case#1 full restores was due to the restore of additional data created by user transactions running during backup. Restoring the transaction logs also added time for the restore process along with full backup restore.

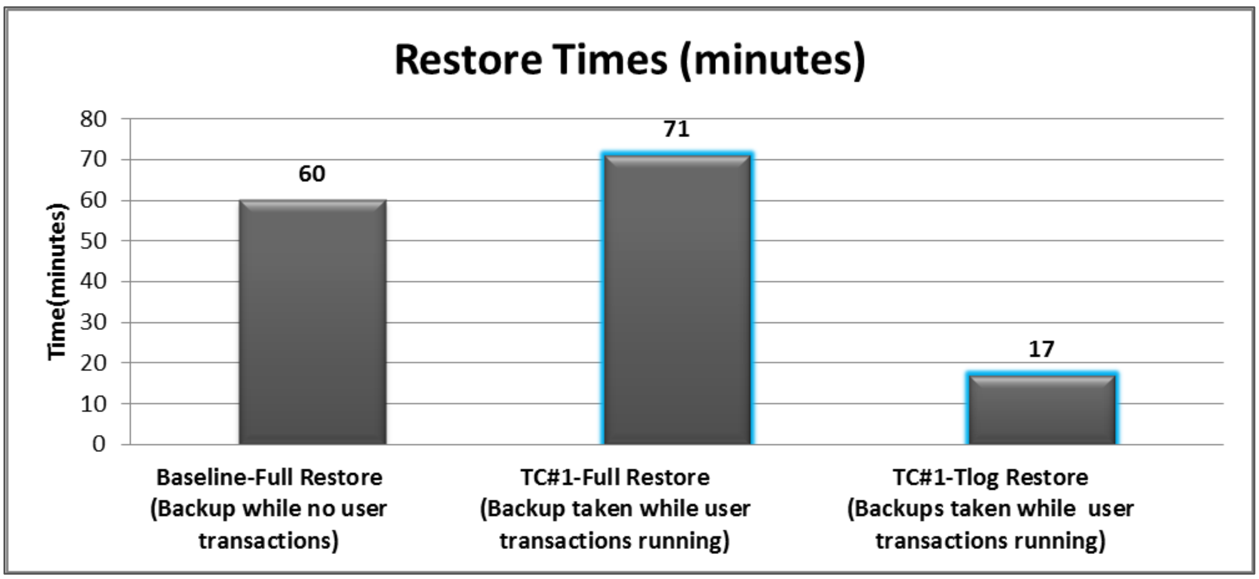

Figure 19 Restore times comparison

Figure 20 shows the breakdown of data restore rates for the backups taken during the user load along with transaction log restores and the backup taken with no user load. There was additional writing involved in Test Case#1 restores since transaction logs were restored after the full backup restores. The read throughput was from the target backup pool and the write throughput was to the data pool. Both the full restore scenarios incurred similar throughput.

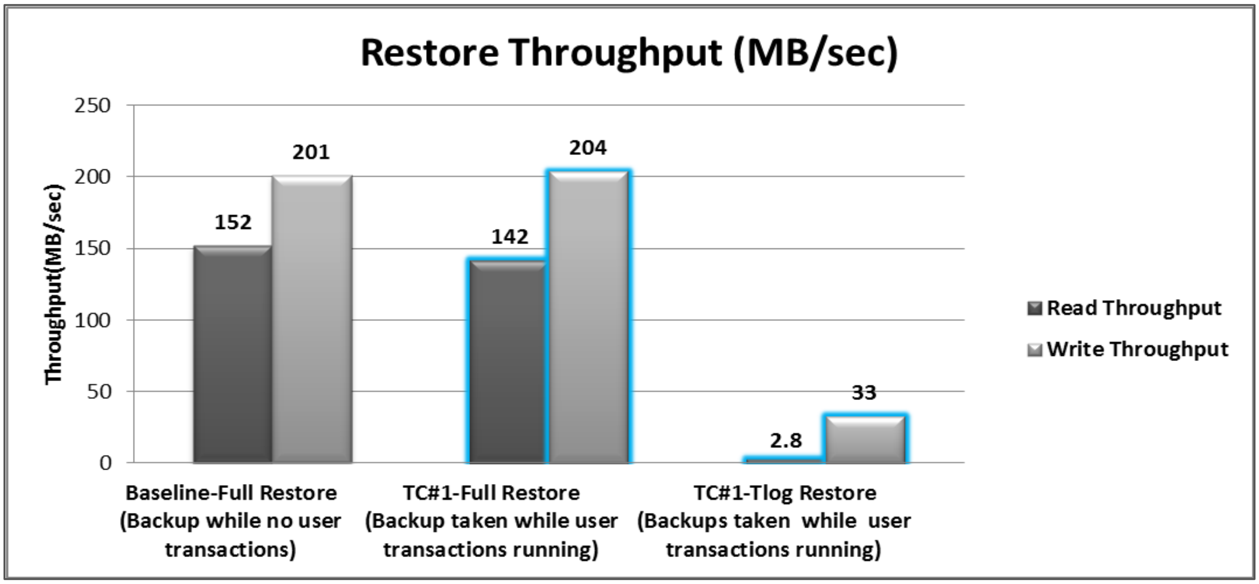

Figure 20 Restore throughput comparisons

4.2.2 Test #2: Performance impact studies during full, tlog, and differential backups

This test simulates an environment with weekly full backups, daily differential backups, and transaction log backups for every 30 minutes assuming the typical production duration to be eight hour workday.

- To simulate this test, we ran the user load from Benchmark Factory for ten hours, scheduled a
  - Full backup after an hour of user load run.
  - After hour and half after the full backup and verify, scheduled transaction log backups for every 30 minutes for seven hours
  - Differential backup after seven and a half hours.
- The backups were verified for integrity and checksums by enabling *Verify backup when finished* and *Perform Checksum before writing to the media* in the backup options screen.
- Performed a restore to a new database from the full, differential, and transaction log backups. The restored database was verified using the T-SQL query, *DBCC CHECKDB*.

For this test we performed full, log, and differential backups as shown in Figure 21. The performance counters measured for this test were CPU utilization, application response times, IO operations at the storage array, backup/restore times from the SQL server, and disk read latency at the storage array. The sections below highlight the performance impact at each step (i.e. full backup, transaction log backup, differential backup, and restore).

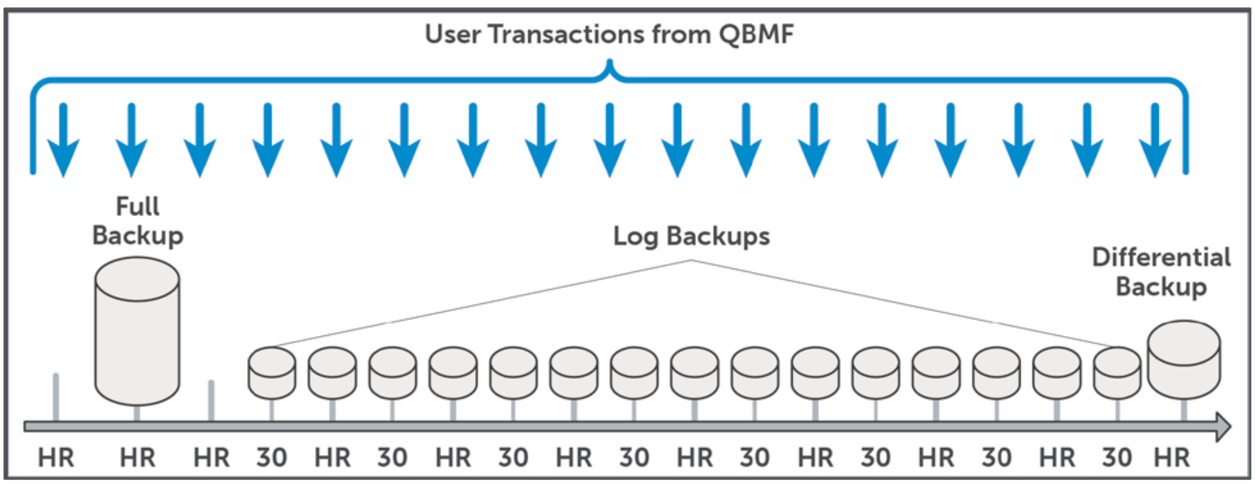

Figure 21 Backup operations in test #2

CPU Utilization

Figure 22 shows increase in CPU utilization during full, tlog, and differential backups. The average CPU utilization percentage increased from 1.7% to 13% during full backup, 1.7% to 10% during verify, and 1.7% to 5% during differential backup. As expected, we saw a significant increase in CPU load on the system during the time of full backup compared to tlog and differential backups. Depending on the overall CPU resources available for SQL server, additional CPU consumption by the backup processes may impact performance of the host server.

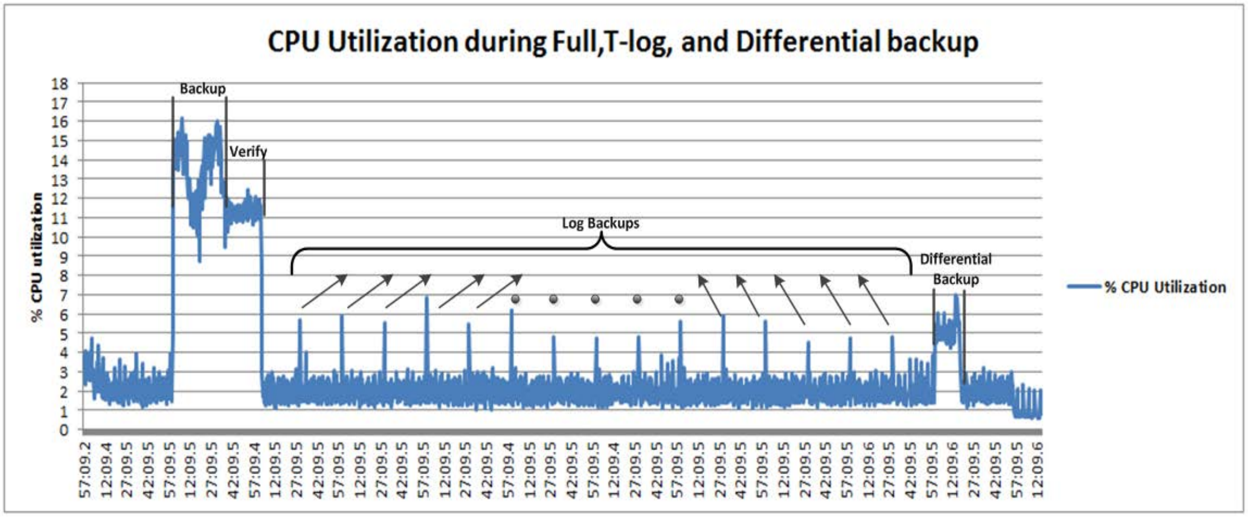

Figure 22 Impact on CPU utilization during full, tlog and differential backups

Application Response Times

Figure 23 highlights the increase in application response time (at QBMF) during full backup. The response time during full backup when user transactions were running was high, compared to before performing a full backup. The application response time increased by 77% during full backup. The impact on the response time during tlog and differential backup were minimal and not shown here.

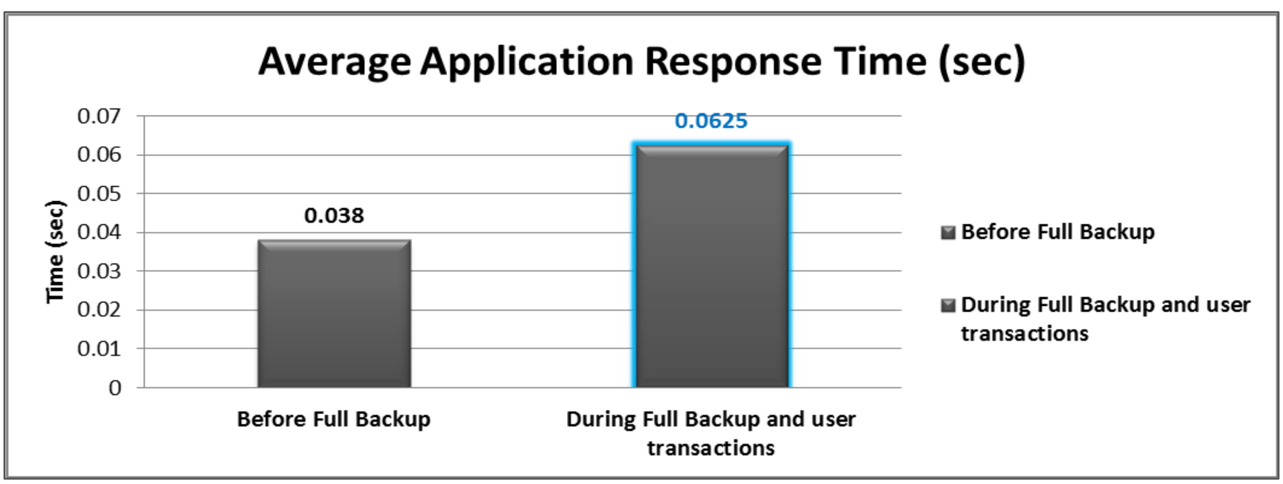

Figure 23 Impact on application response time during full backup and differential backup

Figure 24 shows the IO operations at the storage arrays, recorded using SANHQ. It shows the IOPS before full backup and during full and differential backups. The user transactions have been running throughout; before, during and after backup. As expected the IOPS increased during full and differential backups due to extra read write operations involved during backup process plus IO operations from the user transactions. The figure shows the combined IOPS at both data and backup pools.

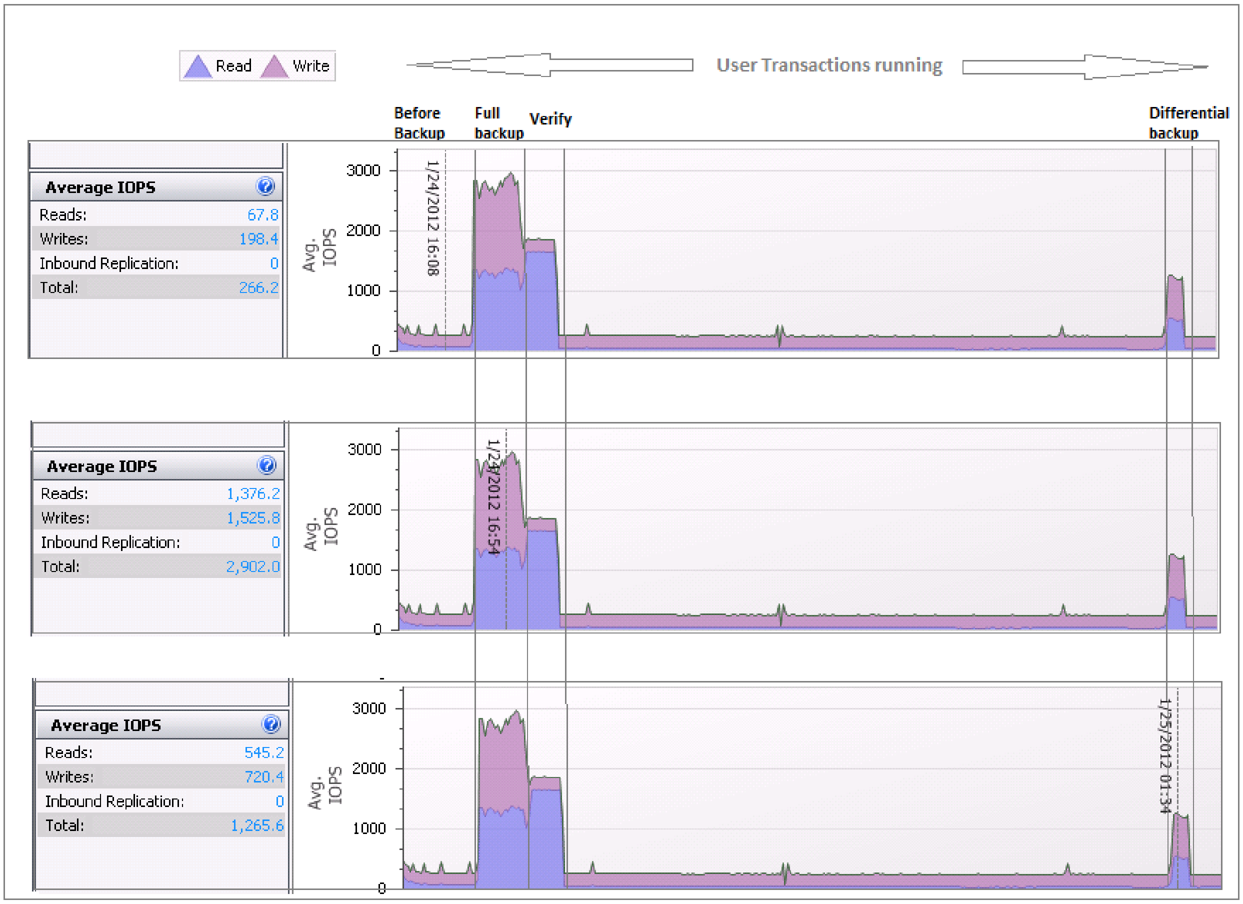

Figure 24 IOPS during full and differential backup

Disk Latency

Figure 25 shows the increase in disk read latency (measured using Perfmon on the SQL Server host) from the EqualLogic array that is part of the data pool during full backup and verify period. The impact on disk read latency was high during full backup compared to during tlog and differential backups because the entire data set backup was taken while the user transactions were running on the same database. Since major impact on disk read latency was during full backup, the impact on disk read latency during tlog and differential (due to the smaller portion of data access and small amount of time used), is not shown in the figure. Read happens across four volumes (source database data spread across four volumes) shown as four drives in Figure 23.

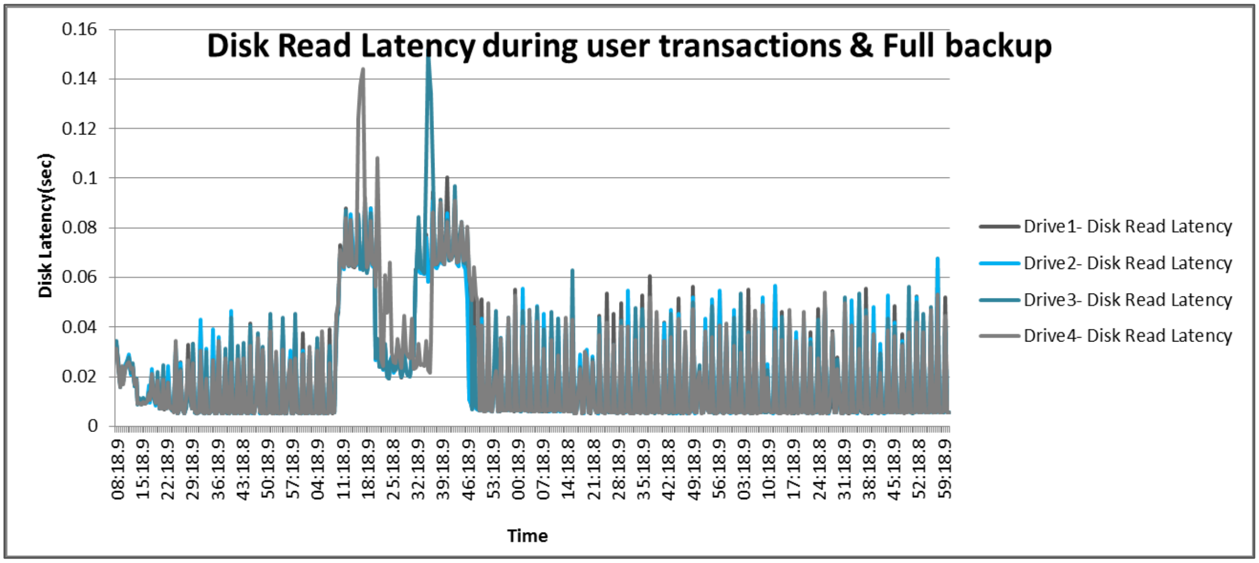

Figure 25 Impact on disk read latency during full backup and verify

Restore time and throughput

The restore duration and throughput of full, transaction log, and differential backup were measured. Table 4 shows the restore operations performed and their descriptions.

Table 4 Restore operations

Restore Operations	Description
Baseline - Full Restore	Restore of a baseline full backup (Backup taken when no user transactions running on the Database)
Test Case#2(TC#2)- Full Restore	Restore of a full DB backup (Full Backup set taken while user transactions running on the Database)
Test Case#2(TC#2)- Differential Restore	Restore of a differential DB backup (Differential Backup taken after full and TLog backups , while user transactions running on the Database)

To recover data to a point in time in an actual restore process, (since full backups were assumed to be taken every week) the previous week's last full backup needs to be restored first, followed by differential backup restore of a specific day, and finally tlog restores until the required point in time.

The "Baseline-Full Restore" in Figure 26 is from the baseline test configuration 2(c) from Figure 7. Figure 26 compares the restore times of baseline and Test Case#2 (Full, Tlog, and Differential Backup). For this test (Test Case#2), it was assumed to restore the data to a specific day in a week and hence the last full backup was restored followed by only the differential backup restore of the specific day (no Tlogs were restored). The full restore time for Test Case#2 took longer compared to the baseline-full restore due to the restore of additional data that was backed up because of the user transactions running during backup. Restoring differential backups, posed additional time for the restore process along with full backup restore.

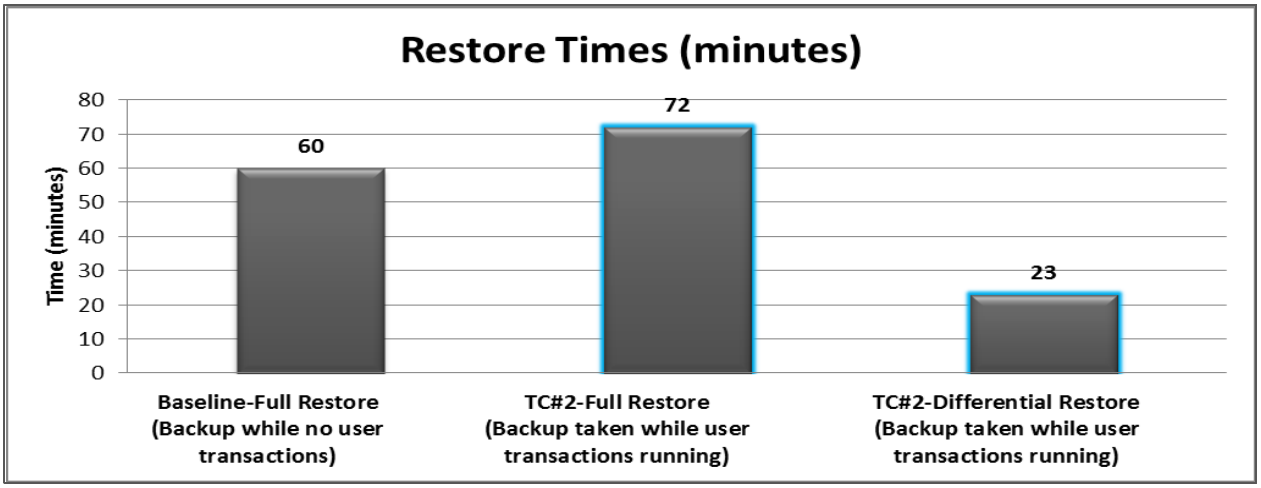

Figure 26 Restore Time Comparisons

Figure 27 shows the breakdown of Test Case#2 data restore rates of the backups taken during the user load and the backup taken with no user load. There was additional writing involved in Test Case#2 restores since differential backups were restored after the full backup restores. The read throughput was collected from the target backup pool and the write throughput from the data pool. The amount of data in the second restore was higher hence it incurred more time, but throughput achieved was almost the same in both the cases.

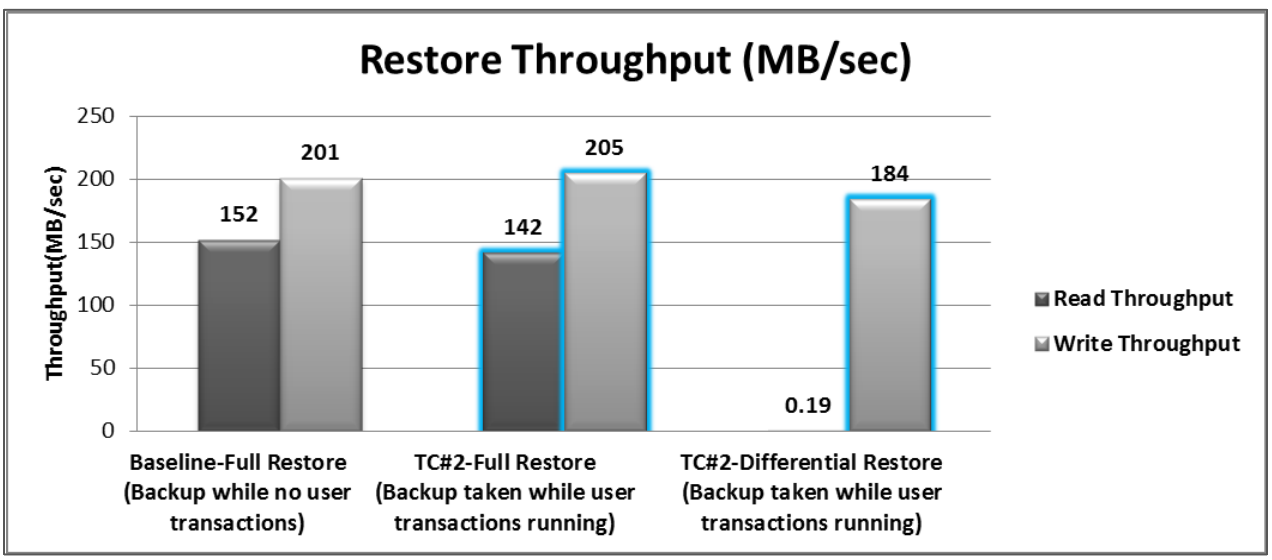

Figure 27 Restore Throughput

4.2.3 Test #3: SQL Server performance impact studies during Compression

SQL compression, if enabled could improve backup storage requirements, however could also impact the resource utilization on the host. With the tests described in this section, the impact of enabling SQL compression on both resource utilization and backup throughput were measured.

For this test the below steps were followed.

- The compression feature was enabled at the server level by choosing it in the *Database Settings* section of the *Server Properties* window.
- Performed a full backup with no user load running.
- Performed a restore of the compressed database backup to a new database.
- The restored database was verified using the DBCC.

Impact of compression on backup and restore

Figure 28, shows the increase in average % CPU utilization from 1.7% to 61% during full backup when compression was enabled. This was due to the additional processing needed for compression at the server .This backup was taken with no user transactions running simultaneously on the database during backup. Database administrators should therefore make sure that if compression is enabled, sufficient CPU resources are available to the SQL host.

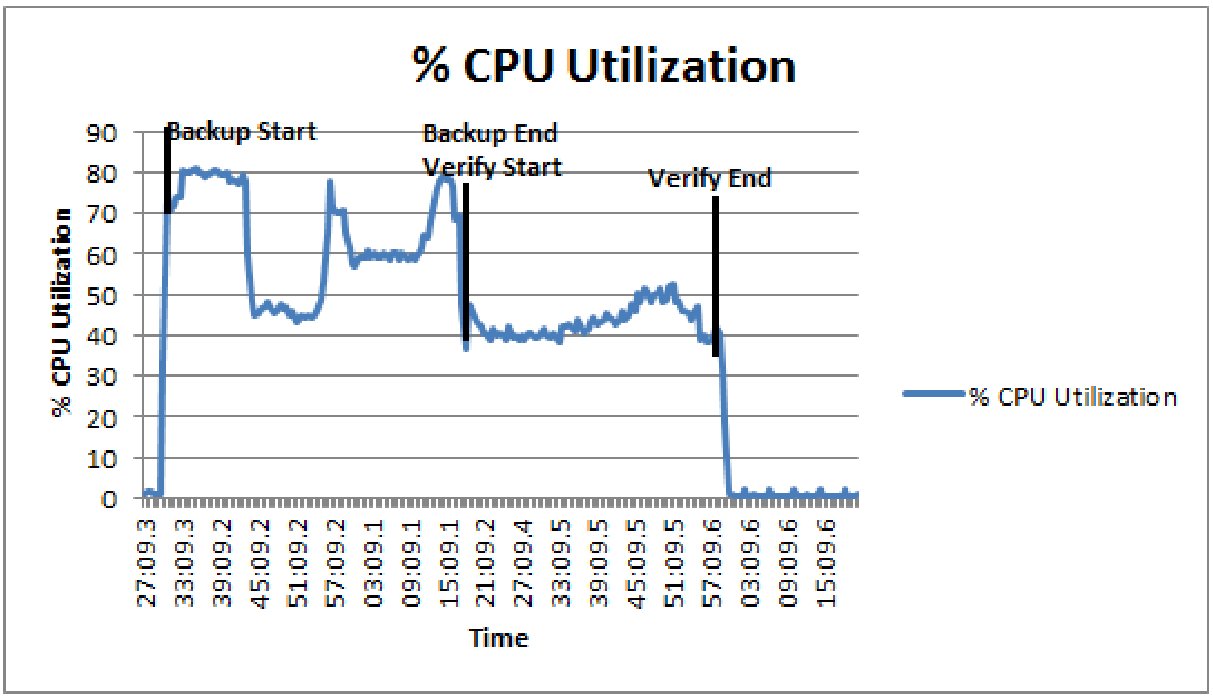

Figure 28 Impact on CPU Utilization during Compression

Backup compression includes processing time and backup time. The backup duration observed during compression was higher than during regular backups due to the higher CPU utilization incurred during compression (Refer Figure 28). However, backup compression reduces the amount of data transferred over the network and also the storage capacity needed to store the backup data.

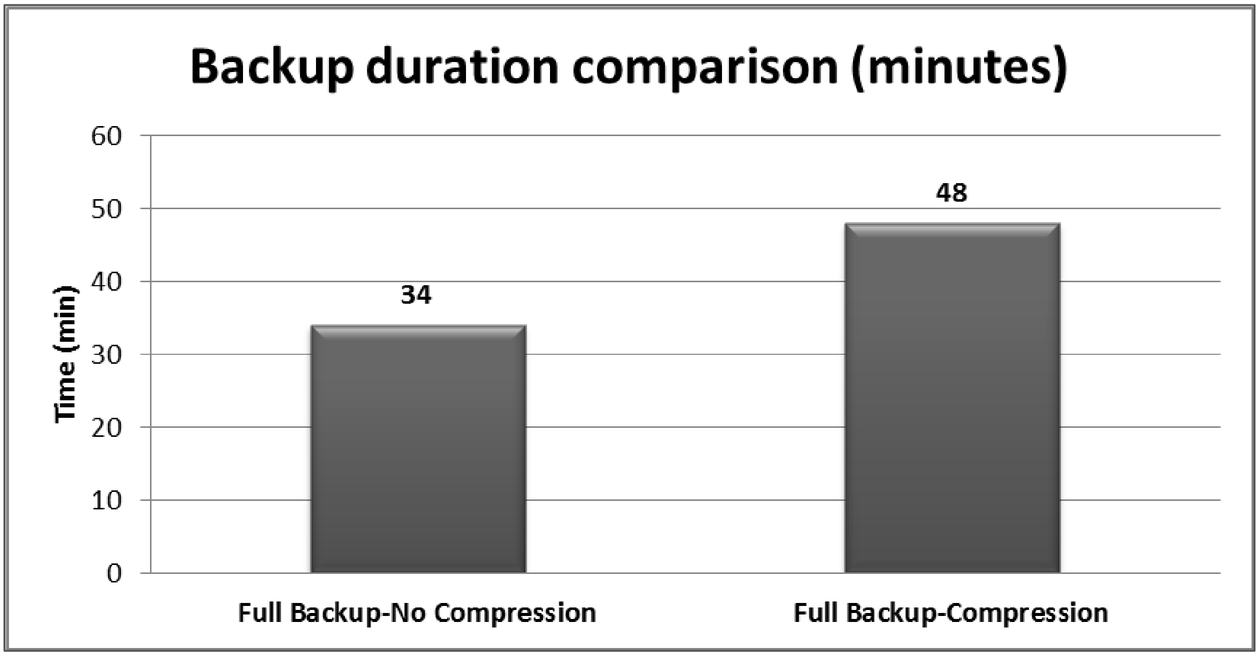

Figure 29 Backup duration comparison

Figure 30 compares the reduction in the backup file size after enabling compression. It also shows the increase in the CPU utilization on the SQL server both during backup and restore.

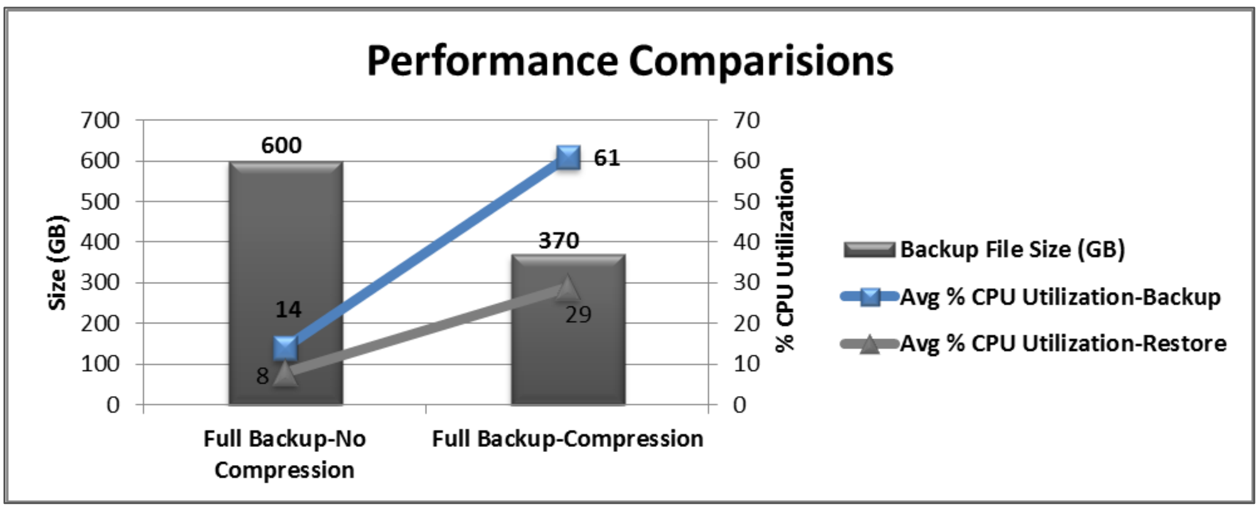

Figure 30 Effects of Compression on Backup and Restore

Note: By default, backup compression significantly increases CPU usage, and the additional CPU consumed by the compression process might adversely impact concurrent operations. The impact could be reduced by creating the backup job as a low-priority compressed backup in a session whose CPU usage is limited by Resource Governor. Refer to http://technet.microsoft.com/en-us/library/cc280384.aspx. If the additional CPU usage is acceptable, a benefit can be achieved in reduced backup sizes and backup/restore elapsed times can be greatly reduced (depending on the compressibility of the database).

4.2.4 Analysis and Conclusion

As expected, we saw a significant increase in CPU utilization, application response times and disk response times during full backup as compared to transaction log and differential backups.

Based on the test results it is evident that running backups during peak periods will have direct impact on:

- The SQL Server host performance: backup processing CPU cycles may impact other application workloads.
- Database performance: increased query response times.

The extra load due to backup processes on the production host is what database administrators want to avoid, hence organizations try to find backup windows or periods of low activity during which backups can be completed. Many organizations even try to use off-host backup capabilities of EqualLogic SAN using backup software solutions such as CommVault Simpana or Symantec Backup Exec. Benefits of using dedicated protection solution in conjunction with EqualLogic snapshots will be discussed in a separate paper.

5 Best Practice Recommendations

5.1 Network Infrastructure

The following are network infrastructure design best practices:

- 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.
- Disable spanning tree on switch ports connecting to end devices such as server ports and storage ports. Enable PortFast for these ports instead.

Note: General recommendations for EqualLogic PS Series array network configuration and performance is provided in the document titled PS Series Array Network Performance Guidelines. http://www.equallogic.com/resourcecenter/assetview.aspx?id=5229.

5.2 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 test configuration in this paper, 15K RPM SAS drives were used on the PS6000XV hosting the database volumes.
- Dedicate a separate storage pool for the production SQL Server data volumes and separate pool for backup volumes for data protection reasons.
- Choose the appropriate RAID type for the PS Series arrays hosting the source database volumes and backup target volumes.
  - Microsoft best practices call for deploying SQL Server log files on RAID 10 volumes when planning for IO intensive workloads such as OLTP where possible for best performance and protection from failures. Since log is write-intensive mostly, RAID 10 would provide better throughput for write-intensive operations .For SQL implementations, RAID 50 can be used to provide maximum storage capacity, in addition to the performance benefits of striping.
  - The test configuration in this paper used RAID 10 for the PS6000XV source array that hosts data and log volumes and used RAID 50 on the PS6500E target array that hosts backup volumes.

Note: 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.* http://www.equallogic.com/uploadedfiles/Resources/Tech_Reports/tr1013-sql-server.pdf

5.3 VMware vSphere ESXi Server/VM

The lab test environment for this paper was comprised of a VMware ESXi server to host SQL Server database virtual machines as well as the QBMF 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.

Virtual machine and guest OS configuration

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, as illustrated in Figure 1 and Figure 2.

- When using the Windows 2008 Server iSCSI initiator within a virtual machine (guest iSCSI), the following best practices apply:
    - Create virtual NICs of type vmxnet3 within the guest VM for connection to iSCSI virtual switches.
    - Enable TSO (TCP Segmentation Offload) and LRO (Large Receive Offload) in the guest VM NICs for iSCSI traffic.
    - Install the Dell EqualLogic 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 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.

5.4 Best practices: Data backup using native SQL server tools

Follow these best practices for data backup using native SQL server tools and EqualLogic arrays.

Database Layout

Ensure that SQL Server databases and associated transaction logs each reside in dedicated volumes. Microsoft recommends separating data and log files onto their own volumes in order to separate random I/O going to the data files from sequential I/O going to the log files. Having database and log files on separate volumes removes the risk of losing both files if the volume fails. For higher resiliency, consider separating data and logs into their own EqualLogic storage pools.

Performance

In order to optimize the backup/restore performance spread the data base data across multiple volumes and backup across multiple target volumes and files. For example, a 1-TB database will fit on one 1-TB volume. However, better performance would be achieved if the database is spread across more volumes. More volumes can result in SQL issuing more I/Os in parallel – thereby improving throughput. Refer to section 4 titled, “Test studies” for details.

The theoretical read I/O rate can be calculated before performing a backup operation to check against the actual values and this would ease in finding the bottleneck areas.

Example: Assuming that the database files are approximately equal in size, the lowest measurement obtained will be the maximum data backup rate that could be expected from the system for that particular database. Theoretical read I/O rate can be achieved by backing up to a null disk; the following T-SQL query can be used for the same.

```
BACKUP DATABASE TPC_C_DB TO DISK = 'NUL' WITH COPY_ONLY
```

It must be noted that a backup to a NUL device is recognized as a valid backup. This means that if a full backup is performed to a NUL device, any differential backups after that are useless unless a full database backup is performed after the NUL backup. The reason is because anything written to NUL is discarded. For instance, if transaction log backups are done to a NUL device, SQL reads over the inactive log records, formats them as for a transaction log backup and hands them off to the operating system which discards the data and sends back an acknowledgement that the data has been written. So SQL thinks that the log chain is intact and discards the log records that were sent to NUL as it would after a normal log backup, because it thinks that they were backed up to disk. Hence log backups to NUL are unusable for restoring the DB because there are log backups missing.

It is worth noting that the backup throughput is first constrained by read throughput. It doesn't matter if the disks can write at a higher rate. If the read throughput is less, then that's the maximum attainable backup throughput. So by having the data spread across multiple volumes and backup across multiple volumes and files, it improves the number of readers and writers involved, thus drastically improving backup performance.

Use a combination of full, differential, and (for the full or bulk-logged recovery model) transaction log backups to minimize recovery time. Differential database backups are normally faster to create than full database backups and reduce the amount of transaction log required to recover the database. In addition to the above, enabling Multipath I/O Device Specific Module (MPIO) using Dell EqualLogic HIT kit at the server, not only provides redundant physical connections to the storage but also enables servers send multiple I/O streams to SAN volumes concurrently thus improving performance of application data hosted on SAN.

General backup strategy best practices

It is very important that backup destinations reside on storage devices (EqualLogic storage pools) that are separate from where the databases are stored. I/O bandwidth can be very critical to the performance of the database. It is recommended to separate the sequential I/O load that is typically generated by backup and restore processing from the random I/O load that is typically generated by transaction processing databases. Besides performance it is necessary for resiliency as well so that disk drives/RAID set failures do not destroy both source and backup data.

- Use the CHECKSUM option of the Backup command. With this option enabled, the backup will verify the page checksums (if they are present), and generate a separate backup checksum for the backup stream that is stored on the backup media. This option will also cause both the backup processing workload and the amount of time to create the backup to increase.
  Example:
  ```
  BACKUP DATABASE TPC_C_DB

  TO DISK = 'B: \Backup\TPC_C_DB.bak

  WITH CHECKSUM
  ```
- Verify the restore process using "DBCC CHECKDB" T-SQL query.
- Always schedule backup operations when database activity is low.
- Back up to a disk first whenever possible before performing a tape backup. Backing up to a disk will greatly increase the performance of the backup process and free the resources of SQL Server. Using file backups also simplifies the restoration process.
- With SQL Server 2008 Enterprise Edition, compressed backups can be done. Compressing database backups has both its advantages and disadvantages. Most of the advantages are related to the reduction of the amount of data stored on the disk and with the speed of the backup and restore processes. The adverse impact on CPU usage could be reduced by creating the backup job as a low-priority compressed backup in a session whose CPU usage is limited by Resource Governor. Refer to http://technet.microsoft.com/en-us/library/cc280384.aspx.

Appendix A Test system component details

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

Table 5 Test Configuration Hardware Components

Test Configuration	Hardware Components
SQL Server® (ESXi01)	One (1) Dell PowerEdge R710 Server running VMware ESXi v4.1, hosting a single SQL Server® Database virtual machine: BIOS Version: 3.0.0 2 x Quad Core Intel® Xeon® E5520 Processors 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 6.2.14, driver version 12.6.0 Two (2) Intel Quad Port VT network adapters (Intel 8257 1Gb). Firmware level 1.3.19.12
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: 3.0.0 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 6.2.14, driver version 12.6.0
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: 3.0.0 Quad Core Intel® Xeon® X5650 Processor 2.26 GHz 48 GB RAM 2 x 146GB 10K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard) – firmware version 6.2.14, driver version 12.6.0
MONITOR SERVER	One (1) Dell PowerEdge R710 Server with Windows Server 2008 R2 for SANHQ: BIOS Version: 3.0.0 Intel® Xeon® X5650 Processor 2.26 GHz 48 GB RAM 2 x 146GB 10K SAS internal disk drives Broadcom 5709c 1GbE quad-port NIC (LAN on motherboard) – firmware version 6.2.14, driver version 12.6.0
Network	2 x Dell PowerConnect 6248 1Gb Ethernet Switch Firmware: 3.2.1.3
Storage	1 x Dell EqualLogic PS6000XV: 14 x 600GB 15K RPM SAS disk drives as RAID 10, with two hot spare disks Dual quad-port 1GbE controllers running firmware version 5.1.1.(H1) 1 x Dell EqualLogic PS6500E: 148x 1TB SATA drives as RAID 50, with two hot spare disks Dual quad-port 1GbE controllers running firmware version 5.1.1(H1)

Table 6 Test configuration software components

Test Configuration	Software Components
Operating systems	Host: VMware vSphere ESXi Server v4.1 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) v3.5.1 installed
Applications	SQL Server® 2008 R2 Enterprise Edition
Monitoring Tools	EqualLogic SAN Headquarters version 2.1 Windows Perfmon
Simulation Tools	Quest Benchmark Factory version 6.5.1

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* at http://www.equallogic.com/resourcecenter/assetview.aspx?id=9831
- *PS Series Array Network Performance Guidelines* at http://www.equallogic.com/resourcecenter/assetview.aspx?id=5229
- *PS Series Groups Deploying Microsoft SQL Server in an ISCSI SAN* at http://www.equallogic.com/uploadedfiles/Resources/Tech_Reports/tr1013-sql-server.pdf
- *Enhancing SQL Server Protection using Dell EqualLogic Smart Copy Snapshots* at http://en.community.dell.com/techcenter/storage/w/wiki/enhancing-sql-server-protection-using-equallogic-smart-copy-snapshots.aspx

The following Microsoft publications are referenced in this document or are recommended sources for additional information.

- *How to: Use Resource Governor to Limit CPU Usage by Backup Compression (Transact-SQL)* http://technet.microsoft.com/en-us/library/cc280384.aspx
- *Optimizing Backup and Restore Performance in SQL Server* http://msdn.microsoft.com/en-us/library/ms190954.aspx
- *Backup Overview (SQL Server)* http://msdn.microsoft.com/en-us/library/ms175477.aspx
- *Backup under the Full Recovery Model* http://msdn.microsoft.com/en-us/library/ms190217.aspx

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