

# Evaluation Report: Supporting Multiple Workloads with the Lenovo S3200 Storage Array

*Evaluation report prepared under contract with Lenovo*

---

## Executive Summary

Virtualization is a key strategy to reduce the capital expense of enterprise hardware. Running multiple applications maximizes the return on investment of servers and storage. Unfortunately, just piling more I/O on to devices such as traditional storage arrays has drawbacks. Hard disk drive (HDD) only arrays have trouble supporting simultaneous diverse workloads. Differing I/O patterns cause conflict with regard to the placement of data, mechanical motion of drive arms, and on-drive cache usage. A traditional way to manage this is by-significantly over-provisioning the number of drives to keep data on the outer cylinders of magnetic platters. This wastes a good deal of the drive space to keep IOPS and bandwidth up, whereas a single solid state drive can deliver the IOPS and bandwidth of an entire shelf or more of HDDs.

The Lenovo S3200 flexible hybrid array is ideal for small and medium businesses seeking to cut costs by consolidating applications onto fewer storage devices. Lenovo offers upgrade options to the S3200 by allowing customers to convert a basic all-HDD array into a hybrid array by replacing some of the drives with flash to accelerate performance instead of deploying expansion shelves. With the addition of a customer-defined number of solid state drives (SSDs), the array can be configured to accelerate read performance with an SSD read cache, or both read and write performance with real-time data tiering.

Demartek evaluated the S3200 supporting a multiple-application workload in a VMware vSphere environment. Running common applications like Microsoft Exchange Server, SQL Server, SharePoint, a web server, and an SMB fileserver simultaneously, we compared the performance of an array with only HDD storage to that with an SSD read cache and with SSD tiering enabled.

We saw storage performance improve for both acceleration technologies. The maximum being a 2.4X bandwidth increase overall (which contributed to a 7X increase for an SQL Server application, specifically). Storage latencies decreased by up to 70%. This contributed to a dramatic 5X reduction in the database application I/O response times with no adverse effect on other applications.

## The Lenovo S3200

Lenovo's S3200 Storage Array (Figure 1) is a dual controller, 2 rack unit storage array supporting either 24 small form factor (SSF) drives or twelve 3.5 inch drives. The S3200 array supports 8 Gb and 16 Gb Fibre Channel as well as 1 Gb and 10 Gb iSCSI. Each array can support seven expansion units for a maximum of 768 TB of total storage capacity.



**Figure 1 - Lenovo S3200**

The base configuration of HDDs is upgradable by replacing some HDDs with flash storage. Lenovo SAN Manager handles the addition of SSDs to an array by enabling the creation of an SSD read cache or an SSD performance tier with Lenovo Intelligent Real Time Tiering, where active data is migrated from HDD to SSD every five seconds.

This evaluation was performed with an S3200 array provisioned with 20 SSF 900 GB 10K RPM HDDs. Four 400 GB SSDs were supplied for read caching and performance tiering upgrades. With this hardware, we executed a 20 HDD baseline test case in which 10 drives were assigned to each controller in a RAID 6 configuration. To evaluate read caching, we added a single SSD to each controller, configured as single drive read cache disk groups. For measuring the effect of SSD tiering, two SSDs in a RAID 1 disk group were added to each controller.

Three volumes were created from these drives and added to vSphere as datastores. The first was used for all virtual machine files. The other two, much larger volumes, were dedicated application datastores. Application virtual drives were provisioned evenly from disk space managed by each controller to best spread the I/O load across the array. Appendix B diagrams the array configuration and volume mapping to vSphere datastores.

## The Mixed Workload

Virtualization is a common, and arguably preferred, method of consolidating multiple applications onto a shared hardware platform. Therefore, we evaluated an S3200 array in a vSphere environment with several application servers running in virtual machines (VMs). We imagined a small to medium sized business of about 150 employees and created several VMs to support common applications that such a business might deploy.

Five application VMs and two infrastructure VMs shared the same S3200 array for virtual machine files and application data. Setup details for the virtual machines are described in Appendix A.

The test workloads were intentionally designed to model what seemed to be a reasonable use case for the business we envisioned. This included some number of the users exercising different applications at the same time. As with real businesses, only a small number of users are likely to be accessing any one IT resource at a given time. We made some educated guesses as to what those user counts could potentially be.

The workload included VMs hosting Microsoft Exchange, SQL Server (executing a transactional workload to drive I/O load), and SharePoint, along with a static content Webserver and an SMB fileserver. We chose to use Microsoft Exchange Jetstress to model Exchange as Jetstress validates storage systems by setting hard limits on the amount of latency database operations may experience. For this reason, we designed the workload intensity such that Jetstress' validation criteria were met by all array configurations, regardless of the performance of the other applications.

With Jetstress, in essence, defining the SLA for the consolidated application environment, the goal was to evaluate the benefit to the other applications by upgrading the array with flash. The assumption was that by adding SSD read caching or tiering, the Lenovo S3200 would accelerate the performance of one of more applications while maintaining or improving the service level of the shared environment as a whole.

Detailed descriptions of the workloads and user counts are available in appendix C.

## Performance Metrics

Key metrics for storage system performance analysis are I/Os per second (IOPS), bandwidth, and latency or response time. These metrics are defined as follows:

- ◆ **IOPS** – I/Os per second – a measure of the total I/O operations (reads and writes) issued by the application servers.
- ◆ **Bandwidth** – a measure of the data transfer rate, or I/O throughput, measured in MegaBytes per second (MBPS).
- ◆ **Latency** – a measure of the time taken to complete an I/O request, also known as response time. This is frequently measured in milliseconds (one thousandth of a

second). Latency is introduced into the SAN at many points, including the server and HBA, SAN switching, and at the storage target(s) and media.

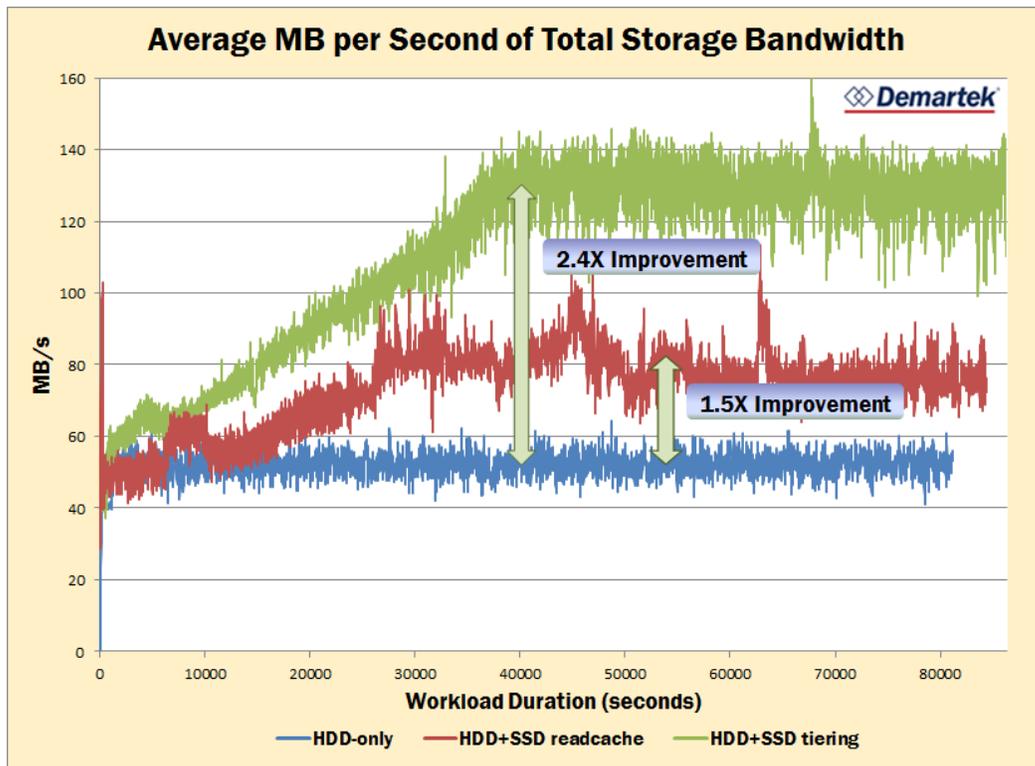
It is important to consider all three metrics when evaluating the performance of storage systems because all three contribute to how the storage will support an application. IOPS drive bandwidth. The number of IOPS times the I/O request size determines the amount of bandwidth delivered.

Latency can have a very significant effect on application performance and user experience. Transactional applications can be quite sensitive. For instance, Microsoft Exchange Jetstress sets an upper limit of 20 milliseconds for database reads and writes to certify any storage system. Unlike IOPS and bandwidth, where more is better, with latency the goal is to keep it as low as possible.

Performance metrics were gathered at the hypervisor, with the vSphere esxtop utility, and within the VMs with Windows Perfmon and Linux systats.

## Results and Analysis

All businesses will place unique demands on storage systems. The workload designed for this evaluation and these results are an example of one possible scenario.



**Figure 2 - Total Bandwidth**

This is a punishing workload to run on a single shelf HDD array. The multiple I/O patterns cause competition for storage resources, such as disk drive head motion, cache space, and controller performance. Without flash enhancement, the HDD-only array was able to deliver about 52 MB/s of total bandwidth (Figure 2), which included all VM files and vSphere infrastructure, as well as all application data.

Array performance contributes to the overall user and application experience. Obviously, low I/O latency at the array will be passed along on to the hypervisor and ultimately the guest VMs. At first glance, latency looks pretty good for an all-HDD array, at an average of 7.1 milliseconds when measured at the hypervisor layer (Figure 3).

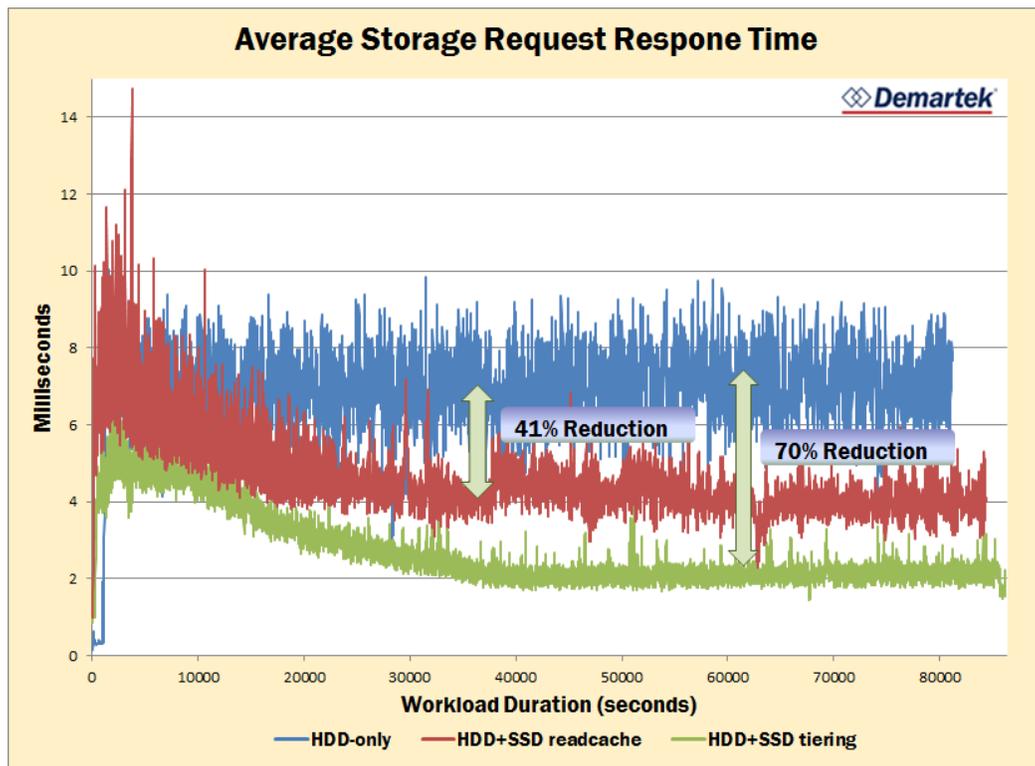
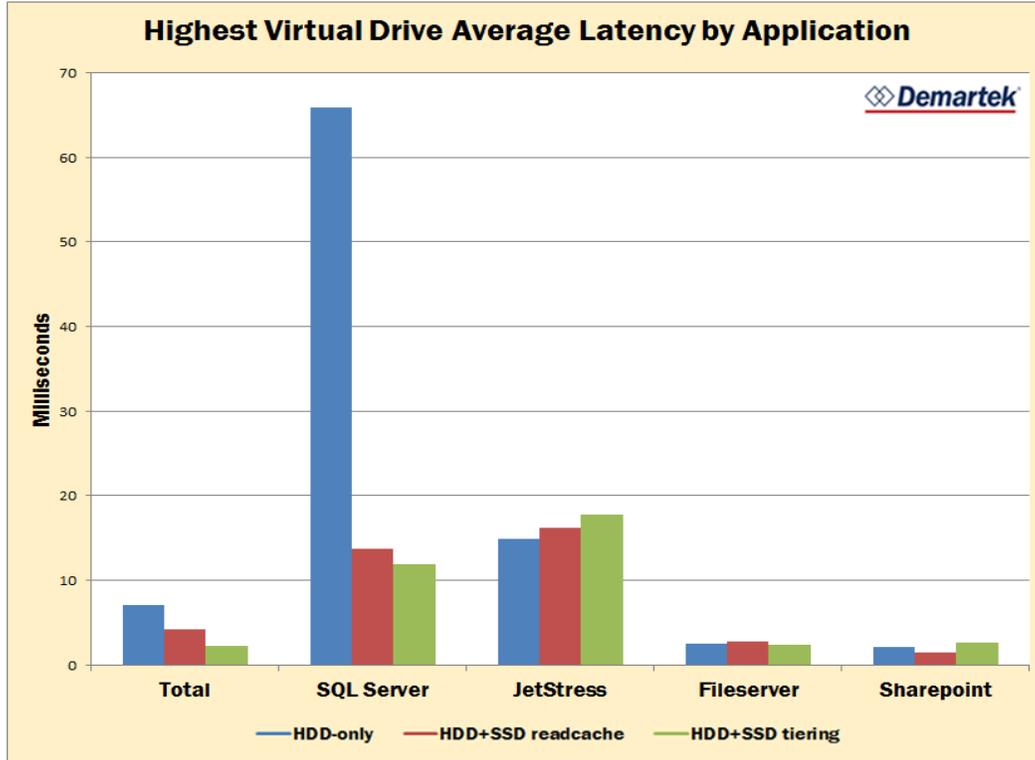


Figure 3 - Storage system response times

However, overhead in the hypervisor and guest operating systems contribute to latency increases experienced by virtual drives within the guest OS. To develop a clearer picture of the application experience, we measured response times of the virtual drives within the VMs as well. These response times were always higher than the raw storage device latency.

Our first observation was that the webserver was very quickly cached in memory and little I/O went to the array, so we focused our attention on the remaining four applications. The highest average database transactional latency experienced by Microsoft Jetstress was 17.7 milliseconds (Figure 4), within the threshold of acceptability set by Jetstress. Therefore, every configuration, with and without SSD acceleration, met our initial criteria for storage validation. Without the benefit of flash acceleration, the SQL server application saw very high virtual storage latencies. This is cause for concern and, if nothing else, should deter our hypothetical company from any further consolidation onto the array. Depending on business impacts, they may be advised to relocate the SQL Server application to another device. However, this situation changed with the addition of SSDs to the array.



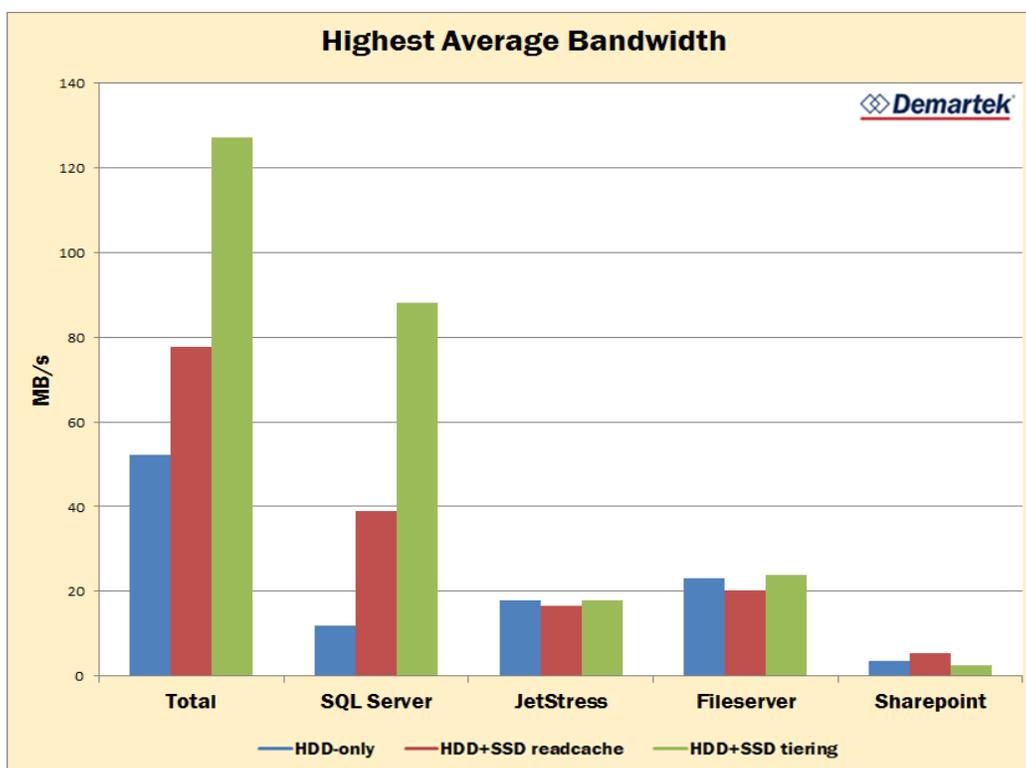
**Figure 4 - Virtual drive latencies**

SQL Server clearly gained the benefit when either flash solution was implemented. Latency decreased by roughly a factor of 5 in both cases. The bigger question is a matter of why this one application received such a boost while the others remained at more-or-less steady state conditions. The OLTP workload, which was used to drive SQL Server I/O, includes a large proportion of random reads, which are very flash-friendly. It seems likely that this I/O was quite compatible with the caching and tiering algorithms employed by the S3200 array. The very large delta between no flash and SSD acceleration also points to the SQL server workload being constrained by the amount I/O the HDD-only configuration could support. The other workloads may have been satisfied with lesser I/O and bandwidth, minimizing I/O request queuing. When we examine the application bandwidth data, this theory will receive additional support.

Curiously, Jetstress, the fileserver, and SharePoint saw latency slightly increase when the array was upgraded with SSDs, in one or both of the configurations. We suspect this is related to SQL Server's strong performance boost. As the database workload runs faster, its transactions require more data. With limited SSD space, some portion of this data must come from the HDDs, which drives up the number of I/O requests to those devices. Since

the HDDs cannot serve any additional I/O over the baseline, queuing occurs. As more I/Os are queued on the HDDs, latency goes up. The impact may be very small on the raw devices, but the added overhead of the hypervisor and guest OS magnify it as seen here.

The same behavior is apparent when we considered the storage bandwidth. Once we enhanced the S3200 array with SSDs and configured either read caching or tiering, the combined bandwidth of entire workload went up. SSD tiering saw the most improvement, 2.4 times the HDD-only baseline, to nearly 130 MB/s while response times dropped by 70% to roughly 2 milliseconds (Figures 2 and 3). Read caching saw more modest gains, but still provided measurable bandwidth and response time improvements.



**Figure 5- Application bandwidths**

As with latency, the SQL Server workload again saw the greatest benefit (Figure 5), with application bandwidth increasing from 12 MB/s to 39 MB/s to 88MB/s with read caching and tiering respectively. This allowed 196 database transactions per second with only HDDs, 409 transactions per second through read caching, and 900 transactions per second with tiering. This very significant improvement, while the other applications remained steady, also points to the HDD-only configuration imposing I/O limits on the SQL Server

application that were not experienced by the others. The Jetstress, Sharepoint, and filesaver applications appear to have been satisfied by the array without additional acceleration, leaving SQL Server to reap the benefits of the flash upgrades.

## Summary and Conclusion

Consolidation of workloads into a virtualized environment can drive up the return on investment for computing hardware, but it needs to be approached with a strong understanding of the workload I/O demands. Different businesses will have different requirements, which may change over time. A flexible solution is needed to address this. Lenovo's S3200 offers three solutions—all-HDD, SSD read caching, and SSD performance tiering—in one storage array, and the amount of flash to deploy is the customer's prerogative.

So long as the business SLA expectations are met, a solution can generally be considered appropriate for the use case. Using our workload as an example, we declared upfront that Exchange Jetstress thresholds had to be met for the storage to be validated. This condition was satisfied in all three configurations. However, combining applications that compete for storage resources can be punishing to a drive array and we saw that our database was suffering from high I/O latencies as a result. When we enabled flash acceleration, database storage bandwidths improved dramatically, by a factor of 3.5 when read caching and a factor of more than 7 with SSD tiering. While the database application was experiencing such significant improvement, the remaining application workloads carried on with no performance degradation.

SSD performance tiering is clearly the stronger solution for our test case. A business with a different set of applications might find read caching a better choice if their I/O needs included a different read profile with potential "hotspots" or frequently accessed data. The great strength of the S3200 array is that it is easily configured for either option, and the number of SSDs can be chosen to fit a business' specific needs.

There is also a cost savings to consider. An all HDD solution would have required 48 drives to duplicate the performance of 20 HDDs and four SSDs configured for tiering.<sup>1</sup> That is another entire drive shelf, with all the associated costs for purchase, deployment, and support.

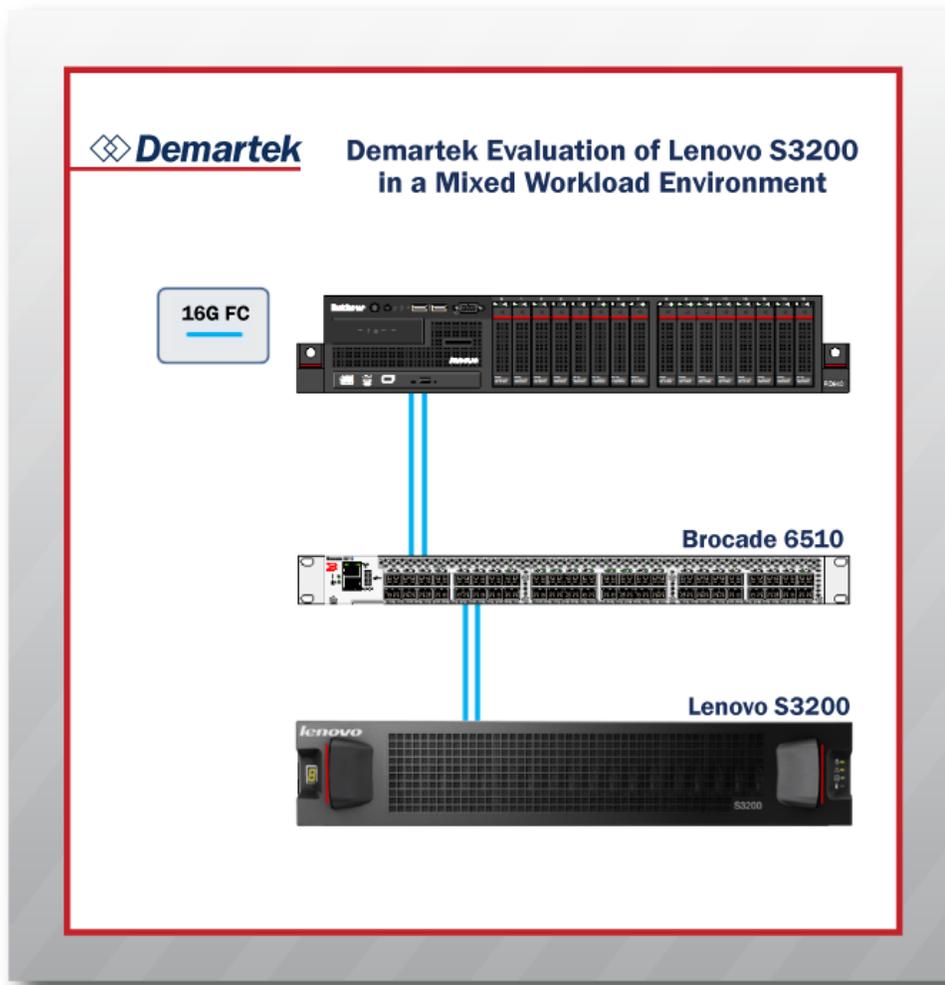
A successful business is not likely to see application usage go down over time and wise IT management will consider anticipated storage growth and performance needs when

---

<sup>1</sup> With 20 HDDs and four rather small SSDs configured for tiering, aggregate performance increased 2.4 times (bandwidth and IOPS). If the relationship of spindles to performance remains linear, another 28 HDDs would be needed to achieve the same performance values as with SSD tiering. (i.e., 20 HDDs x 2.4 = 48 HDDs)

planning purchases and upgrades. The Lenovo S3200 provides the benefits of flash storage, to support current needs and to future-proof the enterprise, without the expense of an all-flash array or the inconvenience and waste of short-stroking multiple HDD shelves to achieve the performance demanded.

## Appendix A – Test Description and Environment



**Figure 6 – Test Environment**

- ◆ Dual processor rack server
- ◆ 2 Intel E5-5680 3.33GHz CPUs
- ◆ 144 GB RAM
- ◆ 1 16Gb FC dual port HBA
- ◆ 1 Intel x540-AT2 10Gb NIC
- ◆ vSphere 5.5
- ◆ Microsoft Exchange Jetstress 2013, Microsoft Exchange 2013 ESE files

### VMs

- ◆ Active Directory

- Windows 2012 R2, 8GB RAM, 4 cores
- ◆ Exchange Jetstress
  - Windows 2012 R2, 8GB RAM, 8 cores, MS Exchange Jetstress 2013
- ◆ SQL Server
  - Windows 2012 R2, 16GB RAM, 8 cores, SQL Server 2012 SP2, MS Benchcraft
- ◆ SMB Fileserver
  - Windows 2012 R2, 6GB RAM, 2 cores
- ◆ SharePoint
  - Windows 2012 R2, 8GB RAM 8 cores, SharePoint 2013
- ◆ Webserver
  - Ubuntu 14.0.1, 2GB RAM, 2 cores, NGINX Webserver
- ◆ vCenter Server
  - Windows 2008 R2, 8GB RAM, 4 cores

### **Fibre Channel Switch**

- ◆ Brocade 6510 16Gb Fibre Channel Switch

### **Storage Array**

- ◆ Lenovo S3200 array
- ◆ Lenovo SAN Manager Storage Operating System
- ◆ 20 900GB 10k RPM 6Gb SAS HDD
  - ◇ 10 drive RAID 6 per storage controller
  - ◇ 550 GB volume for VM files datastore on controller 1
  - ◇ 5000 GB volume for application datastore on controller 1
  - ◇ 5200 GB volume for application datastore on controller 2
- ◆ 4 400GB SSD
  - ◇ 1 drive per storage controller for read caching
  - ◇ 2 drives RAID 1 per storage controller for SSD tiering
- ◆ 4 16Gb FC target ports per controller (1 port active per controller)

## Appendix B – Array Volume Configuration

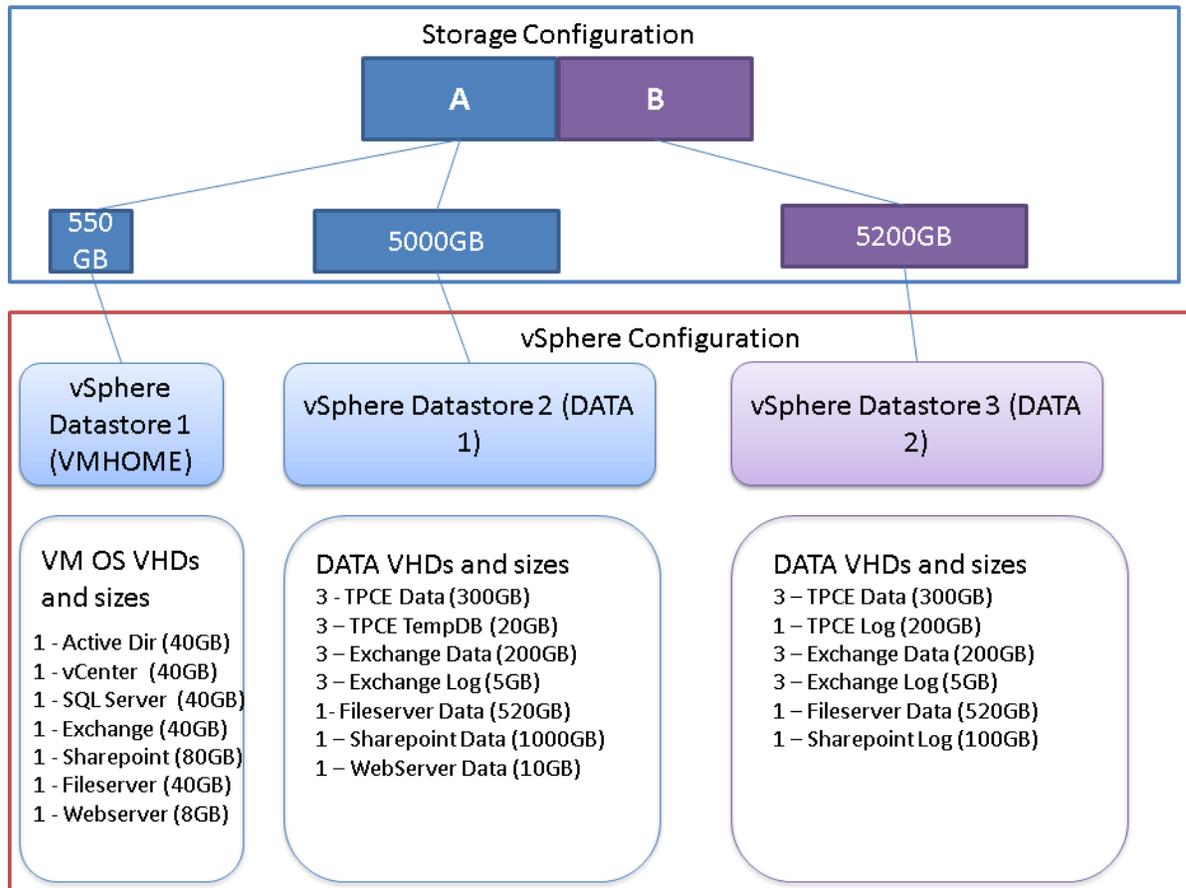


Figure 7 - Array and vSphere volume and datastore configuration

## Appendix C – Workload definitions

### Exchange Workload

We installed Microsoft Exchange Jetstress 2013 to simulate an Exchange server. While not a true mail server, Jetstress models the I/O loading and patterns that would be generated by an Exchange 2013 server, including email database reads and writes as well as logging. Its configuration includes the number and size of mailboxes, number of mailbox databases and logs, and intensity of use, among other things. Jetstress uses the same Extensible Storage Engine files as Microsoft Exchange to ensure that Jetstress modelled performance is representative of a bona fide Exchange Server of the same version. We configured 250 mailboxes and 6 mailbox databases with a single replica each. Every mailbox was allowed up to 2 GB of space. The workload was configured to use all the available drive space equally, which can be cache unfriendly.

Jetstress was used as the primary benchmark by which we determined whether the storage was able to support our multiple workload environment. If Jetstress could not certify the storage for an average mail server load (.2 IOPS per mailbox for Exchange 2013) we would declare the multiple workloads to be too much for the storage array. Jetstress certifies that an environment is able to support a configuration when average database latencies are below 20 milliseconds.

### SQL Workload

SQL Server 2012 SP2 was installed to support a transactional database workload. This workload performs real transactions that might be executed by database application users as well as background transactions from automated processes. The On-Line Transaction Processing (OLTP) workload models a financial brokerage firm with customers who generate transactions related to trades, account inquiries, and market research. The brokerage firm in turn interacts with financial markets to execute orders on behalf of the customers and updates relevant account information. This workload consists of a mixture of mostly reads with some writes to its database.

The database was built with 1TB of data and logs. For our hypothetical business, we assumed that 10 users might be simultaneously accessing this application at any one time. It was not our intention to benchmark a database or database system. The data gathered by this testing is significant to this evaluation only and cannot be compared with any published database benchmarks.

## Microsoft SharePoint Workload

Microsoft SharePoint is a common workspace collaboration tool, accessed through a browser from a variety of client devices. We assumed a sixth of our hypothetical company, 25 users, would be simultaneously using SharePoint. With Neotys NeoLoad web performance testing tools, we created a workload that included general navigation of the collaboration space plus uploads and downloads of files. Many of the users were assumed to be accessing and updating related documents, as project teams would do in real-life.

## Web Workload

Using the NGINX open source webserver software and an Ubuntu server virtual machine, we deployed a read-only webserver. Twenty five client device connections were maintained at all times, hitting web pages in a semi-random fashion, using the Neotys NeoLoad web performance testing tools to drive the demand.

## Fileserver workload

A basic Windows SMB network share of 1TB was used to create a network fileshare. A server outside of the test environment executed an Iometer script against the network drive. 12 workers ran Iometer access specifications performing 4KB and 8KB random and sequential I/O. A 10 Gb/s Ethernet connection was deployed between the client and fileserver VM. We simulated “hotspots” where several users would perform the same I/O across the same limited portion of the virtual drive, mimicking how certain files tend to be accessed more regularly than others in normal operations.

## Appendix D – Fileserver Iometer test file

```

Version 1.1.0
'TEST SETUP
=====
Test Description

Run Time
'      hours  minutes  seconds
      24     10     0
Ramp Up Time (s)
0
'Default Disk Workers to Spawn
NUMBER_OF_CPUS
'Default Network Workers to Spawn
0
Record Results
ALL
'Worker Cycling
'      start  step  step type
      1      1    LINEAR
'Disk Cycling
'      start  step  step type
      1      1    LINEAR
'Queue Depth Cycling
'      start  end  step  step type
      1     32   2    EXPONENTIAL
Test Type
NORMAL
'END test setup
'RESULTS DISPLAY
=====
Record Last Update Results,Update Frequency,Update Type
DISABLED,1,WHOLE_TEST
'Bar chart 1 statistic
Total I/Os per Second
'Bar chart 2 statistic
Total MBs per Second (Decimal)
'Bar chart 3 statistic
Average I/O Response Time (ms)
'Bar chart 4 statistic
Maximum I/O Response Time (ms)
'Bar chart 5 statistic
% CPU Utilization (total)
'Bar chart 6 statistic
Total Error Count
'END results display
'ACCESS SPECIFICATIONS
=====
'Access specification name,default assignment
4 KiB; 100% Read; 25% random,NONE
'size,% of size,% reads,% random,delay,burst,align,reply
4096,100,100,25,0,1,0,0
'Access specification name,default assignment
4 KiB; 10% Read; 0% random,NONE
'size,% of size,% reads,% random,delay,burst,align,reply
4096,100,10,0,0,1,0,0
'Access specification name,default assignment
8 KiB; mixed; hotspots,NONE
'size,% of size,% reads,% random,delay,burst,align,reply
8192,10,100,10,0,1,0,0

```

8192,10,0,10,0,1,0,0  
8192,20,10,0,0,1,0,0  
8192,60,100,10,0,1,0,0

'END access specifications  
'MANAGER LIST

=====  
'Manager ID, manager name  
1,WIN-08GAJTULK4J  
'Manager network address

'Worker

Worker 1

'Worker type  
DISK

'Default target settings for worker

'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value  
1,DISABLED,1,DISABLED,0

'Disk maximum size,starting sector,Data pattern  
0,0,1

'End default target settings for worker

'Assigned access specs  
4 KiB; 100% Read; 25% random

'End assigned access specs

'Target assignments

'Target  
F:\\10.0.9.104\

'Target type  
DISK

'End target

'End target assignments

'End worker

'Worker

Worker 2

'Worker type  
DISK

'Default target settings for worker

'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value  
1,DISABLED,1,DISABLED,0

'Disk maximum size,starting sector,Data pattern  
0,0,1

'End default target settings for worker

'Assigned access specs  
4 KiB; 100% Read; 25% random

'End assigned access specs

'Target assignments

'Target  
F:\\10.0.9.104\

'Target type  
DISK

'End target

'End target assignments

'End worker

'Worker

Worker 3

'Worker type  
DISK

'Default target settings for worker

'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value  
1,DISABLED,1,DISABLED,0

'Disk maximum size,starting sector,Data pattern  
0,0,1

```
'End default target settings for worker
'Assigned access specs
    4 KiB; 100% Read; 25% random
'End assigned access specs
'Target assignments
'Target
    F:\\10.0.9.104\
'Target type
    DISK
'End target
'End target assignments
'End worker
'Worker
    Worker 4
'Worker type
    DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
    1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
    0,0,1
'End default target settings for worker
'Assigned access specs
    4 KiB; 100% Read; 25% random
'End assigned access specs
'Target assignments
'Target
    F:\\10.0.9.104\
'Target type
    DISK
'End target
'End target assignments
'End worker
'Worker
    Worker 5
'Worker type
    DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
    1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
    0,0,1
'End default target settings for worker
'Assigned access specs
    4 KiB; 10% Read; 0% random
'End assigned access specs
'Target assignments
'Target
    F:\\10.0.9.104\
'Target type
    DISK
'End target
'End target assignments
'End worker
'Worker
    Worker 6
'Worker type
    DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
    1,DISABLED,1,DISABLED,0
```

```
'Disk maximum size,starting sector,Data pattern
    0,0,1
'End default target settings for worker
'Assigned access specs
    4 KiB; 10% Read; 0% random
'End assigned access specs
'Target assignments
'Target
    F:\\10.0.9.104\
'Target type
    DISK
'End target
'End target assignments
'End worker
'Worker
    Worker 7
'Worker type
    DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
    1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
    0,0,1
'End default target settings for worker
'Assigned access specs
    4 KiB; 10% Read; 0% random
'End assigned access specs
'Target assignments
'Target
    F:\\10.0.9.104\
'Target type
    DISK
'End target
'End target assignments
'End worker
'Worker
    Worker 8
'Worker type
    DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
    1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
    0,0,1
'End default target settings for worker
'Assigned access specs
    4 KiB; 10% Read; 0% random
'End assigned access specs
'Target assignments
'Target
    F:\\10.0.9.104\
'Target type
    DISK
'End target
'End target assignments
'End worker
'Worker
    Worker 9
'Worker type
    DISK
'Default target settings for worker
```

```
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
  1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
  10737418240,0,1
'End default target settings for worker
'Assigned access specs
  8 KiB; mixed; hotspots
'End assigned access specs
'Target assignments
'Target
  F:\\10.0.9.104\
'Target type
  DISK
'End target
'End target assignments
'End worker
'Worker
  Worker 10
'Worker type
  DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
  1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
  10737418240,0,1
'End default target settings for worker
'Assigned access specs
  8 KiB; mixed; hotspots
'End assigned access specs
'Target assignments
'Target
  F:\\10.0.9.104\
'Target type
  DISK
'End target
'End target assignments
'End worker
'Worker
  Worker 11
'Worker type
  DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
  1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
  10737418240,0,1
'End default target settings for worker
'Assigned access specs
  8 KiB; mixed; hotspots
'End assigned access specs
'Target assignments
'Target
  F:\\10.0.9.104\
'Target type
  DISK
'End target
'End target assignments
'End worker
'Worker
  Worker 12
'Worker type
```

```
DISK
'Default target settings for worker
'Number of outstanding IOs,test connection rate,transactions per connection,use fixed seed,fixed seed value
    1,DISABLED,1,DISABLED,0
'Disk maximum size,starting sector,Data pattern
    10737418240,0,0
'End default target settings for worker
'Assigned access specs
    8 KiB; mixed; hotspots
'End assigned access specs
'Target assignments
'Target
    F:\\10.0.9.104\
'Target type
    DISK
'End target
'End target assignments
'End worker
'End manager
'END manager list
Version 1.1.0
```

-----

The original version of this document is available at:  
[http://www.demartek.com/Demartek\\_Lenovo\\_S3200\\_Evaluation\\_2016-01.html](http://www.demartek.com/Demartek_Lenovo_S3200_Evaluation_2016-01.html) on the Demartek website.

Intel and Xeon are registered trademarks of Intel Corporation.

Lenovo and S3200 are trademarks or registered trademarks of Lenovo Systems.

Neotys and NeoLoad are registered trademarks of Neotys Corporation

Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Demartek is a trademark of Demartek, LLC.

All other trademarks are the property of their respective owners.