Microsoft SQL Server 2014 OLTP Performance: Measuring SMB 3.0, iSCSI, and FC Protocols

Ron Pratt and Dan Chilton, NetApp
January 2015 | TR-4367

Abstract

High-speed Ethernet, enhancements to the SMB 3.0 protocol, and shared storage enable new data center designs for enterprise applications such as Microsoft Hyper-V and SQL Server. The NetApp® clustered Data ONTAP® 8.3 storage operating system provides a unified, multiprotocol storage solution capable of both SAN and Ethernet protocol access. This report describes a performance test that compares the OLTP performance of the 10Gb iSCSI, 8Gb Fibre Channel (FC), and 10Gb SMB 3.0 storage protocols. We present the results of these tests, including throughput, latency, and server CPU utilization. These results show that customers with demanding OLTP database workloads can obtain good performance from the iSCSI, FC, and SMB 3.0 storage protocols.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

The connectivity options available for deploying enterprise applications on shared storage continue to grow. Ethernet speeds have risen rapidly from 10Gb to 40Gb and are likely to hit 100Gb in the next few years. In addition, FC network connectivity speeds have risen from 8Gb to 16Gb, and a target of 32Gb is in sight.

For server infrastructure deployed on shared storage, such as Microsoft Hyper-V or enterprise applications such as Microsoft SQL Server, customers can now choose from file protocols such as SMB 3.0 or block protocols such as iSCSI, Fibre Channel over Ethernet (FCoE), or FC. Although some storage systems are only architected for NAS or SAN, NetApp clustered Data ONTAP provides flexibility in storage networking design because it is multinetwork and multiprotocol capable. With all of these choices, customers can create powerful SAN deployments on clustered Data ONTAP or leverage SMB 3.0 for all-Ethernet network designs, while retaining data protection, storage efficiency, and performance capabilities.

1.1 Microsoft SQL Server 2014

When the term *mission critical* is used, one of the first applications that comes to mind is an OLTP database served by Microsoft SQL Server 2014. Businesses depend on these applications for use cases such as stock market trading, flight reservation systems, and real-time retail transactions. The application workloads are usually small block and I/O intensive, requiring powerful server, caching, and storage resources to meet demand. SQL Server 2014 comes with important enhancements that increase performance and availability:

- **In-Memory OLTP engine.** The new In-Memory OLTP engine provides significant OLTP performance improvements by moving selected tables into memory.
- **Improved scalability.** SQL Server 2014 can scale up to 640 logical processors and 4TB of memory on a physical server. It can scale to 64 virtual processors and 1TB of memory when running in a virtual machine (VM).
- **Enhancements to AlwaysOn Availability Groups.** AlwaysOn Availability Groups can now have up to eight replicas (increased from four in SQL Server 2012).
- **Buffer Pool Extension.** The new Buffer Pool Extension feature allows you to use high-speed flash drives to extend SQL Server 2014's buffer pool, thereby improving performance in systems that have maxed out their memory capacity.

1.2 SMB Protocol

The CIFS (SMB 1.0) protocol has humble origins as a Microsoft client file-sharing protocol developed in the 1990s. To share files over the network, CIFS was known to be chatty, requiring a high level of overhead for metadata operations. Slow network speeds of 1GbE and a single network connection between the client and the server also made CIFS a slow protocol. SMB 3.0 greatly reduces protocol chattiness, leverages multiple network connections, and provides faster Ethernet network speeds.

In the past, SQL Server workloads that were deployed on shared storage solutions could only be deployed with block protocols such as iSCSI or FCP. Microsoft continued to enhance the SMB protocol as it moved from CIFS to SMB 2.0 and SMB 3.0.

Protocol enhancements optimized performance and added resiliency and high availability. SMB performance was improved with features such as compound operations and a reduction in overall operation types. Resiliency was added to the protocol with continuously availability shares, SMB witness, and clustered client failover. With these enhancements, SMB 3.0 is now appropriate for the most demanding SQL database workloads. The next generation of Microsoft SQL deployments will be built on hybrid clouds that extend the database from on-site private clouds to Microsoft Azure and back again.

1.3 NetApp Clustered Data ONTAP 8.3

Consistency, dependability, and high performance are essential for SQL databases deployed on shared enterprise storage. These features must be coupled with nondisruptive operations (NDO), high availability, scalability, and storage efficiency. Customers can depend on clustered Data ONTAP 8.3 to provide these essential elements. The clustered Data ONTAP unified scale-out solution incorporates NDO, high availability, and strong performance by design, with the ability to scale up or scale out as business grows.

Data ONTAP has been a leader since 2012, helping NetApp to become a global enterprise scale-out storage and data management fabric provider. With new choices come new opportunities, and NetApp now offers SMB 3.0 as a data center storage protocol in addition to traditional block protocols such as iSCSI, FCoE, and FC. On-site ready and cloud-connected, clustered Data ONTAP is a complete solution that is future-proof in a rapidly changing technology environment.

1.4 Microsoft SQL Server on NetApp: Testing Performance Across Protocols

Although multiprotocol comparative tests have demonstrated good performance for Hyper-V deployed on SMB 3.0, we wanted to test SQL Server as the primary workload. It is possible that SMB 3.0 is ready for SQL database workloads, but might not be able to compete with FC in terms of heavy transactional OLTP performance. To address these concerns, we built several rigorous performance test plans to evaluate SQL deployed over SMB 3.0, iSCSI, and FC.

We completed two exhaustive rounds of comparative performance tests. In the first round of tests, we used multiple small databases and found performance to be comparable between SMB 3.0, iSCSI, and FC. However, we wanted to confirm these results with larger databases more typical of real world situations. Therefore, we completed a second exhaustive round of performance tests with a heavy OLTP transactional database workload. The results of this second round of tests are related in this technical report.

2 Executive Summary

We set up the test environment according to Microsoft and NetApp best practices. When architecting a high-performance, fault-tolerant SAN environment, we always use multiple cards and switches, multipathing, clustering, and so on. We applied a similar architecture to the SMB 3.0 design by using multiple 10GbE network interface cards on the servers and by using storage logically connected into separate private VLANs on the switch. The SQL Server database we used contained four database files, and we created four SMB connections across the network interfaces to distribute the network connections for performance.

Our tested configuration evenly distributed the I/O traffic across multiple paths, resulting in optimal performance. Deploying a different SMB network configuration may result in performance that differs significantly from the results shown in this report.

For the SQL Server database server hardware, we used a single Fujitsu RX300 S7 server with two 6-core Intel Xeon E5-2630 2.30GHz processors and 128GB of RAM. The server was installed with Microsoft Windows Server 2012 R2 and Microsoft SQL Server 2014. For the storage platform, we used a NetApp FAS8040 storage system, which DCIG recently ranked as best-in-class in its 2014 buyer's guide.[1]

---

[1] DCIG 2014–15 Midrange Unified Storage Array Buyer's Guide.

The pair of highly available FAS8040 nodes was installed with clustered Data ONTAP 8.3. The data storage drives deployed were 10K RPM SAS drives enhanced with NetApp Flash Cache™ 512GB controller-attached PCIe intelligent caching modules. By design, clustered Data ONTAP acts as a hypervisor of storage pool resources, allowing true storage virtualization into storage virtual machines (SVMs) and allowing logical network virtualization of both Ethernet ports and FC HBAs into logical interfaces (LIFs). We deployed a single SVM with multiple LIFs, volumes, and LUNs striped across the two active-active storage controllers.

We used high-speed, best-in-class switches from key NetApp partners for both Ethernet networking and FC SAN. A Cisco Nexus 5596 switch was deployed with full 10Gb network speeds and multiple VLANs and ports used for both server and storage connectivity. SAN connections were provided by a Brocade 300e switch with an HBA port speed of 8Gb.

2.1 Test Workload and Metrics Measured

The load-generating software we used simulated an OLTP database deployed in a real-world configuration. This simulation included actual SQL Server databases, logs, temp space, and so on. The workload emulated an online stock brokerage system with multiple tested load points. The number of stock-trade transactions processed per minute was a high-level throughput metric generated by the load-generating software.

Although we considered several SQL Server load-generating tools, the tool we selected had to meet certain requirements. First, for the test to run, a full set of SQL Server database, logs, and temp files had to be created and connected. Rather than drive an SQL-like workload mix against a file, we wanted to test against a real database like a customer would deploy. Also, we wanted a workload pattern that was consistent and recognized as typical of an OLTP database.

We tested at multiple load points with multiple iterations to confirm valid results. Upon the completion of these tests, we exhaustively tested and analyzed the performance of each solution at each layer of the data center stack.

The following statistics were measured for each protocol (FC, iSCSI, and SMB 3.0):

- Latency at the SQL Server
- Latency at the storage controller
- Throughput in transactions
- Throughput in IOPS
- SQL Server CPU utilization

2.2 Performance Summary

Performance on SMB 3.0 was impressive. Although, we tested with both large and small databases, single and multiple connections, and a mix of working set sizes, the trend remained the same. SMB 3.0 performance was equivalent to traditional block protocols.

The test results demonstrated that:

- SMB 3.0 provided low-latency performance comparable with FC and iSCSI.
- The number of completed transactions for the test period was equivalent for all protocols, with FC being 2% and 3% better than SMB 3.0 and iSCSI, respectively.
- Throughput in IOPS was equivalent across all protocols.
- CPU utilization was not significantly different for the server and storage controller resources when using FC, iSCSI, or SMB 3.0.

As mentioned earlier, our tested configuration evenly distributed I/O traffic across multiple paths resulting in optimal performance. Deploying a different SMB network configuration might result in performance that differs significantly from the results shown in this report.

3 Performance Comparison Details

As mentioned previously, the workload that we used was an intensive OLTP workload that simulated a stock market. We performed the tests multiple times and analyzed results from the top to the bottom of the stack. The results in this section are normalized by taking the actual numbers and placing them on a relative scale with 8Gb FC as the baseline. On the charts, FC is valued as 100, and SMB 3.0 and iSCSI results are calculated relative to that value.

3.1 Microsoft SQL Server Latency

We measured latency both at the server level and at the storage level for the following reasons. Latency is often more important than throughput for demanding OLTP workloads. We wanted to make sure that there were no hidden bottlenecks in the stack between the components that included the test suite, the SQL Server application, the server, the protocol, the switch, and storage. We made sure that latency remained low and comparable between the components.

We then took the results and created charts with FC as the baseline and with the other latencies relative to FC. For example, in Figure 1, when the latency for SMB 3.0 is shown as 100, it is not 100 milliseconds of latency. Instead, it indicates a normalized latency of 100 that is equal to the latency of FC. Actual latency figures in milliseconds are not shown.

Figure 1) Relative SQL Server average latency (lower numbers are better).

■ iSCSI	100
▨ FC	100
■ SMB3	100
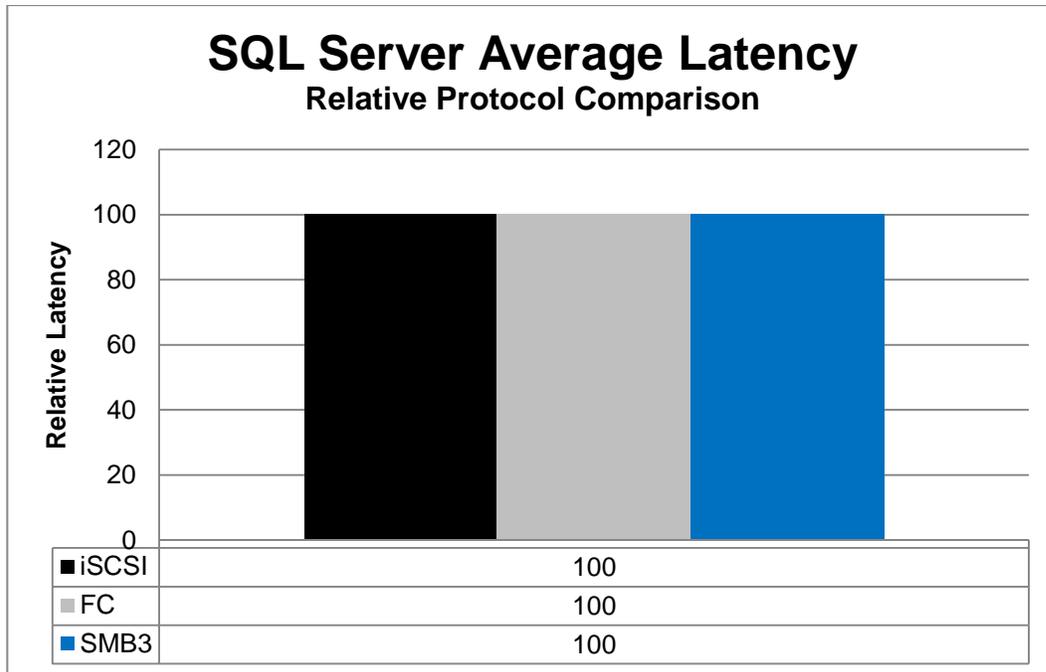
3.2 Storage Controller Latency

We found the storage controller latency to be nearly equivalent to the latency measured at the database and at SQL Server. We anticipated these results, meaning that there were no unnecessary latency sources in the stack. Figure 2 shows that FC and SMB 3.0 latencies were equivalent, with iSCSI being slightly higher.

Figure 2) Relative storage controller average latency (lower numbers are better).

Storage Controller Average Latency

Relative Protocol Comparison

■ iSCSI	108
■ FC	100
■ SMB3	100
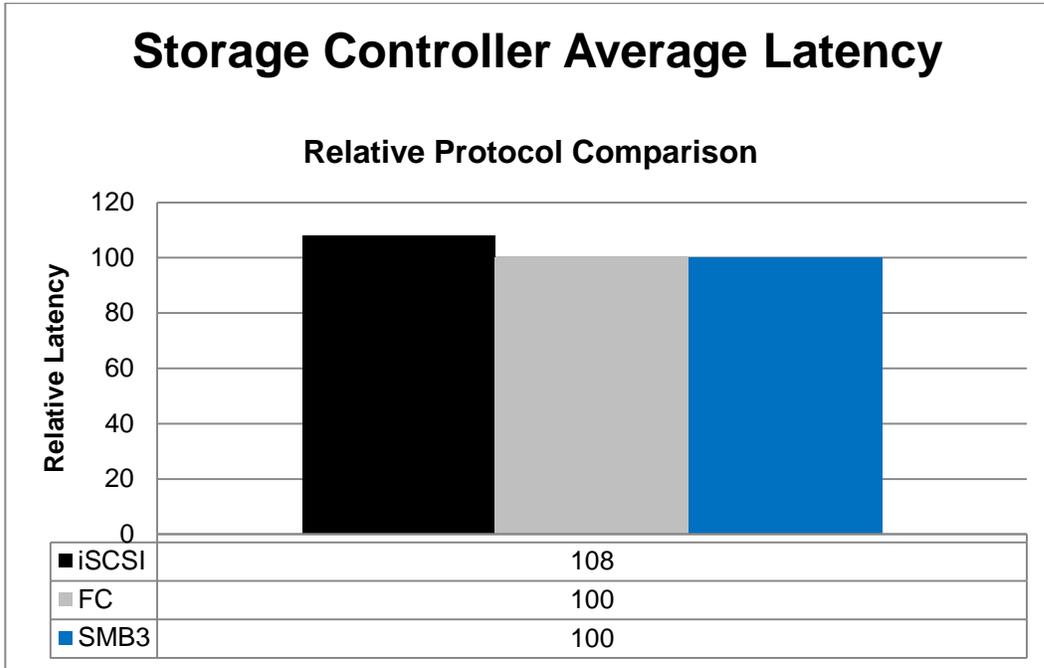
3.3 Stock Market Database Transaction Throughput

The number of stock trade transactions was determined during the test time period. This was normalized to a scale of 100 with FC as the baseline. As is shown in Figure 3, the FC transaction count had a slight performance lead over SMB 3.0 and iSCSI, which were measured at 98 and 97, respectively.

Figure 3) Relative SQL Server average transactions (higher numbers are better).

SQL Server Average Transactions
Relative Protocol Comparison

|	
■ iSCSI	97
■ FC	100
■ SMB3	98
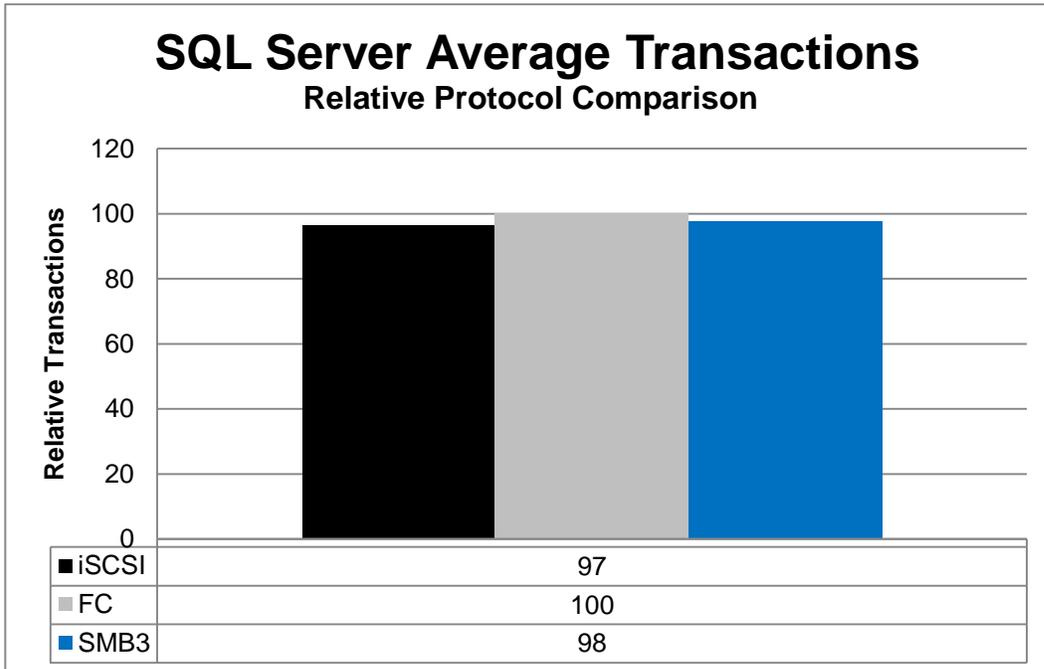
3.4 Storage Controller Throughput in IOPS

The IOPS was measured at the storage controller level to determine whether each protocol required the same amount of IOPS to complete the transactions. CIFS (SMB 1.0) required metadata operations such as the `open`, `close`, and `check` attributes to complete reads or writes. Therefore, SMB was known as a chatty protocol with significant overhead. More efficient protocols have less overhead, using fewer operations to accomplish the same throughput level.

Many performance improvements were made as the SMB protocol progressed from 1.0 to 3.0. In the SMB 3.0 protocol, chattiness has been eliminated through multiple improvements such as compound operations. SMB 3.0 requires a quantity of IOPS similar to FC and iSCSI to achieve the same high-level transaction count. As can be seen in Figure 4, storage controller throughput is equivalent across the protocols.

Figure 4) Relative storage controller average throughput in IOPS.
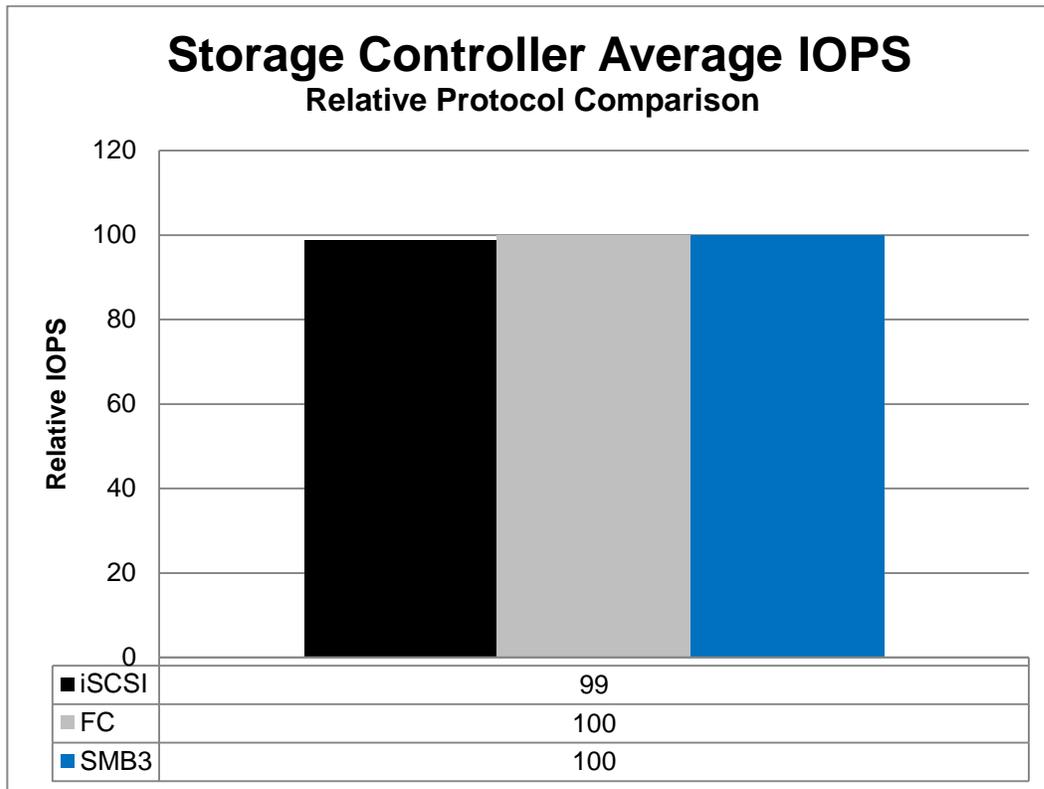
3.5 Microsoft SQL Server CPU Utilization Comparison

SQL Server CPU utilization was consistently approximately 67% when tested with each protocol. Storage controller CPU utilization is not presented in this report, but it was also consistent and lower than 50% for each protocol tested. Clustered Data ONTAP is a highly parallelized operating system that distributes work across multiple cores for each storage controller. The test determined that the storage controller CPU cores were evenly used for each of the protocols tested.

Figure 5) Microsoft SQL Server 2014 average CPU utilization.

SQL Server CPU Utilization
Protocol Comparison

	Relative CPU Utilization
■ iSCSI	67
■ FC	69
■ SMB3	67
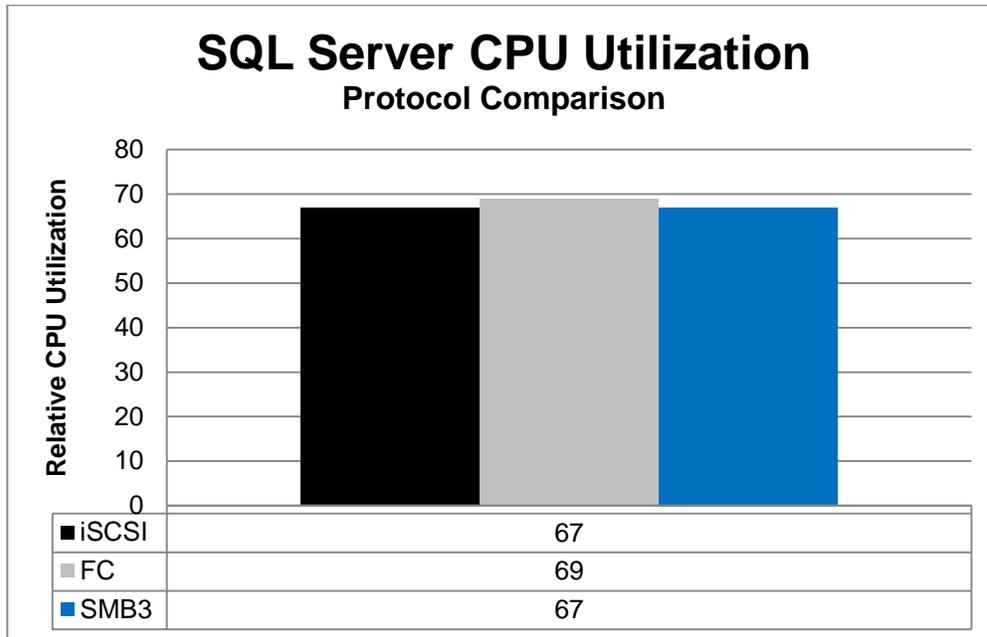
4   Test Bed Configuration and OLTP Workload

This section provides hardware and software configuration details for the test bed configuration. CPU processing speeds, memory counts, network interfaces, and so on were chosen to provide power and to avoid bottlenecks, with the goal of testing the capabilities of each protocol for an SQL Server OLTP workload. This section also provides details about database layout and the OLTP workload and test methods.

4.1   Hardware and Software

Table 1, Table 2, and Table 3 show the hardware and software components that we used for the test bed configuration. The configuration included one medium-sized SQL Server connecting to a large SQL Server database that was split into multiple database files and striped across an HA pair of NetApp storage controllers. The load-generating test suite was installed on a single physical server and a single VM.

Table 1) Microsoft SQL Server hardware and software.

Hardware or Software Component	Details
SQL Server	1 x Fujitsu RX300 S7
Server operating system	Windows 2012 R2 Standard
SQL Server database version	SQL Server 2014 Enterprise Edition
Processors	2 x 6 core Xeon E5-2630 @ 2.30GHz
Memory	128GB
FC network	8Gb
FC HBA	QLogic QLE2562 dual-port PCIe FC

Hardware or Software Component	Details
Dedicated NICs for 10GbE iSCSI and SMB 3.0	Intel 82599 10 Gigabit Ethernet dual-port network connection
Dedicated public 1GbE ports	2 x Intel I350GbE ports
8Gbps FC switch	Brocade 300E
10GbE switch	Cisco Nexus 5596

Table 2) NetApp storage controller hardware and software.

Hardware or Software Component	Details
Storage controller	2 x FAS8040 configured as a highly available (HA) pair
Clustered Data ONTAP	Version 8.3RC1
Total number of drives	144
Size of drives	600GB
Drive speed	10K RPM
Drive type	SAS
Flash Cache	2 x 512GB (1 per node)
FC target ports	4 x 8Gb (2 per node)
Ethernet ports	4 x 10Gb (2 per node)
SVMs	4 (1 x admin, 2 node, 1 x data)
Ethernet LIFs	4 x 10Gb data (2 per node connected to separate private VLANs)
FC LIFs	4 x 8Gb data

Table 3) Load-generating server hardware and software.

Hardware or Software Component	Details
Load-generating server	1 x Fujitsu RX300 S6
Server operating system	Windows 2012 R2 Standard
Processors	2 x 6 core Intel Xeon E5645 @ 2.4GHz
Memory	48GB
Public network connection	1 x Intel I350 Gig ports
Private network connection	1 x Intel 82599 10GbE ports

Table 4) Hyper-V server hardware and software

Hardware or Software Component	Details
Hyper-V server	1 x Fujitsu RX300 S7
Server operating system	Windows 2012 R2 Standard

Hardware or Software Component	Details
Processors	2 x 6 core Xeon E5-2630 @ 2.30GHz
Memory	48GB
Public network connection	2 x Intel I350 Gigabit Ethernet ports
Private network connection	2 x Intel 82599 Gigabit Ethernet ports
Load-generating client	1 x VM
VM operating system	Windows Server 2012 R2
Processors	12 x vCPU
Memory	12GB
Public network connection	1 x Intel I350 Gigabit Ethernet ports
Private network connection	1 x Intel 82599 10GbE ports

4.2 Network Design

This section provides the network connectivity details for the configurations tested. Figure 6 shows that the SMB 3.0 and iSCSI connections to the Cisco Nexus 5596 switch were through two 10GbE ports on the SQL Server. Each storage controller node also had two 10GbE connections, for a total of four connections across the HA pair, connected to the same switch. These connections were logically separated into two distinct private VLANs.

The FC SAN was deployed with a Brocade 300e 8Gb FC switch, and each server and storage controller node had two ports connected into the SAN. Multiple ports used in both Ethernet and FC SAN configurations provided both with high availability through multipathing and increased bandwidth. At no point in testing was network connectivity a bottleneck. The storage controller nodes were also connected by a separate high-speed interconnect for intracluster communication.

Figure 6) Network design.
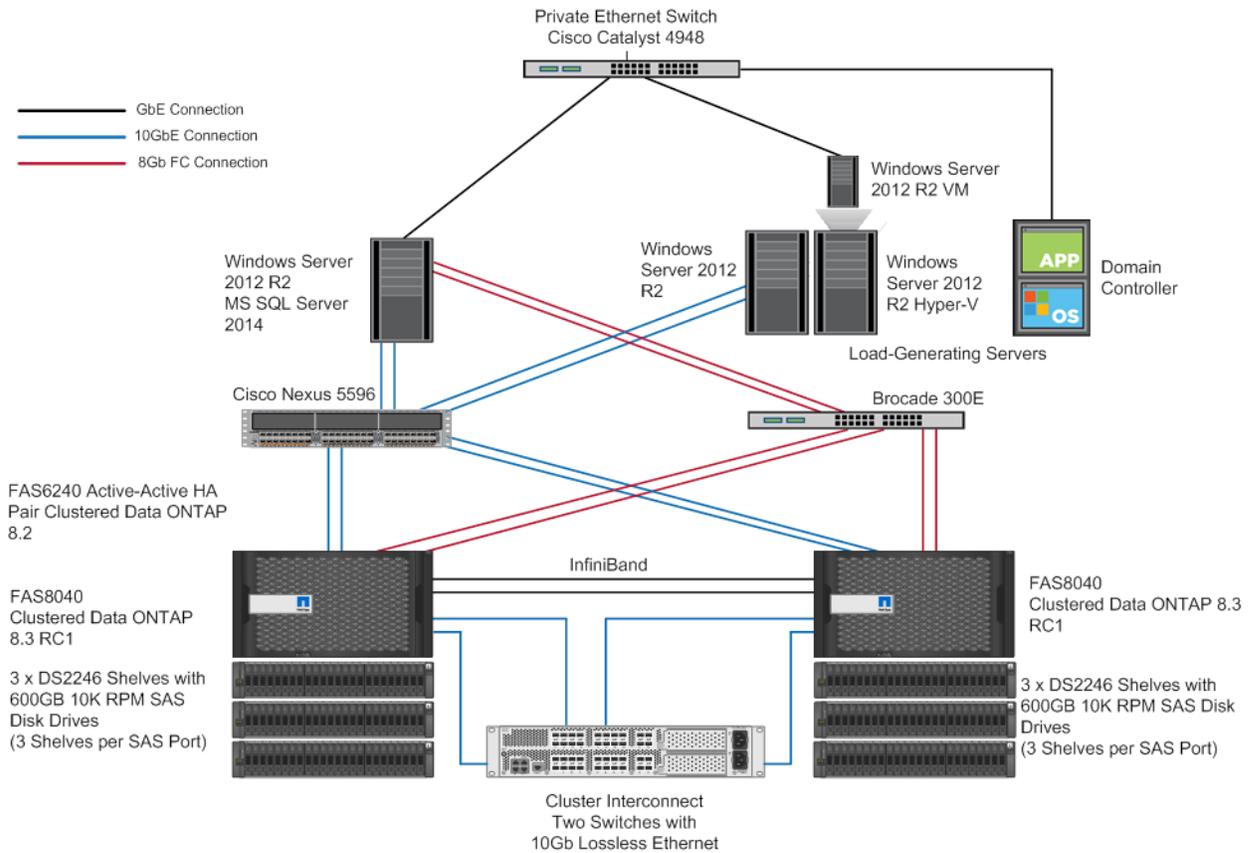
4.3 Database Layout and Storage Provisioning Design for FC and iSCSI

Table 5 summarizes the disk layout for the databases. The storage controller had one spare disk and a small three-disk aggregate for the operating system and configuration files. The database used to host the simulated stock market transaction environment was approximately 1.25TB in size and contained 4 files.

Each controller had a large data aggregate. Each data aggregate contained sixty 600GB 10K RPM SAS drives that were subdivided into 4 NetApp RAID DP® groups with 15 drives each. The Controller 1 data aggregate stored the first two database files and the logs. The Controller 2 data aggregate stored the last two database files and the temp database (`tempdb`). `tempdb` included 8 database files along with the temp logs. Storage processing power was enhanced by a 512GB enterprise-grade flash memory Flash Cache cache connected through PCIe to each controller.

Each database file was contained in a separate LUN. A NetApp Snapshot® copy of the volume containing the LUN was created, which allowed the LUN to be restored to the original initialized state after each test. This allowed easy testing and retesting of the dataset. One igroup containing all of the initiators for iSCSI and another igroup for FC were configured on the cluster. The same LUNs were used for both the FC and iSCSI tests by unmapping the LUNs from their respective igroup for FC and remapping them to an igroup for iSCSI. The LUNs were formatted by Disk Management with GPT and the NTFS file system by using a 64KB block size.

The FC SAN was configured according to typical best practices zoning recommendations on the Brocade switch. The Microsoft native multipath input/output (MPIO) driver and Microsoft DSM were used to provide multipath access to the storage controllers. Clustered Data ONTAP provided Asymmetric Logical Unit

Access communication to the initiators so that optimal paths were used for host I/O access according to the MPIO load-balancing policies on the host.

Table 5) Logical storage layout for FC and iSCSI.

Storage Controller	Aggregate Name	Volume Name	LUN	Size	Description
Controller 1	aggr0			3 disks	
		Vol0		442.4GB	Root volume
	aggr1_n1			60 disks	
		Db1	Lundb1		Database file 1
		Db2	Lundb2		Database file 2
		Logs	Lunlogs		Transaction logs
Controller 2	aggr0_sqlsmb3_02_0			3 disks	
		Vol0		442.4GB	Root volume
	aggr1_n2			60 disks	
		Db3	Lundb3		Database file 3
		Db4	Lundb4		Database file 4
		Tmp1	Luntmp1		Temp database file 1
		Tmp2	Luntmp2		Temp database file 2
		Tmp3	Luntmp3		Temp database file 3
		Tmp4	Luntmp4		Temp database file 4
		Tmp5	Luntmp5		Temp database file 5
		Tmp6	Luntmp6		Temp database file 6
		Tmp7	Luntmp7		Temp database file 7
		Tmp8	Luntmp8		Temp database file 8
		Tmplog	Luntmplog		Temp database file transaction log

We mounted the database files over multiple paths by using multiple initiators and targets.

4.4   Database Layout and Storage Provisioning Design for SMB 3.0

Table 6 summarizes the disk layout for the databases. The storage controller had one spare disk and a small three-disk aggregate for the operating system and configuration files. The database used to host the simulated stock market transaction environment was approximately 1.25TB and consisted of 4 files.

Each controller had a large data aggregate. Each data aggregate contained sixty 600GB 10K RPM SAS drives that were subdivided into four RAID DP groups with 15 drives each. The Controller 1 data aggregate stored the first two database files and the logs. The Controller 2 data aggregate stored the last two database files and the temp database (tempdb). The temp database was composed of eight

database files along with the temp logs. Storage processing power was enhanced by a 512GB enterprise-grade flash memory Flash Cache connected through PCIe to each controller.

Each database file was stored in a separate volume on the data aggregate. A Snapshot copy of the volume was created, which allowed the volume to be restored to the original initialized state after each test. This allowed easy testing and retesting of the dataset. Each volume had a CIFS (SMB) share attached to it so that the SQL Server could mount the databases. There were no special permissions configured on the volume or CIFS share.

Table 6) Logical storage layout for SMB 3.0.

Storage Controller	Aggregate Name	Volume Name	Size	Description
Controller 1	`aggr0`		3 disks	
		`Vol0`	442.4GB	Root volume
	`aggr1_n1`		60 disks	
		`Db1`	1536GB	Database file 1
		`Db2`	1536GB	Database file 2
		`Logs`	1536GB	Transaction logs
Controller 2	`aggr0_sqlsmb3_02_0`		3 disks	
		`Vol0`	442.4GB	Root volume
	`aggr1_n2`		60 disks	
		`Db3`	1536GB	Database file 3
		`Db4`	1536GB	Database file 4
		`Tmp1`	600GB	Temp database file 1
		`Tmp2`		Temp database file 2
		`Tmp3`		Temp database file 3
		`Tmp4`		Temp database file 4
		`Tmp5`		Temp database file 5
		`Tmp6`		Temp database file 6
		`Tmp7`		Temp database file 7
		`Tmp8`		Temp database file 8
		`Tmplog`		Temp database file transaction log

With a design similar to the FC and iSCSI configurations, the database files were mounted over multiple network connections. Clustered Data ONTAP 8.3 does not include the SMB 3.0 multichannel feature. Nevertheless, to increase parallel processing of commands, multiple SMB sessions can be created from the server to storage over multiple connections.

The connections we created were separate 10Gb interface cards connected into separate private VLANs. This allows enhanced bandwidth and failover capabilities and optimal striping of database file access. Logically, this configuration appears as follows:

```
\\192.168.1.100\db1\db1.ndf
\\192.168.10.100\db2\db2.ndf
\\192.168.1.100\logs\logs.ldf
\\192.168.1.200\db3\db3.ndf
\\192.168.10.200\db4\db4.ndf
```

The share type was set to be continuously available with oplocks enabled on the storage virtual machine CIFS shares. Continuously available shares are an enhanced SMB 3.0 feature that makes sure that high availability and network path failover are provided to the SQL Server application. When configured correctly with this setting and the appropriate redundant cards, switches, paths, shares, controllers, and servers, SMB 3.0 has the same high availability as iSCSI and FC. The share settings appear as follows from the storage console:

```
vserver cifs share properties add -vserver sqlsvm -share-name sqlshare -share-properties
continuously-available,oplocks
```

Again, our tested configuration evenly distributed the I/O traffic across multiple paths, resulting in optimal performance. Deploying a different SMB network configuration may result in performance that differs significantly from the results shown in this report.

4.5 Workload

As mentioned in section 2.1, "Test Workload and Metrics Measured," the OLTP transactional workload that we used for testing was a leading benchmark test suite that emulated an online stock market. We configured a database with 120,000 customers and simulated a medium to large, heavy OLTP transactional database environment. The load was applied to a full set of SQL Server database, log, and temp files. We analyzed the workload pattern and found it to be random and composed of 90% reads and 10% writes with an average block size of 8K. This workload pattern is typical of OLTP databases.

The workload test steps were consistent across protocols and test runs. To establish the performance boundaries, we tested the iSCSI environment up to a saturation point in preliminary tests. When a bottleneck was reached, the workload was lowered to apply a strong consistent workload level in which no bottlenecks were present. We found that 600 users was the appropriate concurrency level, and this became the baseline. At this load level, latency was kept to less than 5 milliseconds total at the server and storage level. We made sure that there were no bottlenecks across the server, switch, or storage systems. We tested with this load level across all protocols and reported the results in this report.

Note: We took care in these test steps to simulate a real database and customer workload, but we acknowledge that workloads will vary across databases. In addition, these test results were obtained in a closed lab environment where no other competing workloads were running on the same infrastructure. In a typical shared storage infrastructure, other workloads are sharing resources. Therefore, customers might want to deploy the databases that require the heaviest workloads on isolated cluster nodes or deploy NetApp quality of service (QoS) to restrict other workloads from affecting the SQL workload. Finally, your results might vary from those found in the report.

4.6 Test Methods

As mentioned in section 1.3, "NetApp Clustered Data ONTAP 8.3," we completed two sets of tests. In the initial set of tests, we focused on multiprotocol testing and used up to 10 SQL Servers, each connecting to a small 40GB database stored on a pair of storage controllers. The database sizes were small enough so that the data was served almost entirely out of the Flash Cache. As we expected, this workload was served with low latency, because the databases were served from the cache and disk utilization was

minimal after the caches were warmed. We found that the performance results were comparable across the iSCSI, FC, and SMB 3.0 protocols.

In the second set of tests detailed in this technical report, we focused on using a larger database size and chose 1.25TB. In this case, we used a single, medium-sized SQL Server connected to this database. The larger database size had a larger working set so that the database was served ~78% from the cache and Flash Cache and ~22% from disk. With this set of tests, we designed this environment for optimal performance by using multiple connections, files, mounts, LUNs, and so on. After we established the peak workload level, we held a consistent OLTP workload across the tests. We found that the results showed equivalent performance for SMB 3.0, iSCSI, and FC. Figure 1 through Figure 5 show these results.

We completed additional tests by changing each variable to simplify the configuration. We wanted to be certain that the test results obtained in the second set of tests with an optimized configuration would be the same for a simple configuration. We cut the size of the Flash Cache in half, to 256GB, and this prompted much of the working set to be served from disk with disk utilization averaging 90%. The performance trend remained the same, but, as we expected, latencies increased correspondingly across all protocols because disks serve data slower than Flash Cache.

For both the SAN and Ethernet environments, we reduced the number of network connections from the server to the storage controller from 2 down to 1. For SMB 3.0, when we reduced the number of 10GbE links from 2 down to 1, we saw latency increase significantly, but throughput remained the same. This is intuitive, because having 2 fast connections allows better concurrency than having 1 does. We saw the same behavior for FC when we reduced the number of 8Gb ports from 2 to 1 on the server. Again, this shows that performance improves when multiple paths are available for SAN and Ethernet environments.

Without going into all of the details, we reduced the controller count, volume, and share counts incrementally until we had the simplest configuration. This configuration consisted of 1 storage controller, 1 volume, 1 CIFS share, and 1 network connection. In all these tests, the performance trend of equivalent IOPS, latency, and CPU utilization remained the same across protocols.

5 Best Practices for Performance and Resiliency

- Use multiple 10GbE and 8Gb FC links for optimal performance. Do not use 1GbE.
- For the most IOPS-intensive OLTP databases, divide the database into multiple files striped across multiple volumes and LUNs (for block protocols).
- Mount the databases across separate connections and separate private VLANs and shares for latency-sensitive environments.
- Set the SMB 3.0 shares to be continuously available.
- Set the FC HBA ports with a queue depth of 256.
- Set Load Balancing for the Microsoft DSM to Least Queue Depth.
- Split the database into multiple files located on multiple volumes or LUNs.
- Split the `tempdb` into multiple files located on multiple volumes or LUNs.
- Because they are autotuned, no tuning is required on the server or storage controller side for the TCP receive and send windows.
- For the FC or iSCSI LUNs, use a 64KB block size when formatting the NTFS partition.
- On the NetApp storage controller, set the following option before initializing the aggregate:

```
options wafl.optimize_write_once off
```

This list is not exhaustive. For more information, refer to the relevant Microsoft and NetApp documentation.

6 Conclusion

These solution performance proof points should help customers to choose a SQL Server database network architecture. Because it offers both high-performance SAN and SMB 3.0, customers can rely on clustered Data ONTAP for Microsoft SQL Server deployments. Talk to your Microsoft and NetApp representatives about how they can help you deploy current and future SQL Server workloads in your private cloud and extend them to the public cloud.

7 References

- TR-4108: Installing SQL Server 2012 on SMB Shares on NetApp Storage
  http://www.netapp.com/us/media/tr-4108.pdf
- TR-4218: SnapDrive 7.0 for Windows SMB 3.0: Best Practices and Deployment Guide
  http://www.netapp.com/us/media/tr-4218.pdf
- TR-4353: Best Practices Guide for Microsoft SQL Server and SnapManager 7.1 for SQL Server with Clustered Data ONTAP
  http://www.netapp.com/us/media/tr-4353.pdf
- TR-4175: Microsoft Windows Server 2012 Hyper-V Storage Performance: Measuring SMB 3.0, iSCSI, and FC Protocols
  http://www.netapp.com/us/media/tr-4175.pdf
- TR-3441: Windows Multipathing Options with Data ONTAP: Fibre Channel and iSCSI
  http://www.netapp.com/us/media/tr-3441.pdf

8 Acknowledgments

Special thanks to the following people for their contributions to this study:

- Pat Sinthusan
- Glenn Sizemore
- Saad Jafri
- Scott Lane
- Marty Turner
- Justin Welch
- Manoj D V Kumar

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

NetApp®

www.netapp.com