



CATE CitiStorage

ScaleIO MWDC Tornado++ Resilience and Performance Testing

CSBE-15031-TR

**Test Results
Version 1.0**

VERSION CONTROL

Version and Date of Issue	Authors	Comment
V1.0 19th August 2015	Chris Clayton Eliot Wilson Chris Howarth Scott Torgerson	Initial release

CONTENTS

1	TESTING OBJECTIVES	4
2	TEST FINDINGS	5
2.1	Positive findings	5
2.2	Encouraging findings	7
2.3	Negative findings	7
3	APPENDIX A – DETAILED TEST RESULTS	8
3.1	IOPS Performance	8
3.1.1	OLTP2WH IOPS Testing	8
3.1.2	Random IOPS Testing	10
3.2	QoS throttling	11
3.3	ESX Crash Issue	12
3.4	IO Priority controls	13
3.4.1	Summary of all results	13
3.4.2	SSD	13
3.4.3	SAS	14
3.4.4	NL-SAS	14
3.5	Background device scanner	16
3.6	SDCs seen twice by MDM	17
3.7	“Stuck” rebalances	17
3.8	“Spontaneous” rebuilds	18
3.9	SDS CPU utilization	18
4	APPENDIX B - ISSUES SUMMARY	20

1 Testing Objectives

Testing was undertaken with ScaleIO R1.32.0 and a 4-storage-rack Tornado++ (T++) configuration from Wed 22nd July to Wed 5th August 2015 prior to handing over the MWDC T++ cluster over to CTI for rebuild and rollout.

The test objectives were as follows:

1. **IOPS performance.** Determine more accurately the IOPS capabilities of each of the tiers.
2. **QoS throttling.** Determine if we can increase the IOPS & bandwidth throttle limits currently defined in CATE ScaleIO T++ Best Practice documentation.
3. **ESX host crashes.** Reproduce the ESX crashes first seen at high IOPS in the GTDC Tornado++ ScaleIO testing so that their origin can be determined and fixed by the relevant vendor (EMC, VMware or HP).
4. **IO Priority controls.** Determine the Tornado++ Best Practice (policy and parameter settings) for the R1.32 IO priority controls for rebuild in order to minimize rebuild times whilst minimizing impact on host IO response time.
5. **Background device scanner.** Determine the Tornado++ Best Practice (parameter settings) for the new R1.32 background device scanner
6. **SDCs seen twice by MDM.** Investigate further an issue seen with R1.31.1 in GTDC T++ testing where SDCs were detected twice down different paths, with “one SDC” connected and the “other SDC” disconnected.
7. **“Stuck” rebalances.** Investigate further an issue seen in R1.31.1 GTDC T++ testing where the amount of capacity that still needs to be rebalanced appeared static as reported in the CLI and GUI, even though rebalancing was still occurring. This we refer to as the “stuck” rebalance issue.
8. **“Spontaneous” rebuilds.** Investigate further an issue seen in R1.31.1 GTDC T++ testing where spontaneous rebuilds were seen to occur.
9. **SDS CPU utilization.** Observe SDS CPU utilization on the primary MDM during busy rebalancing activities after loss of a server.

Note that much of the above testing was further investigation of issues discovered during the GTDC T++ testing and determination of R1.32 Best Practice for Tornado++, and hence, as expected, did not result in the opening of a significant number of **new** JIRA tickets, as many of the issues being re-investigated already have JIRA entries.

The following could **not** be tested:

- **Stuck rebuilds.** In GTDC T++ testing, we encountered a significant issue with rebuilds not completing after a server failure even when there was apparently enough free space. This was not re-tested as fixes in this area had not made it into the R1.32.0 code (the latest GA code available at the time of testing).

2 Test findings

2.1 Positive findings

- **IOPS Performance.** Following the more in depth testing in MWDC, IOPS expectations for the 3 tiers of Tornado++ storage have now been revised upwards for the SAS and NL-SAS tiers since the original (limited) GTDC performance testing. The SAS IOPS/UTB has increase from 500 to 600, and for NL-SAS, has increased is from 30 to 85. The SSD figure remains unchanged.

Pool Type	Prior Sustained Host IOPS/UTB	Revised Sustained Host IOPS/UTB
SSD	25,000	25,000
SAS	500	600
NL-SAS	30	85

Revised IOPS/TB capabilities of T++ storage tiers

The above table assumes the following worst-case acceptable limits on IOPS:

- **SSD:** 4ms (note that when SSD and SAS are sharing a disk controller, both latencies increase at very high IOPS).
- **SAS:** 10ms
- **NL-SAS:** 20ms

The increase in NL-SAS IOPS is likely due to the fact that the write-cache for this tier is not shared with any other tier (the NL-SAS tier supported by dedicated nodes, whereas the SSD and SAS tiers share the same higher-end servers).

- **QoS throttling.** Following more comprehensive testing and analysis the IOPS and/or bandwidth throttles can now be increased for all three tiers:

Original Best Practice			Assumed Average Block Size (KB)
	Transactional IOPS/TB limit	Bandwidth MBps/TB limit	
SSD	50,000	1,600	32.77
SAS	1,000	32	32.77
NL-SAS	60	8	136.53

New Best Practice after MWDC T++ testing			Assumed Average Block Size (KB)
	Transactional IOPS/TB limit	Bandwidth MBps/TB limit	
SSD	50,000	2,344	48.00
SAS	1,200	450	384.00
NL-SAS	170	64	384.00

Pool Type	Volumes Size (GB)	Volume IOPS Limit	Volume Bandwidth Limit (MBps)
SSD	2,560	125,000	5,860
SAS	5,120	6,000	2,250
NL-SAS	11,264	1,870	704

New QoS settings for VMware standard LUNs

- **IO Priority controls.** Testing and subsequent discussion has led to the following Best Practice for Tornado++ for rebuild IO priority controls. These are judged to provide the best compromise between minimizing rebuild times and minimizing the impact of rebuilds on host IO.

Pool type	Rebuild IO priority Policy	Concurrent I/O limit	Bandwidth Limit	Note
SSD	Limit concurrent IO	1	N/A	Default
SAS	Limit concurrent IO	2	N/A	Not default
NL-SAS	Limit concurrent IO	1	N/A	Default

Note that rebalancing IO priority policies and bandwidth setting are to be left for all pools with the default configuration of:

Pool type	Rebalance IO priority Policy	Concurrent I/O limit	Bandwidth Limit	Note
SSD	Favor Application IO	1	10MB/s	Default
SAS	Favor Application IO	1	10MB/s	Default
NL-SAS	Favor Application IO	1	10MB/s	Default

- **Background device scanner.** The following Best Practice will be adopted. An increase of the NL-SAS bandwidth limit to 4096KB/s was considered in order to reduce the total scan time for a disk, but as the scanner runs continuously up to this rate regardless of other IO (e.g. rebuilds or host IO), this was considered to be too high a continuous overhead for NL-SAS devices to support. Also, as the NL-SAS device capacity utilization will likely to lower than in other tiers, the time taken to scan a disk will be reduced anyway.

Pool type	Scanner Bandwidth Limit (KB/s/device)	Estimated device scan time for a fully written to disk*	Note
SSD	2048	Up to twice a week	Default
SAS	2048	Up to twice a week	Default
NL-SAS	2048	Every 3-4 weeks	Default

* Commensurately shorter time for partially written to disks

- **SDS CPU usage.** SDS CPU usage during large SAS/SSD rebuilds was well within the capability of the server.

2.2 Encouraging findings

- **Spontaneous rebuilds.** Several times during the GTDC T++ testing with ScaleIO R1.31.1, strange rebuild processes were initiated without any obvious cluster health problems. No such 'phantom' rebuilds were seen during the MWDC testing with ScaleIO R1.32.0.
- **SDCs seen twice by MDM.** During the GTDC T++ testing with ScaleIO R1.31.1, 2 out of 56 dual-connected SDCs were not discovered correctly when initially installed. We did not encounter the same issue when installing the SDCs in MWDC T++ environment.

2.3 Negative findings

- **ESX Crash.** We were able to reproduce the ESX crash (first seen in the GTDC Tornado++ ScaleIO testing) on 3 of the 16 ESX test hosts when testing at high IOPS (>27,000). One of the ESX hosts is particularly susceptible to the issue. We've engaged VMware, EMC, and HP to further troubleshoot the issue and that is ongoing.
- **"Stuck" rebalances.** We observed rebuild activity on one pool impacting rebalancing activity in other pools. At the time of writing, we are still waiting on EMC for some feedback on some debug files that were sent to them regarding an R1.32.0 instance of this problem.

3 Appendix A – Detailed Test Results

3.1 IOPS Performance

3.1.1 OLTP2WH IOPS Testing

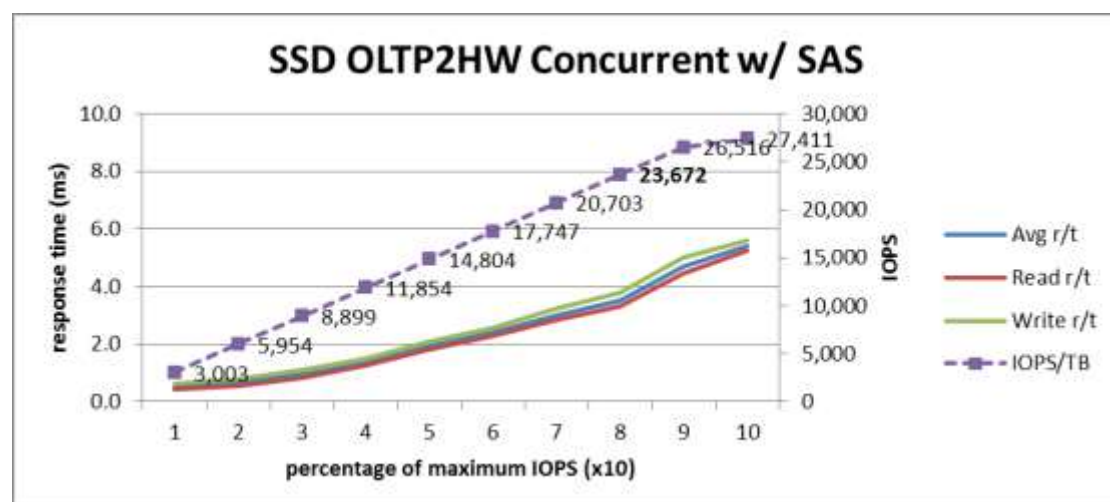
More comprehensive testing with the OLTP2WH workload was conducted. The maximum IOPS for each storage tier were measure using 48 physical SDC clients running the vdbench tool. The SDC clients were collocated on the 48 SAS/SSD SDS hosts.

The IOPS rates achieved with “good” response times will be the basis for marketed tier capabilities and QoS throttles. The “good” response time values are the maximum acceptable values at 4ms for SSD, 10ms for SAS, and 20ms for NL-SAS. IOPS rates were ramped up in incrementally to the maximum that could be achieved. The highest IOPS where neither the read nor write response times exceed the “good” threshold and the expected workload was achieved was consider the maximum achievable.

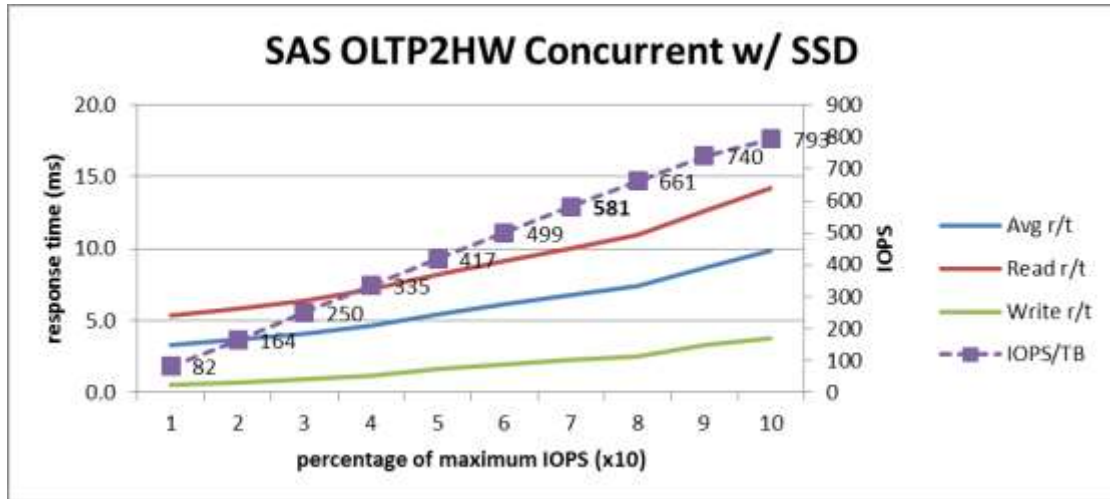
Pool Type	Prior Sustained Host IOPS/TB	Revised Sustained Host IOPS/TB
SSD	25,000	25,000
SAS	500	600
NL-SAS	30	85

IOPS capability of each tier

Since SAS and SSD tiers reside on the same hosts with shared components including disk controllers, the workload was run on both tier concurrently and we observed a reduced maximum achievable IOPS rate for SAS likely due to greater contention on the disk controller and its cache. The SSD measurements are slightly below the 25,000 target but we believe there is contention in this test scenario by running the SDC and SDS on the same host at high IOPS. The maximum IOPS data point is shown below with the data label in **bold**.

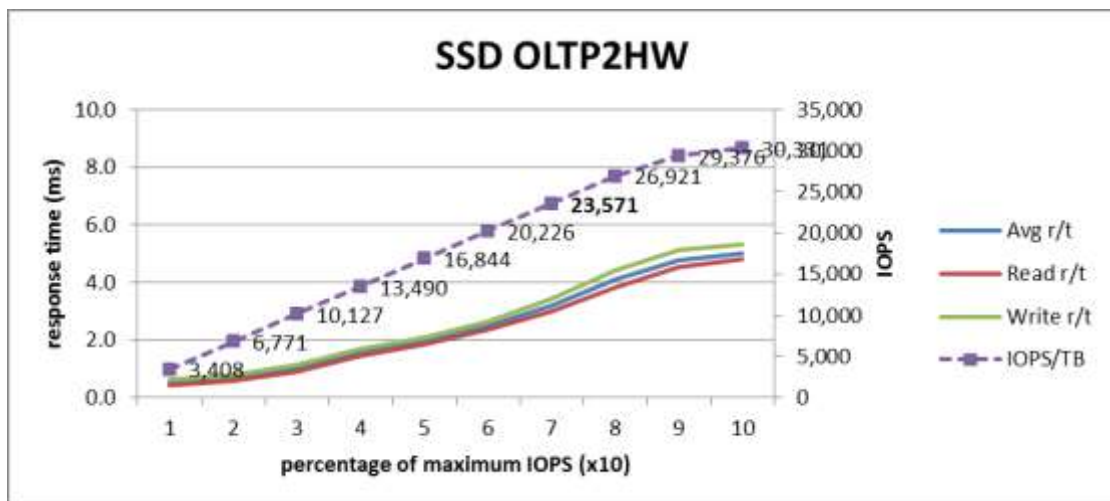


SSD response time (ms) on left axis & IOPS on right axis (concurrent test)

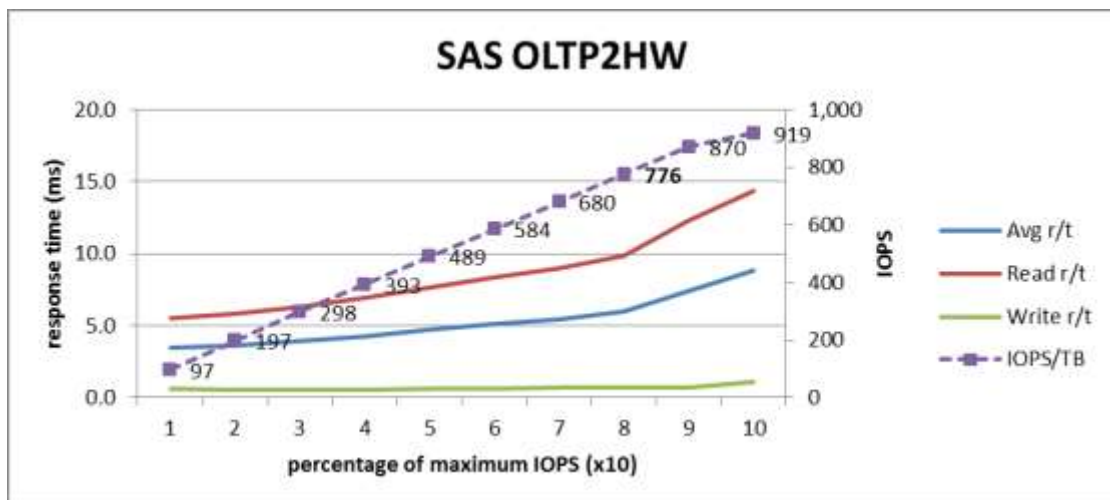


SAS response time (ms) on left axis & IOPS on right axis (concurrent test)

The tests with the workload running independently on each tier did yeild higher IOPS for SAS:

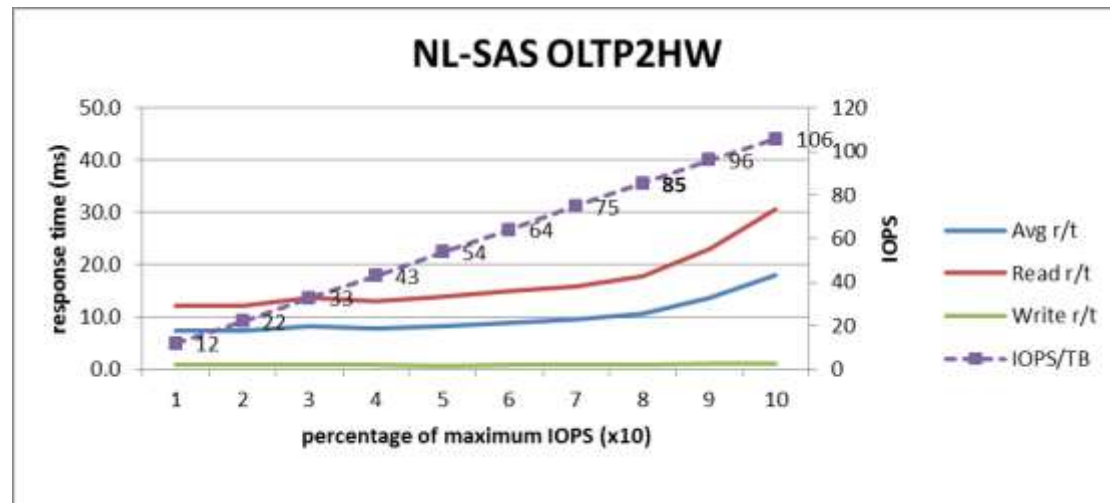


SSD response time (ms) on left axis & IOPS on right axis



SAS response time (ms) on left axis & IOPS on right axis

For NL-SAS we observed a significantly higher maximum IOPS than was estimated from the previous limited tests in GTDC:

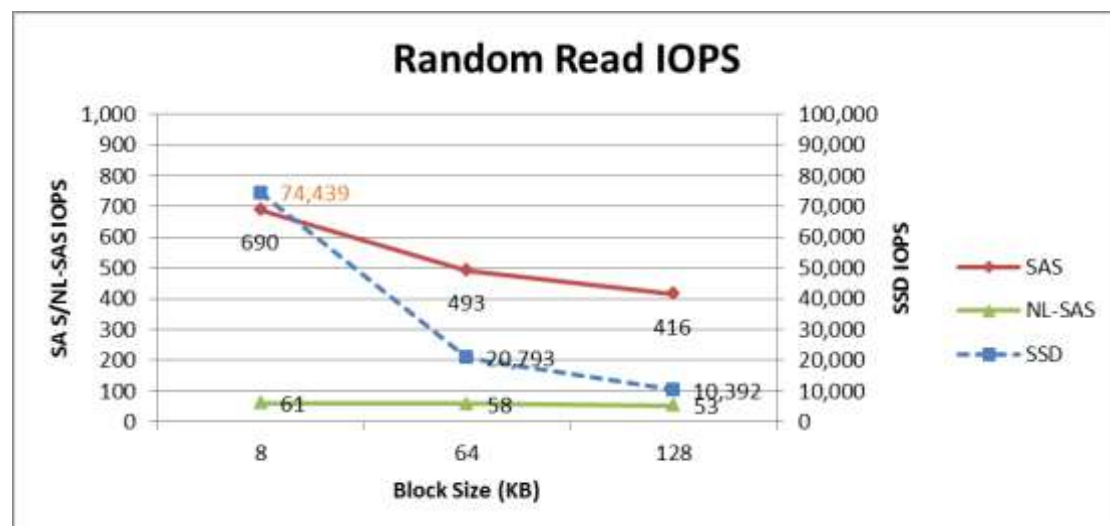


NL-SAS response time (ms) on left axis & IOPS on right axis

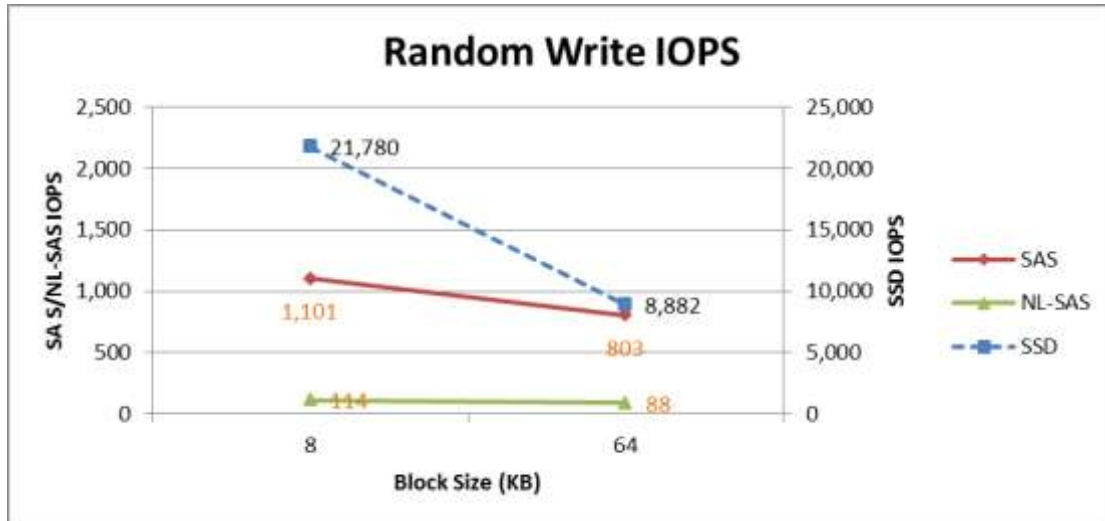
3.1.2 Random IOPS Testing

Additional IOPS testing with fixed block workloads showed expected results:

- As block size increases IOPS drop significantly for SSD
- SSD write IOPS are much lower than read IOPS
- HDD (SAS and NL-SAS) see a smaller impact to IOPS as block size increases
- Some tests performed did not have sufficient threads to achieve maximum IOPS and these are shown with orange data labels



Response time (ms) for SAS/NLSAS on left axis & SSD on right axis



Response time (ms) for SAS/NLSAS on left axis & SSD on right axis



Response time (ms) for SAS/NLSAS on left axis & SSD on right axis

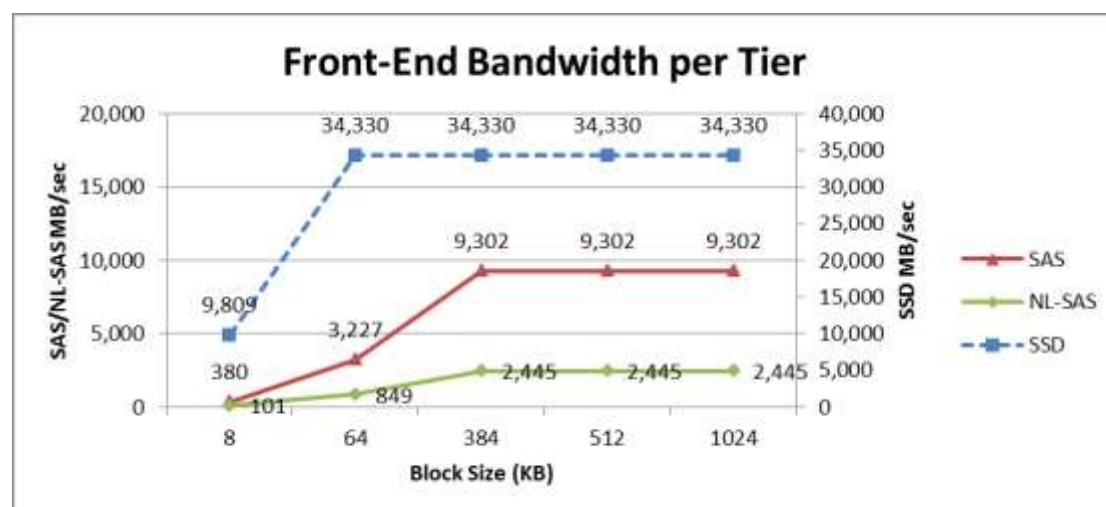
3.2 QoS throttling

Based on the IOPS performance tests in the prior section and analysis of large block performance of SSD, SAS, and NL-SAS drives, we recommend these adjustments QoS throttles:

Original			Assumed Average Block size (KB)
	Transactional IOPS/TB limit	Bandwidth MBps/TB limit	
SSD	50,000	1,600	32.77
SAS	1,000	32	32.77
NL-SAS	60	8	136.53

Proposed			
	Transactional IOPS/TB limit	Bandwidth MBps/TB limit	Assumed Average Block size (KB)
SSD	50,000	2,344	48.00
SAS	1,200	450	384.00
NL-SAS	170	64	384.00

With these throttles as block size increases the total bandwidth will be limited per tier according to the platform ability to handle larger block IOs. SSD will be more constrained than SAS or NL-SAS. The graph below shows the total front end bandwidth that can be achieved with this model assuming 50% of the QoS throttle workload is generated and 50% is at the larger block size indicated:



3.3 ESX Crash Issue

Using a 4KB random read workload to 3 VMDK disks on 3 separate SIO volumes in the SSD pool, we were able to reproduce the PSOD crash on 3 of the 16 test ESX hosts.

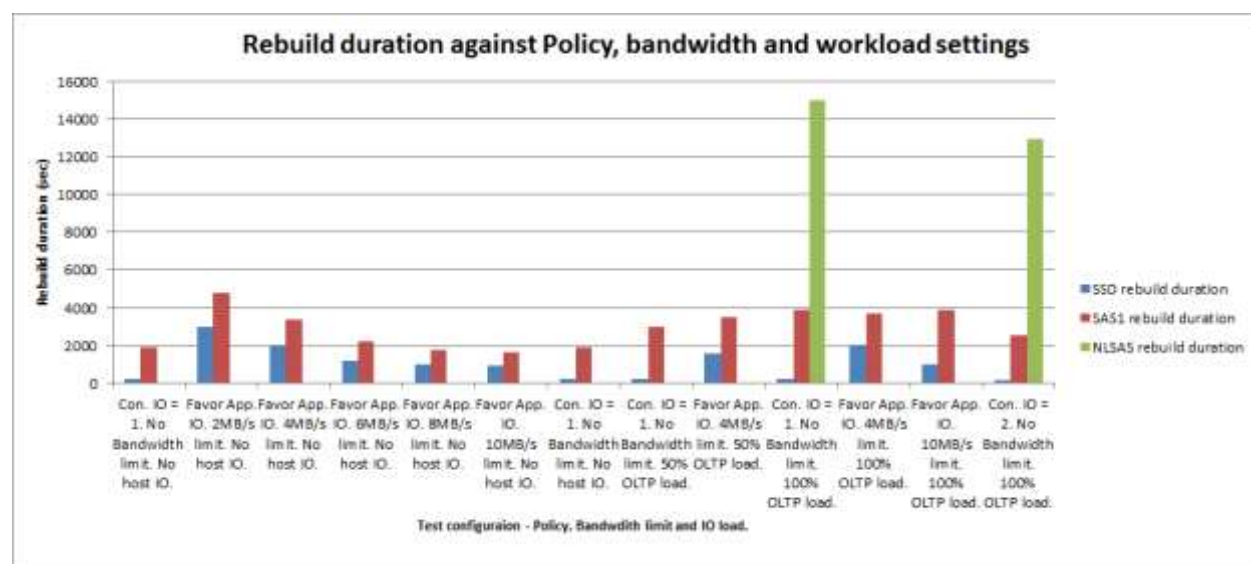
One of the ESX hosts (*fe006) was very sensitive to heavy IOPS and would crash very quickly while the other 2 took many hours of the workload running. The trigger for the crash is that a CPU heartbeat (14 sec default) is triggered so it appears something is not releasing the CPU as expected. Working with EMC:

- Tested with 3 different revisions of the SDC driver with various changes and additional logging
- Tested with the VMware heartbeat timeout set to the maximum of 24 hrs to determine if the issue persisted but avoiding an actual PSOD
- Tested a 4th version of the driver with a parameter to force the driver to release the CPU and this alleviated the issue but with impact to performance
- We setup NFS datastores and found that similar heavy IOPS to those did not result in a crash or CPU heartbeat timeouts
- Determined that 1 of the storage NICs on this blade operates at ~50% of the performance of the other NIC and this appears to be related to the root cause for this host being sensitive. The impact of this is that the host can do just 10% of the IOPS of other hosts at similar latency.
- Swapped the *fe006 blade with the *fe007 blade and confirmed that the slow NIC issue followed the blade and not the slot.
- After swapping the blades we found that *fe007 is susceptible to frequent crashes when running at very high IOPS, crash details send to EMC and VMware

VMware's renewed investigation into the issue identified that there may be interaction with the NIC firmware or driver and requested that we engage HP. We've setup a status call with the 3 vendors to continue troubleshooting the issue and have agreement with the Cloud Ops team to continue using 2 blades.

3.4 IO Priority controls

3.4.1 Summary of all results



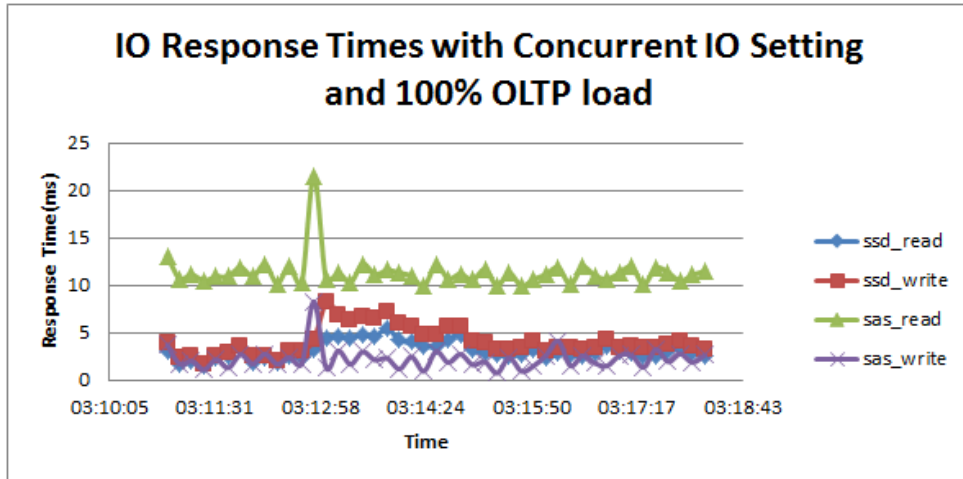
Rebuild Durations as a function of policy, bandwidth limit and workload

3.4.2 SSD

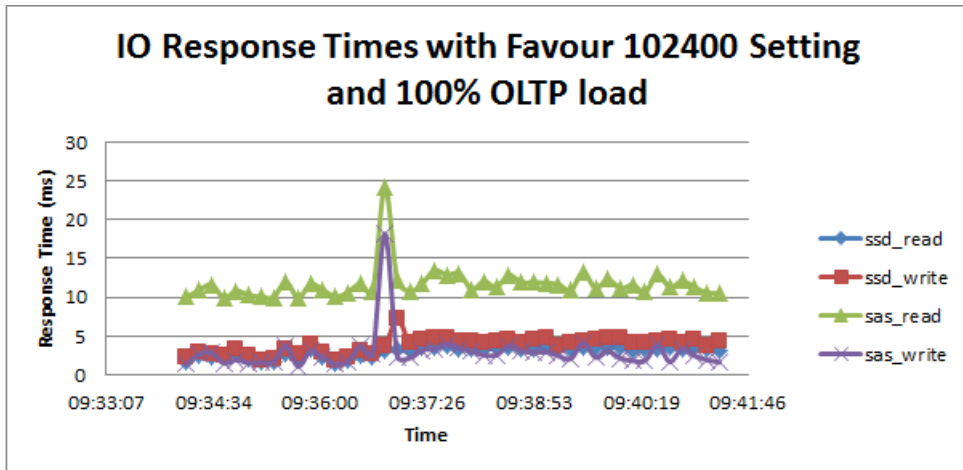
Use of the more benign “favor application IO” policy for SSD did protect the SSD IO response time, but with a significant increase in rebuild time. The choice was between:

- Favor Application IO: SSD write increased from 3ms to 5ms for 16 minutes.
- Concurrent IO limit = 1: SSD write increased from 3ms to 4ms-7ms for only ~3-4 minutes

Hence, we determined to leave at the default value of concurrent IO limit = 1 as it was judged that it was better to have the rebuild complete 4X faster with at 4-7ms response time (and with response time back to 3ms after those 4 minutes) than for the rebuild to take 16 minutes (potentially impacting data availability due to extended period when a second failure would result in data loss), with a much more extended (albeit lesser) impact on response time during that period.



SSD IO response time with concurrent IO limit = 1



SSD IO response time with favor application IO and 10MB/s limit per device

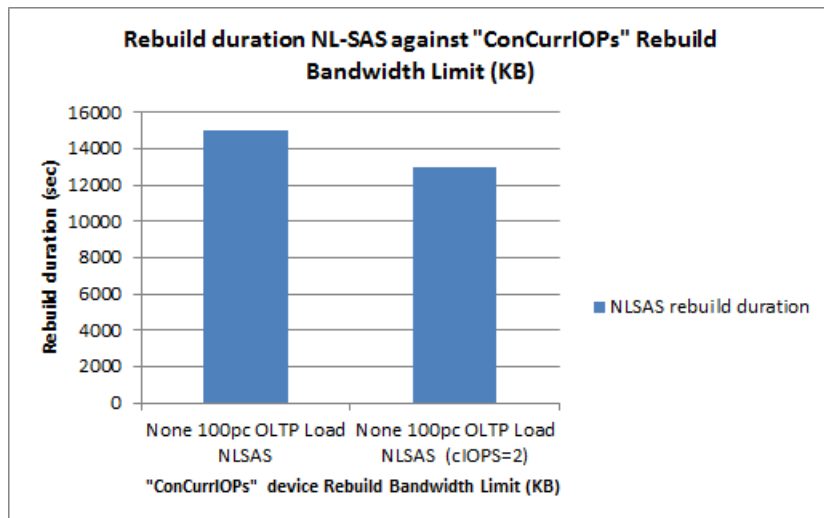
3.4.3 SAS

Use of the more aggressive concurrent IO limit = 2 for SAS significantly reduced the rebuild time (by ~35%) but with only modest host response time impact. Hence, we determined to change at the value of concurrent IO limit to 2 in CATE Best Practice for T++.

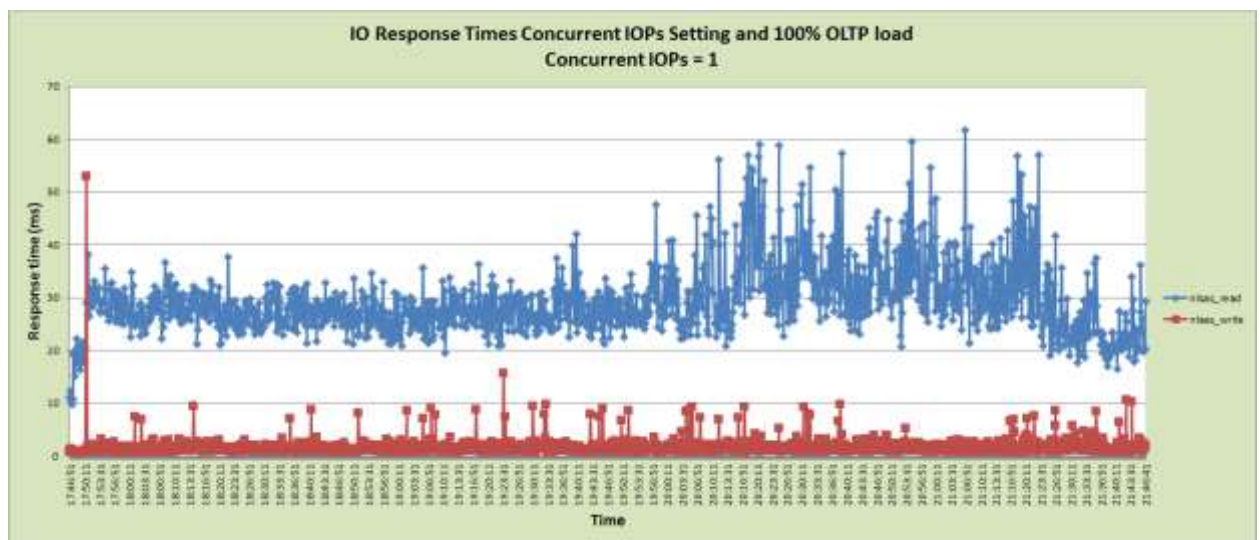
3.4.4 NL-SAS

Use of the more aggressive concurrent IO limit = 2 for NL-SAS only slightly reduced the rebuild time (by ~14%) but resulted in the host response time being significantly affected (about 1/3rd higher). Hence, we determined to leave at the default value of concurrent IO limit = 1.

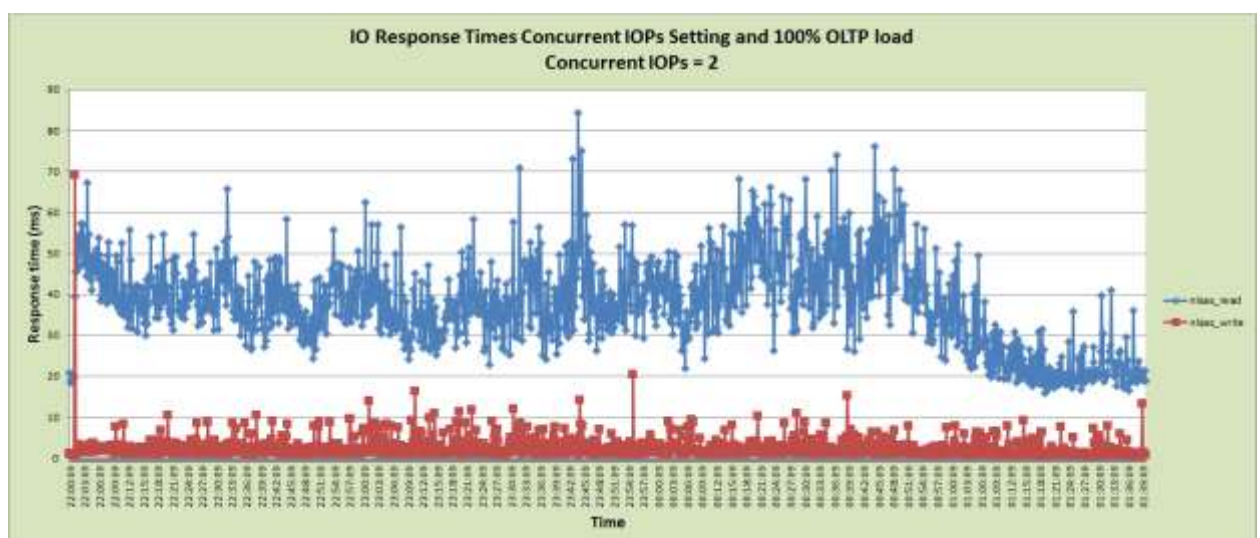
Note that the pool was not fully provisioned for this test, and hence the recorded rebuild times are less than will potentially be seen in a mature NL-SAS pool.



Comparison of NL-SAS rebuild time for concurrent IO limits of 1 & 2

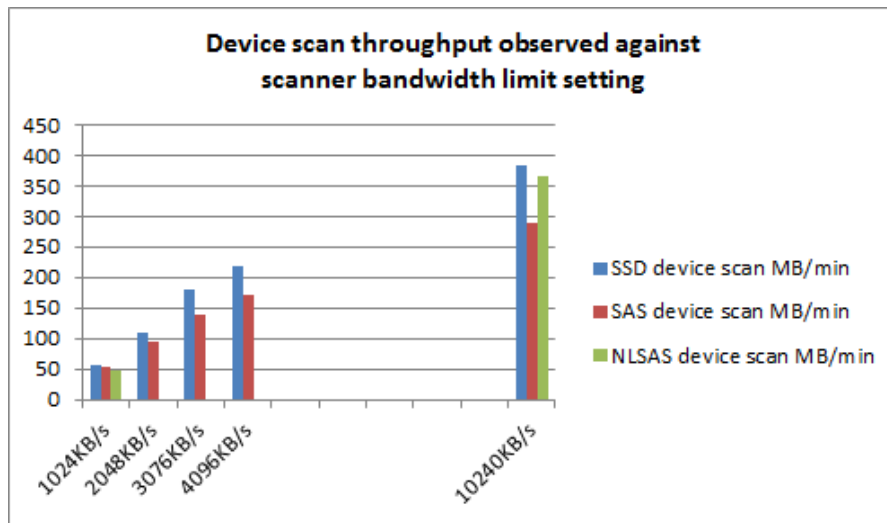


IO response time with 100% OLTP load for NL-SAS with Concurrent IOPS limit =1

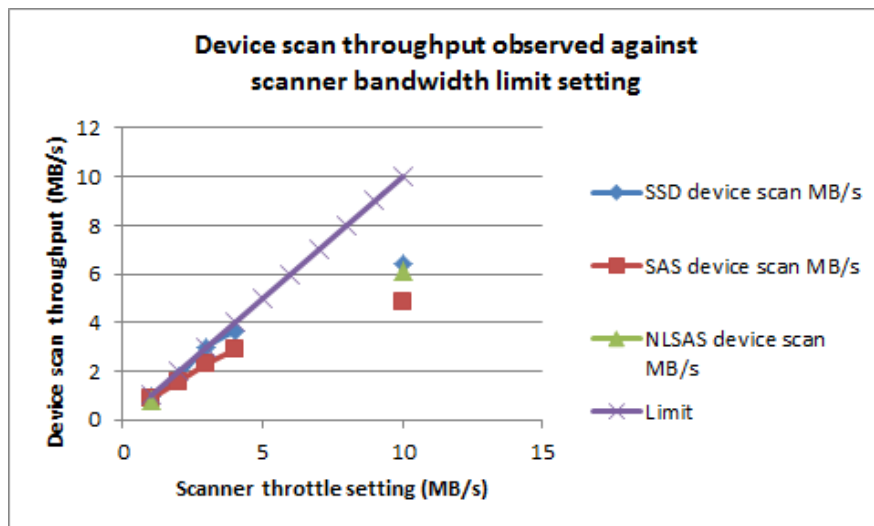


IO response time with 100% OLTP load for NL-SAS with Concurrent IOPS limit =2

3.5 Background device scanner



Device scan rate (MB/min) vs throttle limit set



Device scan rate (MB/s) decay at higher throttle limits

The main findings were:

- The observed device scan rate (ODSR) depends on how fully utilized the volumes are. The closer to the volume being fully written to, the closer the ODSR to the throttle limit set.
- The scan rate increase is non-linear at higher throttle limits i.e. the ODSR falls below the theoretical maximum dictated by the limit set.
- Test had to be done with low host IOPS. There was no discernible impact of scanning on response time (at these low IOPS levels). Any presumed impact will be when the system is highly utilized (host IO and/or during rebuilds). This could not be tested in MWDC, due to zero padding requirement for background device scanner and scheduling limitations.

3.6 SDCs seen twice by MDM

All 56 SDCs connected via a single interface i.e. this time we did not see one disconnected and one connected SDC from different interfaces on the same client as was seen with the GTDC testing.

However, we noticed that 55 of the 56 physical SDCs connected to the MDM via their NIC1 (VLAN 900 (ETH0) – 10.125.32.xx address) whereas one of the SDCs (namwtp1sioa10) connected over its NIC3 (VLAN 901(ETH2) – 10.125.96.xx address).

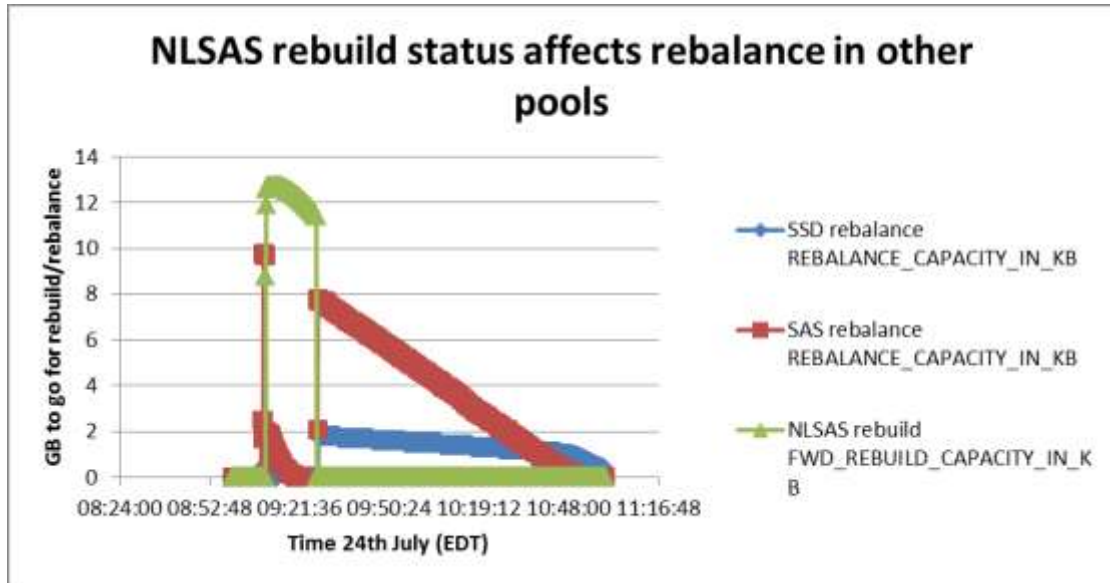
EMC later indicated that this is not unexpected. From SDC and MDM perspective, there is no primary or secondary NIC. Whenever a host boots up, SDC will start before the network is available and will go through the list of MDM IPs trying to connect to an MDM. **When it tries the first IP, there is a chance the MDM or the network are not available yet, therefore sometimes SDC will connect through the second IP.** Please note, this is the way an SDC communicates with MDM, not SDS, with SDS it will use all the available IP addresses simultaneously.

3.7 “Stuck” rebalances

There was limited opportunity to force the cluster into this condition during this testing due to scheduling issues. However, the following was observed:

- One “temporary” stuck rebalance was observed. Logs and previously agreed debug information was uploaded to EMC, but at the time of writing we have not had feedback as to whether EMC could determine the cause of this observed event.
- One observation that I did make in one test is that if pools SSD & SAS are rebalancing, and I took down a separate SDS that supports our NL-SAS pool at 09:11 EDT, then:
 - 1) The NL-SAS pool starts a rebuild (obviously) as shown in green in the graph below below.
 - 2) The SSD and SDS pools that were rebalancing (blue and red lines), are impacted (the REBALANCE_CAPACITY_IN_KB metric for these pools drops).

Then when the NLSAS SDS is brought back on line (at 09:27EDT), causing its rebuild to be cancelled (green line drops to zero), then the reported REBALANCE_CAPACITY_IN_KB for the other pools (red and blue lines) jumps back up again to believable figures i.e. in line with where you might expect it to be (through extrapolation) had the NLSAS rebuild not been triggered.



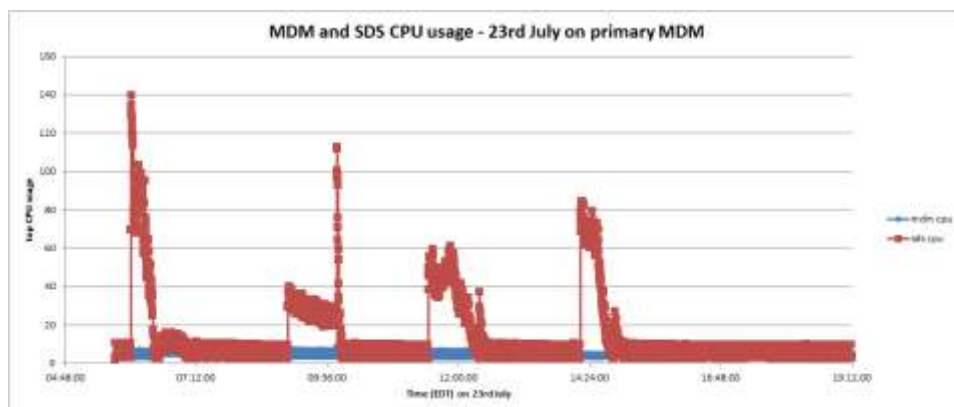
3.8 “Spontaneous” rebuilds

None were observed.

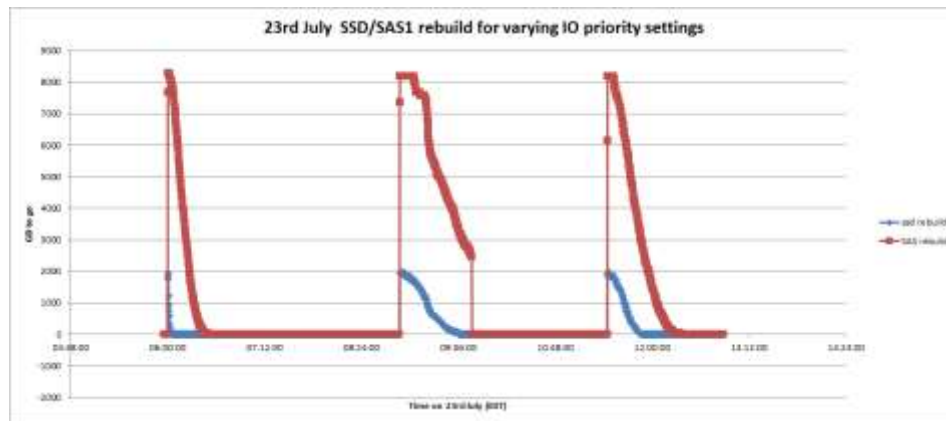
3.9 SDS CPU utilization

The SDS (and MDM) process CPU usage was recorded on one of the SDS servers supporting the SSD and SAS tiers (this is also the primary MDM) during a series of SSD/SAS rebuild following loss of an entire SDS. The SDS CPU usage was well within the capabilities of the server (this server has 8 cores and hence can reach up to 800% CPU usage). We understand that SDS processes are multi-threaded to run over 6 cores by default).

The MDM CPU usage was negligible.



MDM and SDS cpu usage during rebuilds on 23rd July



SSD/SAS rebuilds that correspond to the first 3 above SDS/MDM CPU usage spikes on 23rd July

4 Appendix B - Issues summary

The following table summarises the main issues that were encountered during testing. For further details of any of the items, visit the relevant JIRA entry (see left-most column), or EMC SR number.

Issue	Citi (JIRA) reference	EMC reference	Owner	Issue overview
1	CITISTOR-999	N/A	Chris Clayton	Temporarily "Stuck" rebalance on MWDC cluster (R1.32-402.1)
2	CITISTOR-1000	N/A	Chris Clayton & Chris Howarth	SDCs seen twice by MDM
3	CITISTOR-1004	SR71389984	Eliot Wilson	"Spontaneous" rebuilds
4	CITISTOR-1005	N/A	Eliot Wilson	ESXi crash under high IOPS (R1.32-402.1)
5	CITISTOR-1143	SR73189424	Scott Torgerson	Spontaneous MDM failover (R1.32-402.1)