Optimizing DELL™ PowerVault™ MD1200 Storage Arrays for High Performance Computing (HPC) Deployments

A Dell Technical White Paper

Li Ou, Scott Collier, Onur Celebioglu

Dell HPC Solutions Engineering

Dell HPC Solutions
www.dell.com/hpc
www.dellhpcsolutions.com

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

October 2010

Contents

Figures

Introduction

Cluster computing is currently the most commonly used architecture in high performance computing (HPC)[1]. For clusters to run efficiently in HPC environments, the storage subsystem must be well optimized.

In this paper, we investigate the effects of different configuration settings on the performance of Dell™ PowerVault™ storage arrays, and communicate the results as best practices for setting up similar storage architectures in HPC environments. To perform the study, we configured PowerVault MD1200 storage arrays with various optimizations to determine the impact of overall I/O performance. The study results indicate considerable improvement with optimized RAID configurations and LUN aggregation.

About Storage Arrays

Storage arrays are one of the key building blocks used to architect high-performance I/O systems, since they enable the block level service and serve as the primary data repository for upper-layer components. A typical storage array includes one or more enclosures, and hosts multiple hard drives.

RAID controllers are required as a key interface between hosts and storage arrays to govern everything that occurs within the disk storage enclosure. This includes performing necessary calculations, controlling I/O operations, handling communication with management applications, and storing firmware. RAID controllers are either integrated within the storage arrays (RBODs), or designed as HostRAID adapters that are required on the host side if storage arrays are engineered as JBODs.

Most RAID controllers support various RAID configurations, including levels 0, 1, 5, 6, 10, 50, and 60. RAID controllers present the storage space to hosts as one or multiple Logical Unit Numbers (LUN). Typically, a LUN is created from multiple hard drives contained within the storage array, and directly mapped to a RAID group that is created and managed by the RAID controller,

Storage arrays not only provide a primary data repository, but they also set an I/O performance boundary for the entire system. A storage array's performance depends on several factors including: the number and the type of hard drives within the array, the performance of the RAID controllers managing the array, and the capacity of links between both the controllers and the hard drives and between the hosts and the controllers.

Beyond physical components, the most important factor that impacts storage array performance is how the array is configured to present LUNs to hosts, including how RAID groups are created and aggregated.

RAID Group Creation

Selecting the best storage configuration to satisfy a system's design requirements requires careful balancing of reliability, performance, and capacity. To create a well-optimized RAID group, users must consider how to best configure RAID level, striping width, and chunk size for both reliability and performance. If it is suboptimal to build a single RAID group across the entire storage array, the array may instead be separated into multiple RAID groups.

RAID Group Aggregation

Another issue that may impact performance is the organization and presentation of multiple RAID groups to upper-layer components, such as the cluster file system.

Typically, a RAID group is presented to the host as a LUN. If a file system, such as Lustre, has the ability to aggregate multiple LUNs into a single name space, the process of optimizing the LUNs is transferred to file system configuration. On the other hand, if a file system can only manage a single name space on a single LUN — such as most local file systems, ext3, ext4, and xfs — users may consider investigating whether it is possible to aggregate multiple physical LUNs into a single logical LUN before presenting them to the file system.

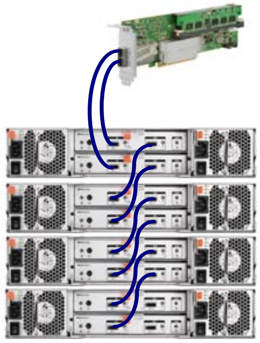
Overview of PowerVault Arrays

PowerVault storage arrays — including MD1200, MD1220, MD3200, and MD3220 — provide a simple way to expand Dell™ PowerEdge™ servers to connect directly to high-performance storage. The MD1200 and MD1220 are JBOD storage arrays that require a PERC H800 HostRAID adapter on the host side. On the other hand, the MD3200 and MD3220 arrays have integrated RAID controllers, and only a SAS adaptor card is necessary within the hosts.

The MD1200 and MD3200 share the same enclosure and contain up to twelve 3.5” hard drives. The MD1220 and MD3220 share the same enclosure and contain up to twenty-four 2.5” hard drives. All of the array models use the same 6-Gb SAS technology supporting various RAID configurations, including levels 0, 1, 5, 6, 10, 50 and 60. Multiple enclosures can be connected to each other by daisy chaining, offering a great flexibility to satisfy both capacity and performance requirements.

Dell 11th Generation (11G) PowerEdge servers support the H800 external HostRAID for access to PowerVault MD1200 and MD1220 JBOD 6-Gb/s SAS enclosures. The H800 adapter has two x4 external mini-SAS ports and a battery backup unit. The adapter is supported by a PCIe 2.0 x8 host interface. The H800 card features redundant paths with I/O load balancing to MD1200/MD1220. Multiple enclosures can be daisy chained from each H800 port.

Figure 1. **Redundant Paths to Connect Storage Arrays.**



Experimental Configurations

In this study, a PowerEdge R710 server with a single H800 adapter was connected to MD1200 storage arrays, which were daisy chained and connected to host adaptors through redundant paths, as shown in Figure 1.

Two types of hard drives were used in this study: Seagate 450GB 15K SAS (representing a performance-oriented option) and Seagate 2TB 7200rpm Nearline SAS (representing an option that balances performance, capacity, and cost). Red Hat Enterprise Linux® (RHEL) 5.4 was installed on the R710 server.

Throughput and input/output operations per second (IOPs) were evaluated with various storage configurations to optimize overall system performance. As benchmarks, the routine Sg_dd[2] was used for throughput and the routine iometer[3] was used for IOPs. Both benchmarks were configured to run on raw LUNs without any file systems to avoid extra overhead.

Within this paper, the convention **RAID Type / Span / # of Drives** is used to specify a RAID group. For example, R60/12/24 means a RAID 60 group with a total of twenty-four drives. Each span contains twelve drives: ten data drives and two parity drives. The last parameter may be expressed as “x” if the total number of drives is unknown; for example, R60/12/x represents a RAID 60 group with a span of twelve drives but no specified total number of drives. The last parameter may also be ignored if a RAID group contains only one span. R6/12 means a RAID 6 group with a total of twelve drives: ten data and two parity.

Additionally, a parameter called chunk size is used to refer to the amount of data that the RAID controller writes on a single physical disk in a RAID group before writing data on the next physical disk.

Creating RAID Groups

One of the main criteria in designing storage subsystems for HPC is price/capacity (e.g., cost per GB). While RAID 0 and RAID 1 may be suitable for special applications, the low usable capacity of RAID 1 and low reliability of RAID 0 make them unpopular for general purpose environments. This study focuses on RAID 5 and RAID 6 since these RAID levels provide a balance between reliability, performance, and capacity.

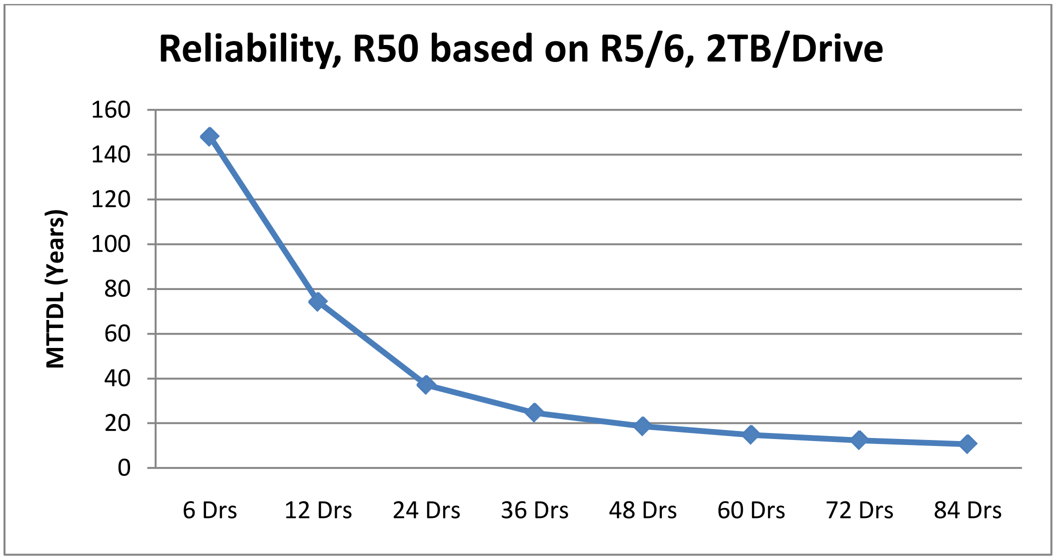
A single MD1200 storage enclosure holds up to twelve 3.5” hard drives. To fully utilize the number of drives and the storage capacity of one enclosure in an HPC environment, either a single RAID group with a span of twelve drives or two RAID groups with a span of six drives is recommended for each enclosure. A RAID 5 group with twelve drives is unlikely to be deployed for applications that require high reliability; due to single parity, its span may become too wide to be considered reliable. Therefore three RAID configurations for MD1200 arrays were evaluated: RAID 5 with six drives (R5/6), RAID 6 with six drives (R6/6), and RAID 6 with twelve drives (R6/12).

Reliability Considerations

In certain applications, multiple RAID 5 or RAID 6 groups must be grouped together as a single R50 or R60 group to present hosts with a single name space. In this scenario, the reliability of the storage array decreases as each R50/R60 group becomes larger. The popularity of high-density hard drives, such as 2TB Nearline SAS drives, worsens the situation due to increased possibility of unreadable segments during reconstruction of the RAID group.

An analysis was conducted to evaluate the reliability of various widths of R50/R60 groups with 2TB drives in terms of mean time to data loss (MTTDL)[4], making the following assumptions: mean time between failures (MTBF) of each drive is 600K hours, sector size is 512 bytes, failed drives are replaced immediately (hot spares), and the bit error rate is 10^{-15}. Although the assumptions are conservative for MTBF and typical for the other parameters, these are statistical estimates and actual reliability may vary greatly depending on usage patterns, environmental factors and other system parameters. It is also assumed that R50/R60 is implemented by striping (RAID0) across groups of R5/R6 created using 6 drives each. The analysis results indicate a ~248,000 year MTTDL of a single R60 group with forty eight drives. However, as shown in Figure 2, an R50 group provides significantly less protection at MTTDL of 18.6 years. Therefore R50 may not be preferable for long term storage.

Figure 2. **Reliability of a R50 Group**



According to the reliability analysis, when multiple R5/R6 groups from an MD1200 are aggregated into a single R50/R60, R6/12 or R6/6 RAID groups and the correspondent R60 group are recommended for general purpose, long-term storage. The R5/6 RAID groups and the correspondent R50 group are practical only for scratch space implementations. However, if multiple R5/R6 groups are presented separately to hosts and the upper-layer file system is able to aggregate them together, then users can consider any of the three configurations and choose a configuration based on other factors, such as performance and capacity.

RAID Group Level

With the configuration of a single H800 and multiple daisy-chained MD1200 arrays, Figure 3 and Figure 4 present sequential read/write throughput for various numbers of hard drives under the three RAID configurations discussed in the previous section. Figure 5 and Figure 6 show the corresponding results for random IOPs. The chunk size was set to 512KB.

For all of the configurations studied in this paper, multiple read/write requests were concurrently posted and each RAID group was assigned at least one request stream.

Results indicated that, on average, an R6/12 RAID group has the best throughput for sequential access. Alternatively, an R5/6 RAID group has the best IOPs for random access. Although R6 is more expensive than R5 in terms of parity maintenance, R6/12 RAID groups have better sequential throughput than R5/6 RAID groups, because the number of R6/12 groups is half that of the R5/6 groups for the same number of drives. Thus, on aggregate, the RAID controller may have less burden to calculate parities for R6/12 groups than for R5/6 groups.

When an R5 and an R6 were compared at the same width, such as R5/6 and R6/6, the R5 still performed better in each test as shown in Figures 3, 4, 5, and 6. Considering the higher capacity overhead and performance deficit, R6/6 is not recommended for HPC storage configurations, especially in configurations that include twenty-four drives or more.

It is important to note that, although R6/12 is a good choice to balance between sequential performance and reliability, R5/6 is still worthy of consideration when random IOP performance is a dominant factor.

Figure 3. **One H800 with MD1200 arrays, Sequential Write**

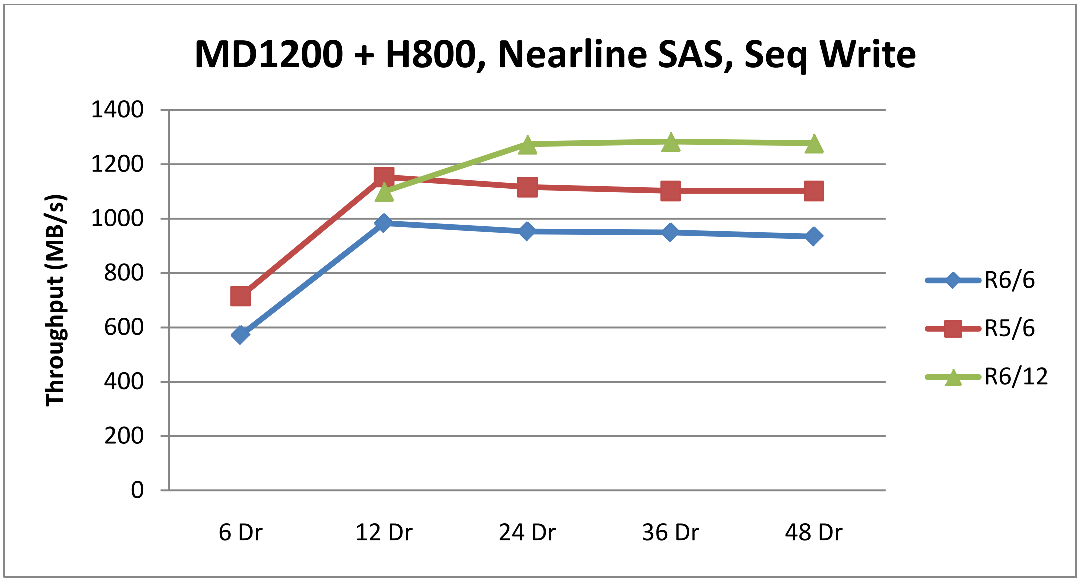


Figure 4. **One H800 with MD1200 arrays, Sequential Read**

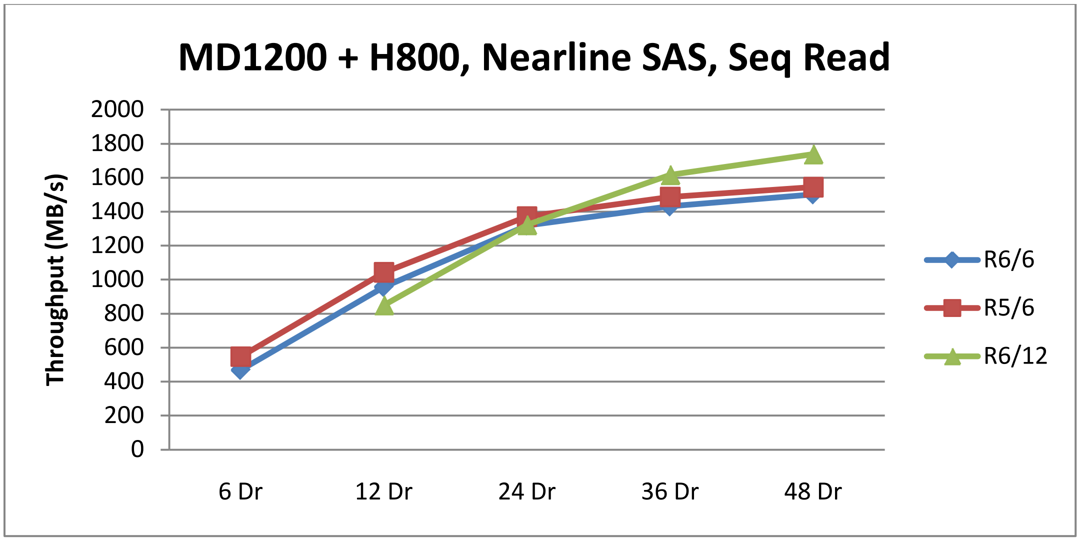


Figure 5. **One H800 with MD1200 arrays, Random Write**

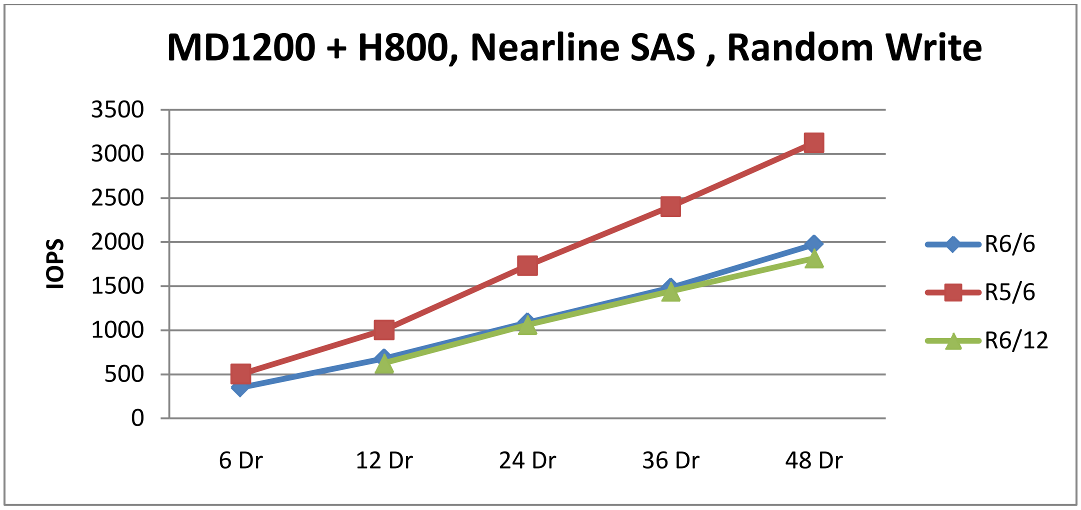
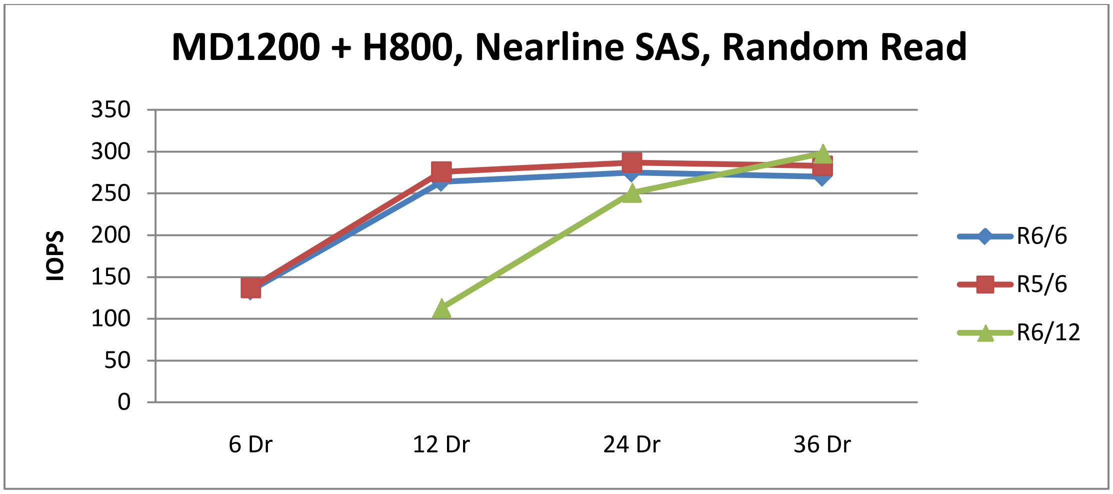


Figure 6. **One H800 with MD1200 arrays, Random Read**



Chunk Size

Beyond RAID level and striping width, chunk size is another parameter that may significantly impact the performance of a RAID group. Because the I/O access patterns of different applications vary, there is not a singular recommended chunk size for all applications. Instead, in this study several different chunk sizes were reviewed to find an optimal setting for general-purpose, sequential throughput-oriented HPC applications for MD1200 storage arrays. To determine the best parameter for your applications, consider the design needs of your specific application.

With one H800 and multiple daisy-chained MD1200 arrays, the sequential read/ write throughput of a single R5/6 RAID group and a single R6/12 RAID group was considerably impacted by the various chunk sizes (Figure 7 and Figure 8). A 64 KB chunk size is the default option to create a RAID group through the H800 management console. The test results showed that a larger chunk size could help boost performance for sequential access patterns. Considering the fact that an excessively large chunk size may hurt the performance of small-size random requests, an optimized chunk size of 256 KB is recommended for an R5/6 group and an optimized chunk size of 512 KB is recommended for an R6/12 group. These recommended values represent the smallest chunk sizes that will help corresponding RAID groups reach their optimal read/write throughput.

Figure 7. **Sequential Throughput of an R5/6 with Various Chunk Sizes**

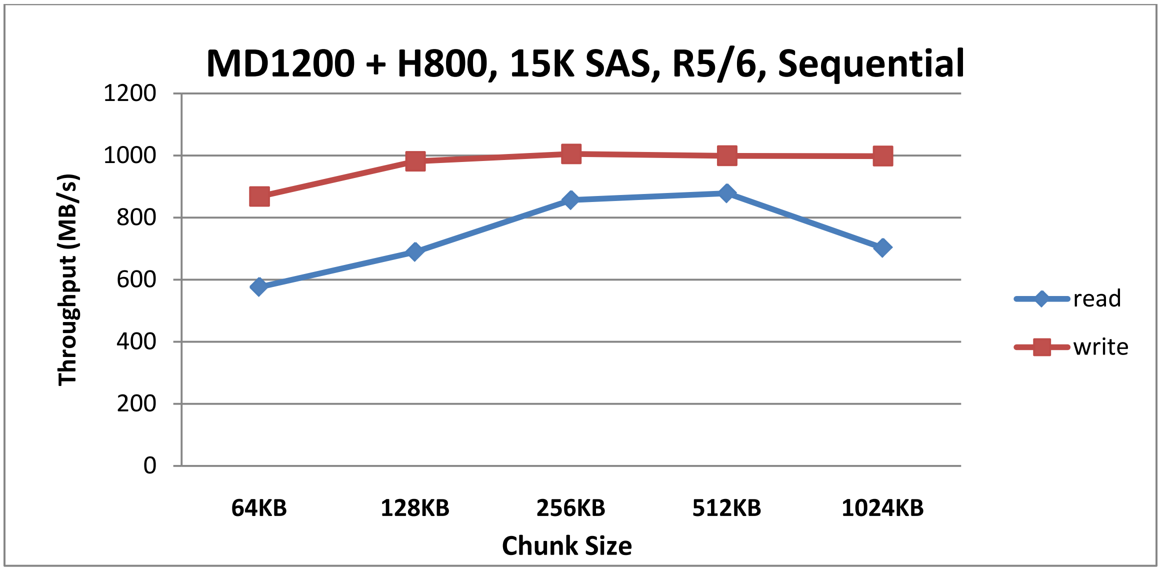
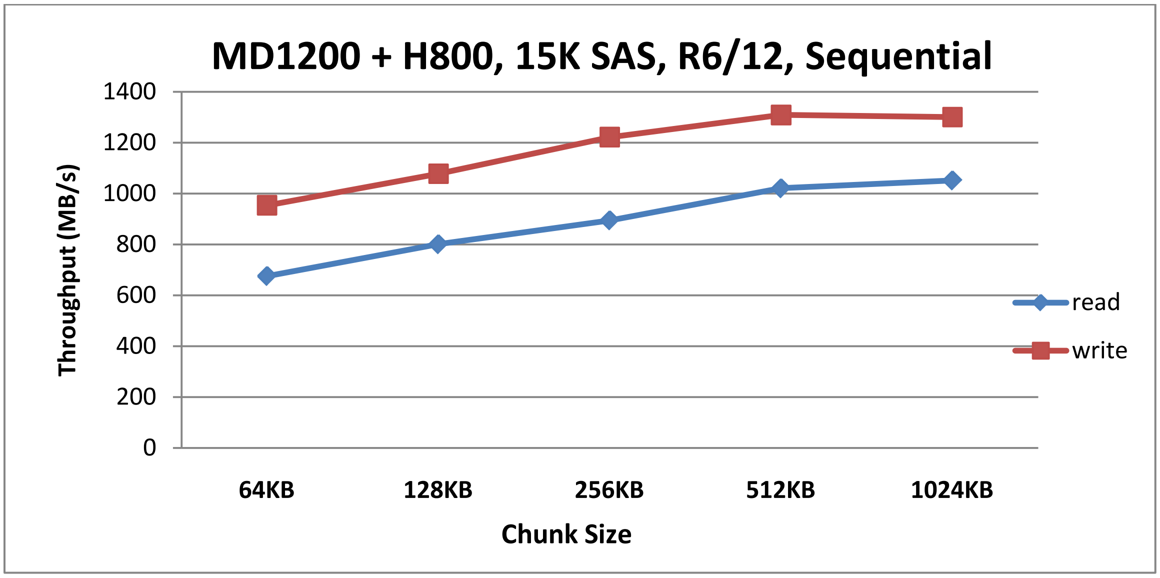


Figure 8. **Sequential Throughput of an R6/12 with Various Chunk Sizes**



Aggregating RAID Groups

Typically, a RAID group is presented to the host as a LUN. As described in previous sections, multiple RAID groups created on a storage enclosure will be mapped to separate LUNs on the host system.

In HPC deployments where a single name space is desired, users need to aggregate physical LUNs into a single logical LUN. Generally, there are two possible places to make this happen. One place is inside qualified RAID controllers, where RAID 50 or RAID 60 can be used to stripe data across multiple RAID 5 or RAID 6 groups by the controller, presenting a single LUN to the host. Another place to accomplish this is within the host Operating System (OS).

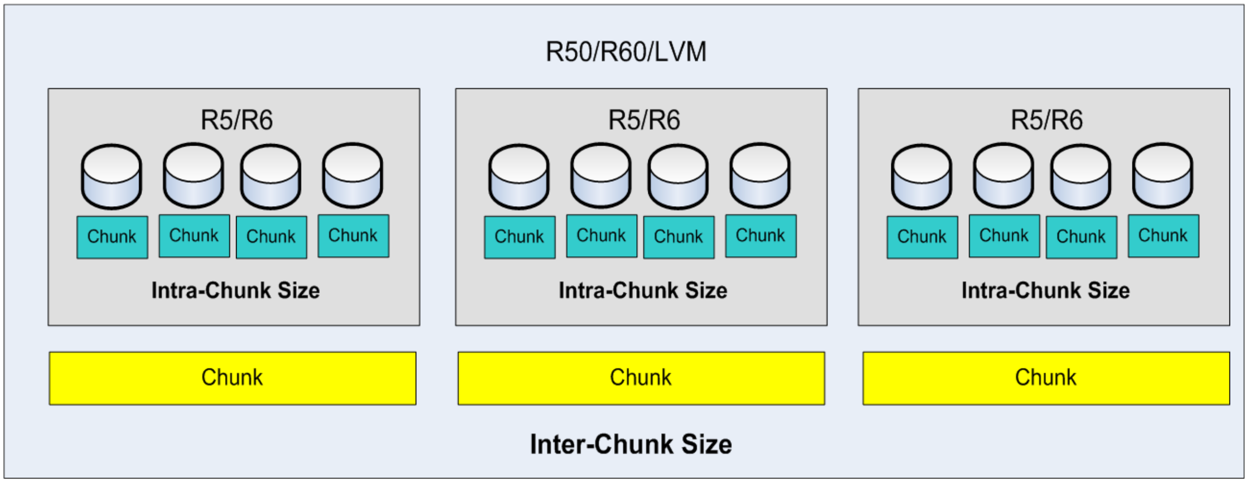
Utilities in popular HPC operating systems, such as the logical volume manager (LVM) in the Linux kernel, can group multiple physical LUNs together. LVM allows users to create single logical volumes from multiple physical LUNs mapped either through a striping mode like RAID 0 or through a concatenated mode. Also, it supports read-only snapshots from logical volumes, and allows resizing logical volumes online. Compared to concatenation, striping may better utilize the performance of underlying physical LUNs.

After physical RAID groups are aggregated via striping mode (Figure 9), multiple layers of chunk sizes exist. In this paper, a chunk size inside a single R5/R6 group is referred to as an intra-chunk size, and a chunk size used by the R50/R60/LVM layer to group individual R5/R6 groups is referred to as an inter-chunk size. The chunk size investigated in the previous section is the intra-chunk size; however, the intra-chunk size and inter-chunk sizes are independent parameters and do not necessarily need to be identical.

An H800 RAID adaptor allows users to build RAID 50/RAID 60 groups on MD1200 arrays, but does not distinguish between inter-chunk and intra-chunk — users can only specify one size for both inter- and intra-chunks. On the other hand, if physical R5/R6 RAID groups are built on an H800 adaptor and then aggregated through the LVM striping mode, intra-chunk and inter-chunk size can be specified separately. In this scenario, the intra-chunk size is specified through the H800 console when creating a RAID group and the inter-chunk size is specified with an LVM command when creating a logical volume (`lcreate –i –I`, where `–i` specifies striping width and `–I` specifies chunk size).

This paper reviews the performance impact of the two different methods described above to aggregate LUNs. The goal was to find the best practices for tuning Dell storage arrays for use in HPC, including setting the optimized chunk size for R50/R60 groups and the optimized inter-chunk size for LVM striping mode. However, as explained in the previous section, there is not a singular recommended chunk size optimized for all applications. The recommended values in this section are for a general purpose HPC environment, and users may still need to tune their system for specific applications.

Figure 9. **Multiple Layers of Chunk Sizes**



Inter-Chunk Size

The first parameter we investigate in this section is the chunk size for R50/R60 RAID groups. With an H800 and multiple daisy-chained MD1200 arrays, the sequential read/write throughputs of a twenty-four drive R50 group with a span of six (R50/6/24) and a twenty-four drive R60 group with a span of twelve (R60/12/24) are both considerably impacted by the various chunk sizes (Figure 10 and Figure 11).

Compared to the intra-chunk size investigated for a single R5/R6 group, the best value for an R60 is still 512 KB, but the best value for an R50 changes from 256 KB for an R5 to 512 KB. Considering small-size random requests, a 512 KB chunk size is recommended for both R50 groups with an internal span of six (R50/6/x) and R60 groups with an internal span of twelve (R60/12/x).

Figure 10. **Sequential Throughput of an R50/6/24 with Various Chunk Sizes**

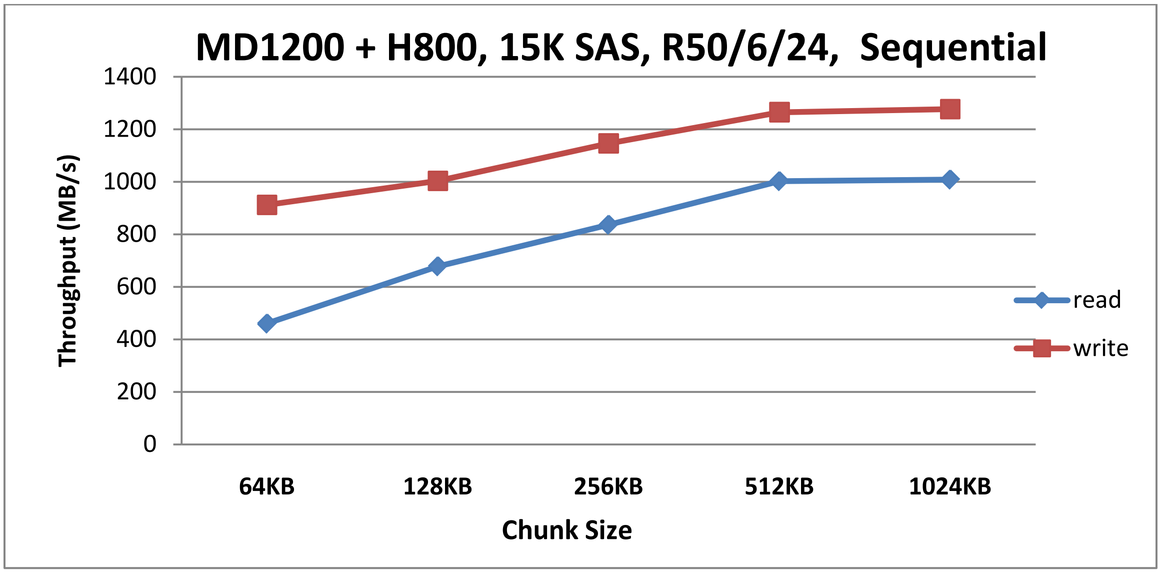
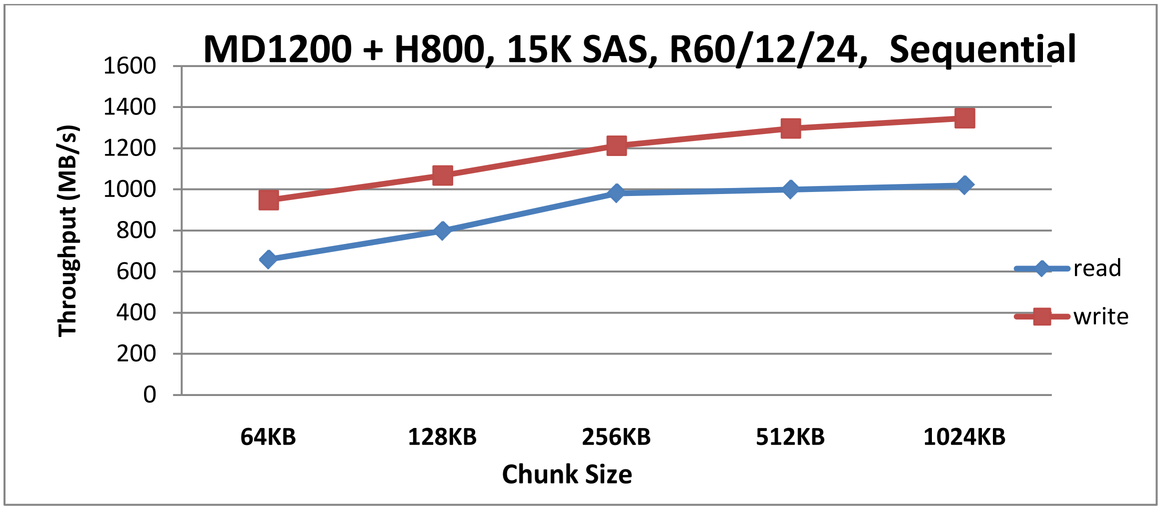


Figure 11. **Sequential Throughput of an R60/12/24 with Various Chunk Sizes**



The second investigation analyzed the inter-chunk size of the LVM striping mode on twenty-four drives, with both a four R5/6 group configuration (Figure 12) and a two R6/12 group configuration (Figure 13). A 64KB chunk size is the default parameter of the LVM striping mode. The intra-chunk sizes for individual R5/6 and R6/12 groups followed the recommendations from the previous section.

Unlike results from previous sections, the write throughput for inter-chunks was not obviously impacted by larger chunk sizes. Instead, a larger chunk size helps boost read performances for sequential access patterns. 1024 KB is the best value for the inter-chunk size of R5/6 groups, and a range from 512 KB to 1024 KB is recommended for the inter-chunk size of R6/12 groups. Normally, it is not recommended to use an inter-chunk smaller than the corresponding intra-chunk.

Figure 12. **Sequential Throughput of Four R5/6 with Various Inter-Chunk Sizes**

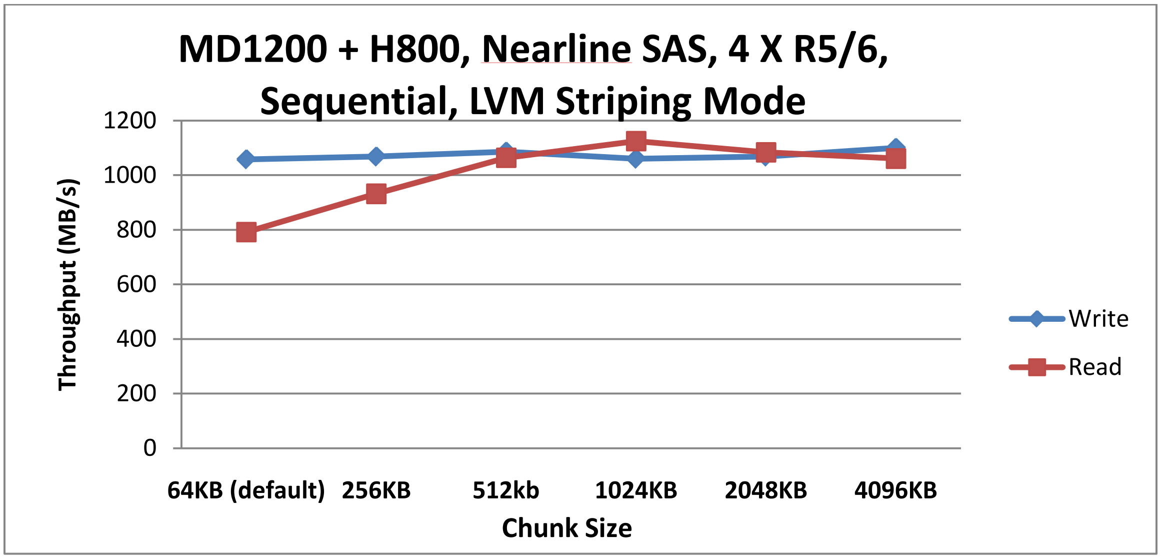
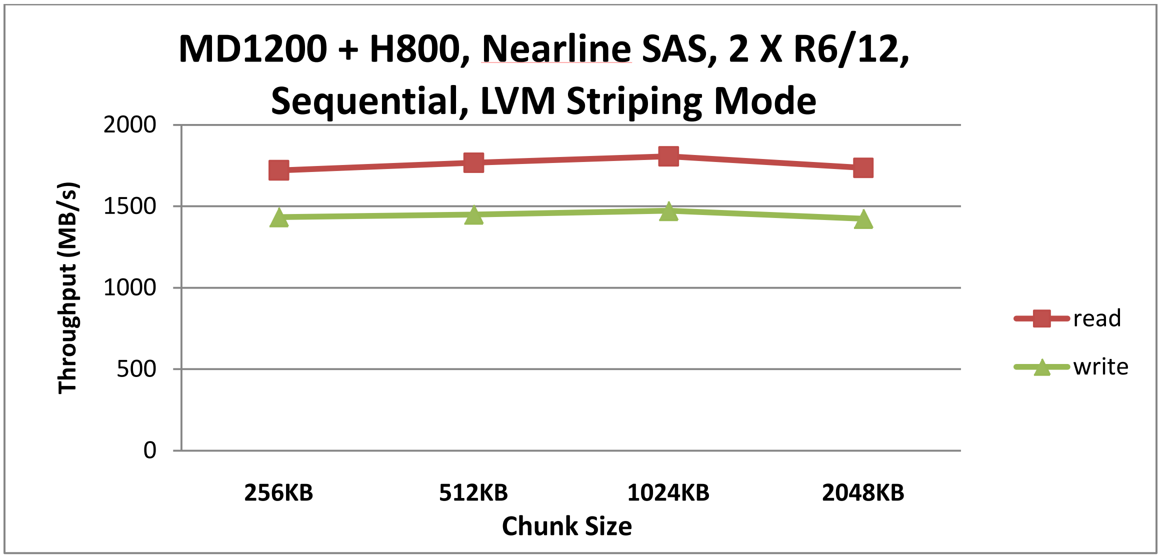


Figure 13. **Sequential Throughput of Four R6/12 with Various Inter-Chunk Sizes**



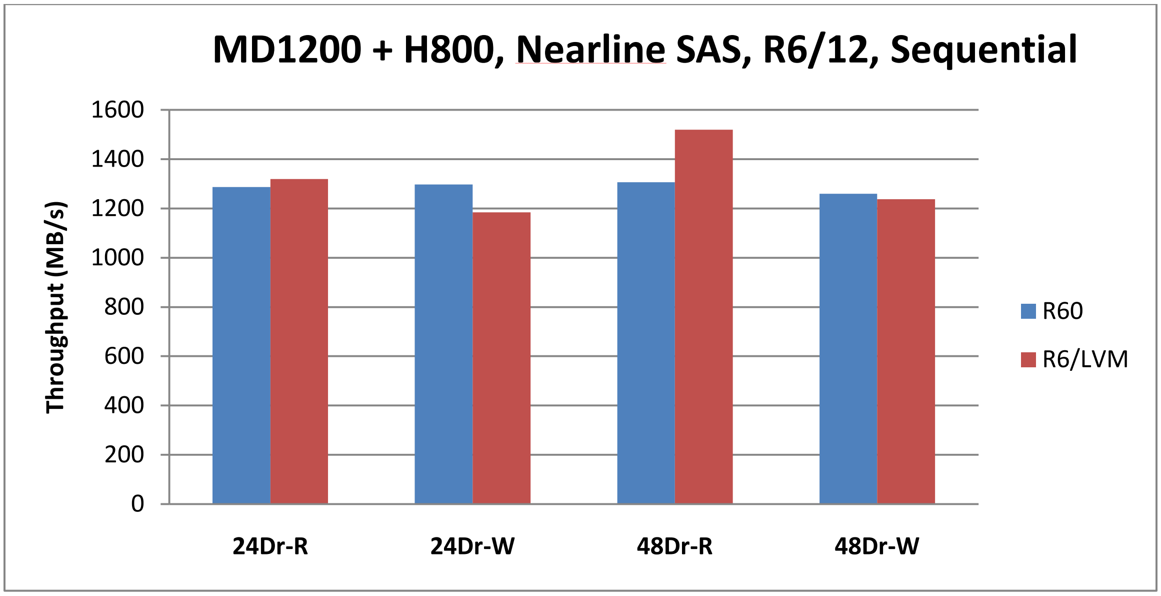
R50/R60 vs. LVM

After determining optimized inter-chunk sizes, another question remains: Which mode is optimal to aggregate physical LUNs? With a configuration of an H800 plus an MD1200 storage array, a comparison between both R50/R60 and LVM is necessary to reach a conclusion.

To answer this question, two storage array configurations were evaluated. One configuration had two MD1200 arrays with twenty-four drives and the other configuration included four MD1200 arrays, for a total of forty-eight drives. R60 with a span of twelve (R60/12/x) was compared with LVM on multiple R6/12 groups for both storage configurations. Optimized chunk sizes determined in previous sections were applied: a 512 KB chunk size for R60/12/x groups a 512 KB intra-chunk size for R6/12 RAID groups, and a 1024 KB inter-chunk size for the LVM configuration.

With the same storage, both R60 and LVM with multiple R6 RAID groups show similar sequential read/write throughput (Figure 14), suggesting that performance may not be a dominant factor in selecting a method for aggregating physical LUNs. R60/R50 groups may be more convenient for configurations because they are built once within RAID controllers. Alternatively, LVM is more flexible and offers more features, such as snapshot. Users should make a decision based on their own specific requirements.

Figure 14. **R60/12/x Vs. LVM on Multiple R6/12**



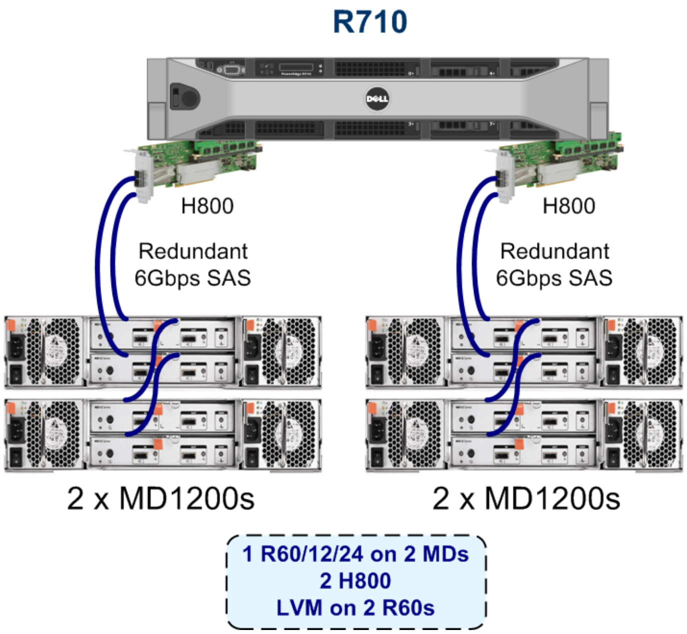
Scaling Up

The studies from the previous sections clearly show the performance bottleneck caused by an H800 RAID adaptor for MD1200 storage arrays (Figure 3 to Figure 6). If the controllers are saturated, the storage performance cannot be improved with additional hard drives. The sequential write throughput of an H800 almost reaches the saturation point with twelve Nearline SAS drives, and is totally saturated at 1300 MB/s with twenty-four Nearline SAS drives. With each extra drive, the read performance is somewhat better, but the throughput improves very slowly.

An intuitive way to overcome this bottleneck and improve resource utilization is to add additional controllers into the storage system. The idea is to limit the number of hard drives managed by a controller to a reasonable range to avoid a performance bottleneck. This configuration requires installation of an extra H800 adaptor in the hosts to double RAID processing capacity. Note that adding extra controllers to boost performance introduces additional complexity if multiple RAID groups from different controllers must be aggregated. RAID 50/RAID 60 does not work in such a scenario.

LVM may be used not only on R5/R6 groups, but also on R50/R60 groups that were pre-aggregated by the controllers. Figure 15 shows an example in which two H800 adapters are used to connect four MD1200s to a host with each H800 managing two MD1200s. In accordance with the previous recommended RAID configurations, an R60 group with twenty-four drives and a span of twelve (R60/12/24) was created on each H800, and two R60 groups are waiting to be aggregated on the host. Two questions must be answered in this configuration: What is the optimized LVM inter-chunk size for multiple R60s and what is the expected performance after R60 groups are aggregated via LVM?

Figure 15. **Two H800 adapters with Four MD1200s, LVM on Two R60 groups**



In this example, both R60 groups were created with an intra-chunk size of 512 KB (recommended from previous sections), and then various inter-chunk sizes for LVM were evaluated (Figure 16). The experimental results indicate that the sequential read/write throughputs are every similar when the inter-chunk size is larger than 256 KB. Considering the size of the intra-chunk, 512 KB and 1024 KB are two recommended values for LVM inter-chunk size to aggregate R60 groups.

Another test was conducted to evaluate the performance improvements by adding the second H800 into the configuration. LVM was used to aggregate RAID groups from different RAID controllers. Four storage array configurations were investigated. On the first configuration of two MD1200s and one H800, an R60 group with twenty-four drives and a span of twelve (R60/12/24) was created, with a 512 KB chunk size. On the second configuration of two MD1200 arrays and two H800 adapters, each H800 was connected to an MD1200 and two R6 groups with twelve drives (R6/12) and a 512 KB intra-chunk size were created (one from each H800), and then aggregated with a 1024 KB LVM inter-chunk size. On the third configuration of four MD1200s and one H800, an R60 group with forty-eight drives and a span of twelve (R60/12/48) was created, with a 512 KB chunk size. On the fourth configuration of four MD1200s and two H800 adapters, each H800 was connected to two MD1200 arrays and two R60 groups with twenty-four drives and a span of twelve (R60/12/24) was created, with a 512 KB intra-chunk size, one from each H800, and then aggregated with a 1024 KB LVM inter-chunk size (same as Figure 15). All four configurations were compared in terms of the sequential read/write throughputs (Figure 17).

With two MD1200 arrays of twenty-four drives, adding an extra H800 and aggregating with LVM yields an observable performance improvement — users may choose this configuration to balance cost and performance. However, with four MD1200 arrays of forty-eight drives, adding an additional H800 and aggregating with LVM boosts overall performance significantly. Two H800 adapters are strongly recommended for four MD1200 arrays.

Figure 16. **Sequential Throughput of Aggregating Two R60 Groups from Two H800 Adapters with Various Inter-Chunk Sizes**

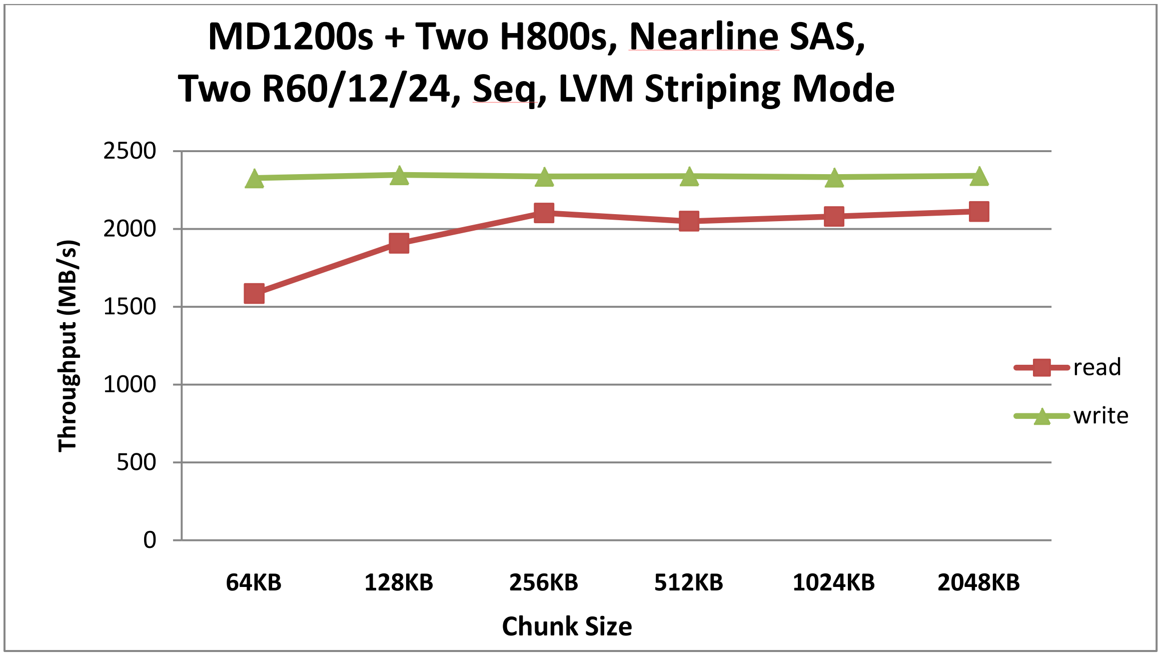
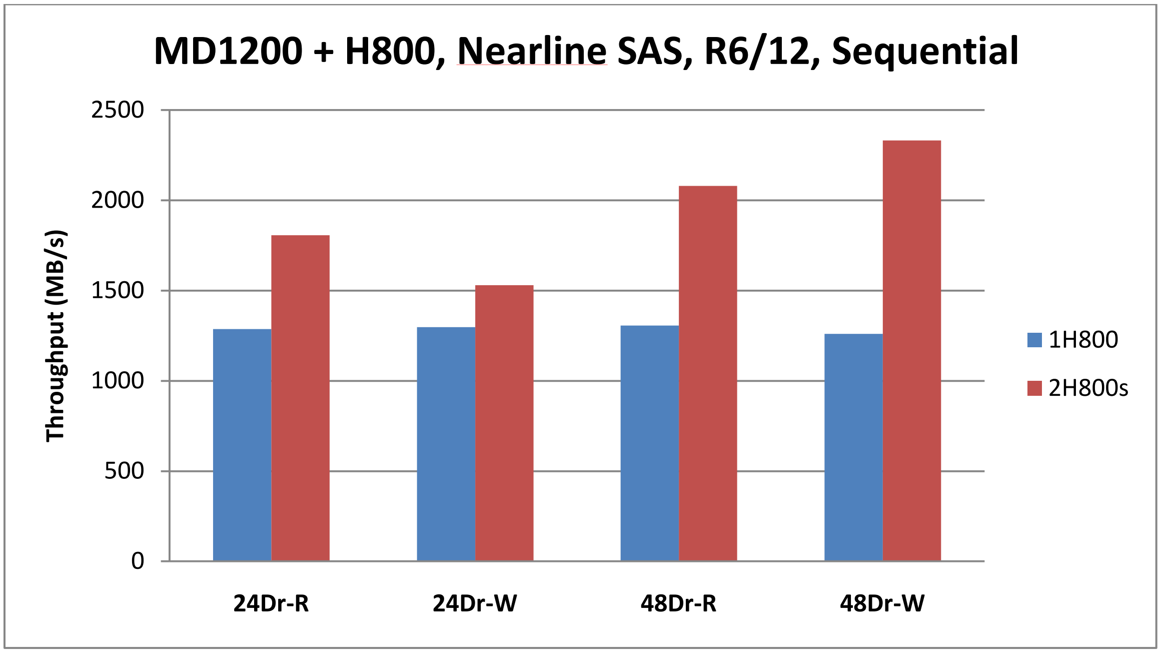


Figure 17. **Sequential Throughput: One H800 Vs Two H800 adapters**



Conclusions

In HPC, storage I/O may define the performance boundary for the entire system, and the configuration of storage arrays in the system may significantly impact overall performance.

The goal of this paper is to establish best practices for tuning MD1200 arrays for use in HPC. These best practices include: creating RAID groups, aggregating physical LUNs, and scaling up performance with additional RAID controllers.

After balancing between reliability, performance, and capacity, a RAID 6 group of twelve hard drives (R6/12) is recommended for general-purpose, long-term storage and a RAID 5 group of six hard drives (R5/6) is recommended for scratch space implementations.

To simplify management and administration, it is often desirable to present a single LUN to the server that can be exported to the compute nodes as a single name space. Here, two methods were investigated: Using RAID 50/60 supported by the RAID controllers and LVM striping on multiple R5/R6 groups. The two methods yielded similar sequential read/write throughput, suggesting that performance may not be a dominant factor in picking among these two methods to aggregate physical LUNs.

Chunk size inside a RAID group and chunk size among multiple RAID groups also impact read/write performance. For general purpose, throughput-oriented HPC applications, a 256 KB intra-chunk size is recommended for an R5/6 group and a 512 KB intra-chunk size is recommended for an R6/12 group. When LUN aggregation is necessary, a 512 KB chunk size is recommended for both an R60 group with a span of twelve drives (R60/12/x), and an R50 group with a span of six drives (R50/6/x). A 512 KB or a 1024 KB inter-chunk size is recommended for a configuration where LVM striping is used on multiple R5/R6 LUNs.

To further optimize storage performance, limiting the number of hard drives managed by a controller is recommended to avoid a performance bottleneck (twenty-four Nearline SAS drives for a single H800). Adding an additional H800 adapter greatly boosts sequential performance if there are four MD1200 enclosures in a system. When multiple R60 groups from different RAID controllers are aggregated via LVM, a 512 KB or a 1024 KB inter-chunk size is recommended. This method also facilitates scaling performance and capacity independently. If capacity is of primary concern, many MD1200 arrays can be daisy-chained behind a single H800.

References

1. Top 500 Supercomputers, http://www.top500.org
2. Sg_dd, Linux Man Page, http://linux.die.net/man/8/sg_dd
3. Iometer project, http://www.iometer.org
4. Ajay Dholakia, Evangelos Eleftheriou, Xiao-Yu Hu, Ilias Iliadis, Jai Menon, K. K. Rao. (2008) “A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors”, ACM Transactions on Storage Vol. 4, No.1, Article 1