Array Tuning Best Practices

A Dell PowerVault™ MD3200 and MD3200i Series of Arrays Technical White Paper

Dell

June 2010

Contents

Tables

Figures

Audience and Scope

This document is intended to guide Dell PowerVault™ MD3200 and MD3200i series of storage arrays customers through the advanced processes behind tuning their storage array to best suit their individual needs.

Performance Tuning Overview

The challenge of storage performance tuning is to understand and control the interacting components (listed below) while accurately measuring the application performance. Because the performance of the storage array accounts for only a portion of the overall application performance, tuning must be performed taking into account the input/output (I/O) characteristics of the application and all of the components involved in the data path, such as the SAS HBA, iSCSI initiator, network switch and host operating system settings. With multiple aspects to consider, the task of performance tuning to maximize performance for even one application can seem formidable. Tuning the system to maximize performance of multiple applications that may share a single storage array can seem even more formidable. To reduce the complexity of tuning, Dell PowerVault™ MD3200 and MD3200i series of storage arrays feature performance monitoring and flexible tuning controls that can be accessed through the Modular Disk Storage Manager (MDSM).

Components That Influence Storage Performance

This white paper provides an overall approach to tuning I/O performance and also provides specific guidelines for using the storage array tuning controls. These recommendations begin with an overall analysis of the elements that determine I/O performance:

- Storage array
- Application software
- Server platform (hardware, operating system, volume managers, device drivers)
- Network (for MD3200i only)

Basic Approach to Performance Tuning

The first principles of I/O performance tuning include the following question:

What should be the performance for my system?

The answers include:

- "It depends…" There are no absolute answers. Each environment is unique and the correct settings depend on the unique goals, configuration, and demands for the specific environment.
- "Actual mileage may vary." Results vary widely because conditions vary widely.

The answers to this question suggest the following basic approach to performance tuning:

1. Configure and test
2. Measure
3. Adjust as required

The performance monitoring features in all MD3200 and MD3200i series of storage arrays and the controls for tuning make them ideally suited for this iterative process. The first step in tuning is to establish a baseline of existing performance. When generating a performance baseline, it is ideal to use a workload that is as similar to the intended final use of the storage solution. This can be as simple as the real application or a SQL Replay with a system performance monitor (perfmon or sysstat/iostat as well as the CLI and state capture performance monitoring) or a synthetic benchmark package that closely imitates the expected I/O profile (Iometer, IOZone, Bonnie). By comparing the baseline data to the estimated needs and capability of the configuration, the user can effectively tune an MD3200 or MD3200i storage array. This white paper provides recommendations for this important first step as well as tuning tweaks to get the full capabilities from the MD3200 and MD3200i series of storage arrays storage systems.

Application Software Considerations

In understanding the I/O characteristics of intended applications using the storage in a manner as close to the expected run-time is required to determine the optimal storage and system configuration, and is critical to tuning the overall solution. This includes, but is not limited to the following:

- Number of discrete I/O sources interacting with the solution
- Randomness of data access by the primary I/O source(s)
- Mean size of typical I/O; this is usually divided into the following three categories:
    - Large Block (≥256KB) transfer size
    - Medium Block (≥32KB and < 256KB) transfer sizes
    - Small Block (< 32KB) transfer sizes
- Burstiness of I/O pattern, i.e. the mean duty-cycle of I/O to the storage array
- Profile of mean I/O direction; this is usually the ratio of reads to writes

Configuring the PowerVault™ MD3200 and MD3200i Series of Arrays

There are two ways to configure the PowerVault™ MD3200 and MD3200i series of storage arrays storage systems. The most common and easiest method is using the MDSM. The MDSM allows automatic configuration settings that provide reasonable configurations with little knowledge of performance tuning required. A manual configuration options are also available in the MDSM providing more flexibility but requiring more knowledge of performance requirements.

Determining the Best Raid Level

The first step involved in tuning an MD3200 or MD3200i storage array is to determine the RAID Level most appropriate for the solutions given the application. Please note that RAID 0 is excluded from most information in this paper due to the lack of data protection. This does not mean the use of RAID 0 is undesired, just that it should only be used for noncritical data. In general, RAID 0 generally provides better performance than RAID 1/10, 5 or 6. Additionally, RAID 6 is not always referenced specifically; most comments applicable to tuning RAID 5 are directly applicable to RAID 6 unless otherwise noted. In situations where the increased fault tolerance provided by RAID 6 are desired, please note that a performance penalty is observed when compared directly to RAID 5 due to the additional parity

calculation and the extra physical disk required for implementation.

An important consideration when determining the appropriate RAID Level is the physical disk-cost required by a RAID Level. Physical disk-cost, is the number of physical drives worth of capacity that are sacrificed to provide the desired data integrity level. The physical disk-cost of each RAID Level is different, and may impact the decision for which RAID Level is most appropriate for a chosen environment. RAID 0, having no level of redundancy has a physical disk-cost of zero. RAID 1/10 has the highest disk-cost in disk groups containing more than 2 drives. Half of the physical drives in a RAID 1/10 are always consumed for the mirror. RAID 5 has a fixed 1-physical disk per disk group cost, i.e. with a RAID 5 set of n disks, only n-1 worth of capacity is available. Similarly, RAID 6 has a fixed two physical disks per disk group cost, or n-2. In RAID 5 and 6, these extra drives account for the space required to maintain the levels of parity information for each stripe.

Physical disk-cost is not the only factor that influences the decision on which RAID Level is most appropriate for a given application. The performance of a chosen RAID Level is heavily interdependent on characteristics of the I/O pattern as transmitted to the storage array from the host(s). With I/O patterns involving write operations, when an I/O burst exceeds 1/3 of available cache memory in size, it should be considered a long I/O. Long writes show the performance of a chosen RAID Level better than short writes. Short write operations can be handled entirely in cache, and the RAID Level performance effect is minimized. As long as the write-burstiness is always lower than the cache to disk offload rate, a choice in RAID Level can be a non-issue.

In general, the following outlines which RAID Levels work best in specific circumstances:

- RAID 5 and RAID 6 works best for sequential, large I/Os (>256KB)
- RAID 5 or RAID 1/10 for small I/Os (<32KB)
- For I/O sizes in between, the RAID Level is dictated by other application characteristics:
    - RAID 5 and RAID 1/10 have similar characteristics for most read environments and sequential writes.
    - RAID 5 and RAID 6 exhibit the worst performance mostly by random writes.
    - In random I/O applications consisting of more than 10% write operations, RAID 1/10 provides the best performance.

Table 1 provides a summary of these points for an ideal environment. An ideal environment consists of aligned stripes or segments reads and writes, as well as burst I/O where cache memory and RAID Controller Modules are not overly saturated by I/O operations.

Table 1. I/O Size and Optimal Raid Level

	Significantly Random		**Significantly Sequential**	
Block Size	**Read**	**Write**	**Read**	**Write**
Small (<32 KB)	1/10, 5, 6	1/10	1/10, 5, 6	1/10, 5
Medium (Between 32 and 256 KB)	1/10, 5, 6	1/10	1/10, 5, 6	5
Large (>256 KB)	1/10, 5, 6	1/10	1/10, 5, 6	5

Selecting a RAID Level - High Write Mix Scenario

In random I/O applications with a >10% mix of write operations and a low degree of burstiness, RAID

1/10 provides the best overall performance for redundant disk groups.

RAID 1/10 performance can be >20% better than RAID 5 in these environments, but has the highest disk-cost; for instance, more physical disks must be purchased. RAID 5 provides protection and minimizes disk-cost for net capacity but is heavily impacted by the write performance overhead of parity updates.

RAID 6, while providing better protection than RAID 5 with minimal disk cost, is more heavily impacted by write performance overhead for the double parity calculations it requires.

In sequential I/O applications with relatively small write transfer sizes, RAID Level does not make a large difference. With medium transfer sizes, RAID 1/10 can provide an advantage over RAID 5/6, again with a higher associated disk-cost. In very large sequential writes, RAID 5 can perform equal or better than RAID 1/10 especially when factoring in disk-cost for equivalent capacity. Additionally, better performance is always achieved when the application or OS is capable of buffering or coalescing writes to fill a whole segment or stripe.

Selecting a RAID Level - Low Write Mix Scenario

In random I/O applications with a low mix (<10%) of write operations, RAID 5 offers approximately the same performance as RAID 1/10 in small transfer sizes but at a lower disk-cost. RAID 0 provides somewhat better performance than 5 or 1/10 but offers no data protection. In larger transfer size environments, RAID 1/10 performances can be slightly better than RAID 5 but has a much higher disk-cost.

Choosing the Number of Drives in a Disk Group

When optimizing performance, there are many factors to consider, such as drive type, capacity, and the number of drives.

The following general guidelines can be used in grouping drives in a disk group:

- Separate random and sequential workloads on different disks groups – segregating I/O traffic to minimize sharing of disk groups between virtual disks.
- Choose faster drives – in general, a single 15K RPM drive yields approximately 20% more performance than 10K for mixed sequential and random operations. Please refer to the manufacturer's data sheets to better determine the optimally suitable drive.
- Adding more drives to a disk group while holding the stripe size fixed can increase I/O rate for sequential I/O, up to the point of controller saturation – more drives mean more spindles to service I/O.
- To optimize for data transfer rate, multiply the number of physical data disks by the segment size to equal the I/O size. However, there are always exceptions. For small/medium I/Os care should be taken to avoid splitting up the I/O in such a way that an even smaller I/O is sent to the disk drives. Please note that data disks do not include parity or mirror disks used in a RAID set.

For IOPS or transaction-oriented applications, the number of drives becomes more significant because disk drive random I/O rates are relatively low. Select a number of drives that matches the per virtual disk group I/O rate needed to support the application. Make sure to account for the I/Os required to implement the data protection of the selected RAID Level. Make the segment size at least as large as the typical application I/O size.

The term *segment size* refers to the amount of data written to one drive in a virtual disk group before

writing data to the next drive in the virtual disk group. A segment size of 128K is a reasonable starting point for most applications. In most applications, the greater the number of drives in a disk group, the better the average performance. The drive count of an existing disk group can be increased using the MDSM.

Virtual Disk Location and Capacity

The location of virtual disks within a disk group, the number and location of virtual disks allocated within a disk group, and the capacity of a virtual disk are important factors to consider when optimizing an array for performance.

When using rotating storage medium, the capacity of a virtual disk and its location within a disk group greatly impact achieved performance. This is primarily due to differences in angular velocity in the outside zones. The effect of allocating the outermost edges of a rotational storage medium to increase performance is termed *short-stroking* of a drive. While it is beyond the scope of this white paper to cover the technical details involved around short-stroking, typically the outer third of a rotational medium is the fastest, while the inner most zones are the slowest. Short-stroking can easily be accomplished by creating a disk group consisting of a single virtual disk, which is allocated less than a third of the total capacity. The obvious downside to short-stroking a volume is the loss of additional usable capacity. Thus, this performance gain must be weighed directly against the capacity loss.

In addition to the performance gains of short-stroking, the effect of drive head seek should be taken into account when carving up a disk group into virtual disks. Virtual disks are aligned in series within a Disk Group with the first virtual disk starting in the fastest outer regions and progressing inwards. When taking this into account, a disk group should be designed with as few virtual disks as possible.

Dell™ does not recommend using more than four virtual disks or repositories per disk group for peak performance. Additionally, where performance is critical, isolate virtual disks to separate disk groups when possible. When multiple high traffic virtual disks share a disk group, even with purely sequential usage models, the disk group I/O behavior becomes increasingly random, lowering overall performance. Additionally, when a disk group must be shared, the most heavily trafficked virtual disk always should be located at the beginning of a disk group.

Virtual Disk Ownership

The Dell™ MDSM can be used to automatically build and view virtual disks. It uses optimal settings to stripe the disk group. Virtual disks are assigned to alternating RAID controllers when they are created. This default assignment provides a simple means for load balancing the workload of the RAID controllers. Ownership can later be modified to balance workload according to actual usage. If virtual disk ownership is not manually balanced, it is possible for one controller to have the majority of the work, while the other controller is idle.

Limit the number of virtual disks in a disk group. If multiple virtual disks are in a disk group, consider the following information:

- Consider the impact each virtual disk has on other virtual disks in the same disk group.
- Understand the patterns of usage for each virtual disk.
- Different virtual disks have higher usage at different times of day.

Calculating Optimal Segment and Stripe Size

The choice of a segment size can have a major influence on performance in both IOPS and data transfer rate.

A set of contiguous segments spanning across member drives creates a *stripe*. For example, in a RAID 5, 4 + 1 virtual disk group with a segment size of 128KB, the first 128KB of an I/O is written to the first drive, the next 128KB to the next drive, and so on with a total stripe size of 512KB. For a RAID 1, 2 + 2 virtual disk group, 128KB would be written to each of the two drives (and same for the mirrored drives). If the I/O size is larger than this (the number of physical disks multiplied by a 128KB segment), this pattern repeats until the entire I/O is complete.

For very large I/O requests, the optimal segment size for a RAID volume group is one that distributes a single host I/O across all data drives within a single stripe. The formula for maximal stripe size is as follows:

LUN segment size = Maximal I/O Size ÷ number of data drives

A LUN is a logical unit number which corresponds to a storage volume and is represented within a disk group. The LUN segment size should be rounded up to the nearest supported power of two value.

For RAID5 and 6, the number of data drives is equal to the number of drives in the volume group minus 1 and 2 respectively. For example:

RAID5, 4+1 with a 64KB segment size => (5-1) x 64KB = 256KB stripe size

Optimally, this RAID group is sufficient for handling I/O requests less than or equal to 256KB.

For RAID1, the number of data drives is equal to the number of drives divided by 2. For example:

RAID1/10, 2+2 with a 64KB segment size => (4-2) x 64KB = 128KB stripe size

It is important to remember that depending on the application's I/O parameters, segment and strip size will vary.

For application profiles with small I/O requests, set the segment size large enough to minimize the number of segments (drives in the LUN) that are accessed to satisfy the I/O request, that is, to minimize segment boundary crossings. Unless specified otherwise by the application, starting with the default segment size of 128KB is recommended.

It is imperative that the stripe size be correctly selected so that the host operating system is always making requests properly aligned with full stripes or full segments when possible.

Cache Settings

Read-ahead Cache can be configured in the MDSM and through the CLI. Additionally, the global cache block size for Read and Write Cache can be adjusted through the MDSM and the CLI.

Setting the Virtual Disk-Specific Write Cache and Write Cache Mirroring

Configured through the MDSM by right clicking on a selected virtual disk in the **Logical** tab, the following commands are available:

- Write Cache – Disabling Write Cache puts the controllers into a Write-Through mode, adding additional latency while data is flushed to the disk. Except for specific read-only environments, it is recommended that this setting stay enabled. Write Cache is automatically disabled in the case of cache battery failure or a cache battery learn cycle.
- Write Cache Mirroring – Write Cache Mirroring provides an additional level of redundancy and fault tolerance in the MD3200 and MD3200i series of storage arrays. As a side effect, it reduces available physical memory and intra-controller bandwidth to perform this operation. In select, non data-critical cases, it can be beneficial to adjust this parameter. For normal use, Dell™ always recommends Enabling
- Cache Mirroring - Cache Mirroring is automatically disabled in the event of controller failure or when Write Caching is disabled.

***WARNING**: Data loss can occur if a RAID controller module fails while Write-Caching without cache mirroring is enabled on a virtual disk.*

Setting the Virtual Disk-Specific Read Cache Pre-fetch

Also configured via the MDSM by right clicking on a selected virtual disk in the **Logical** tab, this command is available at the virtual disk level.

- Read Cache pre-fetch – The Read Cache setting can be toggled on a virtual disk level. Disabling read pre-fetch is primarily useful in primarily small transfer size random read environments, where pre-fetch of random data would not provide sufficient value. However, the normal observed overhead of read pre-fetch is negligible. For most environments, Dell™ always recommends enabling Read Cache pre-fetch.

Setting the Storage Array Cache Block Size

Configured through the MDSM – This command is available at the storage array level and effects all virtual disks and disk groups.

Cache Block Size refers to the way cache memory is segmented during allocation and affects all virtual disks in an array. On the MD3200 and MD3200i series of storage arrays, settings of 4KB and 16KB are available with 4KB being the default. A dramatic impact on performance can occur by choosing the correct cache block size setting specific to the system's I/O profile. If the typical I/O size is ≥16KB, which is typical with sequential I/O, set the storage array Cache Block Size to 16. For smaller (≤8KB) I/O, especially in highly random or transactional use cases, the default 4KB setting is preferred. As this setting affects all virtual disks on a storage array, changing it should be done with attention to the I/O needs of the application.

Table 2. Storage Array Standard Configuration Specifications

Option	MDSM GUI configuration templates			CLI Options
	File System	**Database**	**Multimedia**	

Drive Type	Selectable	Selectable	Selectable	Selectable
RAID Level	Selectable	Selectable	Selectable	0, 1/10, 5, 6
Segment Size	8KB, 16KB, 32KB, 64KB, 128KB, 256KB,	8KB, 16KB, 32KB, 64KB, 128KB, 256KB,	8KB, 16KB, 32KB, 64KB, 128KB, 256KB,	8KB, 16KB, 32KB, 64KB, 128KB, 256KB,
Write Cache with mirroring	On or off	On or off	On or off	On or off
Read-ahead Cache	On or off	On or off	On or off	On or off
Cache Block Size	Array Defaults to 4KB			4KB, 16KB

Tuning Using Array Performance Data

Collecting Performance Statistics

The **stateCaptureData.txt** and **performanceStatistics.csv** files, which are available through the MDSM, **Support** tab as part of a Technical Support Bundle, provide valuable statistical data in an easy-to-read format. The following section shows some sample data from the **stateCaptureData.txt** file and suggested configuration recommendations based on the performance considerations outlined in the previous section.

Other useful information is available through the array profile. Open the MDSM and select the Support tab – View Storage Array Profile.

Before collecting performance statistics, the I/O workload under test should be first executed. This will ensure the validity of performance statistics as part of the measure step of proper performance tuning.

> ***Note**: Figures shown below are from using the performance tool Iometer.*

RAID Level

The **stateCaptureData.txt** file provides statistics in the read and writes percentage columns to aid in selecting most appropriate RAID Level. In Figure 2, the small reads and writes I/O percentages provide information regarding the distribution of the I/O types in the tested workload. This data is especially helpful when utilizing Table 1 to determine the applications current read/write mix. The RAID Level chosen can impact the I/O performance.

Generally, RAID 1/10 has the best overall performance, with the highest physical disk-cost. Use the I/O percent distribution and the average block size from the collected data to aid in this determination. These fields can be found in the highlighted regions of Figure 2. It should be noted that the values in these figures are in block notation; block size for the specific virtual disk configuration is listed in the **stateCaptureData.txt** file, and is almost always 512 bytes. The average received I/O is not the I/O size the application uses but what the host sends, so while an application may attempt to send larger I/Os, the host's I/O stack may coalesce or break up I/Os as appropriate. Please see the appropriate separate OS or HBA documentation to determine these values.

Figure 1. RAID Level. File: stateCaptureData.txt

```
Volume 0 Attributes:
     Volume Type:       RAIDVolume
     User Label:        MyRAID10_One
...
     BlockSize:         512 bytes
     LargeIoSize:       4096 blocks
...
     Perf. Stats:       Requests     Blocks  Avg. Blks  IO Percent
        Reads           67456452 5943724625         88      71.20%
        Writes          27283249 1144902648         41      28.80%
        Large Reads            0          0          0       0.00%
        Large Writes           0          0          0       0.00%
        Total           94739701 7088627273         74     100.00%
```

I/O Distribution

I/O can be characterized by its distribution and pattern. The two primary factors in determining the I/O distribution of an application are the randomness of I/O and the direction of I/O. The randomness of I/O indicates how sequential or random the data access is, as well as the patterning of this data access. The direction of the I/O can be simply related to the read and write percentages of I/O, that is, the direction I/O is taking from the storage device. I/O pattern refers to how tightly the variance of sequential or random data access is contained within the volume. This can be purely random across an entire virtual disk, or random within some bounds, such as a large file stored within a virtual disk compared to large non-contiguous bursts of sequential data access randomly distributed within some bounds. Each of these is a different I/O pattern, and has a discrete case to be applied when tuning the storage.

The data from the **stateCaptureData.txt** file can be helpful in determining these characteristics. Sequential read percentage can be determined based on the percentage of total cache hits. If the cache hits and read percentage are high, then first assume the I/O pattern tends to be more sequential I/O. However, since cache hits are not broken out statistically by read and write, some variable experimentation may have to be performed with the representative data set if the pattern is unknown. For single threaded I/O host streams, this behavior can be confirmed by comparing the magnitude of reads to read pre-fetch statistics.

In cases where many sequential read operations are expected, enabling read pre-fetch in cache is recommended. If the cache hits percentage is low, the application tends to be more random and read-ahead should be disabled. Midrange percentages possibly indicate bursts of sequential I/O but do not necessarily denote their affiliation to read or write I/O. Again, testing with read-ahead on/off would be required.

Figure 2. Performance Statistics Broken Out. File: stateCaptureData.txt

```
*** Performance stats ***

   Cluster Reads    Cluster Writes       Stripe Reads
         6252626           3015009            5334257
   Stripe Writes       Cache Hits    Cache Hit Blks
         2040493           4685032          737770040
    RPA Requests        RPA Width        RPA Depth
          982036           3932113          418860162
     Full Writes   Partial Writes        RMW Writes
          653386                 29            328612
No Parity Writes      Fast Writes    Full Stripe WT
                 0                 0                 0
```

Stripe Size

For the best performance, stripe size should always be larger than the maximum I/O size performed by the host. As identified previously, stripes should be sized as even powers of two. The average block size can be identified from the collected data. Additionally, I/Os over 2MB are considered large and broken out separately from smaller I/Os in the statistics. While all RAID Level's benefit from careful tuning of stripe and segment size, RAID 5 and 6 with their parity calculations are the most dependant.

The 'Avg. Blks' column (see Figure 4) represents the average the I/O block size encountered. The 'LargeIoSize' field denotes a 2MB size with zero registered large reads or writes during the sample period.

Figure 3. Volume Attributes Performance Stats: taken from RAID 1 volume. File: stateCaptureData.txt

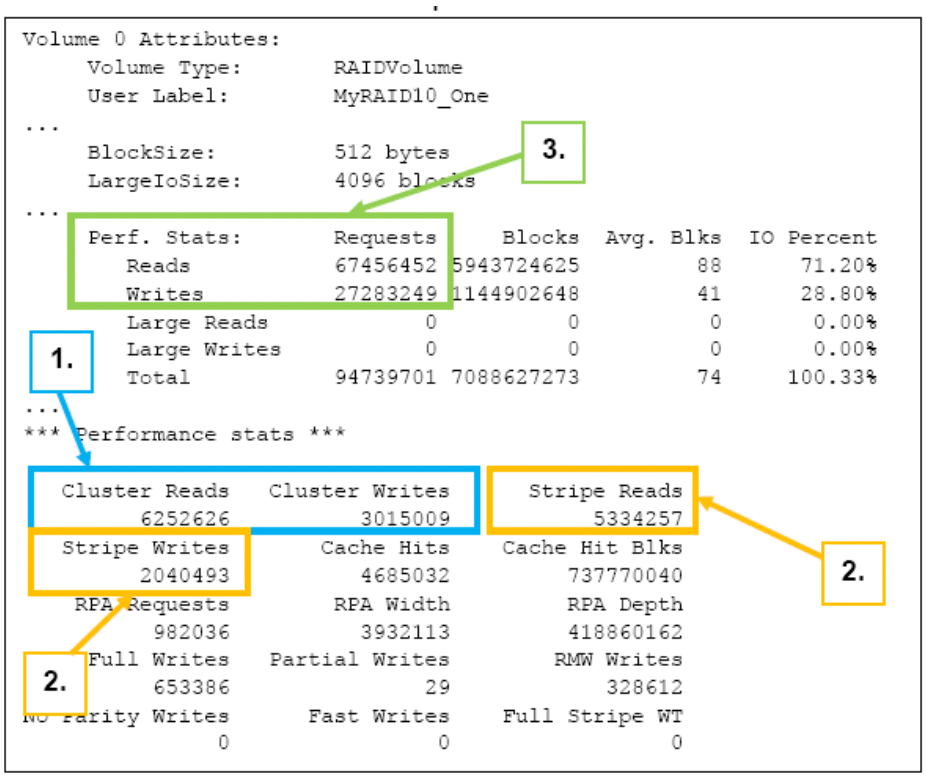
```
Volume 0 Attributes:
     Volume Type:       RAIDVolume
     User Label:        MyRAID10_One
...
     BlockSize:         512 bytes
     LargeIoSize:       4096 blocks
...
     Perf. Stats:       Requests     Blocks  Avg. Blks  IO Percent
        Reads           67456452 5943724625         88      71.20%
        Writes          27283249 1144902648         41      28.80%
        Large Reads            0          0          0       0.00%
        Large Writes           0          0          0       0.00%
        Total           94739701 7088627273         74     100.00 %
```

In addition, the **stateCaptureData.txt** file provides a more granular method for determining the distribution of I/O within stripes and segments. In Figure 5, item 1 is the number of full stripes read and written, and item 2 indicates the number of full clusters or segments read or written. The value of stripes per I/O request within reads and writes is also valuable for determining if stripe or segment settings are optimal for the tested data access pattern. It does not specifically break out the per-I/O ratio in data output, however, it can still be manually calculated simply by dividing the value in item 1

or 2 by the appropriate I/O request value from item 3 indicated in Figure 6.

In most cases, best performance is achieved with segment and stripe per-I/O ratios as close to 1.00. Traditionally, when tuning for maximal I/Os per second, if the segments per-I/O ratio is high, the current segment size may be too low for the application. Similarly, when tuning for highest possible data transfer rate, the stripe per-I/O ratio should ideally be 1.00 or even multiples. If this value is high, increasing the number of physical disks and/or the segment size can improve performance.

Figure 4. Stripe distribution. File: stateCaptureData.txt



Write Algorithm Data

It is important to understand determining the most suitable RAID Level can be a daunting task. Understanding the effect of which write algorithm is in use is an important part of RAID Level balance.

The Full algorithm takes an entire stripe of data and dumps it to the disk, depending on the RAID Level of choice, P or P and Q will get calculated at this point. This is the most efficient type of write that can be performed, and the design of a disk group should be around maximizing full writes.

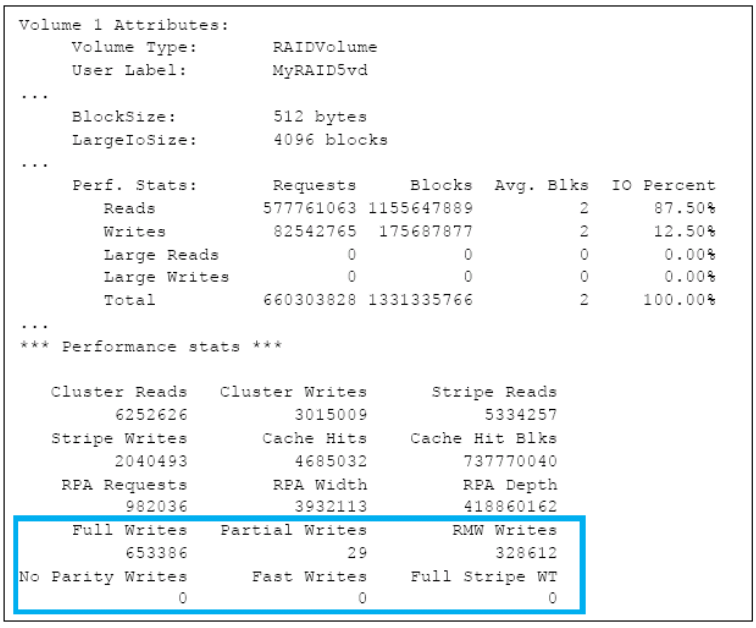
Partial writes are when less than a full stripe of data non-aligned with segment boundaries are modified and written. In RAID Level 5 and 6 this is more complex, as parity data must be recalculated for the whole stripe. Partial writes are a worst-case algorithm and should be minimized. Larger quantities of partial writes than full writes can indicate use of an inappropriate segment size.

RMW, or Read-Modify-Write, is the second-best write algorithm available for RAID 5 and 6. A RMW occurs when a quantity of bits, smaller or equal to an individual segment are modified. This constitutes a two-read operation in RAID 5 and a three-read operation in RAID 6, with one of the segments being modified, and the parity drive(s) are read in. Then parity for the affected region is recalculated and

then the data and parity in the stripe are re-written to disk. In small transactional processing, an extremely large number of RMWs should be expected. These RMW writes can cause a significant loss of performance; however this impact can be lessened by proper tuning of virtual disk stripe size.

RMW2 is used to differentiate Write-to-Cache RMWs and Write-Through RMWs, with RMW2 being the latter, these statistics are consolidated in Figure 6. RMW2's also specifically happen when cache is disabled by force, failed mirroring controller (if policy is active) or failed cache battery. Additionally, it discretely tracks full stripe Write-Through conditions and tracks data on Number of Parity stripes re-calculated.

Figure 5. Write Algorithms



```
Volume 1 Attributes:
     Volume Type:       RAIDVolume
     User Label:        MyRAID5vd
...
     BlockSize:         512 bytes
     LargeIoSize:       4096 blocks
...
     Perf. Stats:       Requests     Blocks  Avg. Blks  IO Percent
        Reads          577761063 1155647889          2      87.50%
        Writes          82542765  175687877          2      12.50%
        Large Reads            0          0          0       0.00%
        Large Writes           0          0          0       0.00%
        Total          660303828 1331335766          2     100.00%
...
*** Performance stats ***

   Cluster Reads   Cluster Writes      Stripe Reads
         6252626          3015009           5334257
   Stripe Writes       Cache Hits    Cache Hit Blks
         2040493          4685032         737770040
    RPA Requests        RPA Width         RPA Depth
          982036          3932113         418860162
     Full Writes   Partial Writes        RMW Writes
          653386              29            328612
No Parity Writes      Fast Writes    Full Stripe WT
                0                0                 0
```

Using the Performance Monitor

Use the Performance Monitor to select virtual disks and RAID controller modules to monitor or to change the polling interval.

Keep these guidelines in mind when using the Performance Monitor:

- The Performance Monitor does not dynamically update its display if any configuration changes occur while the window is open. You must close the *Performance Monitor* window and reopen it for the changes to appear.
- Using the Performance Monitor to retrieve performance data can affect the normal storage array performance depending on the polling interval that you set. For more information, refer to the Learn About Specifying Performance Monitor Settings online help topic

To access the Performance Monitor via the MDSM, use the following steps:

1. *Select **Storage Array >> Monitor Performance**.*

2. *Click **Settings**.*
    a. *Select the items that you will monitor. You can monitor these items:*
        i. RAID controller modules
        ii. Virtual Disks
        iii. Storage array totals
    b. *Select how often you want to update the performance statistics by adjusting the polling interval.*
    IMPORTANT: For an accurate elapsed time, do not use the ***Set RAID controller module Clocks*** option while using the Performance Monitor.
3. *Click **Start**.*

Values appear for the selected storage arrays in the Performance Monitor data table. The table is updated at the interval specified in the ***Polling Interval*** setting. Click ***Update*** to force an immediate poll of the storage array.

4. *Click **Stop** to stop monitoring the storage array*
5. *Click **Save As** on the Performance Monitor main dialog to save the currently displayed performance statistics.*
6. *Select an applicable directory.*
7. *Type a file name in the **File name** text box.*
    NOTE: The *.perf* extension is the default
8. *Select a file type from the **Files of type** list.*

Use the ***Report format (ASCII text)*** file type if you want to save the data to a report form for viewing or printing.
Use the ***Comma Delimited Format*** file type if you want to save the data in a form that can be imported into a commercial spreadsheet application for further analysis. Most leading commercial spreadsheet applications recognize a comma delimiter. These applications use the delimiter to import the data into spreadsheet cells.

9. *Click **Save**.*

Other Array Considerations

Global Media Scan Rate

Use the **Logical** tab to change/set the Media Scan settings in the MDSM. Global Media Scan uses CPU cycles and will affect performance if run at an inappropriate time, for example, during high-use access or during backups.

***Note**: Dell™ recommends turning off media scan for production virtual disks.*

Setting the Virtual Disk-Specific Media Scan

Changing/setting the Media Scan settings in the MDSM is on the Tools tab. To choose to run Media Scan on specific virtual disk, highlight the appropriate virtual disk to be scanned, and select the **Scan selected virtual disks** check box.

Premium Feature Performance

Getting Optimal Performance from Virtual Disk Copy

The Virtual Disk Copy premium feature uses optimized large blocks to complete the copy as quickly as possible. Thus Virtual Disk Copy requires little tuning other than setting the copy priority to the highest level that still allows acceptable host I/O performance. Virtual Disk Copy performance is affected by other controller activity and by the RAID Level and virtual disk parameters of the source virtual disk and the target virtual disk. A best practice for using Virtual Disk Copy is to disable all snapshot virtual disks associated with a source virtual disk before selecting the source virtual disk as a virtual disk copy target volume. Target and source virtual disks should ideally be resident on separate disk groups when possible; keeping them on the same disk group raises the potential for lower performing random I/Os for the copy operation.

Considering the Host Server(s)

Host Hardware Platform

Considering the Server Hardware Architecture

Available bandwidth depends on the server hardware. The number of buses adds to the aggregate bandwidth, but the number of HBAs sharing a single bus can throttle the bandwidth. Additionally, some server hardware has slower-speed PCIE ports (4x) as well as high-speed ports (8x). The Dell SAS6iR HBAs are PCI-E 8x devices and should be installed in 8x slots for maximum performance. Where additional PCI-E slots are available, two SAS HBAs should be used to redundantly attach the I/O host to each storage array controller module to maximize both performance and redundancy.

***Note**: Dell™ provides a bus layout on the lid of all servers. Consult this chart and use a different bus for each HBA installed in the Host.*

Sharing Bandwidth with Multiple SAS HBAs

Each SAS wide port includes four full duplex serial links within a single connector. The individual SAS links run a maximum speed of 6Gb/s. A single path is used as the primary path to a connected device — the second, third, and fourth paths are used as overflow, when concurrent I/Os overload the primary channel. For example, if the first link is transmitting data at 6Gb/s, SAS uses 10-bit encoding versus 8-bit for byte transmission, which makes a single 6Gb/s capped at 600MB/s. If another block of data then needs to be written to disk, for example, and link 1 is still busy, link 2 will manage the overflow of data that cannot be transmitted by link 1. If link 1 finishes its transmission of data, the next block of data will be transmitted on link 1 again, otherwise another link will be used. In this way, for heavy I/O workloads, it is possible that all links are being used at certain times, providing a point-to-point raw wire speed of up to 2.4GB/s. Please note that this raw speed does not take into account the transmission overhead or device operational limits on either side of the SAS wide link and is purely a cached I/O operation.

Additionally, care must be taken of which buses are being used within the host. Installing the HBAs that use the same bus will hamper data transfer rate. Ensure that all HBAs installed in the host are on a different bus.

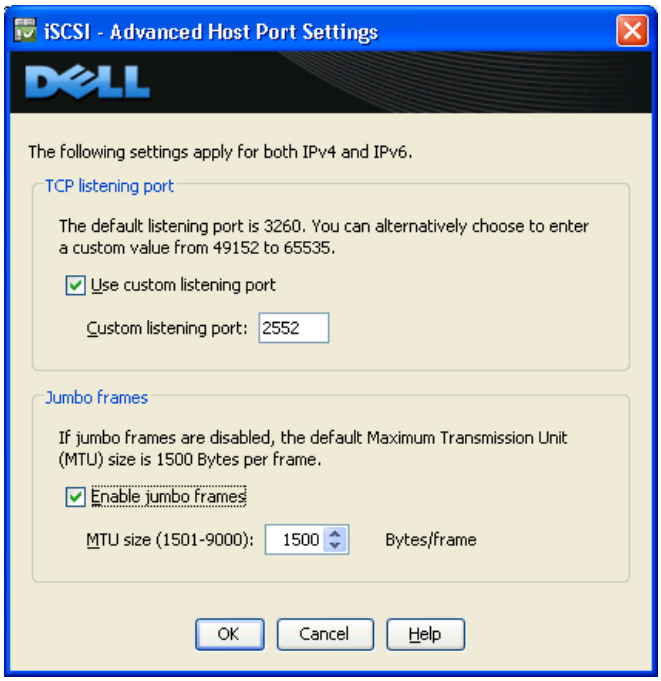
Sharing Bandwidth on the Dell PowerVault™ MD3200i with Multiple NICs

Consider the following when sharing bandwidth on the MD3200i with multiple NICs.

- Give each NIC its own IP address.
- Connect each NIC to a separate switch, or direct attach host NIC(s) to target iSCSI port(s).
- Use separate NICs for public network access and storage array iSCSI traffic. Install additional NICs as needed.
- Ensure the use of separate redundant networks dedicated for iSCSI traffic, if this is not possible, setup a separate VLAN for iSCSI traffic.
- Use Jumbo Frames (Jumbo Frames increase the TCP frame size from 1500 bytes to 9000 bytes).
- The Microsoft iSCSI initiator will *not* work with teamed NICs.
- A single Host cannot mix HBAs and NICs to connect to the same or different arrays.

To edit the Jumbo Frames settings in the MDSM, select the **Setup** tab**, Configure iSCSI Host Ports, Advanced** (see Figure 6). Jumbo Frames can be also be set using the CLI. If using an MD3200i with Jumbo Frames, ensure the Ethernet switches and host NIC(s) also have Jumbo Frames enabled and are set to an equivalent value.

Figure 6. Advanced IPv4 and IPv6 Settings – Jumbo Frames



Considering the System Software

Buffering the I/O

The type of I/O, buffered or un-buffered, provided by the operating system to the application is an important factor in analyzing storage performance issues. Un-buffered I/O (also known as *raw* or *direct* I/O) moves data directly between the application and the drive devices. Buffered I/O is a service

provided by the operating system or by the file system. Buffering improves application performance by caching write data in a file system buffer, which the operating system or file system periodically flushes to non-volatile storage.

Buffered I/O is generally preferred for shorter and more frequent transfers. File system buffering might change the I/O patterns generated by the application. That is, writes might coalesce so that the pattern seen by the storage system is more sequential and more write-intensive than the application I/O itself. Direct I/O is preferred for larger, less frequent transfers and for applications that provide their own extensive buffering, such as Oracle. The I/O performance tool Iometer is another application that can operate un-buffered to test a more raw performance. Regardless of I/O type, I/O performance generally improves when the storage system is kept busy with a steady supply of I/O requests from the host application. Become familiar with the parameters that the operating system provides for controlling I/O, such as maximum transfer size.

Aligning Host I/O with RAID Striping

Most host operating systems require or benefit from varying degrees of partition I/O alignment and avoiding performance-degrading segment crossings. That is, I/Os should not span a segment boundary. Matching of I/O size (commonly, by a power-of-two) to disk group layout helps align I/Os across the entire drive. However, this is only true if the starting sector is properly aligned to a segment boundary. Segment crossing is often seen in the Microsoft Windows operating system, where partitions created by Microsoft Windows 2000 or Microsoft Windows 2003 start at the 64th sector. Starting at the 64th sector causes misalignment with the underlying RAID striping and allows the possibility for a single I/O operation to span multiple segments. Newer versions of the Microsoft Windows OS have a default alignment typically of 2048 blocks that still may require adjustment depending on the environment.

Microsoft provides the diskpar.exe utility as part of the Microsoft Windows 2000 Resource Kit, which was renamed to diskpart.exe in Microsoft Windows 2003 and later. Microsoft has a KB article 929491 covering this, and Dell™ always recommends checking for proper partition alignment to the stripe size of assigned virtual disks. Using these utilities, the starting sector in the master boot record can be set to a value that ensures sector alignment for all I/Os. Use a multiple of 64, such as 64 or 128. Applications such as Microsoft Exchange count on proper partition alignment to the disk stripe boundary.

For Microsoft's usage details about diskpart.exe, go to

http://technet.microsoft.com/en-us/library/aa995867.aspx

***WARNING**: Changes made to the alignment of existing partitions **will** destroy data.*

Appendix

Appendix A: Obtaining Additional Performance Tools

Table 3 shows a number of widely available tools, benchmarks, and utilities. Some of these tools are produced by non-profit organizations and are free.

Table 3. Performance Tools

Name	Description	Platform	Available From
IOBench	I/O data rate and fixed workload benchmark	Unix/Linux	http://www.acnc.com/benchmarks.html
Iometer	I/O subsystem measurement and characterization tool	Unix/Linux	http://www.iometer.org
Xdd	Tool for measuring and characterizing drive subsystem I/O	Windows, Unix/Linux	http://www.ioperformance.com
FIO	Benchmarking and I/O tool	Unix/Linux	http://freshmeat.net/projects/fio/
Bonnie++	I/O benchmark suite	Unix/Linux	http://www.coker.com.au/bonnie++/

Appendix B: Glossary of Terms

Term Definition

- **Burstiness** - A data traffic property defined as the ratio of the peak I/O rate to the average I/O rate; in this case, the mean duty cycle exhibited by I/O transmitted or received from a storage array. Burstiness is adopted from its common usage in describing network workloads.
- **Controller Saturation** - Controller saturation is where an individual RAID Controller Module is reached a maximum operational load and is unable to perform additional operations within its available bandwidth. Additional operations above this maximum point are not lost, they are instead queued up transitionally. This is considered an inflection point where performance hits a hard ceiling.
- **HBA** - Host Bus Adapter
- **HDD** - Hard Disk Drive
- **IOPS** - Input/Output operations per second; Unit of measure in computing benchmarking, quantifies the rate of I/O.

- **iSCSI -** Internet Small Computer Systems Interface. The iSCSI protocol is defined by the IETF in RFC 3720.
- **Long I/O -** Any I/O burst that exceeds 1/3 of the available cache memory size with an increased chance of not being wholly handled in cache.
- **MD3200 -** Dell™ PowerVault MD3200 Expandable Storage Array with SAS front-end.
- **MD3200i -** Dell™ PowerVault MD3200i Expandable Storage Array with iSCSI front-end.
- **MDSM** Dell™ Modular Disk Storage Manager – Host management utility suite for configuring and maintaining a MD3200 and MD3200i storage array.
- **NIC -** Network Interface Controller
- **RAID -** Redundant Array of Inexpensive Disks
- **RAID 0 -** RAID Level 0; RAID 0 is a striped set with no redundant information. It is effectively a fully degraded RAID set with no disk redundancy overhead.
- **RAID 1/10 -** RAID Level 1/10: The RAID 1/10 implementation on the MD3200 and MD3200i following Berkley RAID 1 standard, expanding it to a redundant N+N mirrored set. Functionally, this implementation is set equivalent to a generic nested RAID 1+0 and operates on as few as 2 physical drives. This allows for any number of drive failures as long as one of each drive pair is available at the disk-cost of half the included physical drives.
- **RAID 5 -** RAID Level 5; Involves a block-striping algorithm where n-1 disks per stripe in the raid set contain the data and a parity disk P contains parity or checksum used to validate data integrity provide consistency information. The parity is distributed across all drives in a disk group to provided further fault tolerance. RAID 5 provides protection from single drive failure.
- **RAID 6 -** RAID Level 6; Involves a block-striping algorithm where n-2 disks per stripe in the raid set contain the data and a parity blocks P and Q contains the parity or checksum used to validate data integrity provide consistency information. The parity is distributed across all drives in a disk group to provided further fault tolerance. RAID 6 provides protection from double drive failure.
- **RPA -** Read Pre-fetch Algorithm: An acronym for the read ahead cache used on the MD3200\MD3200i.
- **RMW -** Read, Modify, Write; The second best algorithm available in RAID 5 and RAID 6 write operations. A RMW occurs when a quantity of bits, smaller or equal to an individual segment are modified.
- **RMW2 -** A Firmware Generation One adaptation of RMW specifically for Write-Through conditions where Write Cache is not enabled or requested to a virtual disk.
- **Saturation -** See Controller Saturation
- **SCSI -** Small Computer System Interface, Protocol maintained
- **Segment - T**he data written to one drive in a virtual disk group stripe before writing data to the next drive in the virtual disk group stripe.
- **Short I/O -** Any I/O that consumes less that 1/3 of available cache memory that can be handled within cache, effectively a cached operation.
- **SQL -** Structured Query Language; flexible markup language for computer databases maintained by ANSI and ISO.
- **Stripe -** A set of contiguous segments spanning across member drives.