



Global Distributed Storage Engineering

EMC Symmetrix DMX-3/4 Virtual Provisioning Evaluation

GDSE-08017-E

Evaluation

Version 0.2

1. Version Control

Version	Date	Author	Reviewed	Comments
0.1	May-28-2008	Guy Shimabuku	Quang Le	Peer Review
0.2	May-30-2008	Guy Shimabuku		RFC

2. Legal Notice

Information contained herein is for Internal Use and may be used only for business purposes authorized by Citigroup.

CITI, CITIBANK, CITICORP, CITIGROUP and the Arc Device are trademarks and service marks of Citicorp and are used and registered throughout the world. Citibank, Citicorp, Citigroup and their subsidiaries also claim rights in certain other trademarks and service marks contained in these pages.

3. Contact

Please address any comments or questions regarding the information contained in this document via email to the “*GT Global Distributed Storage Engineering” distribution list or alternatively to gdse@citi.com.

4. Latest Versions

It is recommended that the latest version of the document be used from the [GDSE website](#). Newer versions may include certification on additional platforms, or comments on issues discovered after deployment.

5. Table of Contents:

1. Version Control.....	2
2. Legal Notice	3
3. Contact.....	3
4. Latest Versions.....	3
5. Table of Contents:.....	4
6. EMC Virtual Provisioning	7
6.1. Introduction:.....	7
6.2. I/O Functionality	7
6.3. Volume Table of Contents.....	8
7. Provisioning Recommendations:	9
7.1. Thin Hostile Environments	9
7.2. Thin Friendly Environments	9
7.3. General.....	9
7.4. Storage Pool	10
7.5. Data Migrations	10
7.6. RAID-6 Data Protection	11
7.7. Capacity Utilization	11
7.8. Capacity Oversubscription.....	11
7.9. Metavolume Device Restrictions.....	11
8. Test Summary Highlights:	12
8.1. Throughput Performance	12
PROS:	12
CONS:.....	12
8.2. Input/Output Alignment Performance	12
PROS:	13
CONS:.....	13
8.3. DATADEV Drain Performance.....	13
PROS:	13
CONS:.....	13
8.4. Storage Pool Capacity Utilization.....	14
PROS:	14
CONS:.....	14
8.5. Storage Pool Alerting.....	14
PROS:	15
CONS:.....	15
8.6. Storage Pool Data Security	15
PROS:	15
CONS:.....	15
Circumvention of Virtual Provisioning Objectives	15
PROS:	16
CONS:.....	16

9. Fileserver & Database Simulation and Non-Linear Performance Test.....	16
9.1. File Server Simulation:	19
9.1.1. Description:	19
9.1.2. Test Configuration:	19
9.1.3. Observations:	19
9.1.4. Summary Performance Graphs:	19
9.1.5. Performance Summary Tables:	20
9.2. Database Simulation	21
9.2.1. Description:	21
9.2.2. Test Configuration:	21
9.2.3. Observation:	21
9.2.4. Performance Summary Tables:	23
9.3. Non-linear Performance Load Test:	24
9.3.1. Description:	24
9.3.2. Test Configuration:	24
9.3.3. Observations:	24
9.3.4. Throughput Performance Summary Graphs:	25
9.3.5. Latency Performance Summary Graphs:	29
9.3.6. Throughput Performance Summary Table:	32
9.3.7. Latency Performance Summary Table:	39
9.4. Draining and Degraded Mode Performance	47
9.4.1. Description	47
9.4.2. Test Configuration and Process	47
9.4.3. Observation	47
9.4.4. Conclusion	48
9.4.5. Summary Performance Graphs:	48
9.4.6. Summary Performance Table:	49
9.5. Input/Output Alignment	50
9.5.1. Description:	50
9.5.2. Test Configuration	51
9.5.3. Observations:	51
9.5.4. Conclusion:	52
9.5.5. Summary Graphs:	53
9.5.6. Summary Tables:	55
9.6. Pre-allocation Capacity Throughput Test:	58
9.6.1. Description:	58
9.6.2. Test Configuration:	58
9.6.3. Observations:	58
10. Functional Tests:	60
10.1. Storage Pool Capacity Alerting	60
10.1.1. Description:	60
10.1.2. Observation:	61
10.1.3. Procedure: 70% Fill Threshold Violation	62
10.2. Storage Pool Capacity Utilization	64
10.2.1. Description:	64

10.2.2.	Observation:	64
10.2.3.	Conclusion:	64
10.3.	RAID-6 Data Device Sparing	67
10.3.1.	Description:	67
10.3.2.	Observation:	67
10.4.	Storage Pool Data Distribution	68
10.4.1.	Description:	68
10.4.2.	Observations:	68
10.5.	Circumvention of Virtual Provisioning Objectives	69
10.5.1.	Description:	69
10.5.2.	Observation:	69
10.5.3.	Conclusion:	70
10.5.4.	Data Definition Low-level Copy Test.....	70
10.6.	Storage Pool Data Security	72
10.6.1.	Description:	72
10.6.2.	Observations:	72
10.6.3.	Conclusion:	72
10.6.4.	Data Accessibility Test:	72
10.7.	Modifying Allocated Pool Space	74
10.7.1.	Description:	74
10.7.2.	Observation:	75
10.7.3.	Increase Pre-allocated Capacity Test.....	75
10.7.4.	Freeing Pre-allocation Capacity Test.....	76
11.	Appendix A: Basic Thin Provisioning Commands:	78
12.	Appendix B: General SYMCLI:	79
13.	Appendix C: References	85

6. EMC Virtual Provisioning

6.1. *Introduction:*

EMC's Virtual Provisioning (EVP) product is based on traditional thin provisioning functionality, which uses thin devices presented to the host while consuming physical storage from a shared storage pool. EVP is a feature set introduced with Enginuity 5773 Microcode and Solution Enabler 6.5 for the DMX-3 and DMX-4 storage arrays. EVP provides three distinct advantages over standard provisioning methods:

- **Improves storage capacity utilization** by presenting an application with sufficient storage and allowing the capacity to increase dynamically, thereby allowing the application to mature in parallel with its storage needs.
- **Supports oversubscription** when the presented capacity that is cumulatively advertised exceeds the physical amount of storage available in the storage pool.
- **Increases throughput performance** by striping the data over all storage devices in the thin pool.

EVP utilizes a storage pool to group virtual devices called Thin Devices (TDEV) with the actual storage volumes called Data Devices (DATADEV). The TDEVs function as a typical standard Symmetrix device; however, the TDEV doesn't require any allocation of physical storage. The DATADEVs are created from physical storage that adheres to the same configuration requirements as standard Symmetrix devices (note: at the time of this evaluation EVP supported mirror and Raid-6 protection types). The EVP storage pool sets the oversubscription limits, monitors the allocation status and manages the relationship between the TDEVs and DATADEVs. When the storage pool requires additional storage, new DATADEVs can be created and associated with existing storage pool.

6.2. *I/O Functionality*

EVP widely stripes the data across all available DATADEVs within the storage pool, typically spanning a much greater number of physical spindles than standard device provisioning methods. This distribution of workload across the DATADEVs maintains a consistent level of performance by reducing the number and frequency of hot spots. Random read and write workloads that benefit from wide data striping will see the most performance improvement. Sequential workloads that do not benefit greatly from wide striping will see roughly comparable performance with standard devices, although performance will vary by workload and configuration.

In a write request, the Symmetrix allocates a minimum amount of physical storage from the thin pool. The pool's storage is allocated sequentially, regardless of whether data is written sequentially or randomly to the TDEV. This allocation is performed in small units of storage called "device extents" that are 12 tracks (768 KB) in size. A round-robin mechanism is used to balance the allocation of device extents across all of the DATADEVs in the pool.

A read request retrieves the appropriate data from the DATADEVs utilizing pointers to locate the requested information in the storage pool. Read requests to an area that the thin device that has not been mapped will returned a block where each byte is equal to zero. Reading from an unmapped area will not result in any new storage allocations.

6.3. Volume Table of Contents

Instance Volume Table of Contents (IVTOC) is the low-level formatting of a device and is the underlying processes for the implementation of virtual provisioning on the Symmetrix array. IVTOC utilizes a 2-step process of a logical and physical VTOC to prepare and format storage devices. When storage is allocated to a thin device the logical VTOC process creates a table with the TDEV and DATADEV relational information. When a write request is submitted to the thin device a physical VTOC formats the needed capacity prior to writing the new data.

The write response times for thin devices will be higher than standard devices, due to the thin device first write and partial-track first write penalty. The first write process is where a write is performed to an area of a thin device that has not yet been mapped, at which time the logical VTOC process allocates an extent and updates the information table (IVTOC state). The partial-track first write penalty occurs when the first write fails to completely fill a track (64KB) requiring an additional step by disk adapter to pad the remaining portion of the unfilled track with zeros.

When a TDEV is unbound from the storage pool, the relational information created during the IVTOC process is removed and the storage capacity is returned to the pool as available storage. The original data remains on the DATADEVs following the unbind process; however, retrieving the original data is tremendously remote since the data is widely striped across the storage pool. In addition, the released DATADEV capacity is marked as unmapped requiring a VTOC prior to accessing.

7. Provisioning Recommendations:

7.1. *Thin Hostile Environments*

- Refrain from file systems that do not reuse the space associated with deleted files, resulting in unused space and reducing or eliminating the capacity utilization benefits of EVP.
- Avoid applications that pre-allocate or pre-write their entire storage capacity space by writing zeros or a specific fill words to the available capacity. This pre-allocated space cannot be shared in a thin pool; therefore, reducing or eliminating the capacity utilization benefits.

7.2. *Thin Friendly Environments*

- EVP is a natural fit to ensure that premium flash drive resources are highly utilized. Flash drives can offer both unparalleled response times as well as significant I/Os/GB, and, unlike traditional drives, throughput and response time do not steadily decline as utilization of the drive increases.

Note: Flash drives for the Symmetrix DMX-4 will be evaluated in the 3Q08 timeframe
--

- EVP is appropriate for applications for where some performance variability can be tolerated. Workloads with a high amount of random I/O will likely see performance improvements from device-level wide striping.
- Document and software development/source code repositories, which can tolerate some level of performance variability, will benefit in the storage utilization efficiencies of thin provisioning.

7.3. *General*

- Target **no** higher than 60 to 80 percent capacity utilization per storage pool, leaving a sufficient buffer for unexpected growth or “runaway” application(s) that consume more physical capacity than was originally planned. At a minimum there should be enough free space in the storage pool to cover the largest unallocated thin device. A DATADEV that is disabled or removed requires draining the data to the remaining DATADEVs in the storage pool before the device can be disabled or removed.
- Use concatenated metavolumes when configuring TDEVs larger than 60 GB. Each underlying member of a thin concatenated metadvice is

widely striped across all devices within the pool mitigating the benefits of a striped metavolume configuration. The advantage of concatenated TDEV metavolume is the ability to increase the size of the device without destroying the existing data.

- The DATADEV sizes should be selected to minimize the number of devices required to encompass the desired overall pool capacity. Therefore, DATADEVs should be created using the largest individual volume size of 65520 cylinders (approximately 60GB). DATADEVs do not support metavolumes.
- Segregate applications with incompatible performance demands onto distinct thin device pools.

7.4. Storage Pool

- Capacity expansions should be done with the largest number of DATADEVs as possible. Therefore, DATADEVs should be added to existing thin pool in increments equal to the number of backend loops on the array. For example, a DMX-4 2500 with 16 backend loops should add DATADEVs by increments of 16 (i.e. 16, 32, 48, 64...). Expanding an existing full device pool will essentially concentrate all activity for any new allocations on the newly added devices.
- DATADEVs should be evenly spread across the DAs and drives. The devices in the pool should use an equal amount of space from each drive. The use of sequential device ID's aids in selecting distributed devices on the backend.
- DATADEVs in a pool should all reside on drives with the same rotational speed, and capacity.
- All DATADEVs of a given capacity should be evenly distributed, allowing the round-robin extent allocation policy to balance the load across the array's backend resources.

7.5. Data Migrations

- Host based data aware migration applications provide the most efficient data migration mechanisms. Data aware application refers to transferring the actual data only, as oppose raw block level transfers that send the entire device regardless of actual data utilization.

- No array based device-level copy mechanism is available in the initial EVP release, that is applicable for copying from a standard to a thin device without causing the thin device to become fully allocated.

7.6. RAID-6 Data Protection

- RAID 6 provides the highest level of protection among supported RAID protection types; therefore, minimizing the possibility of data loss in the storage pool. The downside to EVP's wide data striping of the storage pool, is that a single drive failure may corrupt the majority of the TDEVs in the pool.
- Drive rebuild time impacts the availability and performance of the entire storage pool since it is likely that each TDEV will have space allocated on many or every DATADEV in the pool.

7.7. Capacity Utilization

- The initial write to an unallocated device extent adds latency. If predictable response time is required at all times, then pre-allocating the thin device will eliminate some of the subsequent overhead. Pre-allocation is specified during the bind operation and reserves storage space within the pool.
- Pre-allocating capacity is not be appropriate for environments where utilization gains from over-subscription are more important than optimal performance. Pre-allocation reserve extents, whereby reducing the first write penalty.

7.8. Capacity Oversubscription

- Capacity utilization of oversubscribed pools must be monitored to determine when space must be added to the thin pool to avoid out-of-space conditions. Management and monitoring is accomplished via Solutions Enabler and the Symmetrix event log.
- Applications hostile to thin environments should use thin pools with little or no oversubscription. Applications or data managers that pre-write dense patterns across the entire thin device will cause all the associated storage to become allocated.

7.9. Metavolume Device Restrictions

- DATADEVs cannot form metavolumes.

- Metavolumes cannot be formed by combining thin devices and regular devices.
- Metavolumes cannot be formed using thin devices that don't all belong to the same pool.

8. Test Summary Highlights:

8.1. *Throughput Performance*

The fileserver/database simulations and the throughput performance test results are largely determined by the access times and the command queuing performance of the array, rather than bandwidth bottlenecks. As a means to closely simulated server applications, access patterns have been designed to step through several levels of queue depths and transaction sizes. Virtual devices incorporate device-level wide striping to distribute the data across all DATADEVs, whereby obtaining a consistent high-level of through performance. According to EMC, virtual storage has a 10-15% performance advantage for OLTP type workloads over standard devices.

The performance tests conducted in this evaluation compared an equivalent number of standard devices to DATADEVs, which somewhat nullifies the wide striping advantage of EVP. EVP's wide striping benefits from large storage pools, whereby distributing the I/O across all resources available to the pool.

PROS:

- Virtual devices showed highly consistent and predictable throughput at all queue depths tested.
- The virtual devices showed similar throughput performance as transaction sizes approached 32KB and larger.

CONS:

- The throughput test showed standard devices with a slight performance advantage over virtual devices at the small and medium transaction sizes (2-16KB).
- The fileserver/database simulation showed standard devices with a 7-20% throughput advantage over virtual devices.

8.2. *Input/Output Alignment Performance*

I/O alignment specifies the relation between the host transaction size and device partitioning within the array. The proper I/O alignment matches the application

transaction size with the partition sizes on the storage array, whereby reducing the number of partition crossings. Partition crossings increase write latencies by increasing the number of requests placed on the backend of the array. I/O alignment has the greatest effect on random writes, where misalignment may cause multiple reads and writes for each write request on the backend.

PROS:

- Virtual devices showed performance equivalent to standard devices for larger transaction sizes across all alignment sizes tested.
- EMC indicated that future Enginuity versions will provide the user with the capability to adjust the size of the extent, whereby allowing storage pools to be tuned to specific host applications. This ability to tune storage pools for specific applications provides an avenue to tier storage based on application alignment instead of physical disks.

CONS:

- Standard devices showed a significant performance advantage with smaller transaction (<16KB) sizes and alignment sizes less than 64KB. As alignment sizes increase the performance between thin and standard devices equalizes.

8.3. DATADEV Drain Performance

DATADEV(s) can be removed from an existing storage pool; however, this requires that the device be disabled and any allocated space on the device “drained” to the remaining DATADEVs and unmapped from the TDEVs. Only on completion of the drain process can the DATADEV be removed from the storage pool.

PROS:

- DATADEVs can be removed from the storage pool without any data discrepancies.

CONS:

- During the draining process the storage pool showed a **98% reduction in throughput** as compared to normal operation. The draining process has an extreme negative effect on performance which points to the Symmetrix placing a high I/O priority on the draining process. A high priority on the draining process is understandable from the array perspective, since it impacts the integrity of the existing data within the thin pool; however, an equitable balance must be established between the draining and typical I/O within the storage pool. Until EMC develops an equitable I/O sharing capability, disabling DATADEVs should be avoided.

- Removing DATADEVs from a storage pool has a detrimental affect on the distribution of drained data and newly written data to the remaining DATADEVs. The removal of a DATADEV seems to drain the majority of its data to closest neighbor in terms of logical device identifier. The unequal distribution of data within the pool may nullify EVP's performance attributes of the device-level wide striping.

8.4. Storage Pool Capacity Utilization

EMC bases the capacity alerting thresholds on the DATADEV percent full value; therefore, the allocated capacity within the storage pool must be measured accurately. EVP uses thin device extents of 12 tracks (786,432 bytes) each, which is the minimum amount of data that is allocated from the pool when a write occurs to an unmapped area.

PROS:

- EMC provides the capability to view both the DATADEV and TDEV utilization with a single CLI command.

CONS:

- The use of extents for the data writes makes it difficult to determine with any certainty the accuracy of the percent fill value. An initial write request will allocate a full extent of 786,432 bytes to the storage pool; therefore, in a worst case scenario (high number of random writes) the storage pool will depict a percent fill that is 11 times higher than actual. The probability of this capacity misrepresentation is greater with storage pools with a low amount of allocations, versus a highly allocated storage pool where much of the initial extents unused portion is written over with valid data.

8.5. Storage Pool Alerting

Of primary importance for thin provisioning is the ability to monitor the consumptions of pool storage space. EMC recommends that storage pools not surpass 80% of capacity allocation. A certain amount of free space is required to handle "runaway" host applications and/or draining of DATADEV(s). The Symmetrix places alert messages into the Symmetrix log at the following fixed consumption levels:

- 60% - issue a warning alert
- 65% - issue a warning alert
- 70% - issue a warning alert
- 80% - issue a critical alert
- 100% - issue a fatal alert

PROS:

- The storage pool capacity alert messages are properly placed into the Symmetrix event log in the form of informational or error alert message.

CONS:

- Symmetrix event log requires the Symmetrix Event Daemon (storevntd) on the Solutions Enabler servers to be operational. Therefore, the collecting of alert messages into the Symmetrix log relies on the stability of the Solutions Enabler server and the application daemons being operational.

According to EMC, surpassing a capacity consumption level does not warrant a call event. Currently, working with our EMC representatives on a process to alert Citi when an 80% fill is achieved on a storage pool.

8.6. Storage Pool Data Security

The data written to the TDEVs are mapped into the DATADEVs and pointers are used to define the mapping locations. When a TDEV is unbound from the storage pool the mapping to the DATADEVs is also being removed rendering the information on the DATADEVs unintelligible.

PROS:

- The unbinding process removes the pointer relationship between the TDEVs and DATADEVs, thereby dispersing and rendering the original data nearly unintelligible. Unbinding of the TDEVs maybe sufficient to declare the data as irretrievable, which may negate the use of an erasure process when reassigning TDEVs or decommissioning the array.

CONS:

- If the unbinding process is not followed and the TDEV is re-used by masking to another host, the original data written to the TDEV is compromised. Therefore, it is imperative that data on the TDEV be erased or be made unintelligible by the unbinding process.

Circumvention of Virtual Provisioning Objectives

The primary objective of EVP is to provide storage on a demand basis, thereby allowing an application to mature along with its storage needs. However, loopholes exist that allow the user community to circumvent EVP's primary objective by simulating storage demands, whereby providing the false illusion that the entire TDEV capacity was required.

PROS:

- The CLI provides the ability to view allocated status to an individual TDEV. This allows a micro-level view of the storage pool to locate any TDEV that is inexplicably increasing allocation at a rapid rate.

CONS:

- End-users may take advantage EVP's capability of providing storage on demand by using null or substitution bytes to reserve the entire TDEV storage capacity. This misappropriation of virtual storage maybe accomplished by low-level formatting, "dd" (data definition) a disk or creating dummy substitution files to fill-in the device. If the majority of the end-users utilize these types of reservation techniques, the storage pool would be quickly allocated.

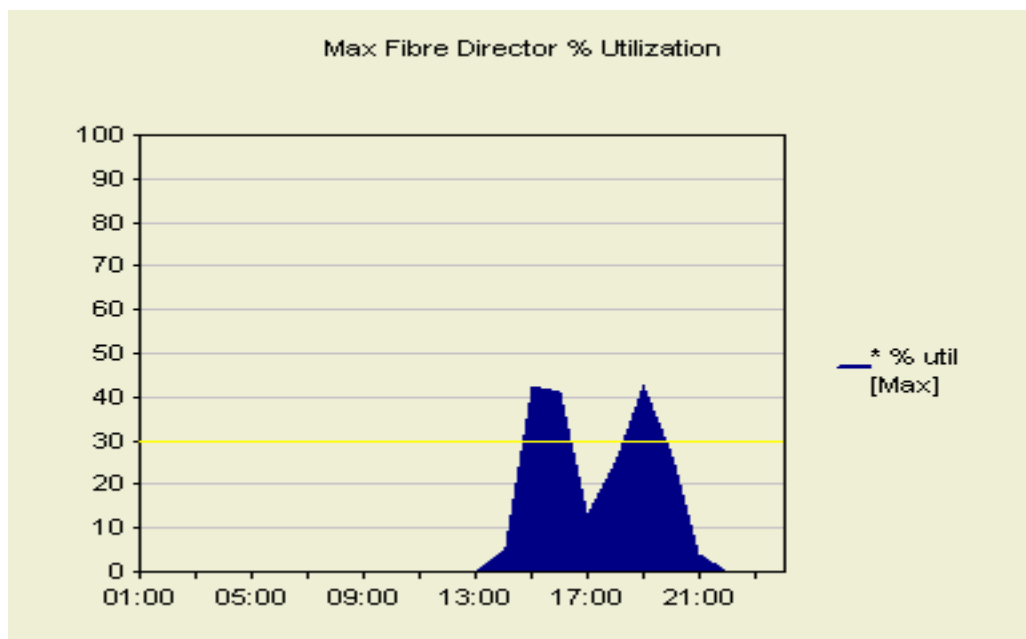
9. Fileserver & Database Simulation and Non-Linear Performance Test

The fileserver and database performance simulation models contain access patterns with a degree of randomness and distribution of transfer sizes created from practical benchmarks. The outcome of the simulations is largely determined by the access times and the command queuing performance of the array, rather than bandwidth bottlenecks. As a means to closely simulate server applications the access patterns will step through several levels of queue depths and transaction sizes. Adjusting the queue depth determines the number of outstanding requests or threads used in the aggregate load test. A queue depth of 1 is an extremely linear load and is not representative any real-world access patterns, while a queue depth of 4 represents the most rudimentary of operations such as loading up an MSWindows calculator. As depicted in the following table, queue depths of 16 – 256 more closely simulates how a fileserver and database application will perform.

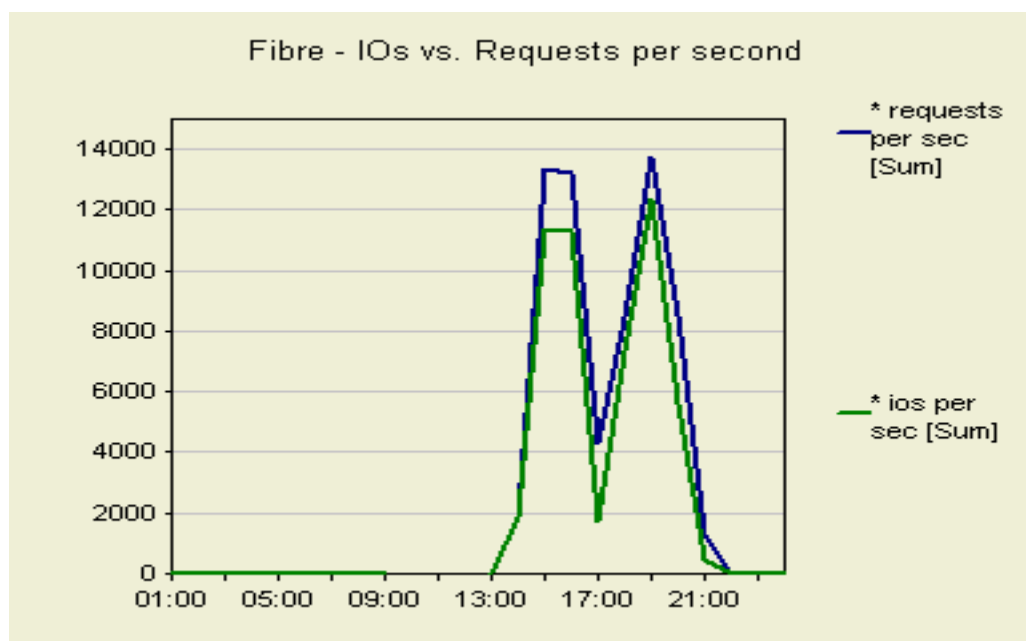
Loads	
Linear	1 Outstanding I/O
Very Light	4 Outstanding I/Os
Light	16 Outstanding I/Os
Moderate	64 Outstanding I/Os
Heavy	256 Outstanding I/Os

IOmeter will be used as the primary I/O generator for this evaluation. IOmeter is a flexible workload modeling tool that allows specific access patterns to be created with sequential/random mixtures and varying transaction sizes. The EMC Performance Manager (e.g. WLA) graphs depicted below, shows the load

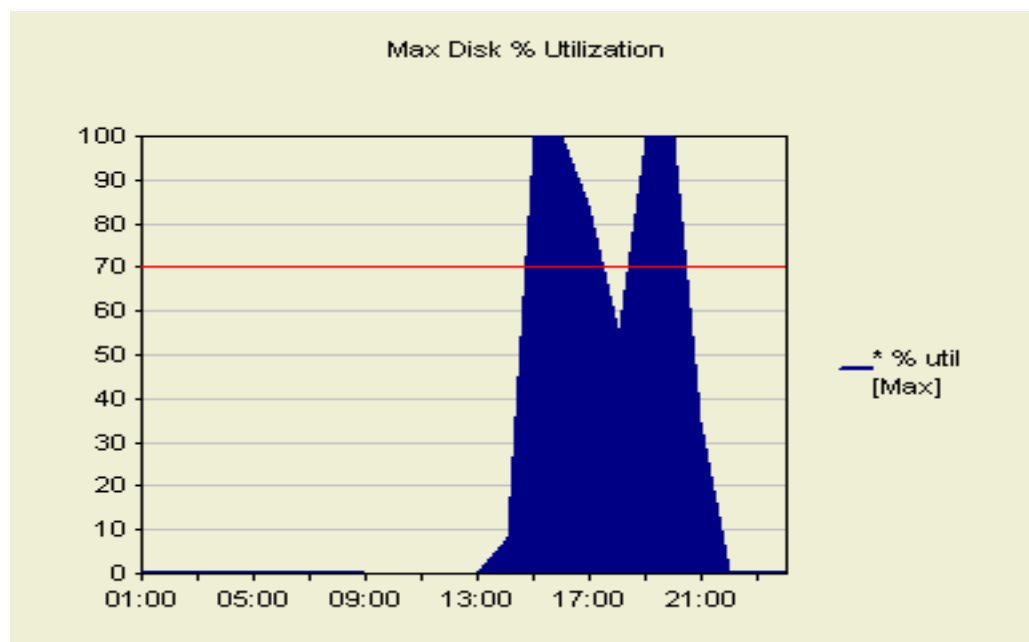
IOmeter placed on the array (see hours 13:00-22:00). Graph 3 shows that various disks involved with the performance tests were at or near maximum utilization, while graph 4 shows Write Pending (WP) reaching 435,000 of the available 721,434 WP slots. This is evident that the IOmeter test configuration places sufficient I/O load to analyze and deduce how an array will perform under load.



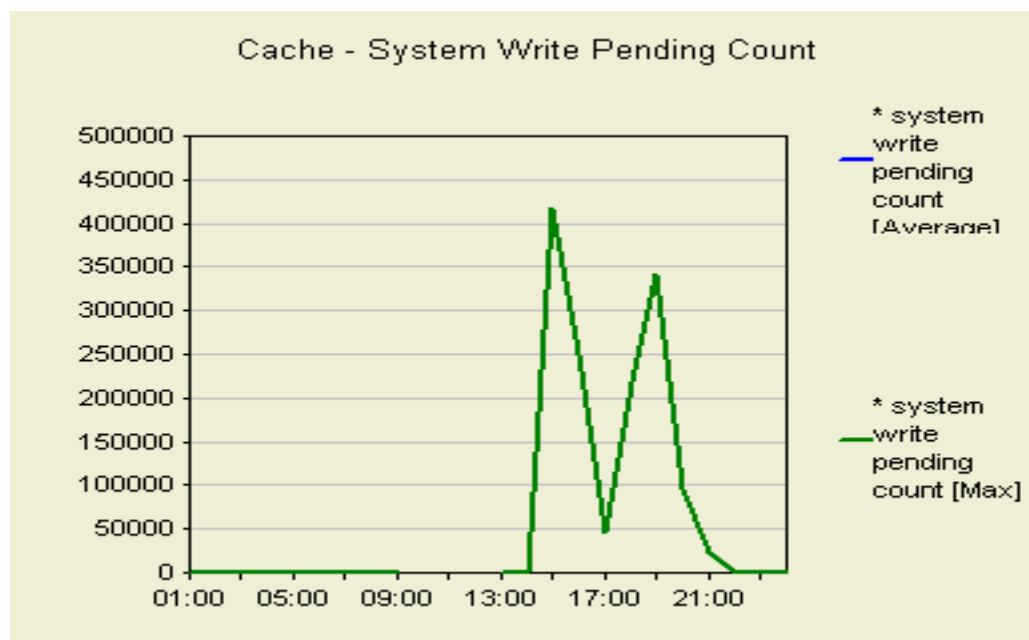
Graph1



Graph 2



Graph 3



Graph 4

Citi is moving toward Raid-6 for the majority of storage assignments, due to the increase protection from data loss and I/O performance through wider stripe distribution on the backend of the array. All performance tests have been accomplished with Raid-6 14+2 volumes at 65742 cylinders on 300GB 15K RPM

drives. The 65472 cylinders volume translates to 61360MB, which uses a hypervolume size of 4384MB across all disks within the Raid group rank.

9.1. File Server Simulation:

9.1.1. Description:

The fileserver simulation consists of a 100 percent random access pattern with an 80/20 percent read/write proportion and transfer sizes starting at 512 bytes and going up to 64KB. 4KB transfers are occurring 60 percent of the time, 512 bytes 10 percent and 64KB 10 percent. The test is performed at gradually increasing queue lengths of 1, 2, 4, 8, 16, 32 and 64.

9.1.2. Test Configuration:

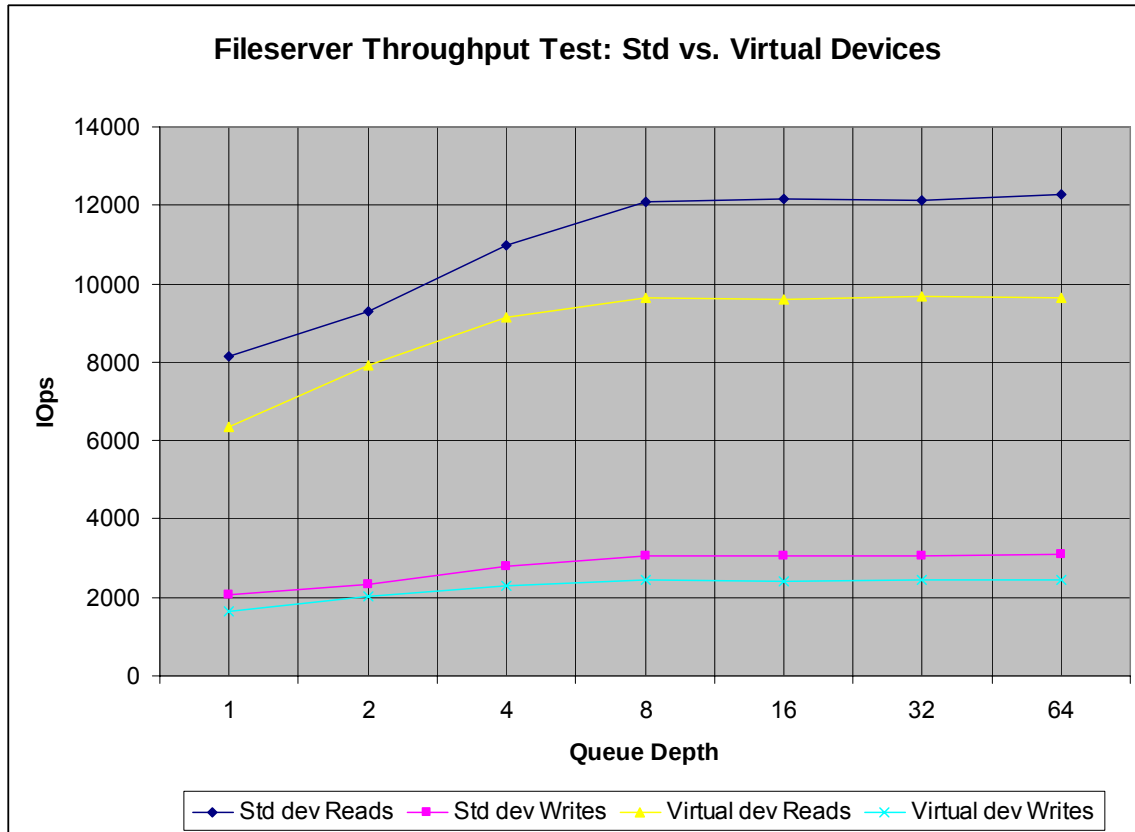
- Solaris 10 server with 4 processors
- 16 standard devices, Raid-6 14+2, 65472 cylinders in size each
- Thin device pool:
 - 16 Raid-6 14+2 DATADEVs at 65472 cylinders
 - 16 TDEVs
- I/O generation specifications:

'size	% of size	% reads	% random	delay	burst	align	Reply
512	10	80	100	0	4	0	0
1024	5	80	100	0	4	0	0
2048	5	80	100	0	4	0	0
4096	60	80	100	0	4	0	0
8192	2	80	100	0	4	0	0
16384	4	80	100	0	4	0	0
32768	4	80	100	0	4	0	0
65536	10	80	100	0	4	0	0

9.1.3. Observations:

- Standard devices showed an average throughput advantage of 19.7% for reads and 19.5% for writes over virtual devices.
- Both standard and virtual throughput values showed consistent plots with increasing an increasing slope from 1 – 8 queue depths and flat lining between 8 and 64 queue depths.

9.1.4. Summary Performance Graphs:



9.1.5. Performance Summary Tables:

Fileserver Standard Devices								
Queue Depth	# Disks	IOPS	Read IOPS	Write IOPS	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time
1	32	10210.40	8146.72	2063.68	87.831	22.297	3.5311	1.5553
2	32	11627.19	9278.74	2348.45	100.606	25.259	6.1670	2.8669
4	32	13747.08	10971.12	2775.96	119.084	30.118	9.9984	6.5392
8	32	15135.48	12079.04	3056.44	130.716	33.274	17.4718	14.6370
16	32	15257.31	12180.09	3077.22	131.524	33.170	33.8006	32.1691
32	32	15199.48	12134.42	3065.06	132.069	33.202	67.6495	65.3915
64	32	15374.84	12275.26	3099.57	132.922	33.368	132.0358	130.5177
Average		13793.11	11009.34	2783.77	119.250	30.098	38.6649	36.2395

Fileserver Virtual Devices								
Queue Depth	# Disks	IOPS	Read IOPS	Write IOPS	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time

1	32	7963.88	6331.53	1632.35	68.355	17.543	4.4699	2.2552
2	32	9930.89	7921.26	2009.64	85.655	21.898	6.9082	4.5912
4	32	11447.79	9145.28	2302.51	99.437	25.218	11.4116	10.2242
8	32	12062.79	9627.24	2435.56	104.168	26.290	21.1567	21.4519
16	32	12003.20	9583.43	2419.77	103.980	26.381	42.5307	43.0725
32	32	12118.03	9665.31	2452.72	104.373	26.359	84.0321	84.9293
64	32	12062.46	9629.73	2432.73	103.857	26.171	168.3827	169.1216
Average		11084.15	8843.40	2240.75	95.689	24.266	48.4131	47.9494

9.2. Database Simulation

9.2.1. Description:

The database simulation consists of a 100 percent random access pattern with a 67/33 percent read/write ratio with 2KB and 8KB transfer sizes. The test is performed at gradually increasing queue lengths of 1 through 256 in powers of two increments.

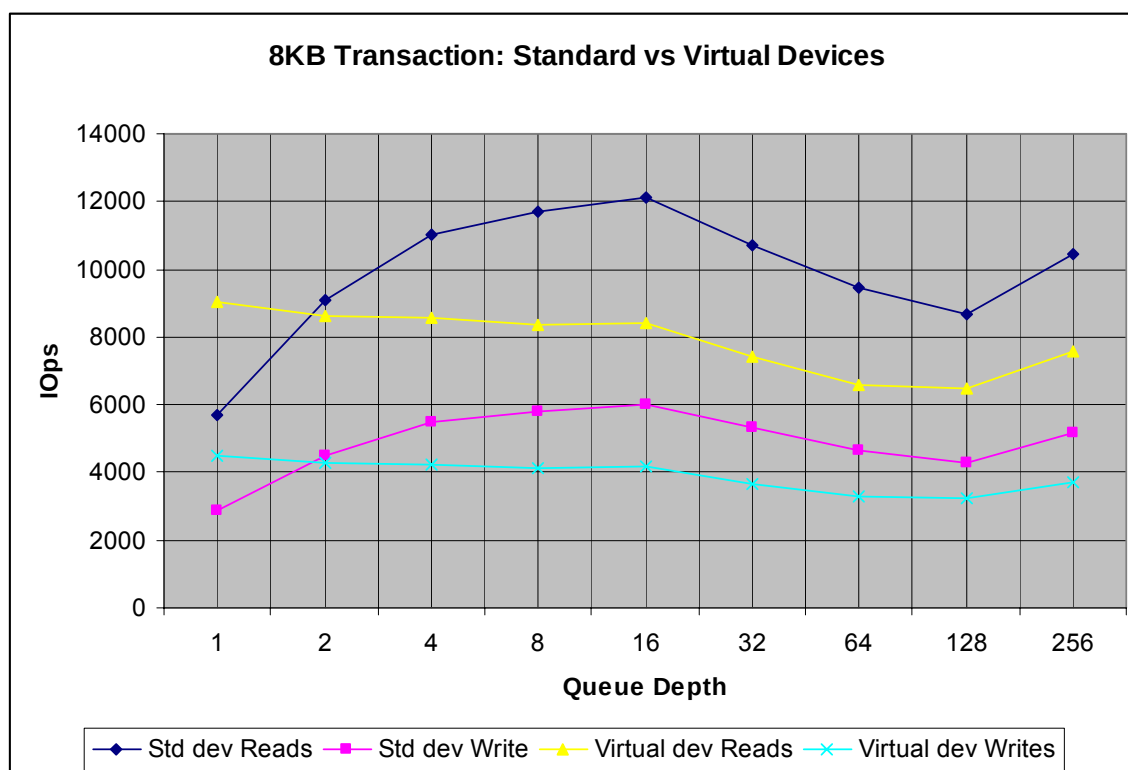
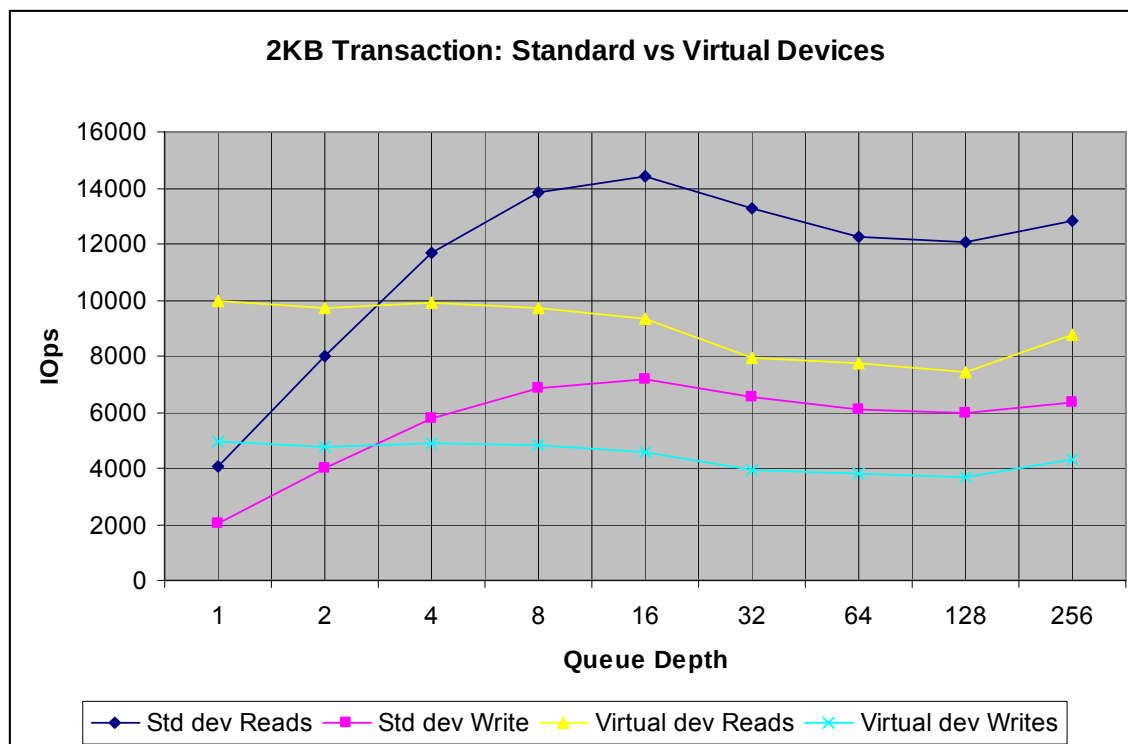
9.2.2. Test Configuration:

- Solaris 10 server with 4 processors
- 16 standard devices, Raid-6 14+2, 65472 cylinders in size each
- Thin device pool:
 - 16 Raid-6 14+2 DATADEVs at 65472 cylinders
 - 16 TDEVs
- I/O generation specifications:

'size	% of size	% reads	% random	delay	burst	align	reply
2048	100	67	100	0	4	0	0
8192	100	67	100	0	4	0	0

9.2.3. Observation:

- The standard devices showed a 13% throughput advantage for reads and writes at 2KB transaction size, and 7% for reads and 12% writes for 8KB transaction sizes over virtual devices.
- The standard devices show a wide dispersion between queue depths as evident by the 41% greater standard deviation over virtual devices for 2KB transactions and 8% for 8KB transactions.
- Virtual devices showed an extremely flat throughput curve that averaged a low standard deviation 961 for reads and 470 for writes for 2KB and 8KB transactions.



9.2.4. Performance Summary Tables:

Standard Devices								
Queue Depth	Access Specification Name	IOps	Read IOps	Write IOps	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time
1	OLTP 2K	6142.26	4079.452	2062.808	7.96768	4.028922	7.411869	0.846387
2	OLTP 2K	12000.11	8009.793	3990.314	15.64413	7.793581	6.91914	2.138696
4	OLTP 2K	17508.31	11705.41	5802.899	22.86213	11.33379	8.577537	4.73579
8	OLTP 2K	20704.31	13842.42	6861.89	27.03599	13.40213	13.06687	10.90267
16	OLTP 2K	21556.10	14407.49	7148.612	28.13963	13.96213	23.55477	24.13196
32	OLTP 2K	19844.01	13275.83	6568.172	25.92936	12.82846	51.34886	51.95085
64	OLTP 2K	18342.95	12267.61	6075.346	23.96017	11.86591	111.3322	111.7474
128	OLTP 2K	18066.99	12074.55	5992.437	23.58311	11.70398	219.0853	218.594
256	OLTP 2K	19208.94	12854.02	6354.913	25.10552	12.41194	289.0809	289.1186
Average		17041.55	11390.73	5650.821	22.24752	11.03676	81.15306	79.35182
1	OLTP 8K	8562.89	5692.86	2870.03	44.47547	22.42211	5.10006	1.025379
2	OLTP 8K	13575.79	9075.154	4500.637	70.89964	35.16122	5.58948	2.92473
4	OLTP 8K	16532.37	11038.59	5493.772	86.23902	42.9201	8.267987	6.667582
8	OLTP 8K	17529.76	11715.3	5814.455	91.5258	45.42543	14.80613	14.13761
16	OLTP 8K	18144.19	12135.76	6008.426	94.81066	46.94083	28.22	28.14665
32	OLTP 8K	16049.91	10727.7	5322.208	83.81015	41.57975	63.74448	63.76153
64	OLTP 8K	14129.55	9460.694	4668.855	73.91167	36.47543	144.4581	144.6405
128	OLTP 8K	12950.54	8667.4	4283.137	67.71406	33.46201	311.171	313.9607
256	OLTP 8K	15620.07	10444.23	5175.842	81.59556	40.43627	372.55	372.7866
Average		14788.34	9884.189	4904.151	77.22023	38.31368	105.9897	105.339

Virtual Devices								
Queue Depth	Access Specification Name	IOps	Read IOps	Write IOps	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time
1	OLTP 2K	14879.27	9955.096	4924.179	19.44355	9.617536	2.147725	1.960864
2	OLTP 2K	14499.37	9707.552	4791.823	18.96006	9.359029	4.410698	4.015332
4	OLTP 2K	14786.92	9901.84	4885.076	19.33953	9.541164	8.652682	8.295158
8	OLTP 2K	14519.9	9723.518	4796.377	18.99125	9.367924	17.62667	17.24398
16	OLTP 2K	13923.89	9322.442	4601.443	18.2079	8.987194	36.76737	36.40242
32	OLTP 2K	11838.82	7924.469	3914.349	15.47748	7.645214	86.42263	86.046
64	OLTP 2K	11611.57	7775.364	3836.206	15.18626	7.49259	176.2547	176.1412
128	OLTP 2K	11131.76	7454.388	3677.369	14.55935	7.182362	355.9054	354.9588
256	OLTP 2K	13059.65	8739.783	4319.869	17.06989	8.437244	384.4231	384.6084
Average		13361.24	8944.939	4416.299	17.47058	8.625584	119.179	118.8525
1	OLTP 8K	13515.19	9047.03	4468.155	70.67993	34.90746	2.364829	2.264506
2	OLTP 8K	12901.13	8630.986	4270.139	67.42958	33.36046	4.957697	4.836477
4	OLTP 8K	12812.86	8572.431	4240.431	66.97212	33.12836	9.985963	9.919672
8	OLTP 8K	12502.64	8367.275	4135.364	65.36934	32.30753	20.47021	20.3471

16	OLTP 8K	12555.66	8400.655	4155.006	65.63011	32.46098	40.77057	40.65433
32	OLTP 8K	11058.85	7405.08	3653.772	57.85219	28.54509	92.55065	92.45117
64	OLTP 8K	9869.972	6589.739	3280.233	51.48234	25.62682	207.5182	207.2527
128	OLTP 8K	9711.467	6480.475	3230.991	50.62872	25.24212	402.4909	401.9595
256	OLTP 8K	11284.9	7555.847	3729.056	59.03005	29.13325	429.8772	429.888
Average		11801.41	7894.391	3907.016	61.67493	30.52356	134.554	134.397

9.3. *Non-linear Performance Load Test:*

9.3.1. Description:

The performance throughput test unlike the simulation test will run through each transaction size with equal I/O specifications. This test provides both positive and negative trends in how an array handles sequential and random I/O load. The load test alternates between sequential and random access patterns with a 70/30 read to write ratio and transfer sizes of 2KB - 128KB. The test is performed at gradually increasing queue depths of 1 through 256 in power of two increments.

9.3.2. Test Configuration:

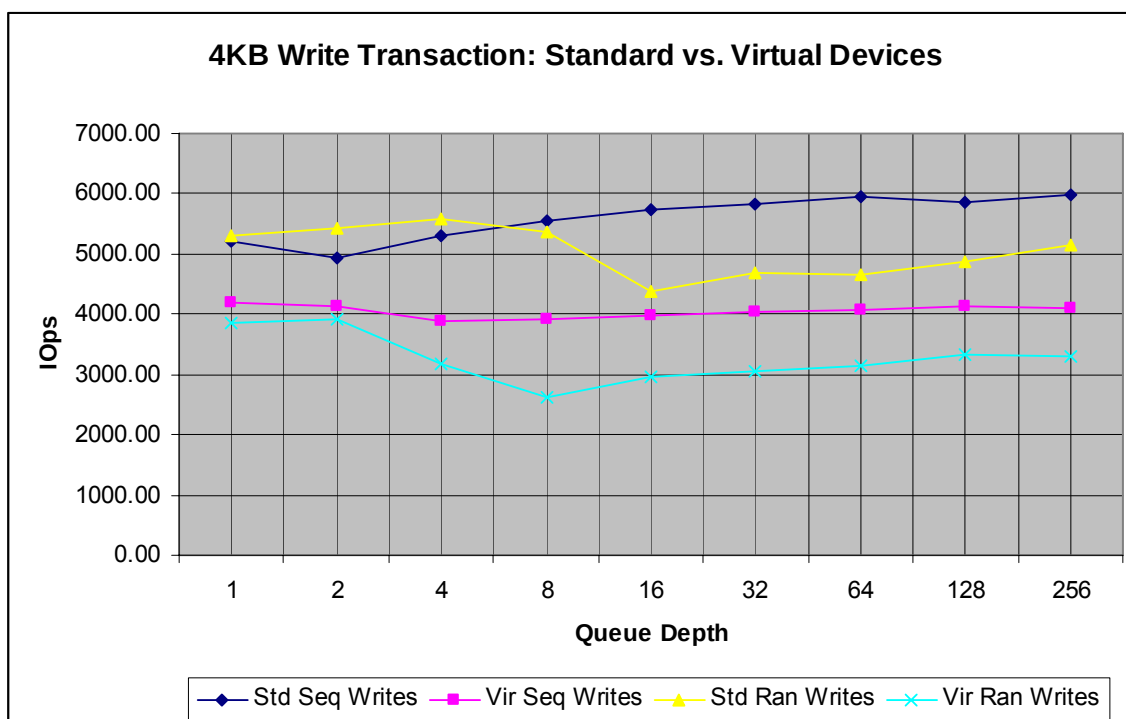
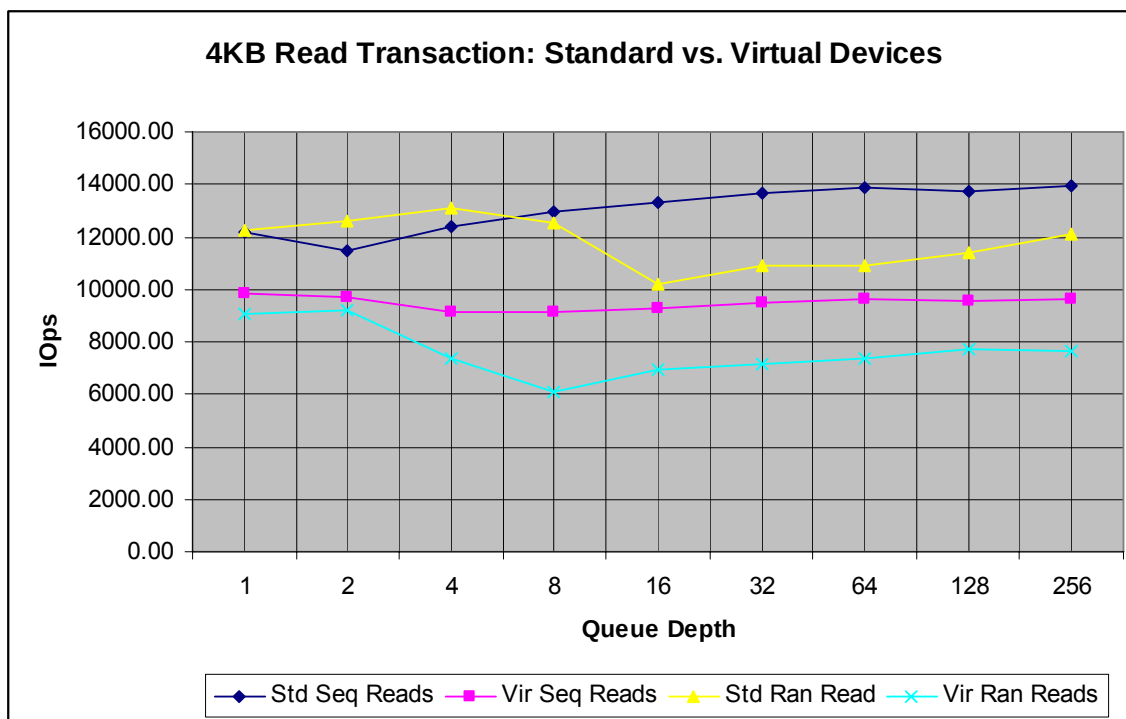
- Solaris 10 server with 4 processors
- 16 Raid-6 14+2 devices at 65472 cylinders form the same symdisk group
- Thin device pool:
 - 16 Raid-6 14+2 DATADEVs at 65472 cylinders
 - 16 TDEVs
- Standard device pool:
 - 16 Raid-6 14+2 devices at 65472 cylinders
- I/O generation specifications:
 - 10sec ramp with 60sec sampling per test instance
 - Transaction size – 2KB, 4KB, 8KB, 16KB, 32KB, 64KB, 96KB and 128KB
 - Queue depth 1 -256, step in power of 2
 - Random and sequential I/O per transaction size

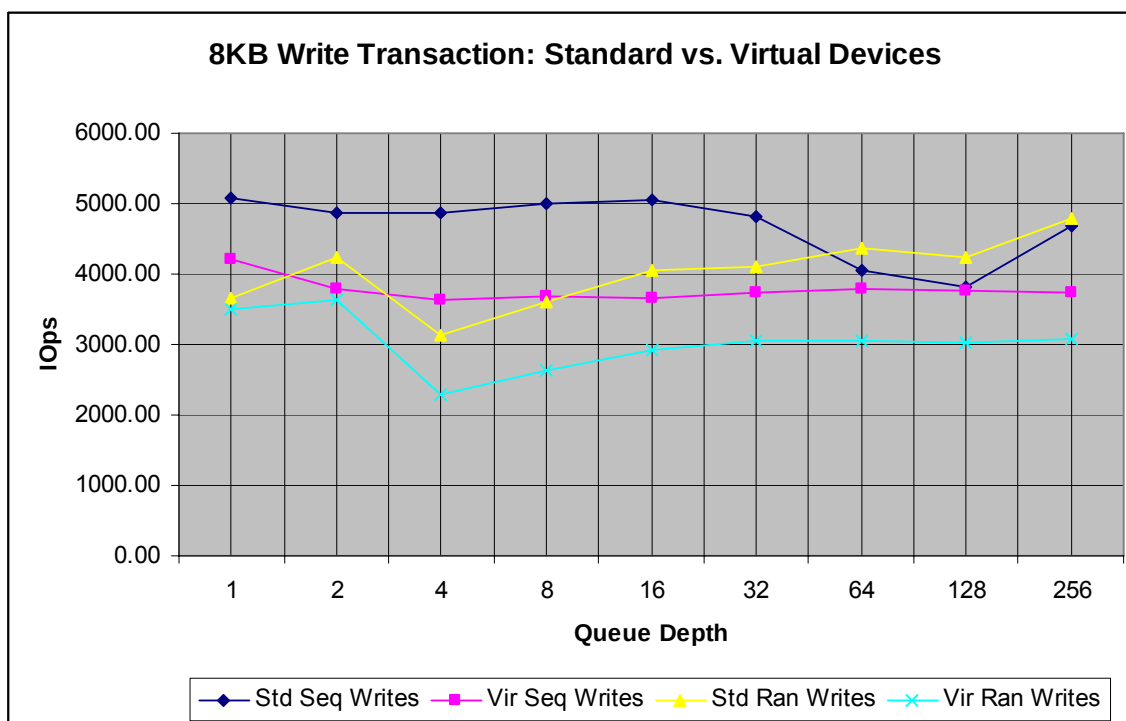
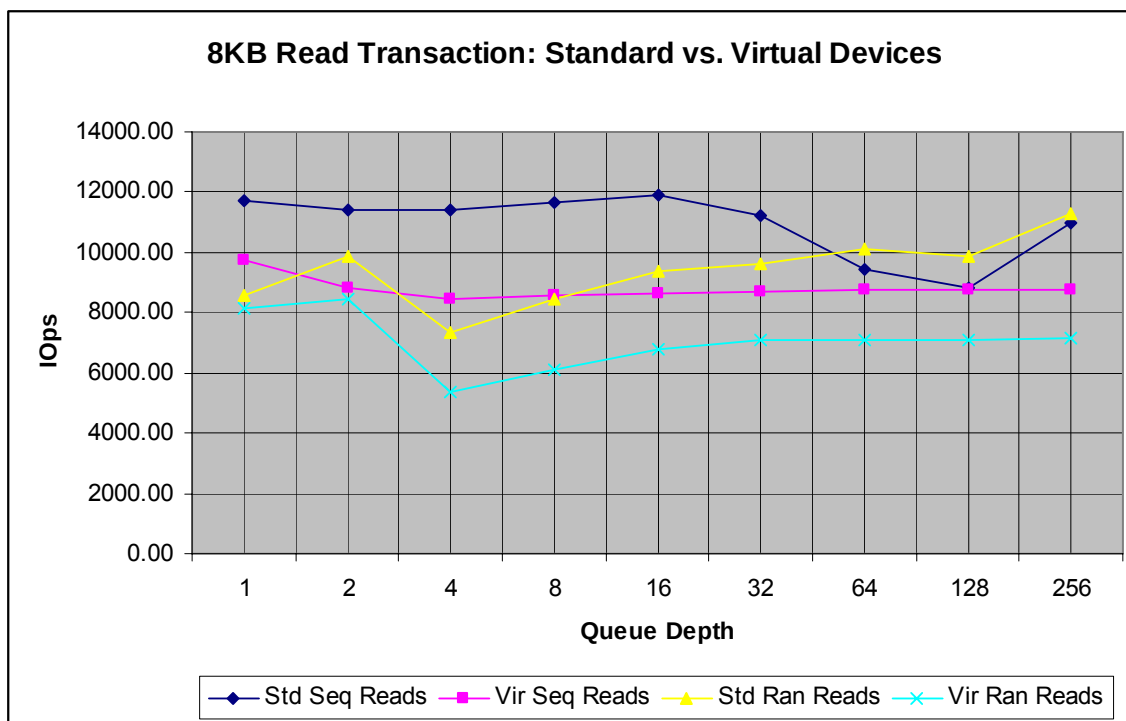
9.3.3. Observations:

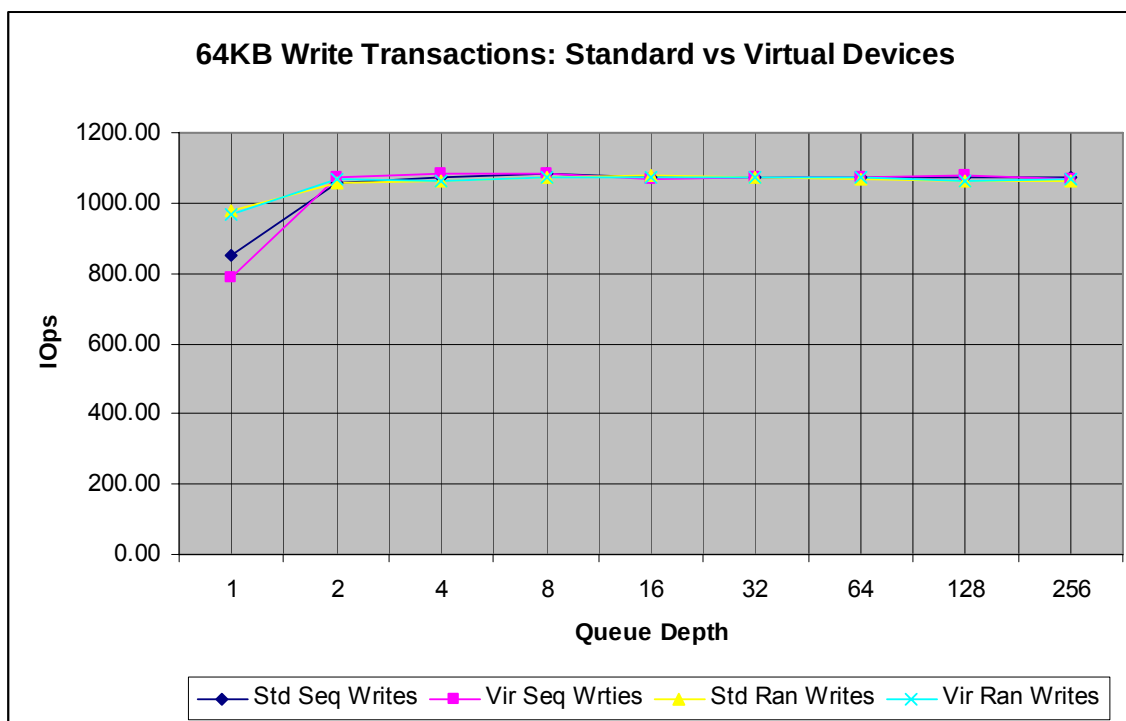
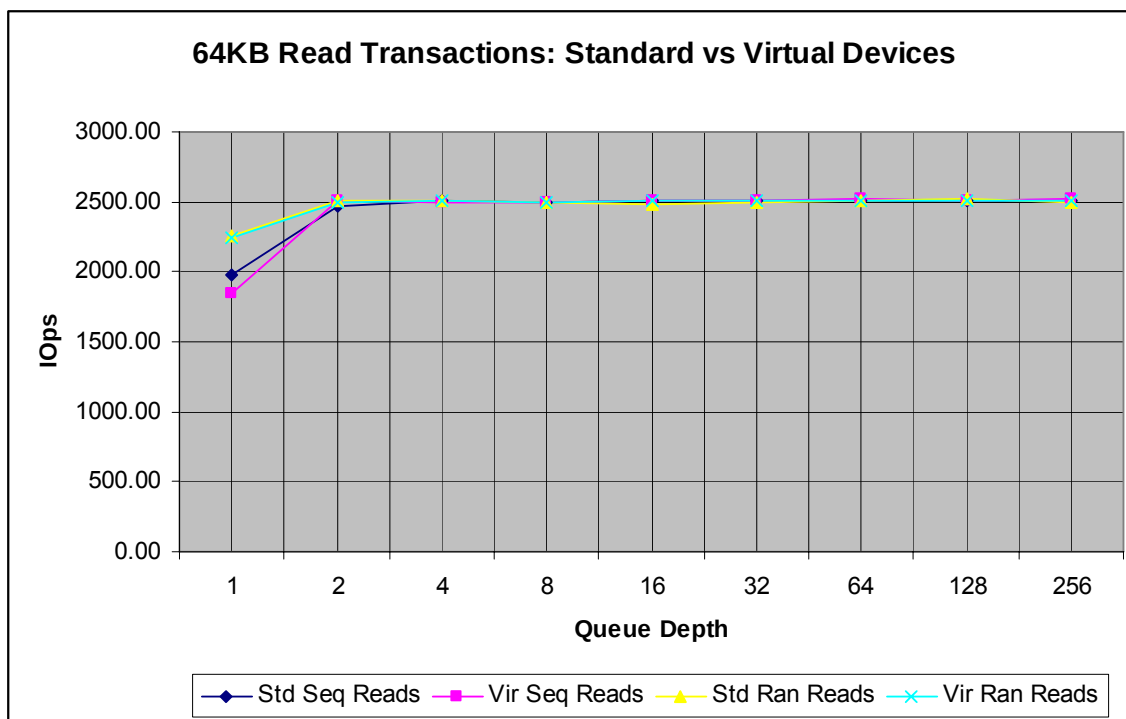
- The 4KB transaction shows standard devices with a 27% sequential and 35% random throughput performance advantage over virtual devices.
- The 64KB transaction shows standard and virtual devices with nearly the same throughput performance.

- The 4KB transactions latency tests shows standard devices with an 11.4% sequential and 9.5% random read advantage over virtual devices.
- The 4KB transactions latency test shows standard devices with an 11.3% sequential and 15.2% random write latency performance advantage over virtual devices.
- The 8KB sequential read and write transaction latency test shows virtual devices with a 19% advantage over standard devices.
- The 8KB random read and write transaction latency test shows standard devices with a 13% advantage over virtual devices.
- The 64KB transaction latency tests shows virtual devices with a 20.9% sequential and 32.4% random read advantage over standard devices.
- The 64KB transaction latency tests shows virtual devices with a 20.9% sequential and 31.6% random read advantage over standard devices.
- Removal of the 256 queue depth results, the 64KB transactions latency performance tests shows standard devices with a 0.10% sequential read advantage, while the virtual devices showed a 2.6% advantage in random reads. These percentage differences are considered negligible.
- Removal of the 256 queue depth results, the 64KB transactions latency tests shows virtual devices with a 0.18% for sequential writes and 2.1% random write advantage over standard devices. These percentage differences are considered negligible.

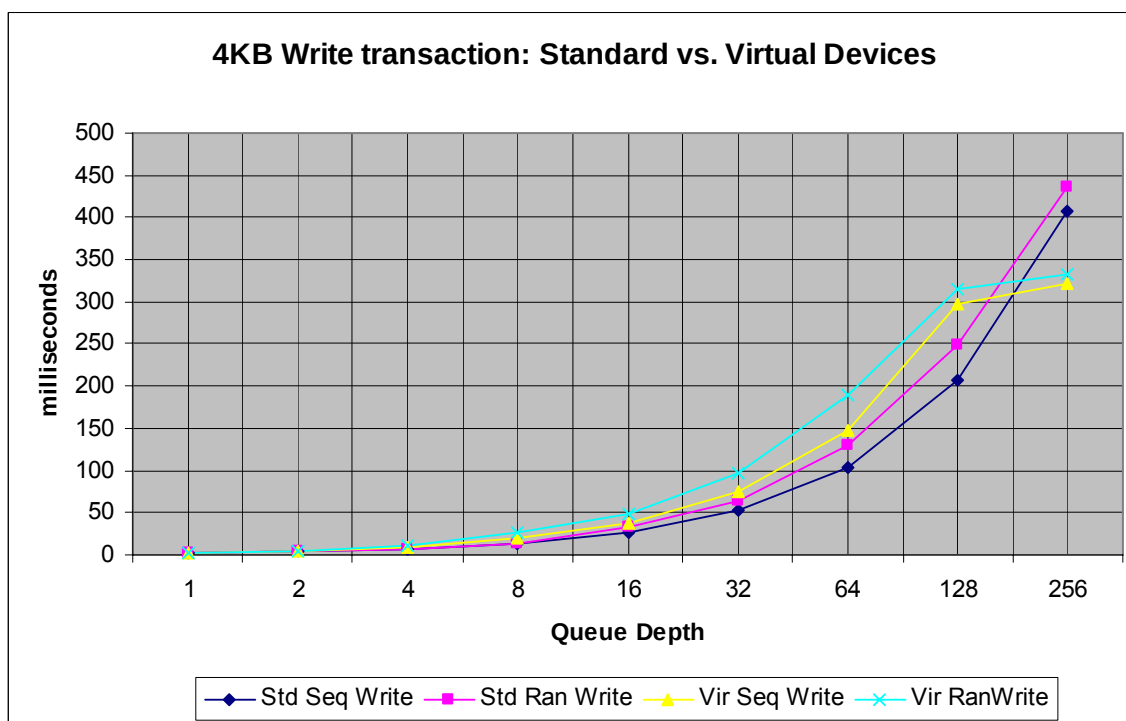
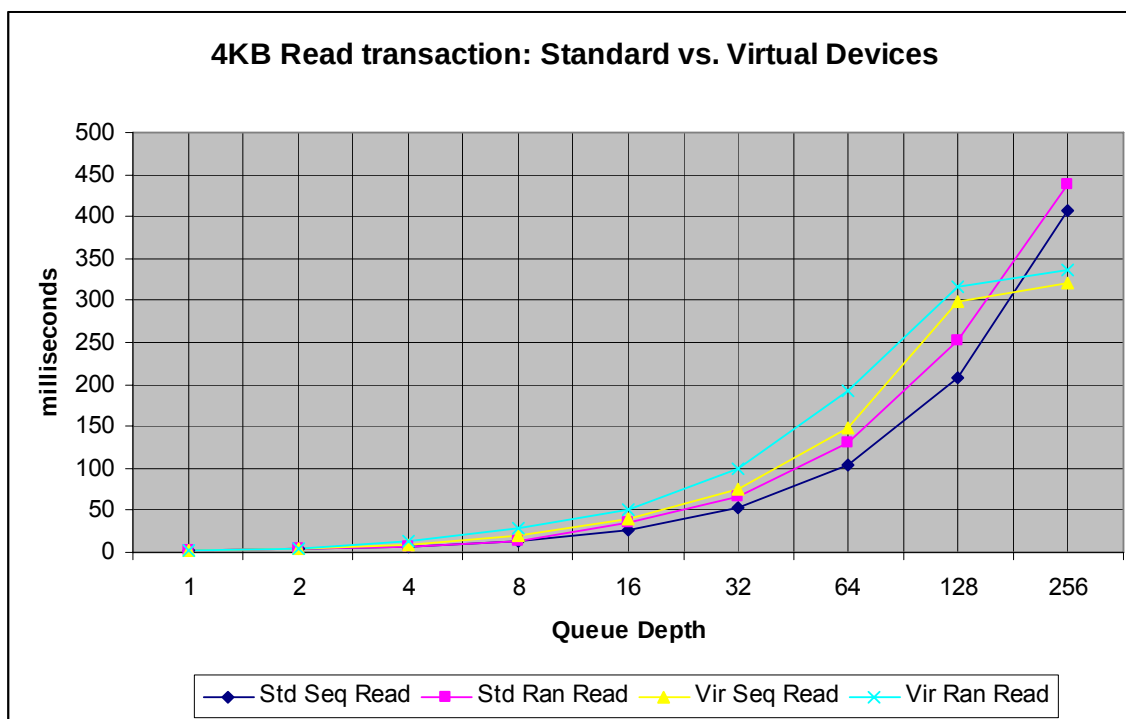
9.3.4. Throughput Performance Summary Graphs:

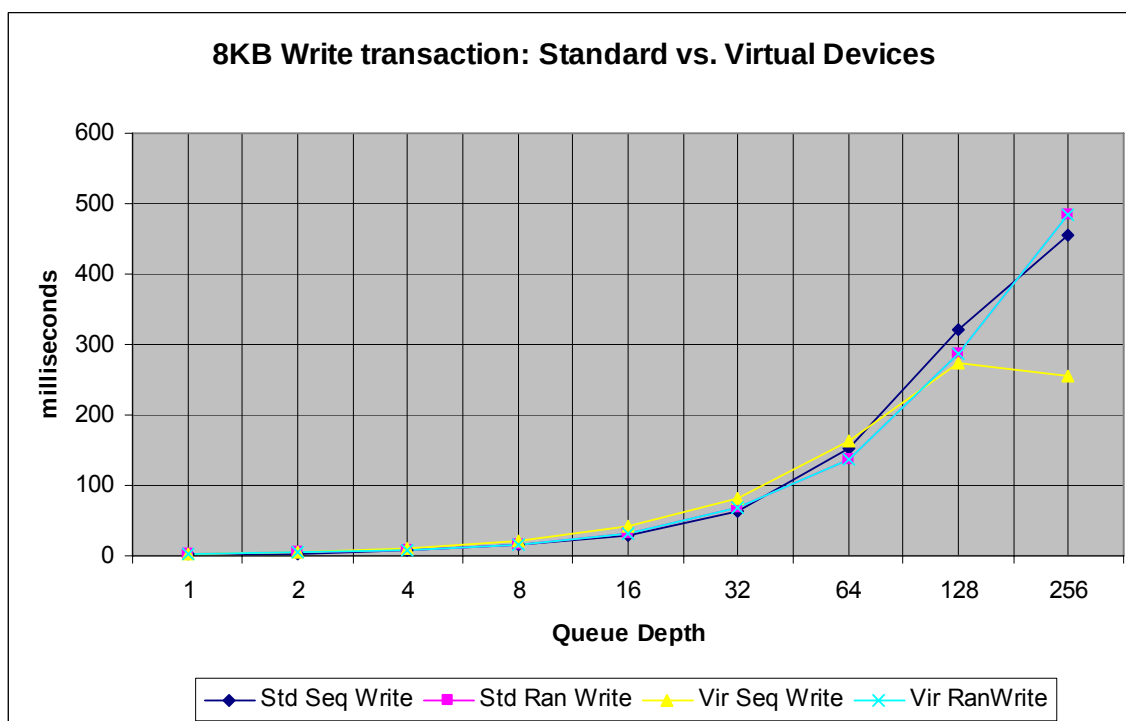
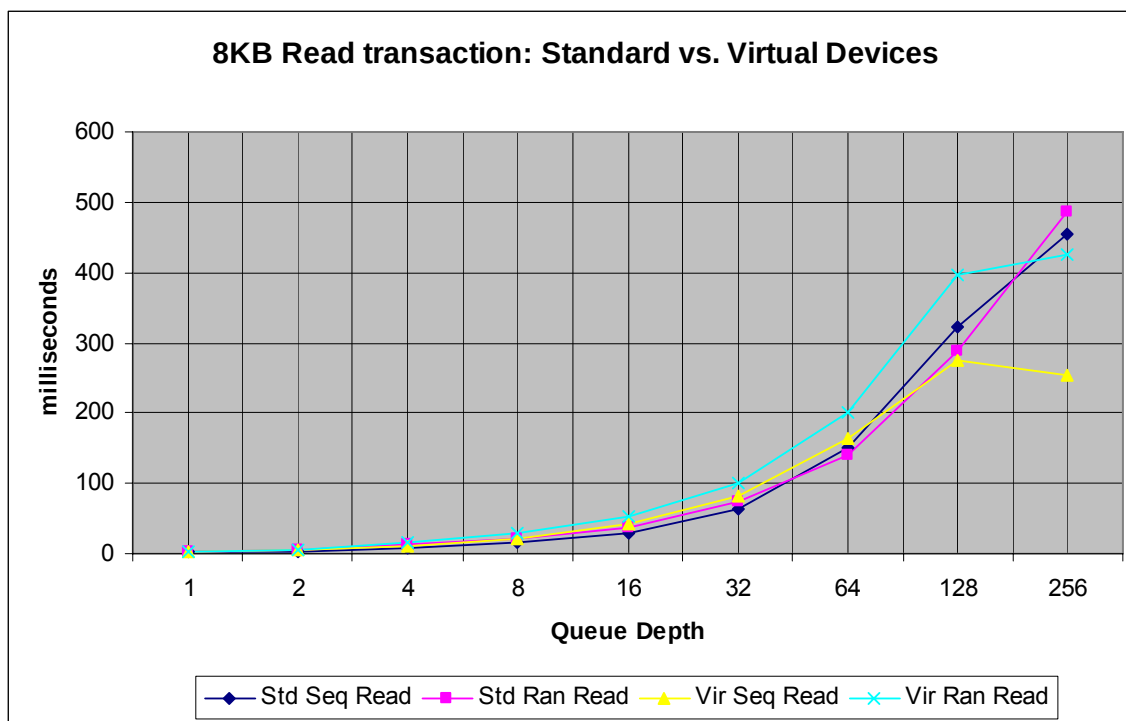


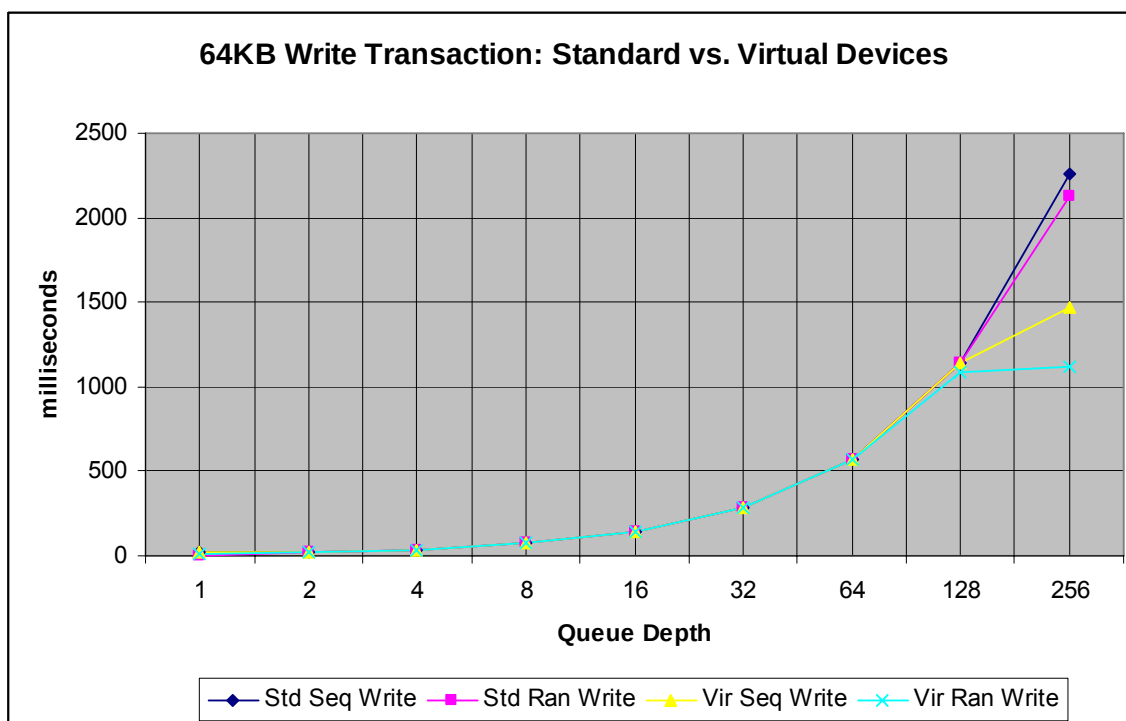
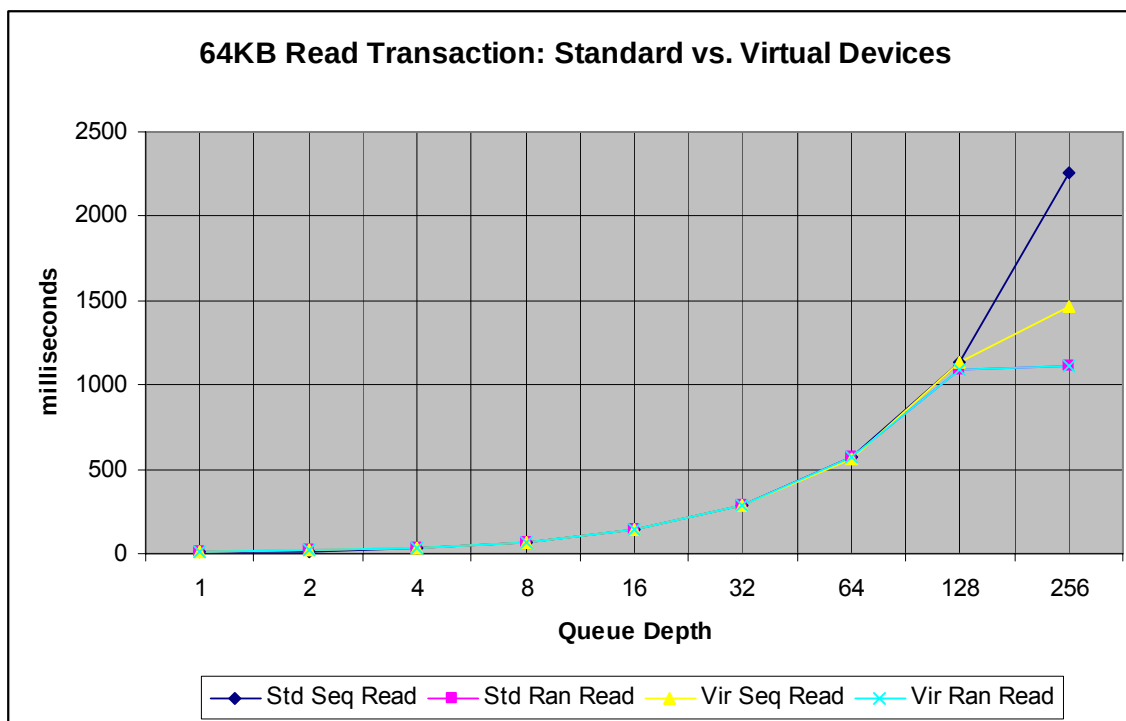




9.3.5. Latency Performance Summary Graphs:







9.3.6. Throughput Performance Summary Table:

Queue Depth	Access Spec Name	IOps	Read IOps	Write IOps	Read MBps	Write MBps
Standard Devices						
1	2KB Seq	18206.63	12753.47	5453.16	24.909	10.651
2	2KB Seq	14754.78	10321.29	4433.49	20.159	8.659
4	2KB Seq	14118.78	9855.42	4263.36	19.249	8.327
8	2KB Seq	14266.01	10000.84	4265.17	19.533	8.330
16	2KB Seq	14407.60	10084.03	4323.57	19.695	8.444
32	2KB Seq	14788.66	10363.97	4424.69	20.242	8.642
64	2KB Seq	15456.11	10830.92	4625.18	21.154	9.034
128	2KB Seq	15511.82	10865.66	4646.16	21.222	9.075
256	2KB Seq	18292.62	12815.96	5476.65	25.031	10.697
	Average	15533.67	10876.84	4656.83	21.244	9.095
Virtual Devices						
1	2KB Seq	16351.84	11478.78	4873.06	22.419	9.518
2	2KB Seq	14236.86	9941.42	4295.45	19.417	8.390
4	2KB Seq	13186.81	9229.77	3957.04	18.027	7.729
8	2KB Seq	12885.51	9054.12	3831.39	17.684	7.483
16	2KB Seq	13034.42	9118.51	3915.92	17.810	7.648
32	2KB Seq	14338.40	10040.56	4297.84	19.610	8.394
64	2KB Seq	14562.81	10174.03	4388.78	19.871	8.572
128	2KB Seq	14554.04	10197.02	4357.02	19.916	8.510
256	2KB Seq	14509.15	10135.26	4373.89	19.795	8.543
	Average	14184.43	9929.94	4254.49	19.394	8.310
Standard Devices						
1	2KB Ran	14395.69	10063.88	4331.81	19.656	8.461
2	2KB Ran	15373.06	10755.06	4618.00	21.006	9.020
4	2KB Ran	16042.00	11219.97	4822.03	21.914	9.418
8	2KB Ran	17310.60	12112.65	5197.95	23.658	10.152
16	2KB Ran	16234.30	11373.06	4861.24	22.213	9.495
32	2KB Ran	16052.36	11230.46	4821.90	21.934	9.418
64	2KB Ran	17206.96	12053.45	5153.51	23.542	10.065
128	2KB Ran	17579.48	12342.85	5236.62	24.107	10.228
256	2KB Ran	17519.77	12271.56	5248.21	23.968	10.250
	Average	16412.69	11491.44	4921.25	22.444	9.612
Virtual Devices						
1	2KB Ran	11067.30	7756.79	3310.52	15.150	6.466
2	2KB Ran	11835.08	8274.71	3560.37	16.162	6.954
4	2KB Ran	12652.00	8845.92	3806.09	17.277	7.434
8	2KB Ran	9599.04	6723.16	2875.88	13.131	5.617
16	2KB Ran	10321.30	7227.31	3093.99	14.116	6.043
32	2KB Ran	10560.96	7414.09	3146.87	14.481	6.146
64	2KB Ran	10932.01	7647.11	3284.90	14.936	6.416
128	2KB Ran	11087.40	7762.78	3324.62	15.162	6.493
256	2KB Ran	11377.71	7981.54	3396.17	15.589	6.633
	Average	11048.09	7737.04	3311.04	15.111	6.467

Standard Devices						
1	4KB Seq	17356.50	12146.89	5209.60	47.449	20.350
2	4KB Seq	16421.12	11476.24	4944.88	44.829	19.316
4	4KB Seq	17671.74	12364.95	5306.80	48.301	20.730
8	4KB Seq	18480.07	12931.99	5548.08	50.516	21.672
16	4KB Seq	19082.83	13342.08	5740.75	52.117	22.425
32	4KB Seq	19491.39	13653.72	5837.67	53.335	22.803
64	4KB Seq	19842.52	13895.72	5946.81	54.280	23.230
128	4KB Seq	19638.81	13766.19	5872.62	53.774	22.940
256	4KB Seq	19976.37	13980.14	5996.24	54.610	23.423
Average		18662.37	13061.99	5600.38	51.023	21.876
Virtual Devices						
1	4KB Seq	14041.62	9834.98	4206.64	38.418	16.432
2	4KB Seq	13810.06	9665.02	4145.04	37.754	16.192
4	4KB Seq	13016.63	9121.69	3894.94	35.632	15.215
8	4KB Seq	13038.45	9109.90	3928.55	35.586	15.346
16	4KB Seq	13219.60	9249.57	3970.04	36.131	15.508
32	4KB Seq	13566.10	9512.41	4053.70	37.158	15.835
64	4KB Seq	13677.74	9595.02	4082.72	37.481	15.948
128	4KB Seq	13704.88	9587.08	4117.81	37.450	16.085
256	4KB Seq	13699.79	9611.92	4087.87	37.547	15.968
Average		13530.54	9476.40	4054.14	37.017	15.837
Standard Devices						
1	4KB Ran	17580.10	12278.28	5301.82	47.962	20.710
2	4KB Ran	18064.72	12629.83	5434.89	49.335	21.230
4	4KB Ran	18665.79	13074.22	5591.57	51.071	21.842
8	4KB Ran	17941.69	12561.64	5380.04	49.069	21.016
16	4KB Ran	14609.47	10221.94	4387.53	39.929	17.139
32	4KB Ran	15624.85	10937.40	4687.45	42.724	18.310
64	4KB Ran	15560.03	10914.13	4645.90	42.633	18.148
128	4KB Ran	16253.68	11373.14	4880.55	44.426	19.065
256	4KB Ran	17253.59	12091.51	5162.08	47.232	20.164
Average		16839.33	11786.90	5052.43	46.043	19.736
Virtual Devices						
1	4KB Ran	12923.86	9054.70	3869.16	35.370	15.114
2	4KB Ran	13125.92	9209.68	3916.24	35.975	15.298
4	4KB Ran	10513.82	7350.24	3163.58	28.712	12.358
8	4KB Ran	8730.40	6100.28	2630.12	23.829	10.274
16	4KB Ran	9900.11	6945.52	2954.59	27.131	11.541
32	4KB Ran	10203.93	7140.03	3063.90	27.891	11.968
64	4KB Ran	10538.40	7378.33	3160.07	28.822	12.344
128	4KB Ran	11065.09	7734.90	3330.18	30.214	13.009
256	4KB Ran	10948.63	7662.96	3285.68	29.933	12.835
Average		10883.35	7619.63	3263.73	29.764	12.749
Standard Devices						
1	8KB Seq	16803.65	11734.66	5068.99	91.677	39.601
2	8KB Seq	16310.30	11430.23	4880.08	89.299	38.126

4	8KB Seq	16262.71	11396.56	4866.15	89.036	38.017
8	8KB Seq	16632.68	11640.69	4991.99	90.943	39.000
16	8KB Seq	16985.04	11923.48	5061.56	93.152	39.543
32	8KB Seq	16061.02	11249.25	4811.76	87.885	37.592
64	8KB Seq	13481.44	9434.11	4047.32	73.704	31.620
128	8KB Seq	12666.15	8850.06	3816.09	69.141	29.813
256	8KB Seq	15653.74	10957.38	4696.36	85.605	36.690
	Average	15650.75	10957.38	4693.37	85.605	36.667
Virtual Devices						
1	8KB Seq	13967.73	9770.10	4197.63	76.329	32.794
2	8KB Seq	12581.63	8796.40	3785.23	68.722	29.572
4	8KB Seq	12070.74	8438.11	3632.63	65.923	28.380
8	8KB Seq	12256.35	8573.88	3682.46	66.983	28.769
16	8KB Seq	12298.76	8639.26	3659.50	67.494	28.590
32	8KB Seq	12443.36	8715.90	3727.46	68.093	29.121
64	8KB Seq	12555.06	8773.64	3781.42	68.544	29.542
128	8KB Seq	12540.83	8784.19	3756.63	68.627	29.349
256	8KB Seq	12500.09	8755.80	3744.29	68.405	29.252
	Average	12579.39	8805.25	3774.14	68.791	29.485
Standard Devices						
1	8KB Ran	12223.76	8574.34	3649.41	66.987	28.511
2	8KB Ran	14126.25	9884.69	4241.56	77.224	33.137
4	8KB Ran	10453.92	7325.97	3127.95	57.234	24.437
8	8KB Ran	12015.97	8419.67	3596.30	65.779	28.096
16	8KB Ran	13418.53	9376.71	4041.83	73.256	31.577
32	8KB Ran	13717.62	9607.25	4110.37	75.057	32.112
64	8KB Ran	14491.45	10132.71	4358.74	79.162	34.053
128	8KB Ran	14117.59	9885.49	4232.10	77.230	33.063
256	8KB Ran	16057.99	11256.10	4801.89	87.938	37.515
	Average	13402.56	9384.77	4017.79	73.319	31.389
Virtual Devices						
1	8KB Ran	11641.93	8129.55	3512.38	63.512	27.440
2	8KB Ran	12103.86	8474.65	3629.21	66.208	28.353
4	8KB Ran	7668.29	5368.16	2300.13	41.939	17.970
8	8KB Ran	8743.52	6124.84	2618.68	47.850	20.458
16	8KB Ran	9708.18	6790.05	2918.13	53.047	22.798
32	8KB Ran	10116.65	7062.70	3053.94	55.177	23.859
64	8KB Ran	10133.11	7091.61	3041.50	55.403	23.762
128	8KB Ran	10119.16	7080.76	3038.40	55.318	23.738
256	8KB Ran	10269.72	7183.34	3086.38	56.120	24.112
	Average	10056.05	7033.96	3022.08	54.953	23.610
Standard Devices						
1	16KB Seq	7093.38	4964.52	2128.87	77.571	33.264
2	16KB Seq	10343.55	7261.89	3081.67	113.467	48.151
4	16KB Seq	11102.81	7775.48	3327.33	121.492	51.989
8	16KB Seq	10622.34	7438.15	3184.19	116.221	49.753
16	16KB Seq	10756.27	7541.29	3214.98	117.833	50.234

32	16KB Seq	10662.62	7467.38	3195.23	116.678	49.926
64	16KB Seq	8887.48	6217.61	2669.87	97.150	41.717
128	16KB Seq	8268.45	5790.52	2477.93	90.477	38.718
256	16KB Seq	10029.77	7036.33	2993.44	109.943	46.773
Average		9751.85	6832.57	2919.28	106.759	45.614
Virtual Devices						
1	16KB Seq	7555.58	5281.40	2274.18	82.522	35.534
2	16KB Seq	10484.95	7344.05	3140.90	114.751	49.077
4	16KB Seq	10607.98	7406.14	3201.83	115.721	50.029
8	16KB Seq	10810.27	7560.13	3250.14	118.127	50.783
16	16KB Seq	10886.86	7603.54	3283.32	118.805	51.302
32	16KB Seq	10992.62	7713.37	3279.25	120.521	51.238
64	16KB Seq	10999.41	7693.03	3306.37	120.204	51.662
128	16KB Seq	10924.85	7628.18	3296.66	119.190	51.510
256	16KB Seq	10848.71	7577.80	3270.91	118.403	51.108
Average		10456.80	7311.96	3144.84	114.249	49.138
Standard Devices						
1	16KB Ran	5008.48	3495.47	1513.00	54.617	23.641
2	16KB Ran	8676.82	6072.53	2604.29	94.883	40.692
4	16KB Ran	11873.81	8303.87	3569.93	129.748	55.780
8	16KB Ran	12915.51	9044.00	3871.50	141.313	60.492
16	16KB Ran	13346.59	9329.64	4016.95	145.776	62.765
32	16KB Ran	13420.88	9398.53	4022.35	146.852	62.849
64	16KB Ran	13467.54	9420.94	4046.59	147.202	63.228
128	16KB Ran	13347.23	9345.02	4002.21	146.016	62.535
256	16KB Ran	13134.63	9205.64	3928.99	143.838	61.390
Average		11687.94	8179.52	3508.42	127.805	54.819
Virtual Devices						
1	16KB Ran	3927.03	2740.46	1186.57	42.820	18.540
2	16KB Ran	7166.62	5011.16	2155.46	78.299	33.679
4	16KB Ran	8981.42	6298.80	2682.62	98.419	41.916
8	16KB Ran	10262.43	7187.15	3075.28	112.299	48.051
16	16KB Ran	10162.19	7101.98	3060.22	110.968	47.816
32	16KB Ran	8186.78	5714.34	2472.44	89.287	38.632
64	16KB Ran	8900.29	6209.17	2691.12	97.018	42.049
128	16KB Ran	8898.30	6243.93	2654.37	97.561	41.475
256	16KB Ran	9308.49	6509.32	2799.16	101.708	43.737
Average		8421.51	5890.70	2530.80	92.042	39.544
Standard Devices						
1	32KB Seq	6014.37	4209.46	1804.92	131.546	56.404
2	32KB Seq	6942.52	4877.66	2064.86	152.427	64.527
4	32KB Seq	6996.90	4896.74	2100.16	153.023	65.630
8	32KB Seq	6997.91	4899.09	2098.82	153.096	65.588
16	32KB Seq	6992.95	4876.37	2116.58	152.386	66.143
32	32KB Seq	6995.53	4897.35	2098.18	153.042	65.568
64	32KB Seq	7001.27	4907.30	2093.98	153.353	65.437
128	32KB Seq	6952.10	4869.11	2082.99	152.160	65.094

256	32KB Seq	6995.34	4895.26	2100.08	152.977	65.628
	Average	6876.54	4814.26	2062.29	150.446	64.446
Virtual Devices						
1	32KB Seq	3977.13	2788.63	1188.50	87.145	37.141
2	32KB Seq	5831.50	4082.17	1749.33	127.568	54.666
4	32KB Seq	5668.11	3958.82	1709.29	123.713	53.415
8	32KB Seq	5843.81	4086.67	1757.14	127.709	54.911
16	32KB Seq	5940.46	4165.54	1774.92	130.173	55.466
32	32KB Seq	6250.38	4376.13	1874.25	136.754	58.570
64	32KB Seq	6565.90	4590.93	1974.97	143.467	61.718
128	32KB Seq	6759.00	4719.04	2039.96	147.470	63.749
256	32KB Seq	6947.56	4856.57	2090.99	151.768	65.343
	Average	5975.98	4180.50	1795.48	130.641	56.109
Standard Devices						
1	32KB Ran	5537.73	3868.79	1668.93	120.900	52.154
2	32KB Ran	6738.02	4720.63	2017.39	147.520	63.043
4	32KB Ran	6987.64	4900.56	2087.08	153.142	65.221
8	32KB Ran	6991.62	4892.80	2098.82	152.900	65.588
16	32KB Ran	7001.75	4900.26	2101.49	153.133	65.672
32	32KB Ran	7032.97	4940.62	2092.35	154.394	65.386
64	32KB Ran	5869.84	4115.18	1754.66	128.599	54.833
128	32KB Ran	5516.29	3855.74	1660.55	120.492	51.892
256	32KB Ran	6394.65	4479.78	1914.87	139.993	59.840
	Average	6452.28	4519.37	1932.90	141.230	60.403
Virtual Devices						
1	32KB Ran	5445.89	3806.90	1638.99	118.966	51.218
2	32KB Ran	6579.96	4604.32	1975.64	143.885	61.739
4	32KB Ran	6900.66	4821.60	2079.06	150.675	64.971
8	32KB Ran	6997.89	4895.70	2102.20	152.991	65.694
16	32KB Ran	7020.88	4922.07	2098.82	153.815	65.588
32	32KB Ran	7032.80	4919.55	2113.25	153.736	66.039
64	32KB Ran	7036.20	4950.86	2085.34	154.714	65.167
128	32KB Ran	7029.12	4928.98	2100.13	154.031	65.629
256	32KB Ran	7026.31	4922.80	2103.51	153.837	65.735
	Average	6785.52	4752.53	2032.99	148.517	63.531
Standard Devices						
1	64KB Seq	2832.97	1980.53	852.44	123.783	53.278
2	64KB Seq	3532.45	2473.67	1058.78	154.605	66.174
4	64KB Seq	3579.85	2508.08	1071.77	156.755	66.986
8	64KB Seq	3580.29	2496.08	1084.21	156.005	67.763
16	64KB Seq	3581.50	2509.58	1071.92	156.849	66.995
32	64KB Seq	3579.44	2508.33	1071.11	156.771	66.945
64	64KB Seq	3580.39	2505.32	1075.07	156.583	67.192
128	64KB Seq	3585.77	2512.72	1073.05	157.045	67.066
256	64KB Seq	3583.20	2512.62	1070.59	157.039	66.912
	Average	3492.87	2445.21	1047.66	152.826	65.479
Virtual Devices						

1	64KB Seq	2624.78	1838.98	785.79	114.936	49.112
2	64KB Seq	3579.84	2507.74	1072.10	156.733	67.006
4	64KB Seq	3580.82	2499.33	1081.49	156.208	67.593
8	64KB Seq	3582.13	2498.62	1083.51	156.164	67.720
16	64KB Seq	3582.61	2513.83	1068.78	157.115	66.799
32	64KB Seq	3582.26	2510.22	1072.05	156.888	67.003
64	64KB Seq	3590.57	2518.41	1072.16	157.400	67.010
128	64KB Seq	3582.68	2503.75	1078.93	156.484	67.433
256	64KB Seq	3587.21	2517.41	1069.80	157.338	66.863
	Average	3476.99	2434.25	1042.74	152.141	65.171
Standard Devices						
1	64KB Ran	3226.33	2250.64	975.69	140.665	60.981
2	64KB Ran	3567.34	2510.06	1057.28	156.879	66.080
4	64KB Ran	3569.94	2508.84	1061.10	156.803	66.319
8	64KB Ran	3568.77	2496.95	1071.82	156.059	66.989
16	64KB Ran	3566.65	2486.59	1080.06	155.412	67.504
32	64KB Ran	3570.31	2496.45	1073.87	156.028	67.117
64	64KB Ran	3581.20	2512.72	1068.49	157.045	66.780
128	64KB Ran	3582.88	2519.54	1063.34	157.471	66.459
256	64KB Ran	3552.66	2491.32	1061.35	155.707	66.334
	Average	3531.79	2474.79	1057.00	154.674	66.062
Virtual Devices						
1	64KB Ran	3208.61	2241.32	967.29	140.082	60.456
2	64KB Ran	3564.81	2497.57	1067.24	156.098	66.702
4	64KB Ran	3568.89	2504.92	1063.97	156.557	66.498
8	64KB Ran	3569.23	2496.72	1072.51	156.045	67.032
16	64KB Ran	3575.01	2503.53	1071.48	156.470	66.968
32	64KB Ran	3575.97	2503.73	1072.24	156.483	67.015
64	64KB Ran	3576.81	2503.53	1073.27	156.471	67.080
128	64KB Ran	3575.69	2512.66	1063.04	157.041	66.440
256	64KB Ran	3574.95	2508.86	1066.09	156.804	66.631
	Average	3532.22	2474.76	1057.46	154.672	66.091
Standard Devices						
1	96KB Seq	2394.03	1683.07	710.96	157.788	66.652
2	96KB Seq	2395.49	1681.24	714.25	157.616	66.961
4	96KB Seq	2392.89	1662.34	730.54	155.845	68.488
8	96KB Seq	2394.38	1674.15	720.23	156.952	67.521
16	96KB Seq	2393.13	1672.07	721.06	156.757	67.600
32	96KB Seq	2396.44	1696.66	699.78	159.062	65.604
64	96KB Seq	2395.69	1673.06	722.63	156.849	67.746
128	96KB Seq	2392.34	1665.64	726.70	156.153	68.129
256	96KB Seq	2393.01	1664.00	729.01	156.000	68.345
	Average	2394.15	1674.69	719.46	157.002	67.450
Virtual Devices						
1	96KB Seq	2391.54	1679.55	711.98	157.458	66.748
2	96KB Seq	2387.53	1668.65	718.88	156.436	67.395
4	96KB Seq	2392.18	1663.51	728.67	155.954	68.313

8	96KB Seq	2392.54	1669.74	722.80	156.538	67.763
16	96KB Seq	2393.03	1671.46	721.57	156.699	67.647
32	96KB Seq	2397.95	1693.45	704.50	158.761	66.047
64	96KB Seq	2396.47	1677.04	719.43	157.222	67.447
128	96KB Seq	2396.26	1675.80	720.46	157.106	67.543
256	96KB Seq	2394.18	1671.04	723.14	156.660	67.794
	Average	2393.52	1674.47	719.05	156.982	67.411
Standard Devices						
1	96KB Ran	2388.73	1674.12	714.61	156.949	66.994
2	96KB Ran	2394.80	1680.73	714.07	157.568	66.944
4	96KB Ran	2393.93	1669.46	724.47	156.512	67.919
8	96KB Ran	2395.92	1683.50	712.42	157.828	66.789
16	96KB Ran	2392.60	1668.11	724.49	156.385	67.921
32	96KB Ran	2392.37	1666.40	725.97	156.225	68.060
64	96KB Ran	2399.33	1677.12	722.20	157.230	67.707
128	96KB Ran	2389.35	1678.56	710.80	157.365	66.637
256	96KB Ran	2385.23	1680.18	705.05	157.517	66.098
	Average	2392.47	1675.35	717.12	157.064	67.230
Virtual Devices						
1	96KB Ran	2386.08	1667.93	718.16	156.368	67.327
2	96KB Ran	2394.10	1676.02	718.08	157.127	67.320
4	96KB Ran	2396.08	1684.68	711.40	157.939	66.694
8	96KB Ran	2393.76	1672.95	720.82	156.839	67.577
16	96KB Ran	2393.46	1671.41	722.05	156.695	67.692
32	96KB Ran	2396.60	1687.08	709.53	158.163	66.518
64	96KB Ran	2393.12	1676.59	716.53	157.180	67.175
128	96KB Ran	2392.60	1676.31	716.29	157.154	67.153
256	96KB Ran	2392.79	1673.15	719.64	156.858	67.466
	Average	2393.18	1676.23	716.94	157.147	67.214
Standard Devices						
1	128KB Seq	1803.09	1265.85	537.24	158.232	67.155
2	128KB Seq	1803.76	1262.07	541.69	157.759	67.711
4	128KB Seq	1805.56	1270.15	535.41	158.769	66.926
8	128KB Seq	1804.09	1265.66	538.43	158.207	67.304
16	128KB Seq	1802.55	1266.39	536.16	158.299	67.020
32	128KB Seq	1800.70	1256.51	544.19	157.064	68.024
64	128KB Seq	1801.85	1257.39	544.46	157.174	68.057
128	128KB Seq	1802.68	1267.14	535.54	158.392	66.942
256	128KB Seq	1803.80	1264.12	539.68	158.015	67.460
	Average	1803.12	1263.92	539.20	157.990	67.400
Virtual Devices						
1	128KB Seq	1801.81	1257.75	544.07	157.219	68.008
2	128KB Seq	1803.54	1259.39	544.15	157.424	68.019
4	128KB Seq	1803.49	1259.13	544.36	157.391	68.045
8	128KB Seq	1804.30	1269.47	534.83	158.684	66.854
16	128KB Seq	1802.83	1262.80	540.03	157.850	67.504
32	128KB Seq	1802.71	1263.32	539.39	157.915	67.424

64	128KB Seq	1803.46	1266.93	536.54	158.366	67.067
128	128KB Seq	1800.82	1256.65	544.17	157.081	68.021
256	128KB Seq	1803.34	1263.85	539.49	157.982	67.436
	Average	1802.92	1262.14	540.78	157.768	67.598
Standard Devices						
1	128KB Ran	1801.00	1264.10	536.90	158.013	67.112
2	128KB Ran	1802.08	1261.39	540.70	157.673	67.587
4	128KB Ran	1802.86	1263.71	539.15	157.964	67.394
8	128KB Ran	1802.77	1266.83	535.93	158.354	66.992
16	128KB Ran	1800.69	1261.23	539.46	157.654	67.432
32	128KB Ran	1801.43	1267.73	533.69	158.467	66.712
64	128KB Ran	1798.43	1261.05	537.38	157.631	67.172
128	128KB Ran	1795.75	1261.58	534.16	157.698	66.770
256	128KB Ran	1786.19	1244.82	541.36	155.603	67.670
	Average	1799.02	1261.38	537.64	157.673	67.205
Virtual Devices						
1	128KB Ran	1799.13	1254.39	544.75	156.798	68.093
2	128KB Ran	1801.06	1255.86	545.19	156.983	68.149
4	128KB Ran	1800.88	1256.32	544.57	157.040	68.071
8	128KB Ran	1800.27	1254.46	545.81	156.808	68.227
16	128KB Ran	1803.13	1271.44	531.69	158.930	66.461
32	128KB Ran	1800.53	1261.66	538.88	157.707	67.360
64	128KB Ran	1800.35	1268.41	531.93	158.552	66.491
128	128KB Ran	1798.80	1256.43	542.37	157.054	67.796
256	128KB Ran	1797.40	1258.30	539.10	157.287	67.388
	Average	1800.17	1259.70	540.48	157.462	67.560

9.3.7. Latency Performance Summary Table:

Queue Depth	Access Spec Name	Average Response Time	Average Read Response Time	Average Write Response Time
Standard Devices				
1	2KB Seq	1.754319	1.526303	2.287586
2	2KB Seq	4.334497	3.806534	5.563609
4	2KB Seq	9.061989	8.309759	10.800886
8	2KB Seq	17.940297	17.149611	19.794273
16	2KB Seq	35.527362	34.791125	37.244516
32	2KB Seq	69.216883	68.416368	71.091928
64	2KB Seq	132.388546	131.742764	133.900803
128	2KB Seq	264.113855	263.800316	264.84711
256	2KB Seq	417.563763	417.117426	418.608229
	Average	105.77	105.18	107.13
Virtual Devices				
1	2KB Seq	1.95539	1.733523	2.47801
2	2KB Seq	4.493629	4.172607	5.236604
4	2KB Seq	9.703832	9.232113	10.804112

8	2KB Seq	19.862183	19.287154	21.221058
16	2KB Seq	39.270295	38.737111	40.511854
32	2KB Seq	71.268241	70.976289	71.950295
64	2KB Seq	139.898213	139.495052	140.832817
128	2KB Seq	242.594174	242.172895	243.580122
256	2KB Seq	225.069964	224.494334	226.403826
	Average	83.79	83.37	84.78
Standard Devices				
1	2KB Ran	2.220117	2.385218	1.836546
2	2KB Ran	4.157307	4.370205	3.661481
4	2KB Ran	7.971185	7.389145	9.325483
8	2KB Ran	14.771121	13.39568	17.976277
16	2KB Ran	31.510224	28.281491	39.063962
32	2KB Ran	63.71055	59.933875	72.506626
64	2KB Ran	118.650272	116.784585	123.013884
128	2KB Ran	231.983547	230.134857	236.340938
256	2KB Ran	460.990561	459.34043	464.84886
	Average	104.00	102.45	107.62
Virtual Devices				
1	2KB Ran	2.889432	2.900646	2.863158
2	2KB Ran	5.402236	5.39336	5.422864
4	2KB Ran	10.106395	10.022667	10.300992
8	2KB Ran	26.647954	23.1851	34.743338
16	2KB Ran	49.571364	46.377842	57.031162
32	2KB Ran	96.777149	93.586911	104.293416
64	2KB Ran	186.518963	183.658901	193.177061
128	2KB Ran	357.631552	354.79542	364.253748
256	2KB Ran	314.402521	312.226642	319.516184
	Average	116.66	114.68	121.29
Standard Devices				
1	4KB Seq	1.841087	1.697482	2.175922
2	4KB Seq	3.89464	3.492767	4.827319
4	4KB Seq	7.239865	6.75024	8.380702
8	4KB Seq	13.847519	13.431096	14.818157
16	4KB Seq	26.722389	26.436111	27.387729
32	4KB Seq	52.500975	52.263089	53.057365
64	4KB Seq	103.091408	102.831552	103.698607
128	4KB Seq	207.253217	207.128582	207.545379
256	4KB Seq	407.842167	407.463462	408.725113
	Average	91.58	91.28	92.29
Virtual Devices				
1	4KB Seq	2.277333	2.17997	2.504965
2	4KB Seq	4.632481	4.345181	5.302379
4	4KB Seq	9.831106	9.44103	10.744636
8	4KB Seq	19.629513	19.230861	20.553946
16	4KB Seq	38.717965	38.402384	39.45322
32	4KB Seq	75.374809	75.129508	75.950433

64	4KB Seq	148.680157	148.343884	149.470454
128	4KB Seq	297.941936	297.646367	298.630079
256	4KB Seq	321.157972	320.824699	321.941605
	Average	102.03	101.73	102.73
Standard Devices				
1	4KB Ran	1.817227	1.727549	2.02491
2	4KB Ran	3.538521	3.472101	3.692872
4	4KB Ran	6.849372	6.830709	6.89301
8	4KB Ran	14.257624	12.913902	17.395027
16	4KB Ran	35.0206	32.023148	42.00398
32	4KB Ran	65.463619	63.255244	70.616507
64	4KB Ran	131.457872	130.250445	134.294345
128	4KB Ran	251.490851	249.930898	255.126156
256	4KB Ran	437.499898	436.49564	439.852328
	Average	105.27	104.10	107.99
Virtual Devices				
1	4KB Ran	2.474015	2.341952	2.783069
2	4KB Ran	4.872402	4.684399	5.314522
4	4KB Ran	12.169951	10.411503	16.255523
8	4KB Ran	29.301642	25.570356	37.955937
16	4KB Ran	51.663084	48.452902	59.209421
32	4KB Ran	100.185133	97.074842	107.433262
64	4KB Ran	193.336031	190.307437	200.407389
128	4KB Ran	316.528253	314.111085	322.142523
256	4KB Ran	335.569869	333.018149	341.521068
	Average	116.23	114.00	121.45
Standard Devices				
1	8KB Seq	1.901711	1.844885	2.03326
2	8KB Seq	3.921033	3.639431	4.580607
4	8KB Seq	7.867412	7.555844	8.597106
8	8KB Seq	15.386412	15.2032	15.813638
16	8KB Seq	30.136594	29.994848	30.470503
32	8KB Seq	63.728563	63.494973	64.274668
64	8KB Seq	151.628178	151.425698	152.100159
128	8KB Seq	322.296001	322.142505	322.651994
256	8KB Seq	455.45826	455.349846	455.711208
	Average	116.92	116.74	117.36
Virtual Devices				
1	8KB Seq	2.289387	2.212591	2.468133
2	8KB Seq	5.084991	4.891575	5.534466
4	8KB Seq	10.601592	10.386357	11.101554
8	8KB Seq	20.88216	20.741338	21.210036
16	8KB Seq	41.60948	41.508609	41.847616
32	8KB Seq	82.199324	82.139447	82.339334
64	8KB Seq	162.618587	162.56796	162.736052
128	8KB Seq	273.605994	273.686086	273.418714
256	8KB Seq	254.859394	254.543191	255.59882

	Average	94.86	94.74	95.14
Standard Devices				
1	8KB Ran	2.614296	3.03224	1.632332
2	8KB Ran	4.526866	4.71641	4.085147
4	8KB Ran	12.230917	9.101483	19.560363
8	8KB Ran	21.282235	17.063261	31.159707
16	8KB Ran	38.099418	32.580872	50.901971
32	8KB Ran	74.4726	68.869917	87.567888
64	8KB Ran	140.99732	137.73445	148.582476
128	8KB Ran	289.032444	287.799002	291.91354
256	8KB Ran	485.197982	484.391643	487.088087
	Average	118.72	116.14	124.72
Virtual Devices				
1	8KB Ran	2.744555	2.795735	2.626097
2	8KB Ran	5.281425	5.300887	5.235978
4	8KB Ran	16.685682	12.920533	25.472984
8	8KB Ran	29.256496	25.916585	37.068214
16	8KB Ran	52.656918	50.214047	58.341114
32	8KB Ran	100.94924	98.871452	105.754442
64	8KB Ran	201.336723	198.725051	207.426143
128	8KB Ran	396.282603	393.82546	402.008785
256	8KB Ran	425.691821	423.447191	430.916037
	Average	136.77	134.67	141.65
Standard Devices				
1	16KB Seq	4.506095	3.42943	7.01688
2	16KB Seq	6.181067	5.592915	7.567036
4	16KB Seq	11.524224	11.122787	12.462323
8	16KB Seq	24.093233	23.888682	24.571055
16	16KB Seq	47.580488	47.352871	48.114403
32	16KB Seq	95.957918	95.612372	96.765472
64	16KB Seq	230.118505	229.9404	230.53327
128	16KB Seq	493.933219	493.891431	494.030882
256	16KB Seq	764.182195	763.952305	764.722573
	Average	186.45	186.09	187.31
Virtual Devices				
1	16KB Seq	4.232806	2.469154	8.328598
2	16KB Seq	6.101056	5.931145	6.498344
4	16KB Seq	12.059771	11.841118	12.565534
8	16KB Seq	23.673683	23.443312	24.209549
16	16KB Seq	47.009149	46.820699	47.445561
32	16KB Seq	92.948111	92.806738	93.280646
64	16KB Seq	185.176111	185.191944	185.139272
128	16KB Seq	318.659253	318.555355	318.899663
256	16KB Seq	263.648326	263.533721	263.913833
	Average	105.95	105.62	106.70
Standard Devices				
1	16KB Ran	6.386298	8.350411	1.848623

2	16KB Ran	7.371932	8.93828	3.719615
4	16KB Ran	10.759626	11.744453	8.468863
8	16KB Ran	19.8072	20.43293	18.345466
16	16KB Ran	38.327967	38.591339	37.716268
32	16KB Ran	76.231632	76.454177	75.711639
64	16KB Ran	151.926804	152.006793	151.740579
128	16KB Ran	306.372459	306.537909	305.986138
256	16KB Ran	620.495892	620.0863	621.455579
	Average	137.52	138.13	136.11
Virtual Devices				
1	16KB Ran	8.146439	10.21639	3.36575
2	16KB Ran	8.925157	10.396438	5.504621
4	16KB Ran	14.239874	15.294153	11.764427
8	16KB Ran	24.922179	25.589244	23.363201
16	16KB Ran	50.308782	49.593762	51.968161
32	16KB Ran	124.861214	120.864255	134.099048
64	16KB Ran	229.234601	226.134401	236.387647
128	16KB Ran	457.004881	453.186505	465.986905
256	16KB Ran	513.779213	510.667806	521.014649
	Average	159.05	157.99	161.49
Standard Devices				
1	32KB Seq	5.314611	5.733146	4.338497
2	32KB Seq	9.211907	9.253355	9.113999
4	32KB Seq	18.28873	18.118465	18.685722
8	32KB Seq	36.566074	36.319107	37.142546
16	32KB Seq	73.112819	72.721743	74.013816
32	32KB Seq	146.073746	145.679201	146.994651
64	32KB Seq	291.776848	291.553034	292.301364
128	32KB Seq	586.258012	585.331026	588.424916
256	32KB Seq	1164.444985	1163.321352	1167.064171
	Average	259.01	258.67	259.79
Virtual Devices				
1	32KB Seq	8.041968	4.650728	15.998945
2	32KB Seq	10.970152	11.259632	10.294633
4	32KB Seq	22.57146	22.315878	23.163404
8	32KB Seq	43.770558	43.307457	44.847617
16	32KB Seq	86.125329	85.686029	87.156322
32	32KB Seq	163.393502	163.188756	163.871554
64	32KB Seq	310.8253	310.046308	312.636113
128	32KB Seq	601.839449	601.552312	602.503685
256	32KB Seq	616.153361	616.730085	614.813854
	Average	207.08	206.53	208.37
Standard Devices				
1	32KB Ran	5.775715	6.710997	3.607614
2	32KB Ran	9.491125	9.799981	8.768409
4	32KB Ran	18.310042	18.12678	18.740349
8	32KB Ran	36.603704	36.337921	37.223303

16	32KB Ran	73.085378	72.878874	73.566904
32	32KB Ran	145.488726	145.292644	145.951729
64	32KB Ran	348.675125	348.835975	348.29788
128	32KB Ran	739.656771	738.899669	741.414786
256	32KB Ran	1171.443423	1170.003374	1174.812541
	Average	283.17	282.99	283.60
Virtual Devices				
1	32KB Ran	5.872625	6.590545	4.205103
2	32KB Ran	9.70663	9.997946	9.027704
4	32KB Ran	18.53495	18.516676	18.577331
8	32KB Ran	36.558206	36.418167	36.884336
16	32KB Ran	72.88926	72.665755	73.413415
32	32KB Ran	145.367311	145.363446	145.376308
64	32KB Ran	290.221751	289.91574	290.948261
128	32KB Ran	468.534171	468.539436	468.521815
256	32KB Ran	494.802716	494.073919	496.508305
	Average	171.39	171.34	171.50
Standard Devices				
1	64KB Seq	11.28796	6.067772	23.416371
2	64KB Seq	18.1021	15.484964	24.216647
4	64KB Seq	35.733632	35.097829	37.221489
8	64KB Seq	71.459122	70.755785	73.078348
16	64KB Seq	142.818634	141.903721	144.960642
32	64KB Seq	284.97971	283.956975	287.374746
64	64KB Seq	569.020165	567.870327	571.699726
128	64KB Seq	1137.950746	1136.90218	1140.406113
256	64KB Seq	2254.108499	2253.291079	2256.02691
	Average	502.83	501.26	506.49
Virtual Devices				
1	64KB Seq	12.187022	6.11319	26.401518
2	64KB Seq	17.871372	17.453337	18.849189
4	64KB Seq	35.734688	35.108289	37.182295
8	64KB Seq	71.421191	70.607745	73.297022
16	64KB Seq	142.677839	141.842264	144.643158
32	64KB Seq	284.982385	283.819152	287.706112
64	64KB Seq	567.348331	566.562276	569.194701
128	64KB Seq	1139.446103	1138.836917	1140.859768
256	64KB Seq	1468.586372	1467.655643	1470.776518
	Average	415.58	414.22	418.77
Standard Devices				
1	64KB Ran	9.914425	11.941835	5.237776
2	64KB Ran	17.926669	17.369711	19.248924
4	64KB Ran	35.844952	34.969587	37.914646
8	64KB Ran	71.696176	70.881804	73.593355
16	64KB Ran	143.428883	142.435377	145.716207
32	64KB Ran	286.158075	285.254757	288.258044
64	64KB Ran	570.131907	570.187727	570.000633

128	64KB Ran	1140.615075	1141.218808	1139.18441
256	64KB Ran	2131.54003	2131.559339	2131.494705
	Average	489.70	489.54	490.07
Virtual Devices				
1	64KB Ran	9.969795	11.366808	6.732774
2	64KB Ran	17.949268	17.47113	19.068213
4	64KB Ran	35.855421	35.126163	37.57232
8	64KB Ran	71.689805	70.904339	73.518314
16	64KB Ran	143.110449	142.192828	145.254479
32	64KB Ran	285.631705	284.552705	288.151215
64	64KB Ran	570.145176	568.934097	572.970152
128	64KB Ran	1086.411939	1085.405872	1088.78994
256	64KB Ran	1112.436079	1110.131704	1117.859036
	Average	370.36	369.57	372.21
Standard Devices				
1	96KB Seq	13.363637	12.451323	15.523371
2	96KB Seq	26.710629	25.464478	29.643903
4	96KB Seq	53.471885	52.046488	56.715382
8	96KB Seq	106.832588	105.416484	110.124302
16	96KB Seq	213.670648	212.353805	216.724265
32	96KB Seq	424.973741	422.834324	430.160905
64	96KB Seq	850.274848	848.114076	855.277654
128	96KB Seq	1702.002653	1701.612438	1702.897041
256	96KB Seq	3063.995208	3063.615114	3064.862781
	Average	717.26	715.99	720.21
Virtual Devices				
1	96KB Seq	13.374004	12.615216	15.163975
2	96KB Seq	26.785074	25.873927	28.900013
4	96KB Seq	53.482907	52.31738	56.143716
8	96KB Seq	106.918412	105.595936	109.973448
16	96KB Seq	213.608465	212.223072	216.817609
32	96KB Seq	424.933544	423.576545	428.195454
64	96KB Seq	850.801012	849.92376	852.84594
128	96KB Seq	1452.066634	1451.27907	1453.89853
256	96KB Seq	1776.5587	1772.597434	1785.712441
	Average	546.50	545.11	549.74
Standard Devices				
1	96KB Ran	13.392801	13.994164	11.983978
2	96KB Ran	26.718102	25.524691	29.52708
4	96KB Ran	53.449399	52.198331	56.332324
8	96KB Ran	106.766844	105.357407	110.097465
16	96KB Ran	213.663646	211.996943	217.501158
32	96KB Ran	426.839883	425.525767	429.856309
64	96KB Ran	850.022164	846.992395	857.057916
128	96KB Ran	1695.907431	1694.684446	1698.794819
256	96KB Ran	2975.187247	2971.701635	2983.494231
	Average	706.88	705.33	710.52

Virtual Devices				
1	96KB Ran	13.406642	13.650687	12.839843
2	96KB Ran	26.726735	25.534031	29.510531
4	96KB Ran	53.400913	52.276568	56.063498
8	96KB Ran	106.867199	105.51126	110.014192
16	96KB Ran	213.613192	212.708981	215.70627
32	96KB Ran	425.743909	424.262027	429.267454
64	96KB Ran	848.800634	846.340616	854.55675
128	96KB Ran	1553.229725	1551.552619	1557.154576
256	96KB Ran	1470.295858	1467.079132	1477.774655
	Average	523.56	522.10	526.99
Standard Devices				
1	128KB Seq	17.74396	16.239312	21.289244
2	128KB Seq	35.472062	33.635164	39.751843
4	128KB Seq	70.859428	68.979156	75.319991
8	128KB Seq	141.775698	139.596765	146.897571
16	128KB Seq	283.404234	281.318134	288.33154
32	128KB Seq	566.325489	565.340404	568.600012
64	128KB Seq	1129.821698	1125.796715	1139.117108
128	128KB Seq	2258.904459	2257.060738	2263.266937
256	128KB Seq	3920.772427	3919.732934	3923.207303
	Average	936.12	934.19	940.64
Virtual Devices				
1	128KB Seq	17.755771	16.233159	21.275674
2	128KB Seq	35.479041	33.698936	39.598945
4	128KB Seq	70.950056	69.19504	75.009465
8	128KB Seq	141.778433	140.155085	145.631615
16	128KB Seq	283.462669	281.928481	287.050178
32	128KB Seq	565.161069	562.641486	571.062228
64	128KB Seq	1129.214708	1128.952253	1129.83444
128	128KB Seq	2092.092375	2089.437764	2098.222653
256	128KB Seq	2209.229811	2206.976273	2214.509174
	Average	727.24	725.47	731.35
Standard Devices				
1	128KB Ran	17.764511	16.976613	19.619584
2	128KB Ran	35.505316	33.840675	39.388744
4	128KB Ran	70.966244	69.378031	74.68885
8	128KB Ran	141.859939	140.147538	145.907677
16	128KB Ran	283.760618	282.167291	287.485753
32	128KB Ran	565.533848	564.038112	569.086811
64	128KB Ran	1128.138861	1125.261312	1134.891457
128	128KB Ran	2249.292719	2250.075076	2247.44505
256	128KB Ran	3701.584003	3692.834125	3721.704176
	Average	910.49	908.30	915.58
Virtual Devices				
1	128KB Ran	17.752002	17.298206	18.796955
2	128KB Ran	35.526862	33.817152	39.465201

4	128KB Ran	71.048757	69.272252	75.147179
8	128KB Ran	142.069158	140.235285	146.284024
16	128KB Ran	283.325174	281.389765	287.953372
32	128KB Ran	566.112612	564.779688	569.233349
64	128KB Ran	1119.032731	1116.174224	1125.848959
128	128KB Ran	1956.685196	1952.824745	1965.628152
256	128KB Ran	1852.621643	1851.830292	1854.4687
	Average	671.57	669.74	675.87

9.4. *Draining and Degraded Mode Performance*

9.4.1. Description

DATADEVs may be removed from an existing storage pool. However, to remove a DATADEV, it first must be disabled and any allocated space on the device must be “drained” to the remaining devices in the storage pool. Once drained, the data device can be removed. Degraded mode operation occurs following the unplanned disabling and draining of DATADEV(s) from an existing storage pool.

The draining and degraded mode simulation utilizes a 100 percent random access pattern with a 70/30 read to write ratio with an 8KB transfer size. The test is performed at gradually increasing queue depths from 4 through 256 in power of two increments.

9.4.2. Test Configuration and Process

- Solaris 10 server with 4 processors
- 16 Raid-6 14+2 devices at 65472 cylinders
- Thin device pool:
 - 15-16 Raid-6 14+2 DATADEVs at 65472 cylinders
 - 16 TDEVs
- Access pattern 8KB random transactions, 70/30 ratio of read to writes, queue depths 4-256 step in power of 2.
- Logical disabling of DATADEV in the pool:


```
symconfigure -sid xxx -cmd "disable dev 363 in pool <poolname>,
type=thin;" -commit -v -nop
```
- Once the access pattern is completed, verify that the disable DATADEV is fully drained.
- Initiate access pattern with one disable DATADEV for degrade mode testing.

9.4.3. Observation

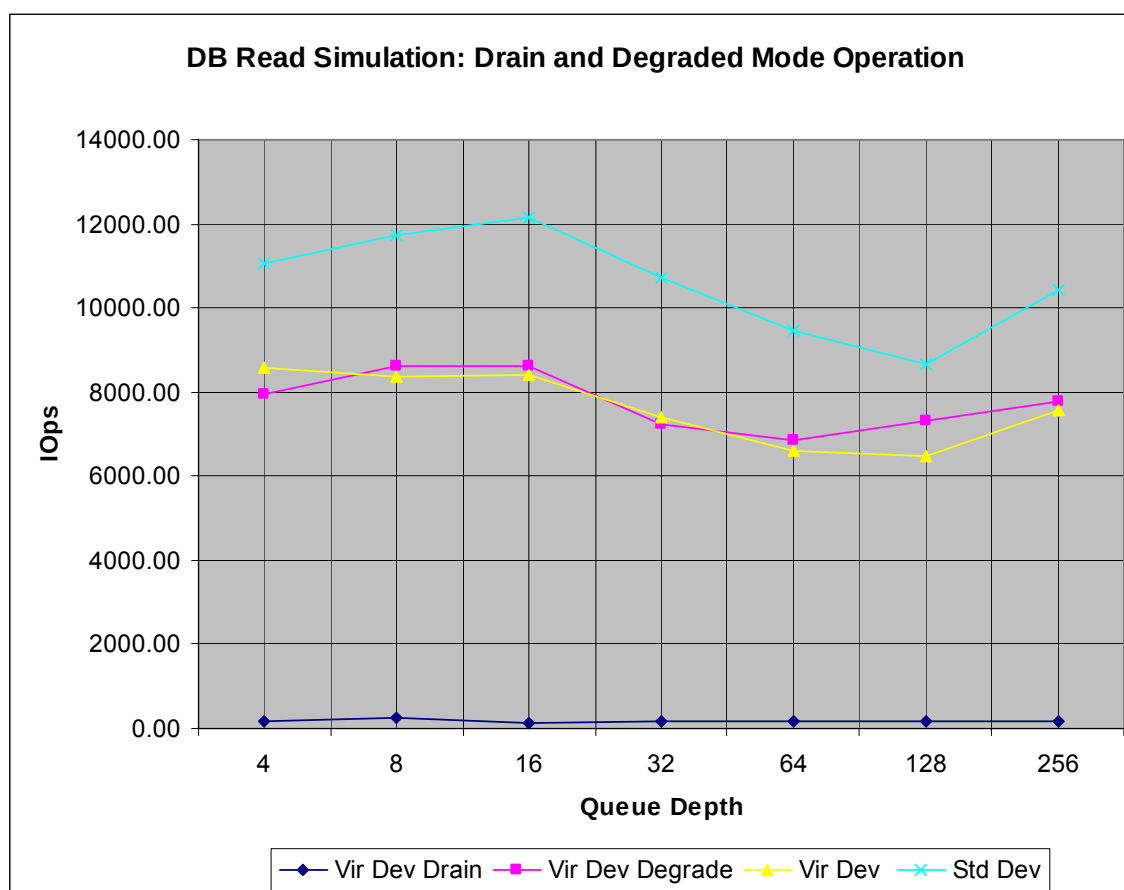
- The affect of draining 1 out of the 16 DATADEVs within the thin pool **showed a 98% reduction in throughput** as compared to normal operation.

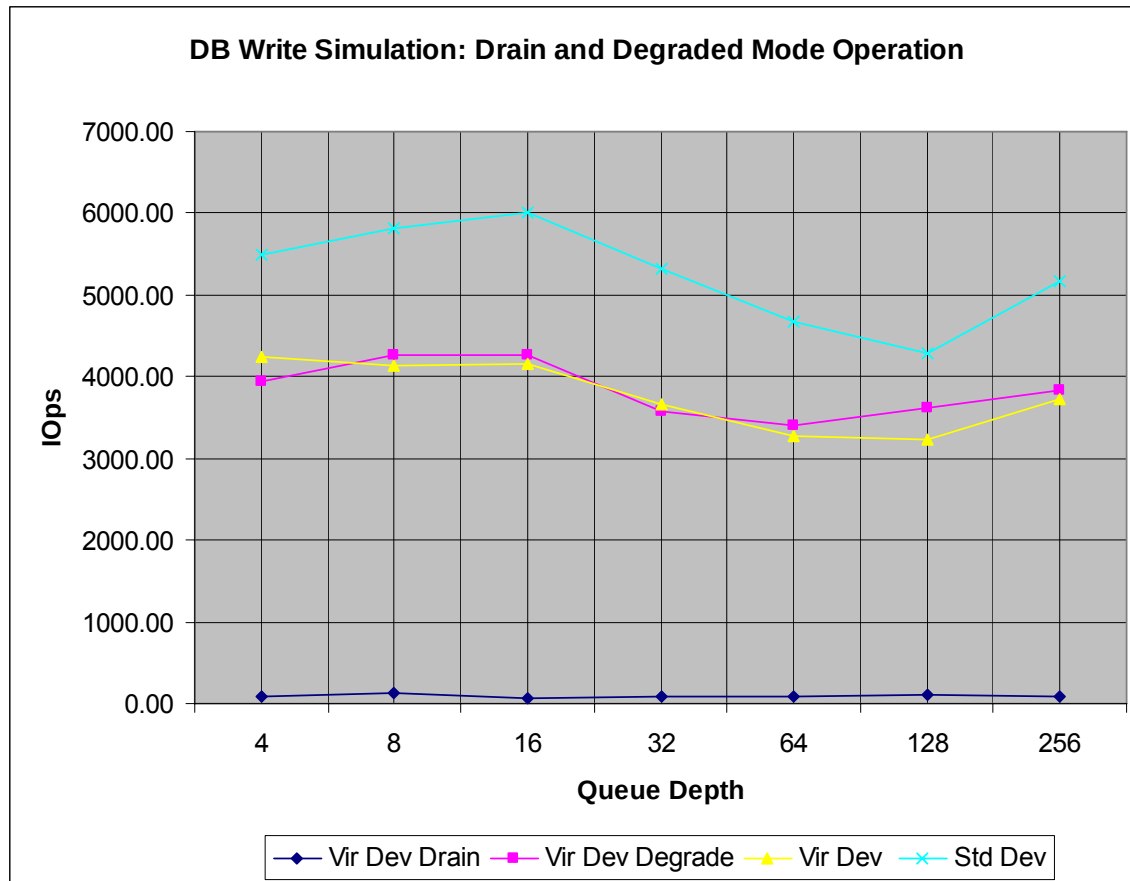
- Degraded mode operation shows a negligible 1.9% detrimental affect on throughput.

9.4.4. Conclusion

Reclaiming storage space from the thin pool requires disabling and draining the DATADEV. This process has an extreme negative affect on performance which points to the Symmetrix placing a high I/O priority on the draining process. Placing a high priority on the draining process is understandable from the array perspective, since it impacts the integrity of the existing data within the thin pool. Therefore, disabling DATADEVs should be avoided; however, if this action is required it should only be done during the maintenance window.

9.4.5. Summary Performance Graphs:





9.4.6. Summary Performance Table:

Drain DATADEV Queue Depth	Access Specification Name	IOPS	Read IOPS	Write IOPS	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time
4	OLTP 8K	231.25	155.22	76.03	1.213	0.594	562.2129	339.6529
8	OLTP 8K	383.59	263.80	119.78	2.061	0.936	645.0283	535.6613
16	OLTP 8K	201.08	139.60	61.48	1.091	0.480	2483.4943	2216.4585
32	OLTP 8K	275.22	187.72	87.51	1.467	0.684	3035.5096	2884.4179
64	OLTP 8K	228.39	147.57	80.82	1.153	0.631	6896.7152	6705.1458
128	OLTP 8K	267.93	170.24	97.68	1.330	0.763	8058.6145	8027.8505
256	OLTP 8K	277.60	188.18	89.42	1.470	0.699	10089.1740	9951.4632
Average		266.44	178.90	87.53	1.398	0.684	4538.6784	4380.0929
Stdev		59.00	41.90	18.27	0.327	0.143	3790.6546	3824.8168
Degrade Pool Queue Depth	Access Specification Name	IOPS	Read IOPS	Write IOPS	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time

4	OLTP 8K	11909.15	7965.38	3943.77	62.230	30.811	11.8109	8.5844
8	OLTP 8K	12905.82	8632.98	4272.83	67.445	33.382	19.9576	19.5433
16	OLTP 8K	12908.93	8636.66	4272.27	67.474	33.377	39.5526	39.8362
32	OLTP 8K	10810.22	7230.93	3579.29	56.492	27.963	94.4965	94.9544
64	OLTP 8K	10255.73	6849.65	3406.08	53.513	26.610	198.9848	199.6835
128	OLTP 8K	10937.92	7313.64	3624.27	57.138	28.315	336.4508	338.2222
256	OLTP 8K	11591.69	7761.55	3830.14	60.637	29.923	337.1795	338.0153
Average		11617.06	7770.11	3846.95	60.704	30.054	148.3475	148.4056
Stdev		1031.28	692.82	338.51	5.413	2.645	143.4497	144.5749
Virtual Dev Queue Depth	Access Specification Name	IOps	Read IOps	Write IOps	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time
4	OLTP 8K	12812.86	8572.43	4240.43	66.972	33.128	9.9860	9.9197
8	OLTP 8K	12502.64	8367.28	4135.36	65.369	32.308	20.4702	20.3471
16	OLTP 8K	12555.66	8400.65	4155.01	65.630	32.461	40.7706	40.6543
32	OLTP 8K	11058.85	7405.08	3653.77	57.852	28.545	92.5507	92.4512
64	OLTP 8K	9869.97	6589.74	3280.23	51.482	25.627	207.5182	207.2527
128	OLTP 8K	9711.47	6480.48	3230.99	50.629	25.242	402.4909	401.9595
256	OLTP 8K	11284.90	7555.85	3729.06	59.030	29.133	429.8772	429.8880
Average		11399.48	7624.50	3774.98	59.566	29.492	171.9519	171.7818
Stdev		1282.53	864.59	418.02	6.755	3.266	179.6438	179.5728
Std Dev Queue Depth	Access Specification Name	IOps	Read IOps	Write IOps	Read MBps	Write MBps	Average Read Response Time	Average Write Response Time
4	OLTP 8K	16532.37	11038.59	5493.77	86.239	42.920	8.2680	6.6676
8	OLTP 8K	17529.76	11715.30	5814.45	91.526	45.425	14.8061	14.1376
16	OLTP 8K	18144.19	12135.76	6008.43	94.811	46.941	28.2200	28.1466
32	OLTP 8K	16049.91	10727.70	5322.21	83.810	41.580	63.7445	63.7615
64	OLTP 8K	14129.55	9460.69	4668.86	73.912	36.475	144.4581	144.6405
128	OLTP 8K	12950.54	8667.40	4283.14	67.714	33.462	311.1710	313.9607
256	OLTP 8K	15620.07	10444.23	5175.84	81.596	40.436	372.5500	372.7866
Average		15850.91	10598.53	5252.39	82.801	41.034	134.7454	134.8716
Stdev		1826.21	1215.75	610.53	9.498	4.770	149.7454	150.6827

9.5. Input/Output Alignment

9.5.1. Description:

I/O alignment specifies the relation between the host transaction size and device partitioning within the array. The proper I/O alignment matches the application transaction size with the partition sizes on the storage array, whereby reducing the number of partition crossings. Partition crossing increases write latencies by increasing the number of requests placed on the backend of the array. According

to a VMware study¹, the proper SAN storage alignment increases maximum throughput by 62% (12% on average) and reduces latency by 33% (10% on average).

I/O alignment has the greatest affect on random writes, where misalignment causes two writes and two reads for each write request on the backend. Random reads will only be affected when partition boundaries are crossed causing two drive side reads from the controller per read request.

Prefetching and fast write caching capabilities shield the bulk of misalignment issues for sequential read and write requests. A small pint of contention may occur at the start and/or end access that requires a read-modify-write process.

9.5.2. Test Configuration

- Solaris 10 server with 4 processors running IOmeter.
- Thin device pool:
 - 16 Raid-6 14+2 DATADEVs at 65472 cylinders each
 - 1 TDEV metavolume at 1047552 cyl.
- Alignment sizes of 8KB, 32KB, 64KB, 768KB, 1536KB and 3072KB.
- Transaction sizes of 8KB and 64KB
- 100% random write access pattern
- 16 queue depth
- Each I/O alignment thin device test instance was prefaced with unbinding and binding of the TDEVs to clear the previously written extents.
- Fat device – 1 metavolume at 1047552 cyl created from 16 Raid-6 14+2 devices at 65472 cylinders each.
- No ramp test - uses a 60 second sample
- Steady state - uses a 180 second ramp-up followed by a 60 second sample

9.5.3. Observations:

- *8KB Transactions:*
 - The no ramp test showed the standard device with a 61% greater throughput and 153% less latency as compared to a thin device.
 - The steady state test showed the standard device with a 75% greater throughput and 368% less latency as compared to the thin device.
 - The steady state test for the thin device showed the larger alignment sizes (768KB-3072KB) with a 5X advantage in throughput and a 6.2X lower latency as compared to the smaller alignment sizes (8KB-64KB).

¹ "Recommendations for Aligning VMFS Partitions" [VMware white paper]

- The thin device no ramp test showed a low standard deviation of 542, indicating consistent throughput regardless of alignment size.

- **64KB Transactions:**

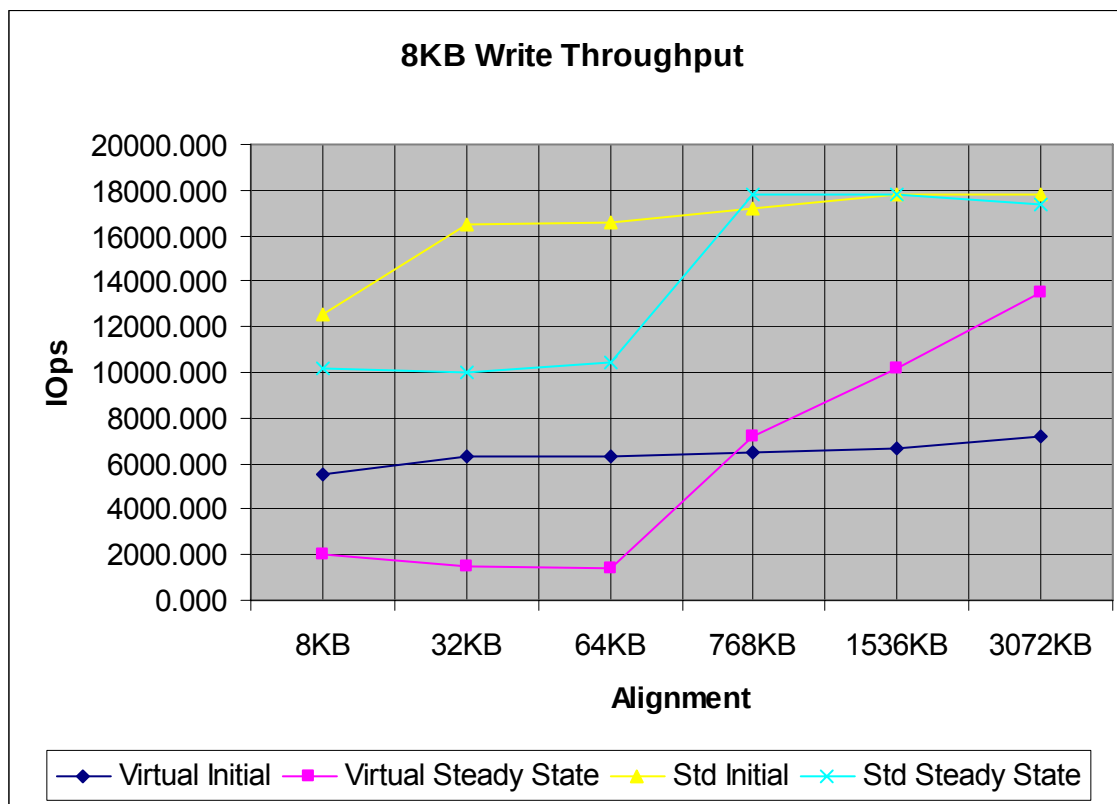
- The no ramp test showed the thin device with a 1.4% greater throughput and 0.7% less latency as compared to a thin device.
- The steady state test showed the standard device with a 9% greater throughput and 22% less latency as compared to the thin device.
- Except for the results of the steady state 8KB alignment test, the results for the remaining 23 tests showed similar performance between thin and standard devices.

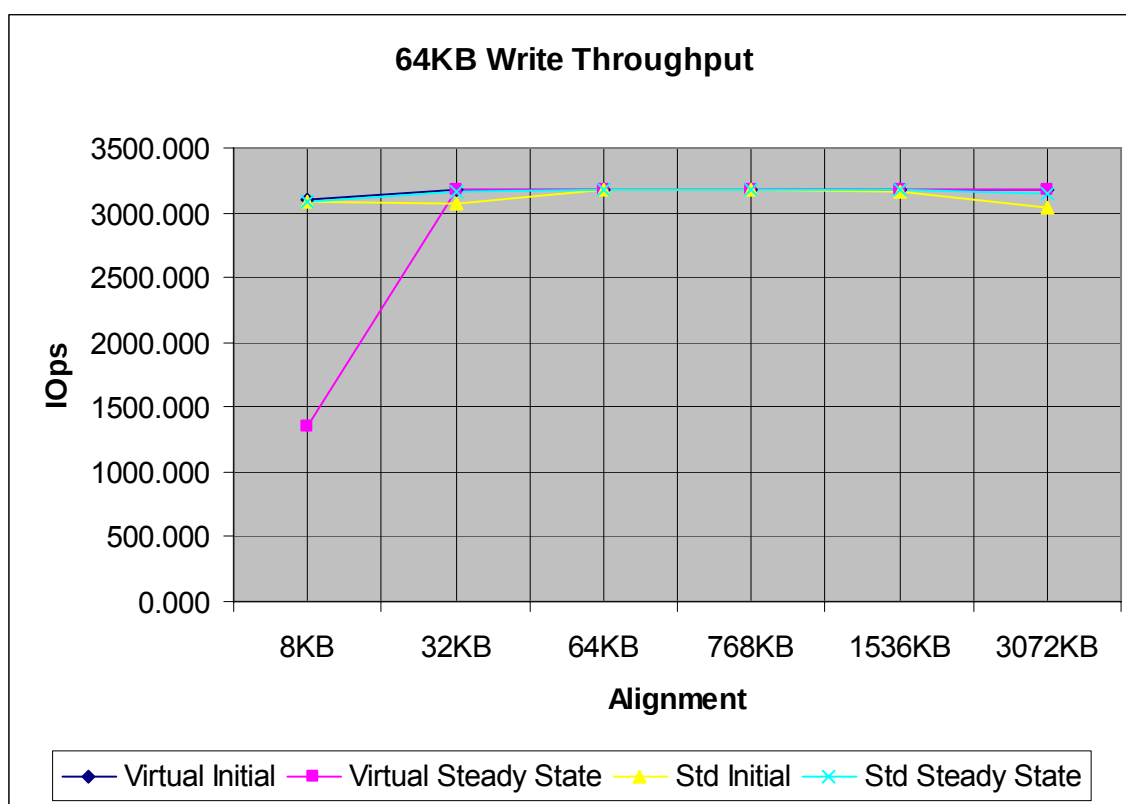
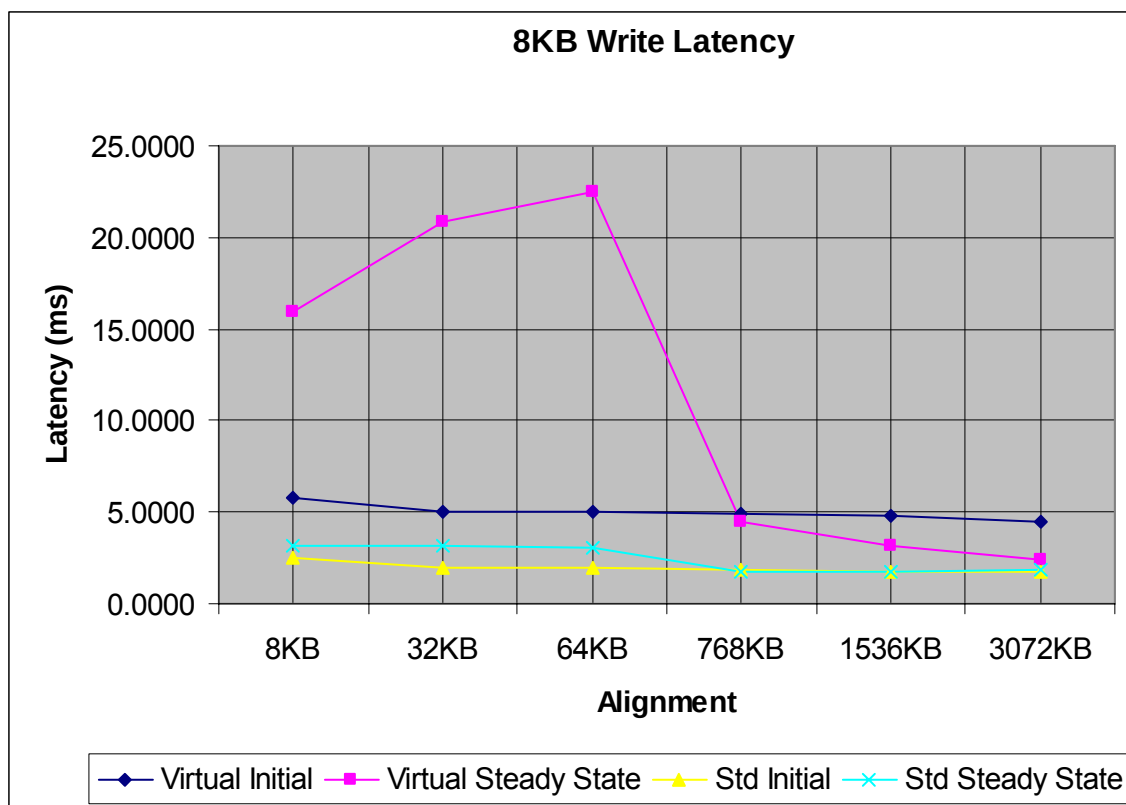
9.5.4. Conclusion:

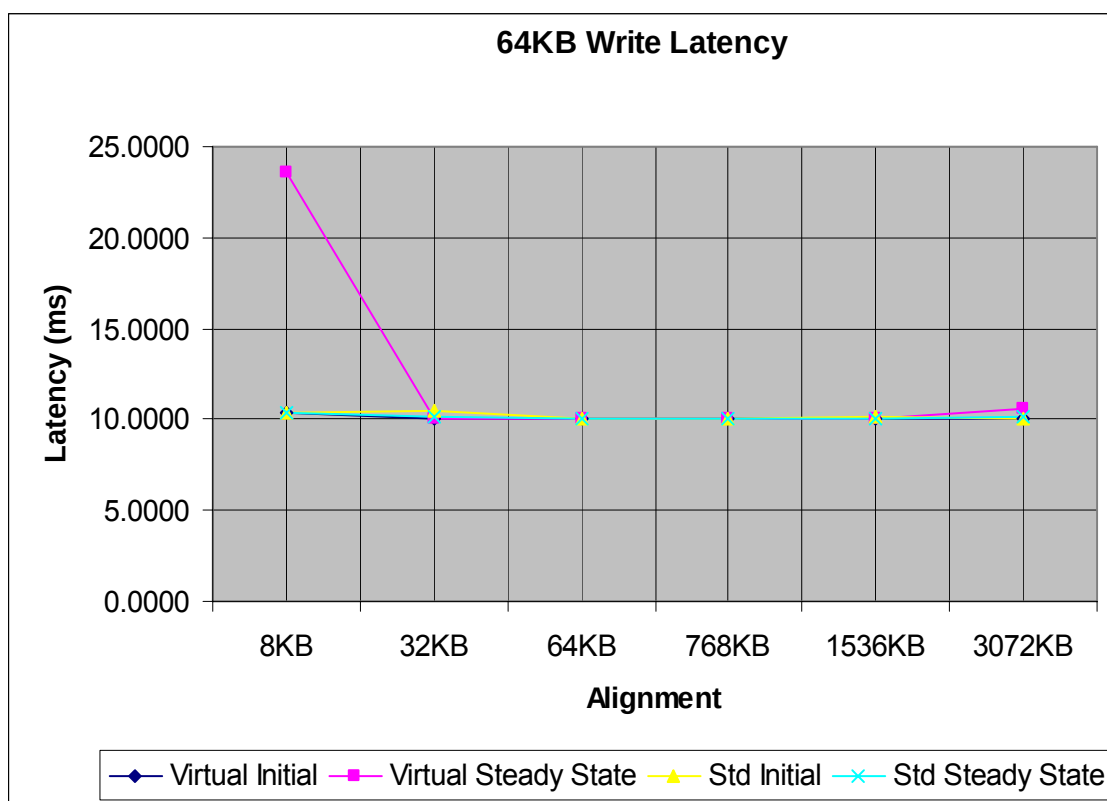
The 8KB transaction steady state test showed a 3.5X increase in throughput performance from sector to 768KB alignment. During the 60 second sampling period the 8KB transaction test wrote 1.94E6 tracks (127GB) at sector alignment and 2.24E6 tracks (147GB) at 768KB alignment. The 64KB transaction steady state test showed a 2.4X increase in throughput performance from sector to 32KB alignment. Results of the 60 second sample period the 64KB transaction test showed sector alignment wrote 1.16E6 tracks (76.17GB) and the 32KB alignment wrote 1.15E6 tracks (75.3GB) for a minuscule 300MB difference. The standard deviation for 8KB transactions is 5135, while the 64KB transaction is 7X less at 744. The low performance showed by the 8KB transaction between sector and 768K alignment maybe caused by a combination of the first write track penalty and I/O generator changing the write profile as the alignment size increased. To further understand this performance dichotomy, the test needs to be refined by removing most of the throughput assistance provide by the cache mechanism.

These test concentrated on the write throughput and response time, due to the first write and partial first write delays inherit with virtual provisioning. To conduct a comparison test on an equal basis, the write cache needs to be limited in order for destaging to occur with greater frequency. Setting the cache partitioning at 10% of available cache for the test DATADEVs and standard devices seemed to be the logical option. Unfortunately, when trying to assign cache partitioning for a DATADEV, an error message is presented indicating that cache partitioning for a thin device is not supported. Therefore, in order to obtain an equal basis test in comparing thin versus standard devices, cache partitioning was disabled for the entire frame.

9.5.5. Summary Graphs:







9.5.6. Summary Tables:

Alignment Test 8KB Random Write Transaction				
Virtual (thin) Device No Ramp				
Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time
16	8KB	5507.531	43.028	5.8087
16	32KB	6318.886	49.366	5.0626
16	64KB	6324.826	49.413	5.0579
16	768KB	6487.580	50.684	4.9310
16	1536KB	6628.336	51.784	4.8262
16	3072KB	7178.216	56.080	4.4564
Average		6407.563	50.059	5.0238
Std Dev		542.921	4.242	0.4446
Virtual (thin) Device Steady State				
Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time

16	8KB	2002.249	15.643	15.9620
16	32KB	1531.751	11.967	20.8687
16	64KB	1419.242	11.088	22.5345
16	768KB	7151.443	55.871	4.4723
16	1536KB	10182.675	79.552	3.1409
16	3072KB	13491.279	105.401	2.3703
Average		5963.106	46.587	11.5581
Std Dev		5135.412	40.120	9.2956

Standard (fat) Device No Ramp

Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time
16	8KB	12514.911	97.773	2.5554
16	32KB	16518.579	129.051	1.9355
16	64KB	16602.834	129.710	1.9258
16	768KB	17173.908	134.171	1.8617
16	1536KB	17796.520	139.035	1.7965
16	3072KB	17849.346	139.448	1.7912
Average		16409.350	128.198	1.9777
Std Dev		1989.894	15.546	0.2896

Standard (fat) Device Steady State

Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time
16	8KB	10132.220	79.158	3.1564
16	32KB	10041.160	78.447	3.1838
16	64KB	10476.311	81.846	3.0525
16	768KB	17828.944	139.289	1.7932
16	1536KB	17801.288	139.073	1.7960
16	3072KB	17373.319	135.729	1.8401
Average		13942.207	108.923	2.4703
Std Dev		4087.013	31.930	0.7251

Alignment Test 64KB Random Write Transaction
Virtual (thin) Device No Ramp

Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time
16	8KB	3094.051	193.378	10.3405
16	32KB	3179.696	198.731	10.0617
16	64KB	3180.135	198.758	10.0607
16	768KB	3180.126	198.758	10.0603
16	1536KB	3179.830	198.739	10.0613
16	3072KB	3174.718	198.420	10.0774

Average		3164.759	197.797	10.1103
Std Dev		34.703	2.169	0.1130
Virtual (thin) Device Steady State				
Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time
16	8KB	1354.951	84.684	23.5950
16	32KB	3179.696	198.731	10.0617
16	64KB	3179.817	198.739	10.0612
16	768KB	3180.126	198.758	10.0603
16	1536KB	3179.830	198.739	10.0613
16	3072KB	3174.718	198.420	10.5352
Average		2874.856	179.679	12.3958
Std Dev		744.601	46.538	5.4898
Standard (fat) Device No Ramp				
Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time
16	8KB	3089.559	193.097	10.3556
16	32KB	3067.102	191.694	10.4314
16	64KB	3179.323	198.708	10.0627
16	768KB	3176.550	198.534	10.0717
16	1536KB	3165.148	197.822	10.1075
16	3072KB	3036.678	189.792	10.0774
Average		3119.060	194.941	10.1844
Std Dev		62.318	3.895	0.1644
Standard (fat) Device Steady State				
Queue Depth	Input/Output Alignment	Write IOps	Write MBps	Average Write Response Time
16	8KB	3084.234	192.765	10.3714
16	32KB	3167.230	197.952	10.1011
16	64KB	3178.098	198.631	10.0668
16	768KB	3178.001	198.625	10.0669
16	1536KB	3176.014	198.501	10.0732
16	3072KB	3148.341	196.771	10.1618
Average		3155.319	197.207	10.1402
Std Dev		36.621	2.289	0.1189

9.6. Pre-allocation Capacity Throughput Test:

9.6.1. Description:

Pre-allocation of storage completes the first half of the IVTOC process by preparing the extents for the write request; therefore, shortening the time required to complete the first-write.

Note: According to EMC the time required for the bind operation is the same for thin devices with or without pre-allocation. The bind operation is followed by the pre-allocation, while the space being pre-allocated is available for use.

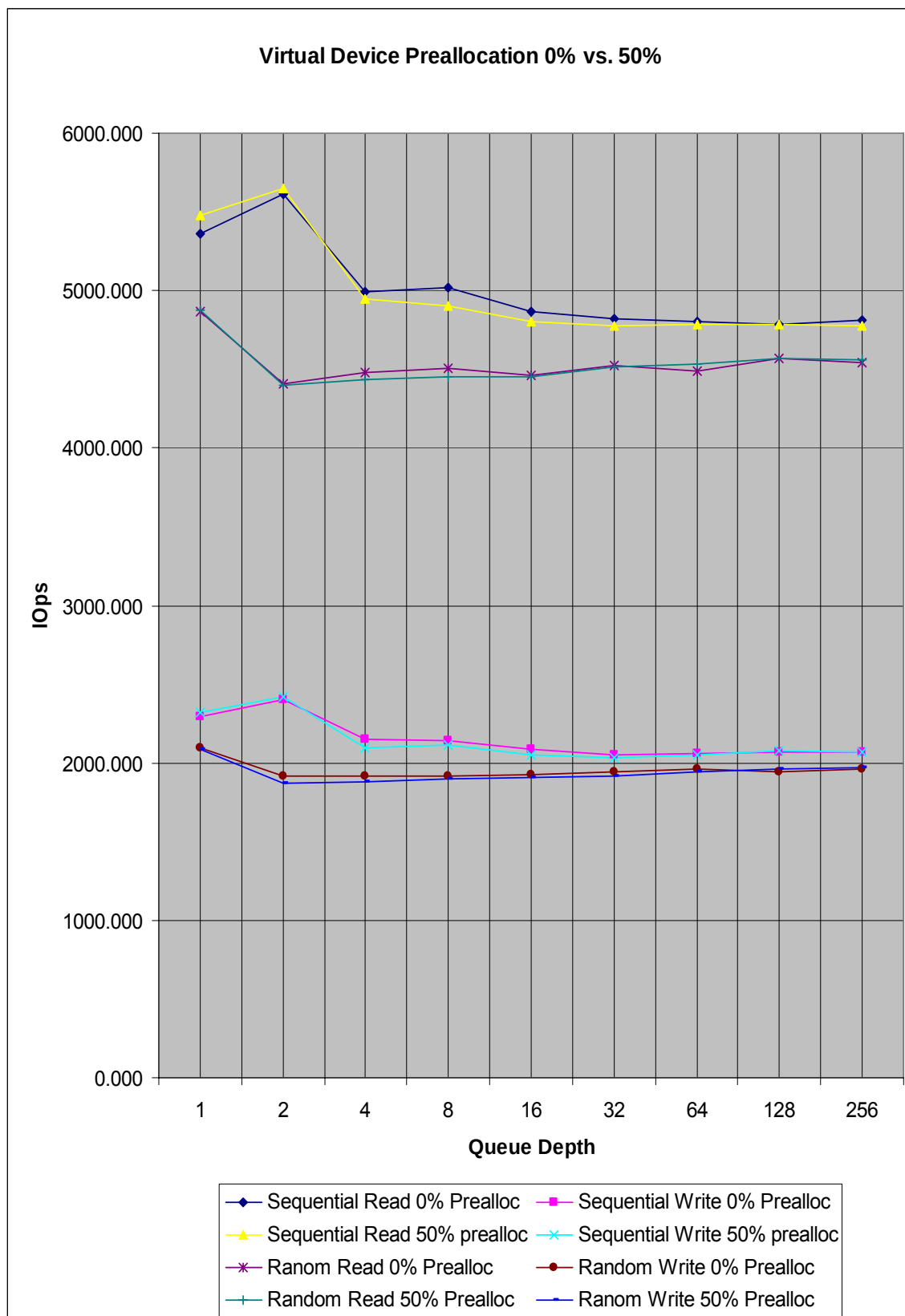
9.6.2. Test Configuration:

- Solaris 10 server with 4 processors
- 16 Raid-6 14+2 devices at 65472 cylinders
- Thin device pool:
 - 16 Raid-6 14+2 DATADEVs at 65472 cylinders
 - 16 TDEVs with no pre-allocation
 - 16 TDEVs with 32,736 cylinder pre-allocation
- I/O generation specifications:

'size	% of size	% reads	% random	delay	burst	align	reply
4096	70	100	0/100	0	4	0	0
4096	30	0	0/100	0	4	0	0
65536	70	100	0/100	0	4	0	0
65536	30	0	0/100	0	4	0	0

9.6.3. Observations:

As depicted in the plots below, pre-allocating capacity within the storage pool has little effect on throughput performance. This indicates that the first portion of the IVTOC process has little to do with the first write and partial-track first write penalty.



10. Functional Tests:

10.1. *Storage Pool Capacity Alerting*

10.1.1. Description:

EVP provides five alert variables to inform the users of state changes and current status of virtual devices and associated pools:

- Device Pool Configuration Change – Alerts the user that the status of the pool has changed as a result of adding or removing data devices or binding or unbinding thin devices.
- Device Pool Free Space – Alerts the user when the amount of free space remaining in a device pool meets or exceeds a specific percentage.
- Device Pool Status – Alerts the user if the state of a pool changes. Example, if the pool state changes from Enabled to Disabled because all of the data devices are disabled by the user, an alert will be triggered.
- Thin Device Allocation – Alerts the user when the amount of allocated space on a thin device meets or exceeds a specific percentage.
- Thin Device Usage – Alerts the user when the amount of used space on a thin device meets or exceeds a specific percentage.

Of primary importance for thin provisioning is the ability to monitor the consumption of pool storage space. EMC highly recommends that storage pools **not** surpass 80% of capacity allocation. A certain amount of free space is required to handle “runaway” host applications and/or draining of DATADEV(s). The Symmetrix places alert messages into the Symmetrix log at the following fixed consumption levels:

- 60% - issue a warning alert
- 65% - issue a warning alert
- 70% - issue a warning alert
- 80% - issue a critical alert
- 100% - issue a fatal alert

A manual method of monitoring the thin pool consumption is through the Solutions Enabler “symcfg monitor” command. The “symcfg monitor” allows a percent fill on a specific storage pool to be configured and monitor at set intervals. Once the percent fill is reached, an action script may be initiated to pass and SNMP alert message or place a message in a log file.

10.1.2. Observation:

- The following alert messages were evident in the log file during the test. The required information of Symmetrix ID, pool name, severity level and percent fill are present in the alert messages:
 - 40% allocation – information message:
[fmt=evt] [evtid=1208] [date=2008-04-1XXXXXXXX] [symid=000190103110]
[TPDataPool=glab1] [sev=normal] = Data Pool utilization is now 40 percent.
 - 65% allocation – minor severity
[fmt=evt] [evtid=1208] [date=2008-04-1XXXXXXXX] [symid=000190103110]
[TPDataPool=glab1] [sev=minor] = Data Pool utilization is now 65 percent.
 - 70% allocation – major severity
[fmt=evt] [evtid=1208] [date=2008-04-1XXXXXXXX] [symid=000190103110]
[TPDataPool=glab1] [sev=major] = Data Pool utilization is now 70 percent.
 - 75% allocation – major severity
[fmt=evt] [evtid=1208] [date=2008-04-1XXXXXXXX] [symid=000190103110]
[TPDataPool=glab1] [sev=major] = Data Pool utilization is now 75 percent.
- Symmetrix event log requires the “storevntd” daemon on the Solutions Enabler servers to be operational. Therefore, collecting and routing of the thin provisioning log messages relies on the stability of the Solutions Enabler server and the “symevntd” being operational.
- The alert messages depicted in the log file were slightly different in than displayed in the manual; however, the primary values such as the Symmetrix ID, pool name, severity level and percent fill were evident in the log messages.
- The “symcfg monitor” command allows a Solution Enabler user to set a threshold on percent fill for a specific storage pool and initiate a script to define an action. This action may be scripted to send a message to a log or central monitoring server. However, since this is a basic SYMCLI command, failure of the host or stopping of the Solution Enabler daemon will halt this function.

According to EMC they did not receive a call home event indicating that a storage pool threshold level has been violated. I have requested EMC to provide an option where surpassing the 75% threshold constitutes a call home event. At this point EMC can alert Citi of this threshold violation through verbal and electronic communication.

10.1.3. Procedure: 70% Fill Threshold Violation

1. Verify the operational status of the “storevntd” daemon:

```
# stord daemon list
```

```
Available Daemons ('[*]': Currently Running):
```

```
[*]storapid      EMC Solutions Enabler Base Daemon
[*]storgnsd      EMC Solutions Enabler GNS Daemon
    storrdfd      EMC Solutions Enabler RDF Daemon
[*]storevntd     EMC Solutions Enabler Event Daemon
[*]storwatchd    EMC Solutions Enabler Watchdog Daemon
    storsrmd      EMC Solutions Enabler SRM Daemon
    storstp      EMC Solutions Enabler STP Daemon
[*]storsrvd      EMC Solutions Enabler SYMAPI Server Daemon
```

2. Disabling a DATADEV causes the storage pool to pass the 70% fill threshold:

```
Symmetrix ID: 000190103110
```

```
Symmetrix ID      : 000190103110
Pool Name         : glab1
Pool Type         : Thin
Dev Emulation     : FBA
Dev Configuration : RAID-6(14+2)
Pool State        : Enabled
# of Devices in Pool : 16
# of Enabled Devices in Pool : 15
Max. Subscription Percent : None
```

```
Enabled Devices(15):
```

```
{
-----
Sym      Total      Alloc      Free Full Device
Dev      Tracks     Tracks     Tracks  (%)  State
-----
0354      982080     626664     355416   63  Enabled
0355      982080     640356     341724   65  Enabled
0356      982080     654768     327312   66  Enabled
0357      982080     653460     328620   66  Enabled
0358      982080     401736     580344   40  Enabled
0359      982080     699252     282828   71  Enabled
035A      982080     682308     299772   69  Enabled
035B      982080     652464     329616   66  Enabled
035C      982080     733476     248604   74  Enabled
035D      982080     787716     194364   80  Enabled
035E      982080     795240     186840   80  Enabled
035F      982080     689868     292212   70  Enabled
0360      982080     890268     91812    90  Enabled
0361      982080     923328     58752    94  Enabled
0362      982080     924012     58068    94  Enabled
-----
Tracks    14731200    10754916    3976284    73
}
```

Disabled Devices(1):

Sym Dev	Total Tracks	Alloc Tracks	Free Tracks	Full (%)	Device State
0353	982080	0	982080	0	Disabled
Tracks	982080	0	982080	0	

Thin Devices(16):

Sym Dev	Total Tracks	Pool Subs (%)	Pool Allocated Tracks (%)	Pool Written Tracks (%)	Status
0363	982080	7	672492	5	78284 1 Bound
0364	982080	7	672264	5	77658 1 Bound
0365	982080	7	672264	5	78106 1 Bound
0366	982080	7	671928	5	77432 1 Bound
0367	982080	7	672552	5	77823 1 Bound
0368	982080	7	672036	5	77670 1 Bound
0369	982080	7	672384	5	78217 1 Bound
036A	982080	7	672108	5	77461 1 Bound
036B	982080	7	672348	5	78278 1 Bound
036C	982080	7	672096	5	77915 1 Bound
036D	982080	7	672084	5	77957 1 Bound
036E	982080	7	672048	5	77543 1 Bound
036F	982080	7	672312	5	77996 1 Bound
0370	982080	7	671952	5	77684 1 Bound
0371	982080	7	672000	5	78035 1 Bound
0372	982080	7	672048	5	77742 1 Bound
Tracks	15713280	107	10754916	73	1245801 8

3. An alert placed in the Symmetrix event log depicts the threshold violation:

Symmetrix Event Log Output:

[fmt=evt] [evtid=1207] [date=2008-04-09T09:45:21]

[symid=000190103110] [TPDataPool=glab1] [sev=info] = Data Pool configuration has changed.

[fmt=evt] [evtid=1208] [date=2008-04-09T09:46:23]

[symid=000190103110] [TPDataPool=glab1] [sev=major] = Data Pool utilization is now 70 percent.

10.2. Storage Pool Capacity Utilization

10.2.1. Description:

EMC bases the storage pool capacity thresholds on the DATADEV percent full value; therefore, the accuracy of this value is important in determining when additional capacity is required for a storage pool.

10.2.2. Observation:

In analyzing the two examples depicted below, it is difficult to derive the actual storage pool allocation with the provided status tools. In example #1, an I/O generator was used to simulate writes to increase the storage pool allocation, which resulted in the DATADEVs showing 33% full while the TDEVs showed 7% written tracks. In example #2, writes to a single TDEV were generated by a low-level format process, which displayed an aggregate percent full of 6% for the DATADEVs and the single TDEV showed 100% (982052 tracks) allocated.

10.2.3. Conclusion:

The I/O profile of the data written to the TDEVs plays an important role in determining an accurate allocated percentage. Both examples depict below shows the TDEV utilization at 6-7%, yet the DATADEV utilization differs by 5.5 times. This dichotomy in DATADEV percentage utilization maybe attributed to the fill amount for each extent created. Much of the accuracy in determining DATADEV percent fill is based on the I/O profile; therefore making it difficult to determining actual allocation capacities.

EMC acknowledges that accurate allocation percentages may not be possible at the infancy of the storage pool, due to the empty tracks in each extent. As the pool matures an allocation increases the extent get filled in and the DATADEV allocation percentage becomes more accurate. Citi needs to have an accurate allocation of every storage pool regardless of maturity; therefore, EMC must provide comprehensive monitoring tool(s) to accurate depict capacity allocation.

1.1.1.1. Example #1: I/O Generator

```
# symcfg show -sid 3110 -pool glab1 -thin -detail -all
```

```
Symmetrix ID: 000190103110
```

```
Symmetrix ID           : 000190103110
Pool Name              : glab1
Pool Type              : Thin
Dev Emulation          : FBA
Dev Configuration     : RAID-6(14+2)
Pool State             : Enabled
# of Devices in Pool   : 16
# of Enabled Devices in Pool : 16
```

Max. Subscription Percent : None

Enabled Devices(16):

{

Sym Dev	Total Tracks	Alloc Tracks	Free Tracks	Full (%)	Device State
0353	982080	336132	645948	34	Enabled
0354	982080	336336	645744	34	Enabled
0355	982080	336216	645864	34	Enabled
0356	982080	336768	645312	34	Enabled
0357	982080	338424	643656	34	Enabled
0358	982080	225540	756540	22	Enabled
0359	982080	336036	646044	34	Enabled
035A	982080	336120	645960	34	Enabled
035B	982080	335628	646452	34	Enabled
035C	982080	335880	646200	34	Enabled
035D	982080	335772	646308	34	Enabled
035E	982080	335916	646164	34	Enabled
035F	982080	335580	646500	34	Enabled
0360	982080	334728	647352	34	Enabled
0361	982080	334416	647664	34	Enabled
0362	982080	334332	647748	34	Enabled

Tracks	15713280	5263824	10449456	33
--------	----------	---------	----------	-----------

}

Thin Devices(16):

{

Sym Dev	Total Tracks	Pool Subs (%)	Pool Allocated Tracks	Pool (%)	Pool Written Tracks	Pool (%)	Status
0363	982080	6	328428	2	67278	0	Bound
0364	982080	6	329112	2	67232	0	Bound
0365	982080	6	328392	2	67275	0	Bound
0366	982080	6	329160	2	67322	0	Bound
0367	982080	6	329052	2	67226	0	Bound
0368	982080	6	329052	2	67382	0	Bound
0369	982080	6	329016	2	67190	0	Bound
036A	982080	6	329520	2	67475	0	Bound
036B	982080	6	329496	2	67486	0	Bound
036C	982080	6	328596	2	67209	0	Bound
036D	982080	6	328896	2	67245	0	Bound
036E	982080	6	328560	2	67203	0	Bound
036F	982080	6	329064	2	67418	0	Bound
0370	982080	6	330072	2	67575	0	Bound
0371	982080	6	328428	2	67169	0	Bound
0372	982080	6	328980	2	67203	0	Bound
Tracks	15713280	100	5263824	33	1076888	7	

}

Note that the written track shows an aggregate written value of 7%, yet each individual TDEV shows 0%. Each TDEV will use a minimum of 0.48% of the capacity by default.

1.1.1.2. Example #2: Low-level Format

```
# symcfg show -sid 3110 -pool glab1 -thin -detail -all
```

```
Symmetrix ID: 000190103110
```

```
Symmetrix ID           : 000190103110
Pool Name              : glab1
Pool Type              : Thin
Dev Emulation          : FBA
Dev Configuration      : RAID-6(14+2)
Pool State             : Enabled
# of Devices in Pool   : 16
# of Enabled Devices in Pool : 16
Max. Subscription Percent : None
```

```
Enabled Devices(16):
```

```
{
-----
Sym      Total      Alloc      Free Full  Device
Dev      Tracks     Tracks     Tracks (%)  State
-----
0353      982080      61404      920676    6 Enabled
0354      982080      61392      920688    6 Enabled
0355      982080      61404      920676    6 Enabled
0356      982080      61404      920676    6 Enabled
0357      982080      61392      920688    6 Enabled
0358      982080      61392      920688    6 Enabled
0359      982080      61380      920700    6 Enabled
035A      982080      61380      920700    6 Enabled
035B      982080      61380      920700    6 Enabled
035C      982080      61380      920700    6 Enabled
035D      982080      61392      920688    6 Enabled
035E      982080      61392      920688    6 Enabled
035F      982080      61392      920688    6 Enabled
0360      982080      61392      920688    6 Enabled
0361      982080      61392      920688    6 Enabled
0362      982080      61392      920688    6 Enabled
-----
Tracks    15713280      982260      14731020    6
}
```

```
Thin Devices(16):
```

```
{
-----
Sym      Total      Pool      Pool      Pool
Dev      Tracks     Subs      Allocated  Written
          (%)      Tracks    (%)      Tracks    (%)  Status
-----
0363      982080      6         12        0         1    0 Bound
0364      982080      6         12        0         1    0 Bound
-----
```

0365	982080	6	12	0	1	0	Bound
0366	982080	6	12	0	1	0	Bound
0367	982080	6	12	0	1	0	Bound
0368	982080	6	12	0	1	0	Bound
0369	982080	6	12	0	1	0	Bound
036A	982080	6	12	0	1	0	Bound
036B	982080	6	12	0	1	0	Bound
036C	982080	6	12	0	1	0	Bound
036D	982080	6	12	0	1	0	Bound
036E	982080	6	12	0	1	0	Bound
036F	982080	6	12	0	1	0	Bound
0370	982080	6	12	0	1	0	Bound
0371	982080	6	12	0	1	0	Bound
0372	982080	6	982080	6	982052	6	Bound

Tracks	15713280	100	982260	6	982067	6	

10.3. RAID-6 Data Device Sparing

10.3.1. Description:

According to EMC Thin Provisioning Best Practice documents, DATADEVs only supports dynamic sparing; unfortunately, RAID-6 only supports permanent drive sparing. The device-level widely striped data encoding makes a DATADEV failure a storage pool wide event.

10.3.2. Observation:

EMC has retracted their statement in their best practices document indicating that permanent member sparing of DATADEVs is not supported. Therefore, EMC like Citi sees tremendous benefit in utilizing RAID-6 from a cost, protection and performance standpoint.

In previous DMX-4 performance tests on standard devices, a 25% loss in throughput was displayed in a single degraded mode (one drive failure) for RAID-6 devices.

The device-level widely striped data encoding is a tremendous benefit to performance by distributing the data across the storage pool. The drawback to widely striped data is a single drive failure has the potential to corrupt the entire pool. To minimize the possibility of data corruption on EVP supported arrays, EMC must replace bad drives within the time specified by the maintenance contract.

10.4. Storage Pool Data Distribution

10.4.1. Description:

To remove DATADEV(s) from a storage pool it must first be disabled, which includes draining of any allocated space to the remaining DATADEVs in the storage pool. The draining procedure allows the TDEVs to remain intact by changing the pointers to the new location in the remaining DATADEVs. Once drained, the DATADEV can be removed from the storage pool.

10.4.2. Observations:

Removing or replacing a DATADEV has a detrimental effect on the distribution of the allocated data. The unequal allocation of data as depicted in the following example, where eight DATADEVs are 100% allocated and four are less than 50% allocated. This unequal distribution may nullify some of the performance aspect of EVP's widely striped data distribution.

Requested that EMC provide some type of redistribution tool that maybe manually or automatically initiated during low I/O periods to equalize the data distribution.

```
# symcfg show -sid 3110 -pool glab1 -thin -detail -all
```

```
Symmetrix ID: 000190103110
```

```
Symmetrix ID           : 000190103110
Pool Name              : glab1
Pool Type              : Thin
Dev Emulation          : FBA
Dev Configuration      : RAID-6(14+2)
Pool State             : Enabled
# of Devices in Pool   : 16
# of Enabled Devices in Pool : 14
Max. Subscription Percent : None
```

```
Enabled Devices(14):
```

```
{
```

Sym Dev	Total Tracks	Alloc Tracks	Free Tracks	Full (%)	Device State
0353	982080	538452	443628	54	Enabled
0356	982080	982080	0	100	Enabled
0357	982080	982080	0	100	Enabled
0358	982080	940512	41568	95	Enabled
0359	982080	982080	0	100	Enabled
035A	982080	982080	0	100	Enabled
035B	982080	982080	0	100	Enabled
035C	982080	982080	0	100	Enabled
035D	982080	982080	0	100	Enabled
035E	982080	265128	716952	26	Enabled
035F	982080	267972	714108	27	Enabled
0360	982080	982080	0	100	Enabled
0361	982080	409344	572736	41	Enabled
0362	982080	476868	505212	48	Enabled

```
Tracks 13749120 10754916 2994204 78
}
```

Disabled Devices(2):

Sym Dev	Total Tracks	Alloc Tracks	Free Tracks	Full (%)	Device State
0354	982080	0	982080	0	Disabled
0355	982080	0	982080	0	Disabled
Tracks	1964160	0	1964160	0	

Thin Devices(16):

Sym Dev	Total Tracks	Pool Subs (%)	Pool Allocated Tracks (%)	Pool Written Tracks (%)	Status
0363	982080	7	672492	5	78285 1 Bound
0364	982080	7	672264	5	77661 1 Bound
0365	982080	7	672264	5	78109 1 Bound
0366	982080	7	671928	5	77434 1 Bound
0367	982080	7	672552	5	77825 1 Bound
0368	982080	7	672036	5	77672 1 Bound
0369	982080	7	672384	5	78220 1 Bound
036A	982080	7	672108	5	77463 1 Bound
036B	982080	7	672348	5	78283 1 Bound
036C	982080	7	672096	5	77918 1 Bound
036D	982080	7	672084	5	77957 1 Bound
036E	982080	7	672048	5	77545 1 Bound
036F	982080	7	672312	5	77996 1 Bound
0370	982080	7	671952	5	77688 1 Bound
0371	982080	7	672000	5	78039 1 Bound
0372	982080	7	672048	5	77744 1 Bound
Tracks	15713280	114	10754916	78	1245839 9

10.5. Circumvention of Virtual Provisioning Objectives

10.5.1. Description:

The primary objective of EVP is to provide storage on a demand bases, thereby allowing an application to mature along with its storage needs. However, loopholes exist that allow users to circumvent EVP's storage saving objective by providing a false illusion that the entire or majority portion of the TDEV capacity is needed.

10.5.2. Observation:

The inappropriate reservation of virtual storage maybe accomplished through low-level formatting, coping zero byte blocks or creating substitution files to fill in the device. A Solaris low-level format showed the TDEV allocation at 982080 tracks or 100% of the TDEVs (device ID) capacity.

The Unix “data definition (dd)” is a low-level copying and conversion of raw fixed blocks sizes can be copied from one device to another. The “dd” command can be used to copy data from an empty source TDEV to a target TDEV to falsely allocate storage pool capacity. As depicted in section 10.5.4, the source TDEV (device ID 0369) did not increase in allocation while the target TDEV (device ID 0363) written tracks increased from 3 to 34773.

EVP returns a block where each byte is equal to zero, when a read request is generated to an area that has not been mapped.

The dd utility copies the specified input file to the specified output with possible conversions. The standard input and output are used by default. The input and output block sizes may be specified to take advantage of raw physical I/O. Sizes are specified in bytes; a number may end with k, b, or w to specify multiplication by 1024, 512, or 2, respectively. Numbers may also be separated by x to indicate multiplication.

10.5.3. Conclusion:

EVP increases storage usage efficiency; however, improper use can nullify any virtual device capacity benefit. This creates a dilemma where instructing the users on procedures **not** to run provides the formula to circumvent EVP’s storage savings objective.

Citi has request of tool with the ability to query the top 10% of TDEVs in terms of rate of allocation and/or TDEVs that are allocating capacity at greater than 10% per day. This query ability should provide the means to uncover users who are falsely reserving virtual storage.

10.5.4. Data Definition Low-level Copy Test

1. Initiate dd process:

```
Inq:
/dev/vx/rdmp/emcpower199s2      :EMC      :SYMMETRIX      :5773
:10369000      :62853120
/dev/vx/rdmp/emcpower200s2      :EMC      :SYMMETRIX      :5773
:1036E000      :62853120

# dd if=/dev/rdisk/emcpower199c of=/dev/rdisk/emcpower202c
bs=1024
^C2225331+0 records in
2225331+0 records out
```

2. Collect virtual device status:

```
Symmetrix ID: 000190103110

Symmetrix ID      : 000190103110
```

```

Pool Name           : glab1
Pool Type           : Thin
Dev Emulation       : FBA
Dev Configuration   : RAID-6(14+2)
Pool State          : Enabled
# of Devices in Pool : 16
# of Enabled Devices in Pool : 16
Max. Subscription Percent : None

```

Enabled Devices(16):

```

{
-----
Sym      Total      Alloc      Free Full  Device
Dev      Tracks     Tracks     Tracks (%)  State
-----
0353     982080     280500     701580  28  Enabled
0354     982080     283620     698460  28  Enabled
0355     982080     289560     692520  29  Enabled
0356     982080     284520     697560  28  Enabled
0357     982080     273696     708384  27  Enabled
0358     982080     269424     712656  27  Enabled
0359     982080     259584     722496  26  Enabled
035A     982080     267120     714960  27  Enabled
035B     982080     271560     710520  27  Enabled
035C     982080     284604     697476  28  Enabled
035D     982080     280932     701148  28  Enabled
035E     982080     268152     713928  27  Enabled
035F     982080     255696     726384  26  Enabled
0360     982080     270156     711924  27  Enabled
0361     982080     272484     709596  27  Enabled
0362     982080     273564     708516  27  Enabled
-----
Tracks   15713280   4385172   11328108  27
}

```

Thin Devices(16):

```

{
-----
Sym      Total      Pool      Pool      Pool
Dev      Tracks     Subs     Allocated  Written
              (%)   Tracks   (%)   Tracks   (%)  Status
-----
0363     982080      6      34800      0      34773      0  Bound
0364     982080      6      300024      2           3      0  Bound
0365     982080      6      300024      2           3      0  Bound
0366     982080      6      300024      2           3      0  Bound
0367     982080      6      300024      2           3      0  Bound
0368     982080      6      300024      2           3      0  Bound
0369     982080      6        36      0          3      0  Bound
036A     982080      6      300024      2           3      0  Bound
036B     982080      6      300024      2           3      0  Bound
036C     982080      6      300024      2           3      0  Bound
036D     982080      6      300024      2           3      0  Bound
036E     982080      6      300024      2           3      0  Bound
036F     982080      6      300024      2           3      0  Bound
0370     982080      6      300024      2           3      0  Bound
0371     982080      6      300024      2           3      0  Bound
0372     982080      6      450024      3           3      0  Bound
-----
Tracks   15713280   100     4385172   28     34818      0
}

```

10.6. Storage Pool Data Security

10.6.1. Description:

Determine the security of the data from being view by another host within the same storage pool.

10.6.2. Observations:

The unbinding process removes the pointer relationship between the TDEVs and DATADEVs, rendering the data inaccessible. The data written to the DATADEVs are intact until written over; however, the device-level wide striping of the data makes recreating the original data nearly impossible. This is evident when unbinding and rebinding the same TDEV to a host, requires the operating system to write a new label indicating that the original disk geometry and drive characteristics information has been removed.

10.6.3. Conclusion:

The unbinding process is required to remove the access table between the TDEV and DATADEV. If the TDEV is not unbound and rebound before being re-used in the same storage pool, the original data written to the TDEV is compromised.

10.6.4. Data Accessibility Test:

1. Unbind TDEV from the pool:

```
# symdev -sid 3110 not_ready 0372

Execute a 'Not Ready' Device operation for device '0372' (y/[n]) ? y

'Not Ready' Device operation successfully completed for the device.

# symconfigure -sid 3110 -cmd "unbind tdev 0372 from pool glab1;" commit -v -nop

A Configuration Change operation is in progress. Please wait...

Establishing a configuration change session.....Established.
Processing symmetrix 000190103110
{
  unbind tdev device 0372 from pool glab1;
}

Performing Access checks.....Allowed.
Checking Device Reservations.....Allowed.
Locking devices.....Locked.
Validating configuration changes.....Validated.
Preparing configuration changes.....Prepared.
Committing configuration changes.....Started.
Committing configuration changes.....Committed.
Terminating the configuration change session.....Done.

The configuration change session has successfully completed.

# symcfg show -sid 3110 -pool glab1 -thin -detail -all
```

Symmetrix ID: 000190103110

Symmetrix ID : 000190103110
 Pool Name : glab1
 Pool Type : Thin
 Dev Emulation : FBA
 Dev Configuration : RAID-6(14+2)
 Pool State : Enabled
 # of Devices in Pool : 16
 # of Enabled Devices in Pool : 16
 Max. Subscription Percent : None

Enabled Devices(16):

```
{
-----
Sym      Total      Alloc      Free Full  Device
Dev      Tracks     Tracks     Tracks (%)  State
-----
0353     982080     50304     931776   5  Enabled
0354     982080     50292     931788   5  Enabled
0355     982080     50304     931776   5  Enabled
0356     982080     50304     931776   5  Enabled
0357     982080     50292     931788   5  Enabled
0358     982080     50292     931788   5  Enabled
0359     982080     50280     931800   5  Enabled
035A     982080     50292     931788   5  Enabled
035B     982080     50280     931800   5  Enabled
035C     982080     50280     931800   5  Enabled
035D     982080     50292     931788   5  Enabled
035E     982080     50292     931788   5  Enabled
035F     982080     50292     931788   5  Enabled
0360     982080     50292     931788   5  Enabled
0361     982080     50292     931788   5  Enabled
0362     982080     50292     931788   5  Enabled
-----
Tracks   15713280    804672    14908608   5
}
```

Thin Devices(16):

```
{
-----
Sym      Total      Pool      Pool      Pool
Dev      Tracks     Subs      Allocated  Written
              (%)      Tracks    (%)      Tracks    (%)  Status
-----
0363     982080      6         12      0         1      0  Bound
0364     982080      6         12      0         1      0  Bound
0365     982080      6         12      0         1      0  Bound
0366     982080      6         12      0         1      0  Bound
0367     982080      6         12      0         1      0  Bound
0368     982080      6         12      0         1      0  Bound
0369     982080      6         12      0         1      0  Bound
036A     982080      6         12      0         1      0  Bound
036B     982080      6         12      0         1      0  Bound
036C     982080      6         12      0         1      0  Bound
036D     982080      6         12      0         1      0  Bound
036E     982080      6         12      0         1      0  Bound
036F     982080      6         12      0         1      0  Bound
0370     982080      6         12      0         1      0  Bound
0371     982080      6         12      0         1      0  Bound
0372     982080      6      800388   5      800360   5  Unbinding
-----
Tracks   15713280    100      800568   5      800375   5
}
```

2. Verified that the original data associated with the TDEV could not be read. As evident by the Unix format command requesting a new label be written to the test volume:

```

42. emcpower191a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@191
43. emcpower192a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@192
44. emcpower193a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@193
45. emcpower194a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@194
46. emcpower195a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@195
47. emcpower196a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@196
48. emcpower197a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@197
49. emcpower198a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@198
50. emcpower199a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@199
51. emcpower200a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@200
52. emcpower201a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@201
53. emcpower202a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@202
54. emcpower203a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@203
55. emcpower204a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@204
56. emcpower205a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@205
57. emcpower206a <EMC-SYMMETRIX-5773 cyl 65470 alt 2 hd 15 sec 128>
    /pseudo/emcp@206

```

```

Specify disk (enter its number): 42
selecting emcpower191a
[disk formatted]
Disk not labeled. Label it now?

```

10.7. *Modifying Allocated Pool Space*

10.7.1. **Description:**

EVP allows for the freeing or expanding of pre-allocated capacity to an online TDEV. Pre-allocation provides for the reservation of storage capacity and preparing the extents for future write request. The initial pre-allocation of capacity is assigned through the TDEV binding process, and once bound the pre-allocated capacity maybe increased or decreased while the TDEV is bound to the storage pool.

10.7.2. Observation:

Expanding the TDEV allocation is possible when the user specifies the region of the TDEV to expand (see example 1). A portion or all of the remaining TDEV capacity maybe pre-allocated.

Capacity that has been pre-allocated but not written to maybe freed and returned the storage pool as unmapped capacity. The process of freeing and increasing allocation proved to be stable and requiring a short period to drain or allocate capacity (see example 2).

10.7.3. Increase Pre-allocated Capacity Test

Increase an existing TDEV's allocation by 10000 cylinders:

```
# symconfigure -sid 3110 -cmd "allocate tdev 372 start_cyl=20000 size=10000;"
commi>
```

A Configuration Change operation is in progress. Please wait...

```
Establishing a configuration change session.....Established.
Processing symmetrix 000190103110
{
  allocate tdev 0372 start_cyl=20000 cyl
    size = 10000 cyl;
}

Performing Access checks.....Allowed.
Checking Device Reservations.....Allowed.
Locking devices.....Locked.
Validating configuration changes.....Validated.
Preparing configuration changes.....Prepared.
Committing configuration changes.....Started.
Committing configuration changes.....Committed.
Terminating the configuration change session.....Done.
```

The configuration change session has successfully completed.

```
root@sangs7(/var/symapi/log)# symcfg show -sid 3110 -pool glab1 -thin -detail -all
```

Symmetrix ID: 000190103110

```
Symmetrix ID           : 000190103110
Pool Name              : glab1
Pool Type              : Thin
Dev Emulation          : FBA
Dev Configuration     : RAID-6(14+2)
Pool State             : Enabled
# of Devices in Pool   : 16
# of Enabled Devices in Pool : 16
Max. Subscription Percent : None
```

Enabled Devices(16):

```
{
-----
Sym      Total      Alloc      Free Full  Device
Dev      Tracks     Tracks     Tracks  (%)   State
-----
0353     982080     307236     674844   31   Enabled
0354     982080     311592     670488   31   Enabled
0355     982080     316608     665472   32   Enabled
0356     982080     311328     670752   31   Enabled
```

```

0357      982080      299136      682944      30 Enabled
0358      982080      295116      686964      30 Enabled
0359      982080      284928      697152      29 Enabled
035A      982080      292296      689784      29 Enabled
035B      982080      297036      685044      30 Enabled
035C      982080      311124      670956      31 Enabled
035D      982080      307932      674148      31 Enabled
035E      982080      294204      687876      29 Enabled
035F      982080      279216      702864      28 Enabled
0360      982080      294276      687804      29 Enabled
0361      982080      298008      684072      30 Enabled
0362      982080      300156      681924      30 Enabled

```

```

-----
Tracks    15713280    4800192    10913088    30
}

```

Thin Devices(16):

```

{
-----
Sym      Total      Pool      Pool      Pool
Dev      Tracks     Subs      Allocated  Written
                               Tracks      (%)      Tracks      (%)      Status
-----
0363      982080      6         150012      1         1      0      Bound
0364      982080      6         300012      2         1      0      Bound
0365      982080      6         300012      2         1      0      Bound
0366      982080      6         300012      2         1      0      Bound
0367      982080      6         300012      2         1      0      Bound
0368      982080      6         300012      2         1      0      Bound
0369      982080      6         300012      2         1      0      Bound
036A      982080      6         300012      2         1      0      Bound
036B      982080      6         300012      2         1      0      Bound
036C      982080      6         300012      2         1      0      Bound
036D      982080      6         300012      2         1      0      Bound
036E      982080      6         300012      2         1      0      Bound
036F      982080      6         300012      2         1      0      Bound
0370      982080      6         300012      2         1      0      Bound
0371      982080      6         300012      2         1      0      Bound
0372      982080      6         450012      3         1      0      Bound
-----
Tracks    15713280    100      4800192    31         16      0
}

```

10.7.4. Freeing Pre-allocation Capacity Test

Remove all pre-allocation capacity from a TDEV:

```
# symconfigure -sid 3110 -cmd "free tdev 363 start_cyl=0 end_cyl=last_cyl;" commit
>
```

A Configuration Change operation is in progress. Please wait...

```
Establishing a configuration change session.....Established.
Processing symmetrix 000190103110
```

```
{
  free tdev 0363 start_cyl=0 cyl
    end_cyl = last_cyl;
}
```

```
Performing Access checks.....Allowed.
Checking Device Reservations.....Allowed.
Locking devices.....Locked.
Validating configuration changes.....Validated.
Preparing configuration changes.....Prepared.
Committing configuration changes.....Started.
Committing configuration changes.....Committed.
```

Terminating the configuration change session.....Done.

The configuration change session has successfully completed.

symcfg show -sid 3110 -pool glab1 -thin -detail -all

Symmetrix ID: 000190103110

```
Symmetrix ID      : 000190103110
Pool Name         : glab1
Pool Type         : Thin
Dev Emulation     : FBA
Dev Configuration : RAID-6(14+2)
Pool State        : Enabled
# of Devices in Pool : 16
# of Enabled Devices in Pool : 16
Max. Subscription Percent : None
```

Thin Devices(16):

```
{
-----
Sym      Total    Pool      Pool      Pool
Dev      Tracks  Subs      Allocated  Written
                               Tracks    (%)    Status
-----
0363      982080    6      106584    1      0 Dealloc'ing
0364      982080    6      300012    2      0 Bound
0365      982080    6      300012    2      0 Bound
0366      982080    6      300012    2      0 Bound
0367      982080    6      300012    2      0 Bound
0368      982080    6      300012    2      0 Bound
0369      982080    6      300012    2      0 Bound
036A      982080    6      300012    2      0 Bound
036B      982080    6      300012    2      0 Bound
036C      982080    6      300012    2      0 Bound
036D      982080    6      300012    2      0 Bound
036E      982080    6      300012    2      0 Bound
036F      982080    6      300012    2      0 Bound
0370      982080    6      300012    2      0 Bound
0371      982080    6      300012    2      0 Bound
0372      982080    6      450012    3      0 Bound
-----
Tracks    15713280    100    4756764    30      16      0
}
```

symcfg show -sid 3110 -pool glab1 -thin -detail -all

Symmetrix ID: 000190103110

```
Symmetrix ID      : 000190103110
Pool Name         : glab1
Pool Type         : Thin
Dev Emulation     : FBA
Dev Configuration : RAID-6(14+2)
Pool State        : Enabled
# of Devices in Pool : 16
# of Enabled Devices in Pool : 16
Max. Subscription Percent : None
```

```
Tracks    15713280    4650192    11063088    29
}
```

Thin Devices(16):

```
{
-----
Sym      Total    Pool      Pool      Pool
                               Allocated  Written
-----
```

Dev	Tracks	(%)	Tracks	(%)	Tracks	(%)	Status
0363	982080	6	12	0	1	0	Bound
0364	982080	6	300012	2	1	0	Bound
0365	982080	6	300012	2	1	0	Bound
0366	982080	6	300012	2	1	0	Bound
0367	982080	6	300012	2	1	0	Bound
0368	982080	6	300012	2	1	0	Bound
0369	982080	6	300012	2	1	0	Bound
036A	982080	6	300012	2	1	0	Bound
036B	982080	6	300012	2	1	0	Bound
036C	982080	6	300012	2	1	0	Bound
036D	982080	6	300012	2	1	0	Bound
036E	982080	6	300012	2	1	0	Bound
036F	982080	6	300012	2	1	0	Bound
0370	982080	6	300012	2	1	0	Bound
0371	982080	6	300012	2	1	0	Bound
0372	982080	6	450012	3	1	0	Bound
Tracks	15713280	100	4650192	30	16	0	

}

11. Appendix A: Basic Thin Provisioning Commands:

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
1.0.0.0	Storage pool creation		CLI		Pass	create pool <poolname>, type=thin;
1.0.0.1	Detail storage pool view		CLI		Pass Cond	symcfg show -pool glab1 -thin -detail -sid xxx -mb -all
1.0.0.2	Create datadev		CLI		Pass	create dev count=1, config=2-way-mir, size=65520, emulation=FBA, attribute=datadev,
1.0.0.3	List DATADEVs		CLI		Pass	symdev -sid xxx list -datadev
1.0.0.4	Create datadev from specific disk group, assign to a group, enable the device with storage group		CLI		Pass	create dev count=1, config=2-way-mir, size=65520, emulation=FBA, attribute=datadev, in pool glab1, member_state=ENABLE, disk_group=002;
1.0.0.5	Add datadev to pool and enable device		CLI		Pass	add dev xxx to pool <poolname>, type=thin, member_state=enable;
1.0.0.6	Create thin devices with pool		CLI		Pass	create dev count=3, config=TDEV,

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
	binding					emulation=FBA, size=0, bind to pool <poolname>;
1.0.0.7	Bind thin device with a pre-allocated size equal to 0		CLI		Pass	bind tdev xxx:xxx to pool <poolname> preallocate size=0 cyl;
1.0.0.8	Bind thin device with a pre-allocated size greater than 0		CLI		Pass	bind tdev xxx:xxx to pool <poolname> preallocate size=22079 cyl; (see note 1)
1.0.0.9	List DATADEVs		CLI		Pass	symdev -sid list -datadev
1.0.0.10	List thin devices		CLI		Pass	symdev -sid list -tdev
1.0.0.11	Set over-subscription percentage		CLI		Pass	set pool <poolname>, type=thin, max_bind_percent=200;
1.0.1.1	Unbind tdevs		CLI		Pass	unbind tdev device xxx:yyy from pool test1;
1.0.1.2	Disable datadev member in a pool		CLI		Pass	disable dev xxx:yyy in pool <poolname>, type=thin; (see note 2)
1.0.1.3	Remove datadevs from storage pool		CLI		Pass	remove dev xxx:yyy from pool PoolName, type=thin;
1.0.1.4	Delete storage pool		CLI		Pass	delete pool <poolname>, type=thin;

12. Appendix B: General SYMCLI:

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
1.0.0.0	Device depiction		CLI		Pass	"symdev show" proper list the capacity of the devices. Request the number of stripes and parity members be shown.
1.0.0.1	Device list with cylinder capacity		CLI		Pass	"symdev list -cyl" proper list the cylinder capacity of the devices.
1.0.0.2	Device list specifying date		CLI		Pass	"symdev list -RAID5/6 - protection m+x"

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
	protection type					
1.0.0.3	List device inventory		CLI		Pass	"symdev -sid list -inventory"
1.0.0.4	List device by disk group		CLI		Pass	"symdev -sid list -disk_group 03"
1.0.0.5	Display configuration information that is not stored in the SYMAPI database and retrieved directly from the configuration server.		CLI		Pass	"symconfigure -sid list -v"
1.0.0.6	Verify freespace in bytes and cylinders		CLI		Pass	"symconfigure -sid list -freespace -units MB/cyl, depicts the amount of free space available on the frame."
1.0.0.7	Verify configuration manager status		CLI		Pass	"symconfigure -sid verify"
1.0.0.8	Verify capacity on back-end directors		CLI		Pass	"symdev -sid list -da all -space -cyl"
1.0.0.9	List VAULT devices		CLI		Pass	"symdev -sid list -all" or "symdev -sid list -all grep VAULT"
1.0.2.0	Listing drive capacity in cylinders		CLI		Pass	"symdisk -sid xxx list -cyl"
1.0.2.1	Listing drive capacity in bytes		CLI		Pass	"symdisk -sid xxx list"
1.0.2.2	List drive information by disk group		CLI		Pass	"symdisk -sid list -by_diskgroup"
1.0.2.3	Listing devices on a particular FA port with LUN ID's		CLI		Pass	"symcfg -sid -FA -p list -address"
1.0.2.4	Host device inquiry		CLI		Pass	"syminq"
1.0.2.5	Display Symmetrix configuration database information		CLI		Pass	"symcfg -sid -db"

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
1.0.2.6	Verbose list of Symmetrix configuration		CLI		Pass	"symcfg -sid list -v"
1.0.2.7	Display all Symmetrix directors		CLI		Pass	"symcfg -sid -dir all list"
1.0.2.8	Listing devices by FA port		CLI		Pass	"symcfg -sid -dir all list - address"
1.0.2.9	Listing host to Symmetrix connections with capacities. Verify HW type, OS and release.		CLI		Pass	"symcfg -sid list - connections"
1.0.2.10	Listing devices without port assignments		CLI		Pass	"symdev -sid list -noport"
1.0.2.11	Listing Symmetrix locks		CLI		Pass	"symcfg -lockn all list"
1.0.2.12	Display number of memory boards and size		CLI		Pass	Symcfg -sid list -memory
1.0.2.13	Display cabinet summary status		CLI		Pass	"symcfg -sid list -env_data"
1.0.2.14	Display System Cabinet Status		CLI		Pass	"symcfg -sid show - env_data SystemBay"
1.0.2.15	Display Drive Cabinet Status		CLI		Pass	"symcfg -sid 3110 show - env_data Bay-1A"
1.0.2.16	Listing host to Symmetrix connections with capacities.		CLI		Pass	"symcfg -sid list - connections -capacity"
2.0.0.0	List Symmetrix daemons		CLI		Pass	"stordaemon list"
2.0.0.1	Symmetrix daemon shutdown		CLI		Pass Cond.	"stordaemon shutdown all" All ECC agents must be shutdown prior to using the shutdown all arguments.
2.0.0.2	Symmetrix daemon startup		CLI		Pass	"stordaemon start <daemon_name"
2.0.0.3	Display Symmetrix code level RAID 5 availability		CLI		Pass	"symconfigure list -v"

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
2.0.0.4	Display cabinet status		CLI		Pass	"symcfg -sid list -env_data"
3.0.0.0	Create RAID 6 device from available free from a specific disk group		CLI			Configuration manager to create multiple RAID 6 devices from freespace: create dev count=3, size=36828, emulation=FBA, onfig=RAID 6, data_member_count=14, mvs_ssid=0000, disk_group=5;
3.0.0.1	RAID 6 Metavolume device creation		CLI			configuration manager to create RAID 5 metavolume: form meta from dev 02E3 config=Striped, stripe_size=1920; add dev 02E4:02E5 to meta 02E3, protect_data=FALSE;
3.0.0.2	Metavolume device dissolve		CLI			configuration manager to dissolve metavolume: dissolve meta dev 02E3;
3.0.1.0	Port mapping of devices		CLI			configuration manager to map metavolume to front-end pair: map dev 02E3 to dir 7A:0 lun=074; map dev 02E3 to dir 10B:0 lun=074;
3.0.1.1	Port unmapping of devices		CLI			configuration manager to remove metavolume from front-end ports: unmap dev 02E3 from dir 7A:0; unmap dev 02E3 from dir 10B:0;
3.0.1.2	Removal of one device from striped meta-volume		CLI			Removing a device from a striped metavolume should not be allowed. Error message: Striped META device membership cannot be modified Device 02E5 generated the failure
4.0.0.0	Symmetrix device group rename		CLI		Pass	symdg rename <orig_name> <new_name>

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
4.0.0.1	List channel director device assignments		CLI		Pass	symmaskdb -sid list assignment -dev xxx:yyy
4.0.0.2	List log records		CLI		Pass	symaudit -sid list
4.0.0.3	List finite number of log records		CLI		Pass	symaudit -sid list -last_n 5
4.0.0.4	Show detail individual log record		CLI		Pass	symaudit -sid list -v -record_number xxxxx
5.0.0.0	Backup VCMDB		CLI		Pass	symmaskdb -sid -file <filename> backup
5.0.0.1	List devices assigned to a wwn in the backup file		CLI		Pass	symmaskdb -file list devs - wwn
5.0.0.2	Display channel adapter location of the specified device		CLI		Pass	symmaskdb -file <filename> list database
5.0.0.3	Display device and FA assignment		CLI		Pass	Symmaskdb -sid list assignment -dev xxx:xxx
5.0.0.4	Display channel adapter and device size and type assigned to a particular host		CLI		Pass	symmaskdb -sid -host <hostname> list capacity
5.0.1.0	Restore VCMDB		CLI			symmaskdb -sid -file restore
5.0.1.1	List connectivity by hosts		CLI		Pass	symcfg list -connections -sorthosts
5.0.1.2	List Symmetrix hosts by application ID		CLI		Pass	symcfg list -application
5.0.1.3	Lists the FA/WWN information		CLI		Pass	symcfg list -sid -FA all
5.0.1.4	Display HBA, WWN and connect FA ports from host		CLI		Pass	symmask list hba
5.0.2.0	Lists the RA/WWN information		CLI		Pass	symcfg list -sid -RA all

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
5.0.2.1	Show available LUN ID's per channel adapter		CLI		Pass	symcfg -sid list -available -address-sa -p
5.0.2.2	Show symm dev address per channel adapter		CLI		Pass	symcfg -sid -fa -p list -address
5.0.2.3	Offline and online an FA port		CLI		Pass	symcfg -sid -SA -p offline symcfg -sid -SA -p online
5.0.2.4	Display RA group connected in RDF fabric		CLI		Pass	symcfg -sid list -RA 8D -switch
5.0.2.5	Display RA group connected in RDF fabric		CLI		Pass	symcfg -sid list -RA 8D -switch
6.0.0.0	List gatekeeper and DB semaphores		CLI		Pass	Symcfg list -semaphores. SE server usually locks 6 semaphores with GK ID's.
6.0.0.1	Log file creation		CLI		Pass	View symapi log files
6.0.0.2	Initialize masking database		CLI			Symmashdb -sid -file init
6.0.0.3	Rebuild masking database		CLI		Pass Cond.	"Symcfg discover" pol the array(s) to rebuild the SE database. Specifying an array "symcfg -sid xxx discover" fails the update the database.
6.0.0.4	Disable/Enable a range of devices		CLI		Pass	Symdg create <testgrp> Symld -g testgrp addall dev -range xxx:xxx -sid Symld -g testgrp not_ready Symld -g testgrp ready Symdg -force delete testgrp
6.0.0.5	List device inventory		CLI		Pass	Symdev -sid list -inventory
6.0.0.6	Display configured network services		CLI			Symcfg list -services
7.0.0.0	Display cache status		CLI			Symstat -type cache -i -c -sid -sa all
7.0.0.1	Display LRU status		CLI			Symstat -type cache -lru all -i -c -sid
7.0.0.2	Display prefetch status		CLI			Symstat -type prefetch -i -c -sid -da all

Item #	Storage Management	Date	GUI Or CLI	Size and Qty	Pass Fail	Comments
7.0.0.3	Display memory status		CLI			symstat -sid -i 5 -c 5 -type memio
7.0.0.4	Display disk status		CLI			symstat -sid -i 5 -c 5 -type disk - disk 2a,c,all
8.0.0.0	Depict individual disk profile and hypers		CLI		Pass	symdisk -sid show 16b,c,06
8.0.0.0	Depict available spare disks		CLI		Pass	symdisk -sid list -hotspares
8.0.1.0	Enables the use of the auto meta config feature. Set the number of cylinders in a metamember.		CLI		Pass	set symmetrix meta_member_size = 36828;
8.0.1.1	Enables the use of the auto meta config feature. Set the meta configuration.		CLI		Pass	set symmetrix meta_config = Concatenated;
8.0.1.2	Set maximum disk hyper value		CLI		Fail	"set symmetrix max_hypers_per_disk=160;" EMC recommends that disk hypers be limited to 160 for RDF frames.

13. Appendix C: References

Document	Location
GDSE Website	http://engineering.citigroup.net/gdse/
GDSE Symmetrix Documentation	http://engineering.citigroup.net/gdse/products/platforms/symmetrix/
EMC Corporation. <i>EMC Symmetrix Virtual Provisioning Implementation and Best Practices</i>	http://powerlink.emc.com