Evaluation of PERC RAID Controller Availability and Performance

Evaluation report prepared under contract with Dell and LSI

Introduction

High availability (HA) configurations are becoming increasingly important, and Dell and LSI are recommending an HA baseline configuration using dual RAID controllers on a split, mirrored backplane. All mirroring in this configuration would be done with software without assistance from the controllers. The PowerEdge Raid Controller (PERC), built on the LSI RAID-on-Chip (ROC) technology, should provide a good underpinning for an HA hypervisor system running VMs and databases on the mirrored drives.

LSI and Dell commissioned Demartek to test dual PERC H710P controllers with a split backplane R820 Dell server, validating system availability when drives or controllers are removed, comparing performance parameters for simple I/O stress tests using one or both raid controllers, and validating performance parameters with two real-world workloads run in VMs.

Executive Summary and Key Findings

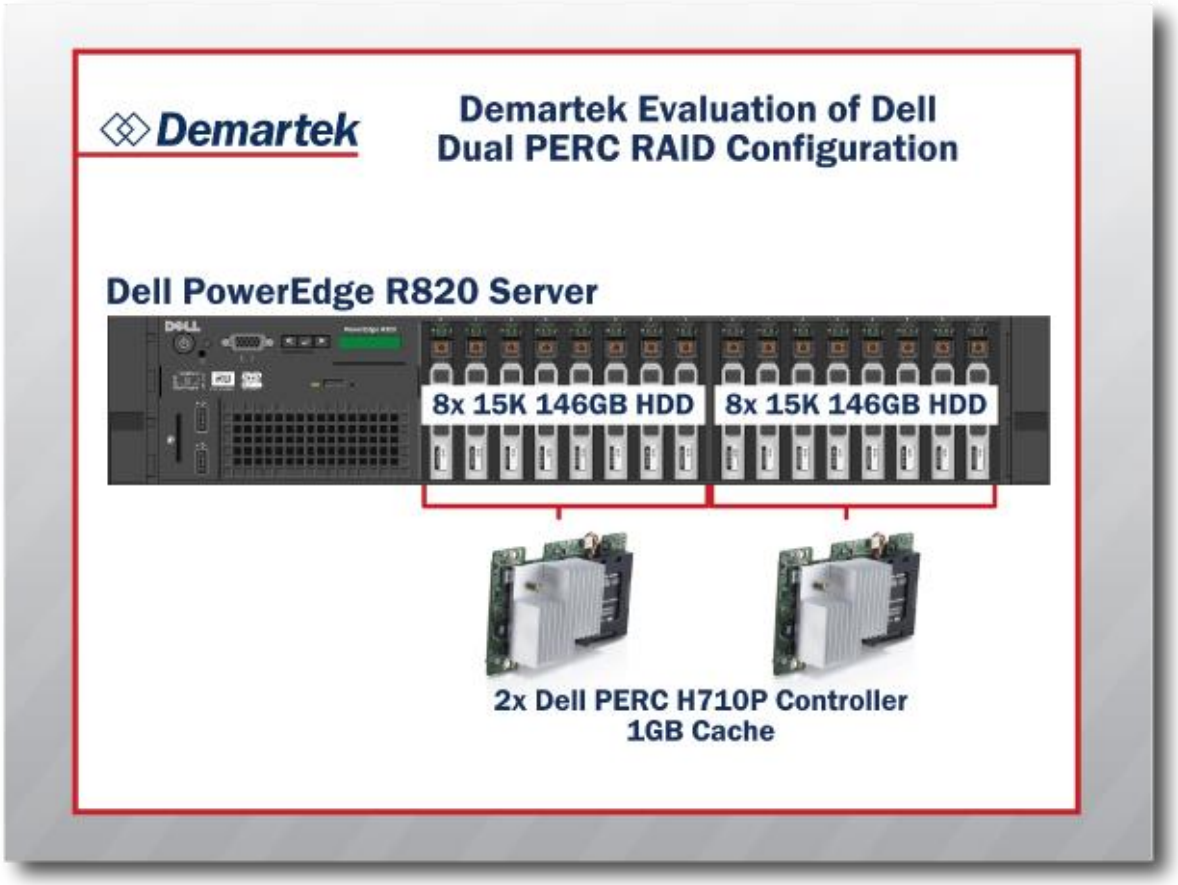
- Mdadm was used to mirror the backplanes. Using MD The system remained available when any drives were removed from one backplane, including operating system drives.
- The IOPS and throughput was up to 96% higher for the mirrored dual backplane configuration for Random Reads and 64% higher for Sequential Reads than it was without mirroring.
- The IOPS and throughput for Random and Sequential Writes differed by 0.2% or less between the mirrored and non-mirrored configurations.
- The highest IOPS, 4,181, and highest throughput, 1,045 MB/s, were reached in the sequential write tests.
- Latency for Random Reads was up to 36% lower for the mirrored dual backplane.
- The VirtIO interface was used for the VM VHDs, enabling real-world workloads to perform reliably.

We believe that this high-availability configuration is reliable and can offer increased performance as compared to a non-mirrored configuration. We recommend this high availability mirrored backplane for use with VMs running OLTP and Exchange server applications.

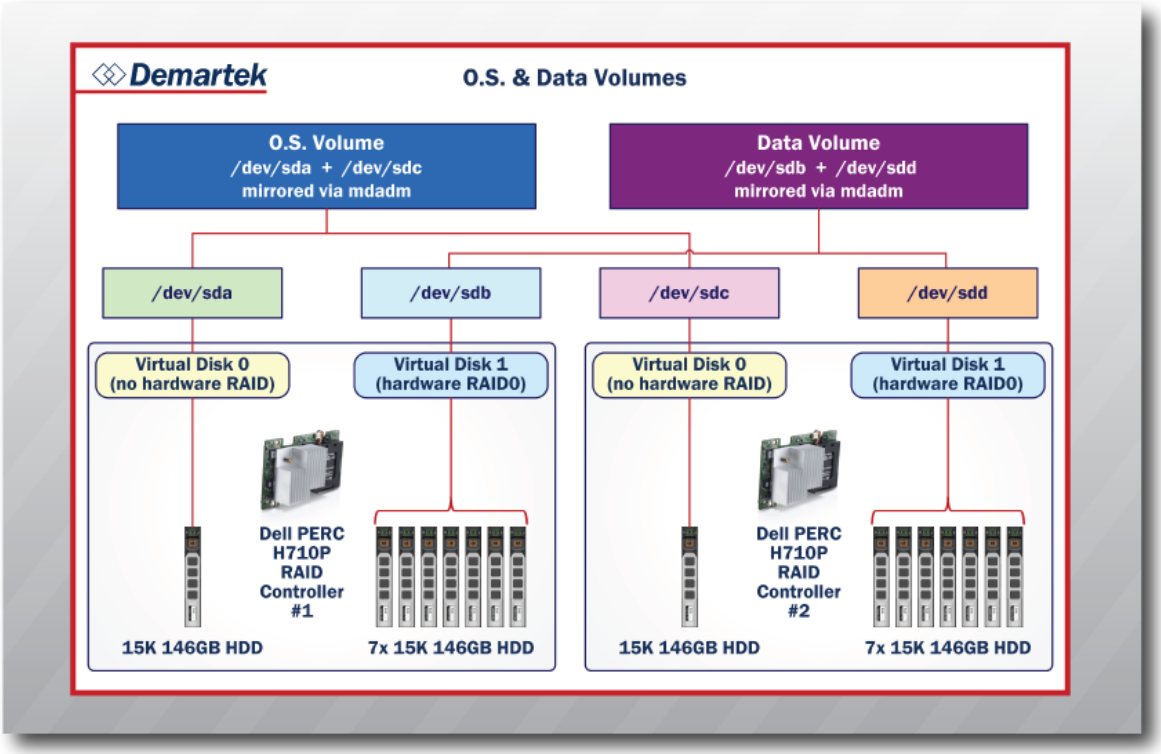
Dell H710P PERC with LSI ROC technology

The Dell H710P Power Edge RAID Controller (PERC) is built with LSI SA2208 dual-core PowerPC RAID-on-Chip (ROC) technology. This provides two processor cores to offload data processing and provide accelerated performance. In addition 1GB of DDR3 cache memory is provided to accelerate performance.

Test Configuration



16 Hard Disk Drives (HDD)s were configured on a dual backplane system with 8 HDDs per back plane. Each backplane was connected to a PERC H710P controller. Each PERC H710P had 1024MB cache memory on it and controlled 8 drives. Each hardware RAID controller was configured to provide two virtual disks. Virtual Disk 0 had only one HDD, from slot 0, and Virtual Disk 1 had 7 HDDs in a RAID 0 configuration.



The Operating System was RedHat Enterprise Linux (RHEL) 6.4. Two virtual drives, Virtual Disk 0 and Virtual Disk 1, were passed to the OS from each RAID controller, giving a total of 4 drives. The Multiple Device Administration utility (mdadm) was used to mirror the drives. Identical partitions were created on each virtual disk 0. Each partition was mirrored using mdadm to create the necessary OS mountpoints as RAID 1 devices. Each Virtual Disk 1 was a RAID0 device already. They were mirrored using mdadm to create the data mountpoint as a RAID 10 device.

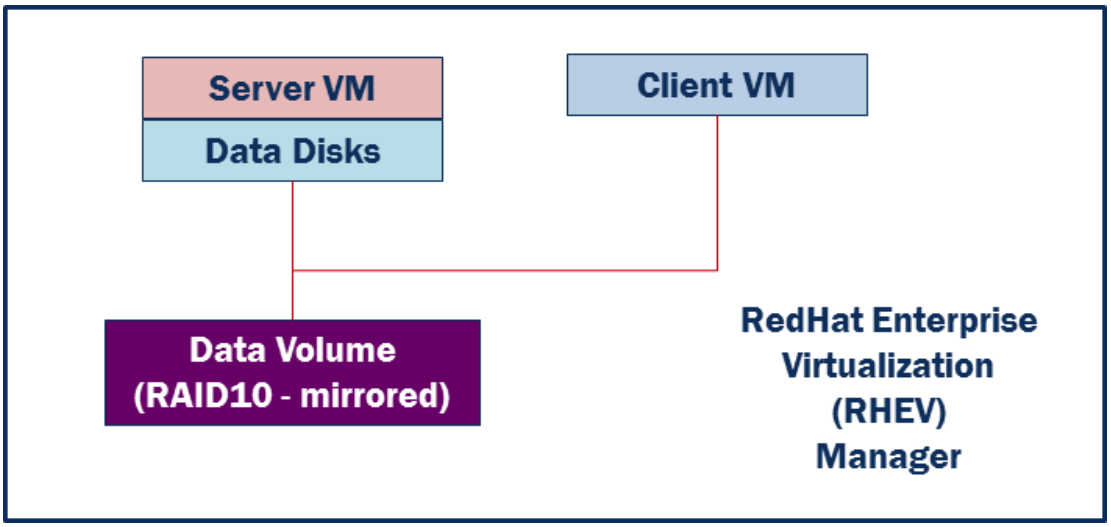
Availability Test Configuration

For the availability tests, individual drives were monitored with IOStat. Any necessary I/O was generated using vdbench filesystem tests. Filesystem tests enable VDbench to be used against an OS without writing over the filesystem, which was important when testing availability when OS drives were removed.

Stress Test Configuration

For the stress tests, individual drives and mirrored devices were monitored with IOStat. I/O was generated using vdbench on the raw drives. These tests were done against the RAID 10 data device only so as to not interfere with the operating system.

Real-world Test Configuration



For the real-world tests, Red Hat Enterprise Virtualization (RHEV) 3.0 was installed on the server, enabling an RHEV Manager (RHEVM) to add the machine as a host. The mirrored Data drives were provided to RHEVM as a Local Storage Domain. Two virtual machines running Windows Server 2012 were created on this Local Storage Domain and were provided storage from this same domain on which to build their databases. One database server virtual machine was given 24 processors, 180GB memory, and was configured to access the databases. One client virtual machine was given 3 processors, 60GB memory, and was configured as a client server. Both machines used the VirtIO interface on pre-allocated virtual hard drives.

It is interesting to note that the VirtIO interface requires a virtual floppy disk containing drivers to be loaded into the RHEVM ISO repository and subsequently provided to the VM during the first run using the "run once" option for OS installation. Without this the Windows 2012 Server installation was not able to recognize the drives. The IDE interface option is available for use and does not require a driver disk. This interface was tested, and found to offer substandard performance in addition to failing when drives are removed from a powered down VM. The IDE interface is not recommended.

Software RAID using mdadm

The software RAID functions were handled by the Linux utility ***mdadm***, which is short for multiple device administration. The ***mdadm*** utility is one of several tools found in the ***raidtools*** package that provides software RAID functions beginning with Linux 2.2

kernels. The Linux software RAID devices are implemented using the md (multiple devices) device driver.

The utility ***mdadm*** is a fairly comprehensive utility for managing software-RAID storage arrays composed of two or more physical disk devices. It can create RAID 0, 1, 4, 5, 6 and 10 disk groups, and has a variety of sub-commands to create and manage these arrays. It can add and remove devices to or from an array, and grow or shrink the volumes stored on those arrays. Details such as chunk size, the number of spare devices to include in the array and other details can be managed. The utility ***mdadm*** can have containers with multiple arrays under its control, very similar to a hardware RAID controller, and is aware of multi-path I/O.

During our testing, we found that ***mdadm*** was better in terms of performance, reliability and recoverability than using ***lvm***, especially for mirroring the boot volume.

In our tests, we used ***mdadm*** to create a mirrored (RAID1) boot volume and mirrored (RAID1) data volume. These volumes were presented to the operating system using two Dell PERC H710P RAID controllers, as described in the diagrams above.

Availability Tests

Two availability scenarios were tested that simulate various failure conditions:

1. Simulate unplugging two drives from one RAID controller disk group using MegaCli64 PDOffline and observe the system behavior. Repeat the test using other RAID controller. The system is expected to remain online and available.
    a. Perform test with no I/O running through the controllers.
    b. Perform test with I/O running through the controllers. Running I/O provided by VDBench filesystem tests.
2. Simulate unplugging the OS drive from one RAID controller disk group using MegaCli64 PDOffline and observe the system behavior. Repeat the test using the other RAID controller. The system is expected to remain online and available.
    a. Perform test with no I/O running through the controllers.
    b. Perform test with I/O running throught he controllers. Running I/O provided by VDBench filesystem tests.
3. Simulate Hot-unplugging one PCIe 3.0 PERC RAID controller using "echo 1 > /sys/bus/pci/devices/{bus}/remove" and observe the system behavior. Repeat the test using other RAID controller. The system is expected to remain online and available.
    a. Perform test with no I/O running through the controllers.
    b. Perform test with I/O running through the controllers. Running I/O provided by VDBench.

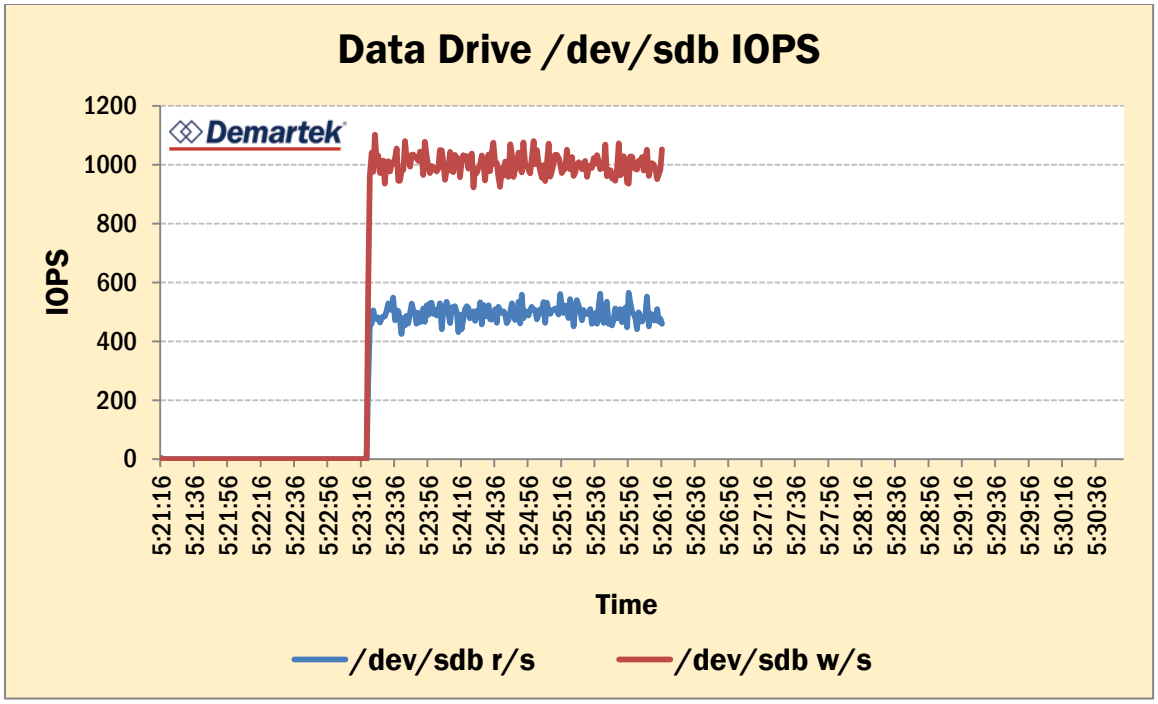
Availability Test Results

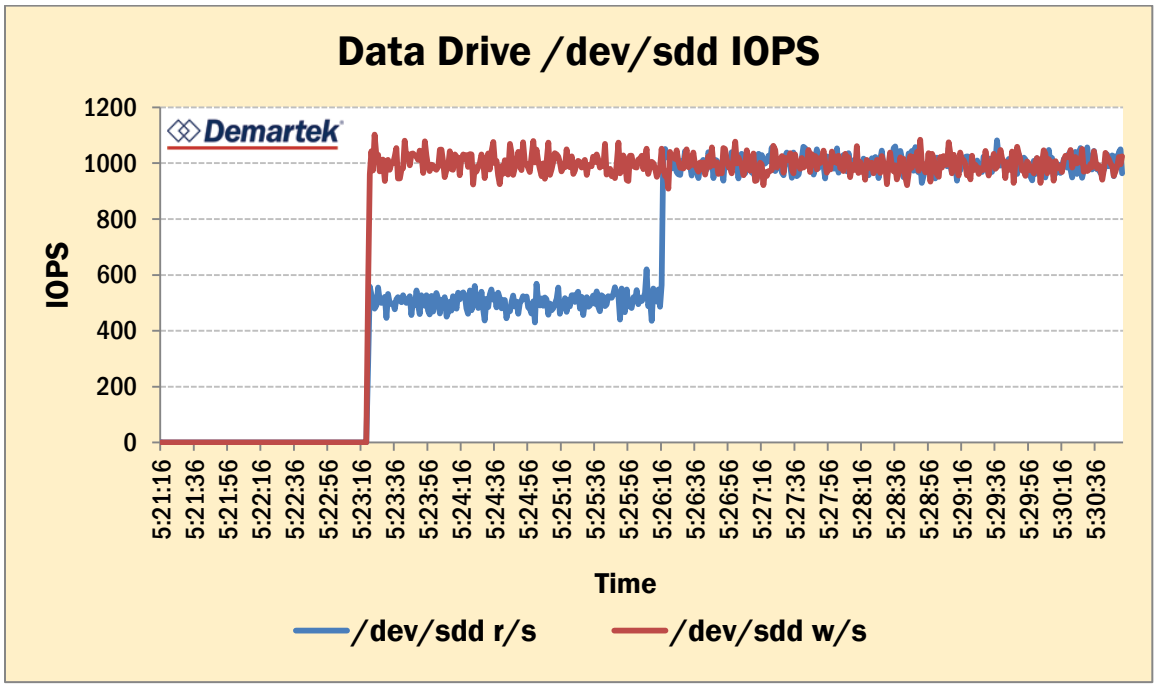
During all availability tests on the mdadm administered RAID devices, the system remained online and available. When two disk drives were unplugged, the associated virtual disk went offline and the system carried on any I/O with the mirrored device. The same was true when a single OS disk was unplugged, the associated virtual disk went offine and the system carried on any I/O with the mirrored device. When an entire controller was taken out, both virtual disks associated with the controller went down and any I/O continued with the mirrored virtual drives.

It is interesting to note that Logical Volume Manager (LVM) can also be used to mirror the virtual disks. However, the Logical Volume Manager configuration did not pass all availability tests and is not recommended.

It is also interesting to note that VDbench was configured to provide equal amounts of read and write I/O to the mirrored devices, but writes per second for each drive were twice the reads per second for each drive. It would appear that the writes must be made to both drives in order for the mirror to not be broken, but the read load could be split between the two drives. One drive's read does not need to be checked against the others as they are trusted to be mirrored. This is confirmed when one drive is removed. After drive removal, the reads per second on the remaining drive increase to the same level as the writes after its twin is no longer available to share the load.

The following is an example from an availability test where two of the physical drives associated with the virtual drive sdb are removed, and its mirror virtual drive sdd sees an increase in reads per second while writes per second remains constant throughout the test.


Data Drive /dev/sdb IOPS
IOPS
1200
1000
800
600
400
200
0
5:21:16
5:21:36
5:21:56
5:22:16
5:22:36
5:22:56
5:23:16
5:23:36
5:23:56
5:24:16
5:24:36
5:24:56
5:25:16
5:25:36
5:25:56
5:26:16
5:26:36
5:26:56
5:27:16
5:27:36
5:27:56
5:28:16
5:28:36
5:28:56
5:29:16
5:29:36
5:29:56
5:30:16
5:30:36
Time
/dev/sdb r/s
/dev/sdb w/s


Data Drive /dev/sdd IOPS
IOPS
1200
1000
800
600
400
200
0
5:21:16
5:21:36
5:21:56
5:22:16
5:22:36
5:22:56
5:23:16
5:23:36
5:23:56
5:24:16
5:24:36
5:24:56
5:25:16
5:25:36
5:25:56
5:26:16
5:26:36
5:26:56
5:27:16
5:27:36
5:27:56
5:28:16
5:28:36
5:28:56
5:29:16
5:29:36
5:29:56
5:30:16
5:30:36
Time
/dev/sdd r/s
/dev/sdd w/s

Stress Tests

Simple I/O stress tests were run using both controllers on Mdadm-managed drives, and then repeated using only one controller. The following performance metrics were captured:

- IOPS
- Throughput (or bandwidth, measured in MBPS)
- Latency (or response time, measured in milliseconds)
- % CPU used

VDbench was used to generate the following workloads:

VDBench Random I/O Parameters

- Block size 4KB
- Threads: 1,4,16,64, and 256 (Queue Depth or QD)
- Read/write profiles: 100%read, 100%write

VDBench Sequential I/O Parameters

- Block size 256KB
- Threads: 1,4,16,64,256 (Queue Depth or QD)
- Read/write profiles: 100%read, 100%write

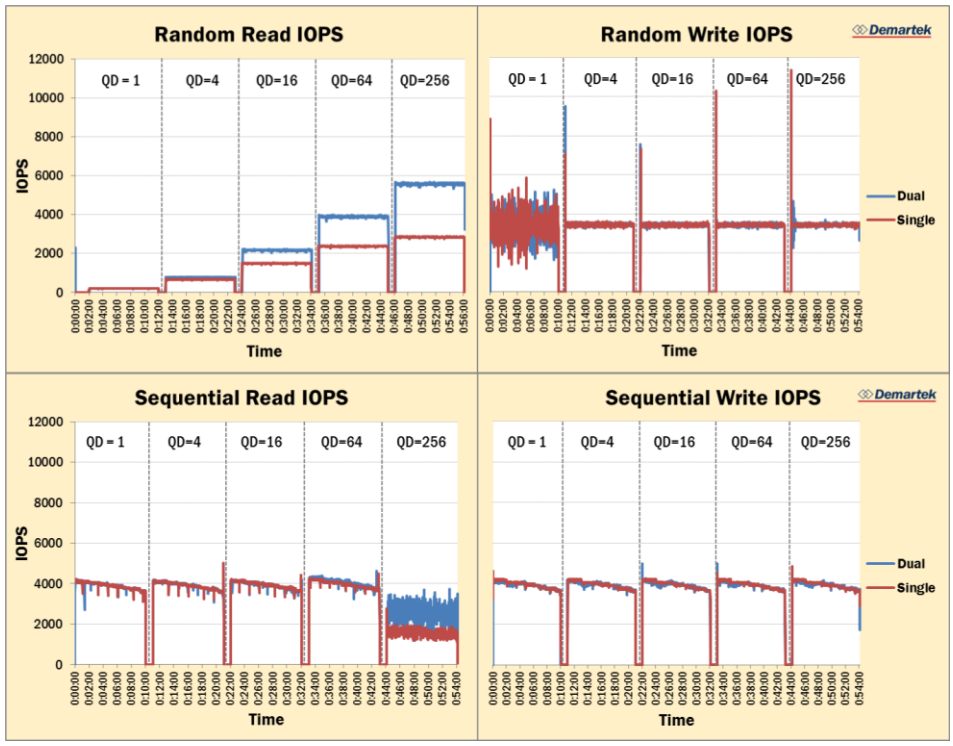
Results

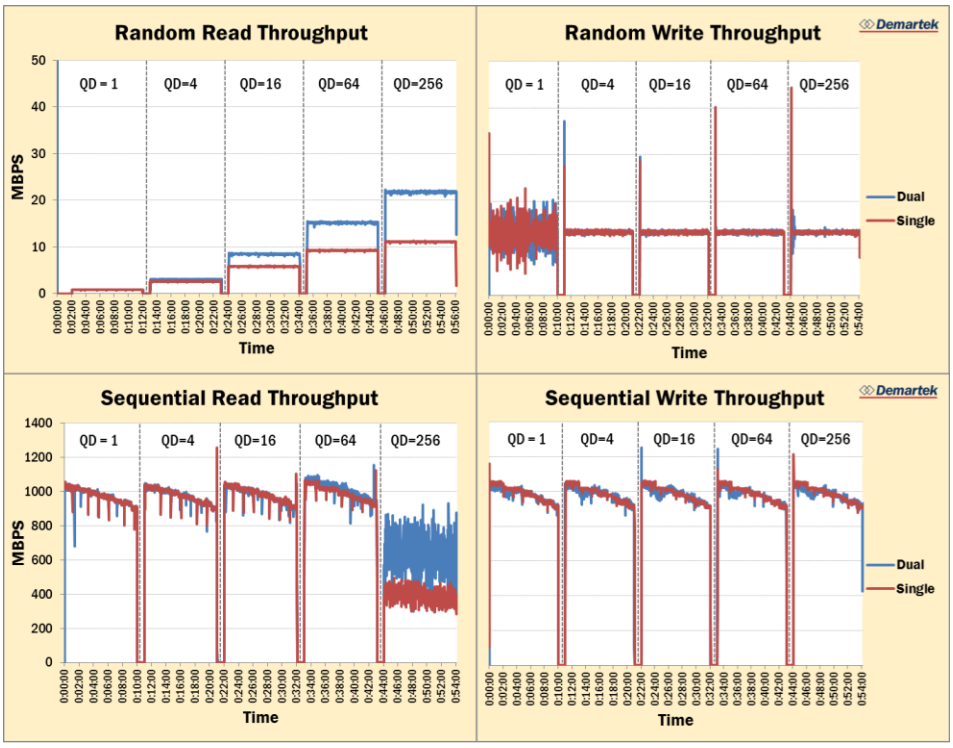
For reads, whether they were random or sequential, the mirrored dual configuration generally had higher performance in terms of IOPS and throughput. As the Queue Depth (thread count) increased, the performance difference increased with it.

For writes, whether they were random or sequential, the mirrored dual configuration did not differ much in terms of IOPS and throughput.

IOPS and Throughput (Bandwidth)

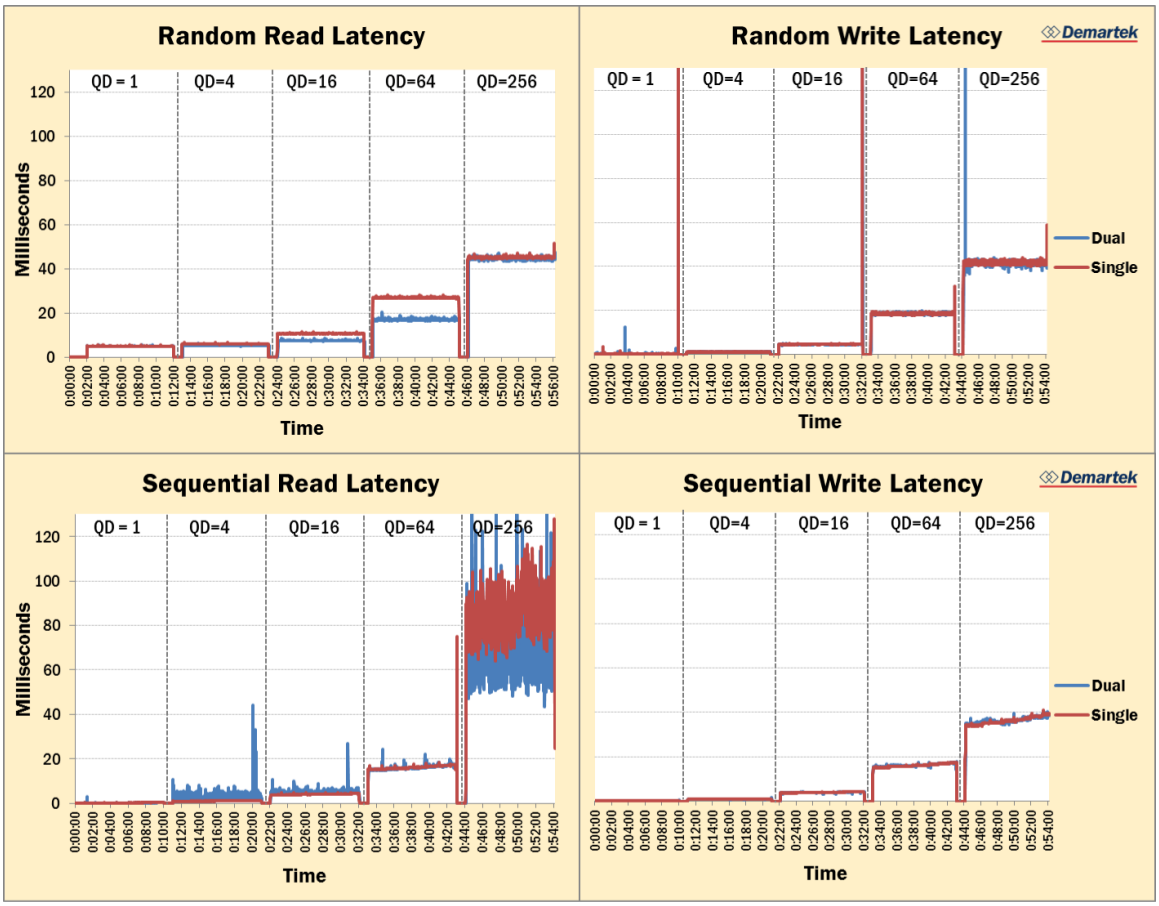
For reads, the mirrored dual configuration generally had higher performance, especially as the queue depth (thread count) increased. For writes, the single and dual configuration performed very similarly.





Latency or Response Time

Latency on the mirrored dual backplane was slightly lower on average for some queue depths of Random Reads and Sequential Reads, but for the most part did not vary much from the latency for single backplane.



CPU Utilization

CPU was monitored to make sure that the server was not being overloaded. At no point did %CPU consume more than 20%, even when the storage was stressed beyond its maximum performance point at Sequential Read Queue Depth 256.

Real-World Workloads

Two sets of tests, Microsoft Exchange JetStress 2010 and a variation of an On-Line Transaction Processing (OLTP) workload were run from Microsoft Windows Server 2012 Virtual Machines.

Microsoft Exchange JetStress 2010

Microsoft Exchange Jetstress 2010 simulates the Exchange Server disk input/output (I/O) load. This tool, provided by Microsoft, verifies the performance and stability of a storage subsystem and its suitability for Microsoft Exchange Server. Jetstress is generally used by customers before deploying Exchange Server to ensure that the storage subsystem can sustain a production workload.

The Jetstress configuration we ran represents a company with 500 employees. These JetStress configurations used 500 mailboxes of size 700MB using the "heavy" user profile of approximately 250 messages per day. There were 5 databases contained in one data drive, each of which was accessed simultaneously. The JetStress tests were run for a minimum of 2 hours each. Seven threads per database kept the latency below the 20ms database read limit.

Jetstress Test Results

Three Jetstress tests were performed with the following results:

	Achieved IOPS	DB Average Read Latency	Log Average Write Latency	CPU Utilization
Test 1	1315.45	16.75	0.33	2.20%
Test 2	1297.72	17.02	0.34	2.16%
Test 3	1298.37	17.00	0.33	2.15%
Test 4	1338.92	16.42	0.33	2.20%
AVERAGE	1312.61	16.80	0.34	2.18%

On-Line Transaction Processing (OLTP) Workload

The On-Line Transaction Processing workload models a 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.

The benchmark is "scalable," meaning that the number of customers defined for the brokerage firm can be varied to represent the workloads of different-size business. The benchmark defines the required mix of transaction the benchmark must maintain. The TPC-E metric is given in transactions per second (tps). It specifically refers to the number of Trade-Result transactions the server can sustain over a period of time.

This TPC-E workload was not used to generate an official TPC benchmark score, but was used to provide an actual database workload for the purposes of comparing the performance of storage systems.

For the test, we limited Microsoft SQL Server to 8GB of RAM even though more was available. We did this in order to limit the effects of memory caching functions and put more stress on the storage.

The workload was run on an OLTP database of 30,000, with 250 active users for the duration of the test. Three data VHD of size 110GB and one log VHD of size 204GB were provided to the database VM for the TPCE database.

Perfmon

The performance data shown in this report was taken from Performance Monitor (PerfMon), running inside the database VM. Perfmon is the standard performance monitoring application that is provided with the Windows operating system. CPU utilization, physical disk metrics, logical disk metrics, and memory usage were captured while the real-world workloads were running.

OLTP Workload Test Results

The OLTP Database workload resulted in an average of 274 SQL Server Transactions per Second, an average of 34.6 MB/s bandwith total across three virtual hard drives, average total IOPS of 4,387 across all Data drives, and a latency of 82 ms average.

Summary and Conclusion

This dual-backplane mirror configuration offers high availability with the possibility of increased read performance:

- Demartek has proved that the configuration presented does enable the system to remain available when one or more drives, including an OS drive, fail.
- The mirrored backplane configuration performance was on par with or exceeded the performance of a single unmirrored backplane, proving that performance was not impeded by the high availability configuration.
- Microsoft Exchange Jetstress 2010 and an Online Transaction Processing (OLTP) workload were run on VMs that were using the mirrored backplane data device as a datastore, proving this to be an effective configuration

We recommend the Dual PERC backplane configuration with RHEV VMs running real-world workloads as a high-availability configuration.

The most current version of this report is available at http://www.demartek.com/Demartek_Dell_LSI_Dual_PERC_RAID_Evaluation_2014-03.html on the Demartek website.