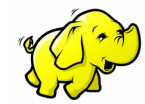
Sven Oehme – oehmes@us.ibm.com
Chief Research Strategist Spectrum Scale

Spectrum Scale performance update

October 2016

- Authentication Access Control
- External key Encryption
- Auto Tiering
- Redundancy High Availability
- Disaster Recovery
- Geo-distributed Caching
- Common Namespace
- Scalability (exabytes, trillion files)
- File Compression
- Backup and Compliant Archive
- De-clustered erasure coding

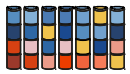


SPARK | HDFS-API | NFS | CIFS | HTTP | POSIX | S3 | Swift | iSCSI**

BigData | File Serving | Object | Block

Spectrum Scale (GPFS)

Shared Storage | **Direct Attached Storage**



External RAID Controller

Erasure Coded GPFS Native RAID

FPO Replicated storage

Mestor Network erasure coded**

**** Statement of direction**

Disclaimer

Non of the following Performance numbers should be reused for sales or contract purposes.

Some of the numbers produced are a result of very advanced tuning and while achievable, not very easy to recreate at customer systems without the same level of effort

A word of caution : The achieved numbers depends on the right Client configuration and good Interconnect and can vary between environments. They should not be used in RFI's as committed numbers, rather to demonstrate the technical capabilities of the Product in good conditions

Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
DeepFlash 150

Spectrum Scale Performance Optimization challenge

- Where we are today :
  - Every new Scale release added new configuration parameters
  - On Scale prior 4.2 we had >700 Parameters
  - Overwhelming majority are undocumented and not supported unless instructed by development, but many of them are used in systems without development knowledge to achieve specific performance targets
  - Tuning Scale systems is considered ‘magic’
  - Changing defaults is impossible due to the wide usage of Scale as impact would be unknown and impossible to regression test due to the number of combined options and customer usage

- So how do we change this ?
  - Significant reduce number of needed parameters to achieve desired performance
  - Auto adjust dependent parameters
  - Provide better ‘new defaults’ when new auto scale features are used
  - Document everything else that is frequently required
  - Provide better insight in ‘bottlenecks’ and provide hints on what to adjust

1st Enhancements implemented as part of 4.2.1 (small subset already in 4.2.0.3)

- Introduce workerThread config variable
  - WorkerThread (don't confuse with worker1Thread) is a new added config variable available from 4.2.0.3+ or 4.2.1.0+
  - Its not just another parameter like others before, it is the first to eliminate a bunch of variables that handle various aspects of tuning around threads in Scale today.
  - Instead of trying to come up with sensible numbers for worker1Threads,worker3Threads,various sync and cleaner threads, log buffer counts or even number of allocation regions, simply set workerThreads and ~20 other parameters get calculated based on best practices and dynamically adjusted at startup time

Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
DeepFlash 150

Spectrum Scale Communication Overhaul

- Why do we need it ?
  - Keep up with the io(not capacity) density of bleeding edge Storage technology (NVMe, etc)
  - Leverage advances in latest Network Technology (100GE/IB)
  - Single Node NSD Server ‘Scale-up’ limitation
  - NUMA is the norm in modern systems, no longer the exception
- What do we need to do ?
  - Implement an (almost) lock free communication code in all performance critical code path
  - Make communication code as well as other critical areas of the code NUMA aware
  - Add ‘always on’ instrumentation for performance critical data, don’t try to add it later or design for ‘occasional’ collection when needed
- What are the main challenges ?
  - How to make something NUMA aware that runs on all Memory and all Cores and everything is shared with everything :-D

High Level Requirements for a Next Gen HPC project

- **2.5 TB/sec single stream IOR**
- **1 TB/sec 1MB sequential read/write**
- **Single Node 16 GB/sec sequential read/write**
- 2.6 Million 32k file creates /sec

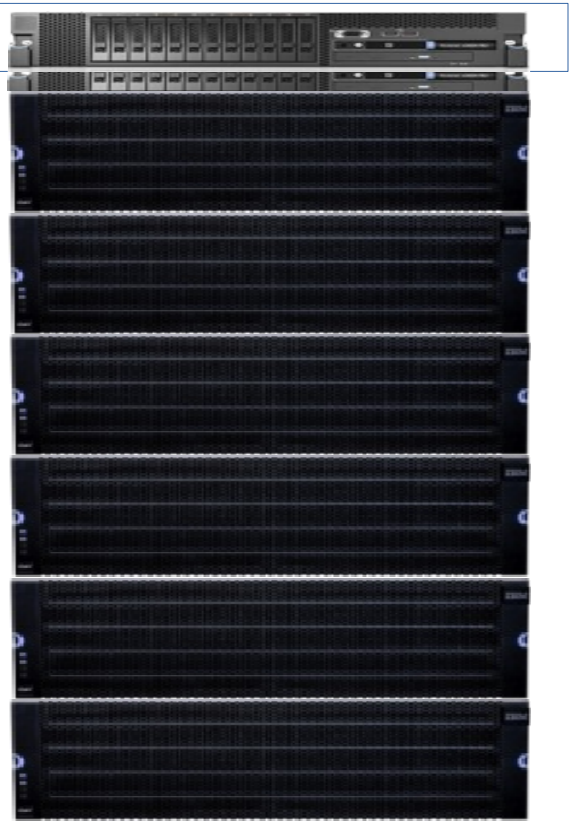
Test Environment Setup

...

12 x3650-M4 Server each with
32 GB of Memory (6 gb Pagepool)
2 FDR Port
2 x 8 core CPU

Mellanox 32 Port
Infiniband EDR switch

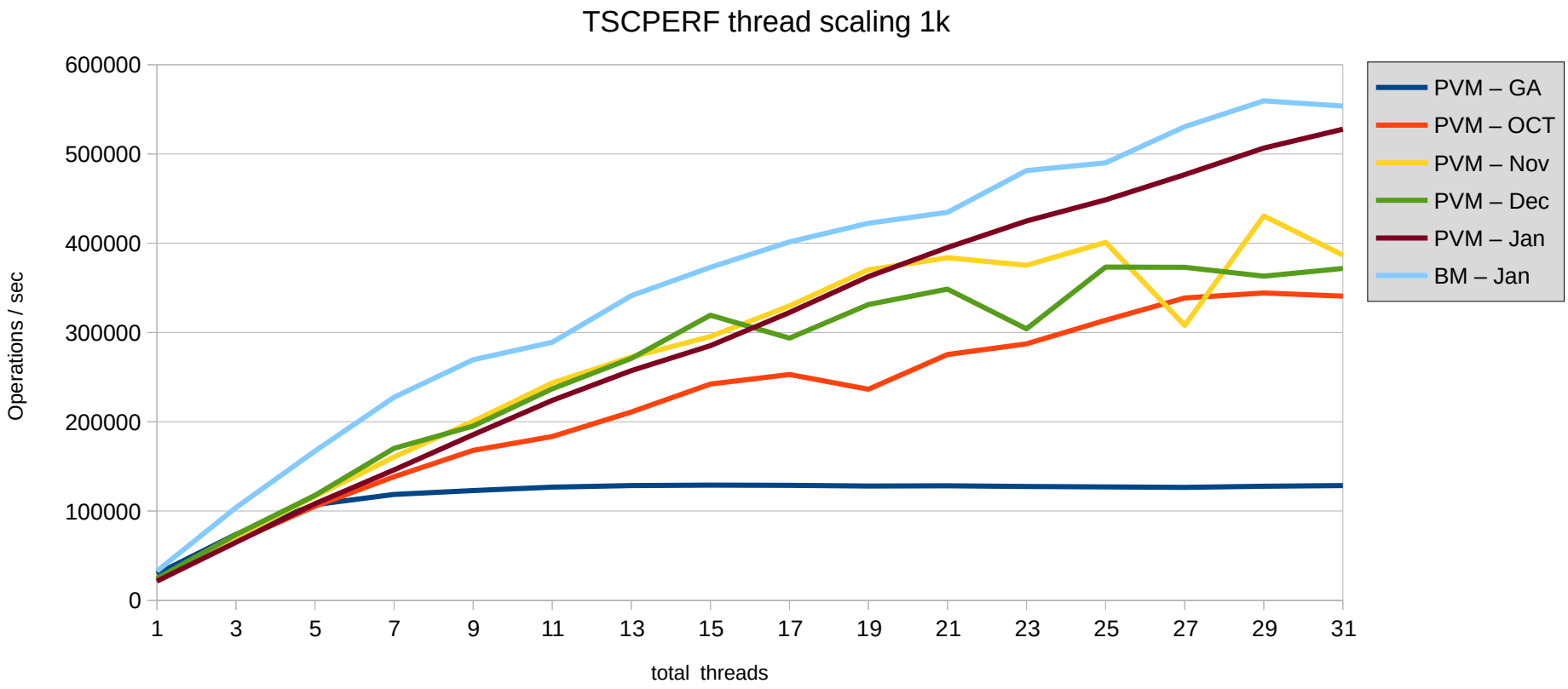
8x



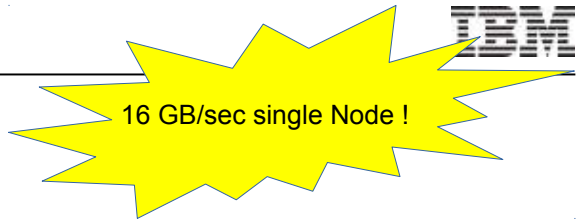
1,2,4 or 6 encl system
4 EDR Ports connected per ESS node

Spectrum Scale Communication Overhaul

Factor 5 improvement



Single thread RPC latency went down by 50%, peak result went up 500%



Single client throughput enhancements

```
[root@p8n06 ~]# tsqosperf write seq -n 200g -r 16m -th 16 /ibm/fs2-16m-06/shared/testfile -fsync
tsqosperf write seq /ibm/fs2-16m-06/shared/testfile
  recSize 16M nBytes 200G fileSize 200G
  nProcesses 1 nThreadsPerProcess 16
  file cache flushed before test
  not using direct I/O
  offsets accessed will cycle through the same file segment
  not using shared memory buffer
  not releasing byte-range token after open
  fsync at end of test
    Data rate was 16124635.71 Kbytes/sec, thread utilization 0.938, bytesTransferred 214748364800
```

Single thread small I/O latency

50-80 usec per small Data RPC

```
[root@client01 mpi]# tsqosperf read seq  -n 1m -r 1k -th 1 -dio /ibm/fs2-1m-07/test
tsqosperf read seq /ibm/fs2-1m-07/test
  recSize 1K nBytes 1M fileSize 1G
  nProcesses 1 nThreadsPerProcess 1
  file cache flushed before test
  using direct I/O
  offsets accessed will cycle through the same file segment
  not using shared memory buffer
  not releasing byte-range token after open
    Data rate was 12904.76 Kbytes/sec, thread utilization 0.998, bytesTransferred 1048576
[root@client01 mpi]# mmfsadm dump iohist |less

I/O history:

 I/O start time RW    Buf type disk:sectorNum     nSec  time ms      tag1      tag2           Disk UID typ      NSD node context   thread
--------------- -- ----------- ---------------- -----  ------- --------- --------- ------------------ --- --------------- --------- ----------
09:26:46.387129  R        data    1:292536326        2    0.081   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387234  R        data    1:292536328        2    0.075   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387333  R        data    1:292536330        2    0.057   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387413  R        data    1:292536332        2    0.057   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387493  R        data    1:292536334        2    0.059   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387576  R        data    1:292536336        2    0.063   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387663  R        data    1:292536338        2    0.059   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387746  R        data    1:292536340        2    0.054   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387824  R        data    1:292536342        2    0.054   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
09:26:46.387901  R        data    1:292536344        2    0.065   8755200         0  C0A70D06:571A90C4 cli  192.167.20.125 MBHandler DioHandlerThread
```

Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
Deepflash 150

High Level Requirements for a Next Gen HPC project

- 2.5 TB/sec single stream IOR
- 1 TB/sec 1MB sequential read/write
- Single Node 16 GB/sec sequential read/write
- **2.6 Million 32k file creates /sec**

Factor 2.5 improvement

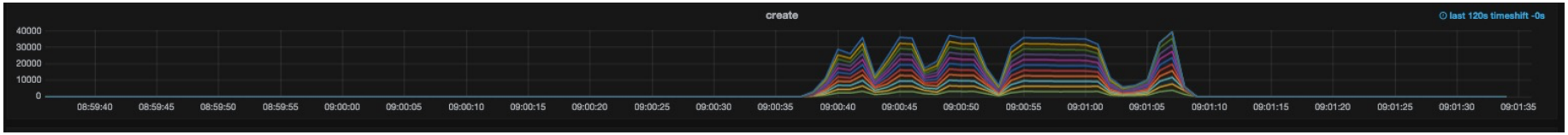
Shared Directory metadata Performance improvement

4.1.1 GA code :

Operation		Max	Min	Mean	Std Dev
File creation	:	11883.662	11883.662	11883.662	0.000
File stat	:	2353513.732	2353513.732	2353513.732	0.000
File read	:	185753.288	185753.288	185753.288	0.000
File removal	:	10934.133	10934.133	10934.133	0.000
Tree creation	:	1468.594	1468.594	1468.594	0.000
Tree removal	:	0.800	0.800	0.800	0.000

4.2 GA code :

Operation		Max	Min	Mean	Std Dev	
File creation	:	28488.144	28488.144	28488.144	0.000	~250%
File stat	:	3674915.888	3674915.888	3674915.888	0.000	~150%
File read	:	188816.195	188816.195	188816.195	0.000	
File removal	:	65612.891	65612.891	65612.891	0.000	~650%
Tree creation	:	501.052	501.052	501.052	0.000	
Tree removal	:	0.497	0.497	0.497	0.000	



*Both tests performed on same 12 node cluster with mdtest -i 1 -n 71000 -F -i 1 -w 1024

Further Shared Directory metadata Performance improvements (tests on DEV code build)

```
-- started at 02/28/2016 16:28:46 --

mdtest-1.9.3 was launched with 22 total task(s) on 11 node(s)
Command line used: /ghome/oehmes/mpi/bin/mdtest-pcmpi9131-existingdir -d /ibm/fs2-1m-
07/shared/mdtest-ec -i 1 -n 71000 -F -i 1 -w 0 -Z -p 8
Path: /ibm/fs2-1m-07/shared
FS: 25.5 TiB   Used FS: 4.8%   Inodes: 190.7 Mi   Used Inodes: 0.0%

22 tasks, 1562000 files

SUMMARY: (of 1 iterations)
   Operation                     Max            Min           Mean        Std Dev
   ---------                     ---            ---           ----        -------
   File creation     :       41751.228      41751.228      41751.228          0.000
   File stat         :     4960208.454    4960208.454    4960208.454          0.000
   File read         :      380879.561     380879.561     380879.561          0.000
   File removal      :      122988.466     122988.466     122988.466          0.000
   Tree creation     :         271.458        271.458        271.458          0.000
   Tree removal      :           0.099          0.099          0.099          0.000

-- finished at 02/28/2016 16:29:58 --
```


NON Shared Directory metadata Performance improvements (tests based on 4.2.1)

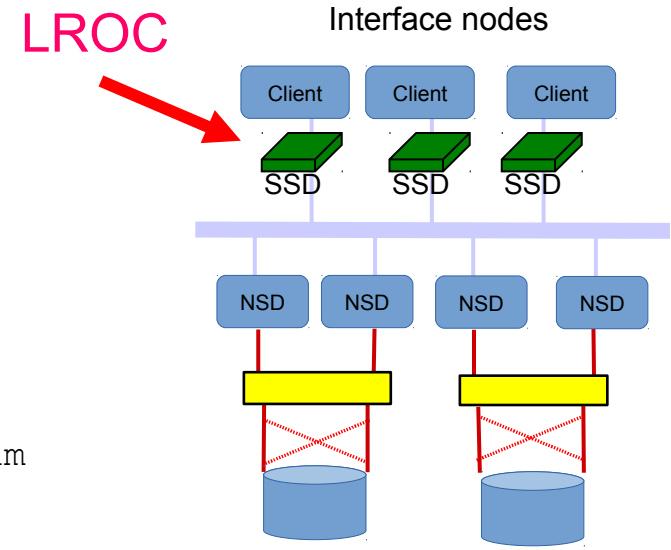
```
-- started at 03/05/2016 05:42:09 --

mdtest-1.9.3 was launched with 48 total task(s) on 12 node(s)
Command line used: /ghome/oehmes/mpi/bin/mdtest-pcmpi9131-existingdir -d /ibm/fs2-1m-
07/shared/mdtest-ec -i 1 -n 10000 -F -i 1 -w 0 -Z -u
Path: /ibm/fs2-1m-07/shared
FS: 22.0 TiB   Used FS: 3.7%   Inodes: 190.7 Mi   Used Inodes: 0.0%

48 tasks, 480000 files

SUMMARY: (of 1 iterations)
   Operation                  Max            Min           Mean        Std Dev
   ---------                  ---            ---           ----        -------
   File creation     :     352119.402     352119.402     352119.402          0.000
   File stat         :    9735705.056    9735705.056    9735705.056          0.000
   File read         :     263264.692     263264.692     263264.692          0.000
   File removal      :     374812.557     374812.557     374812.557          0.000
   Tree creation     :         13.646         13.646         13.646          0.000
   Tree removal      :         10.178         10.178         10.178          0.000

-- finished at 03/05/2016 05:42:14 --
```

Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
Deepflash 150

Spectrum Scale Software Local Read Only Cache (LROC)

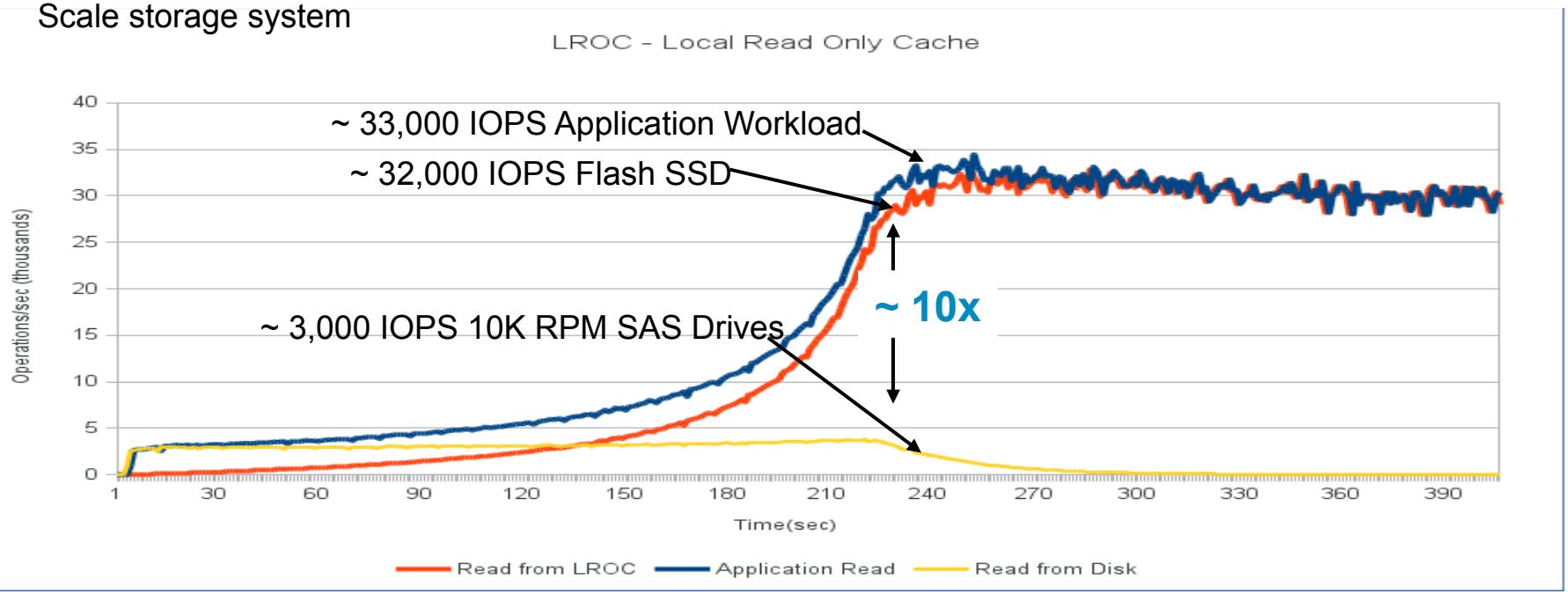
- Many NAS workloads benefit from large read cache
  - SPECsfs
  - VMWare and other virtualization
  - Database
- Augment the Interface Node DRAM cache with SSD
  - Used to cache:
    - Data
    - Inodes
    - Indirect blocks
  - Cache consistency insured by standard Spectrum Scale tokens
  - Assumes SSD device is unreliable, data is protected by checksum and verified on read
  - Provide low-latency access to file system metadata and data
- Implement with consumer flash for maximum Cache/$
  - Enabled by FLEA's LSA (Data is writte Sequential to Device, to eliminate wear leveling)
  - Reach small File performance leadership compared to other NAS Devices



Add 100's of GBs of SSD to each interface node

LROC Example Speed Up

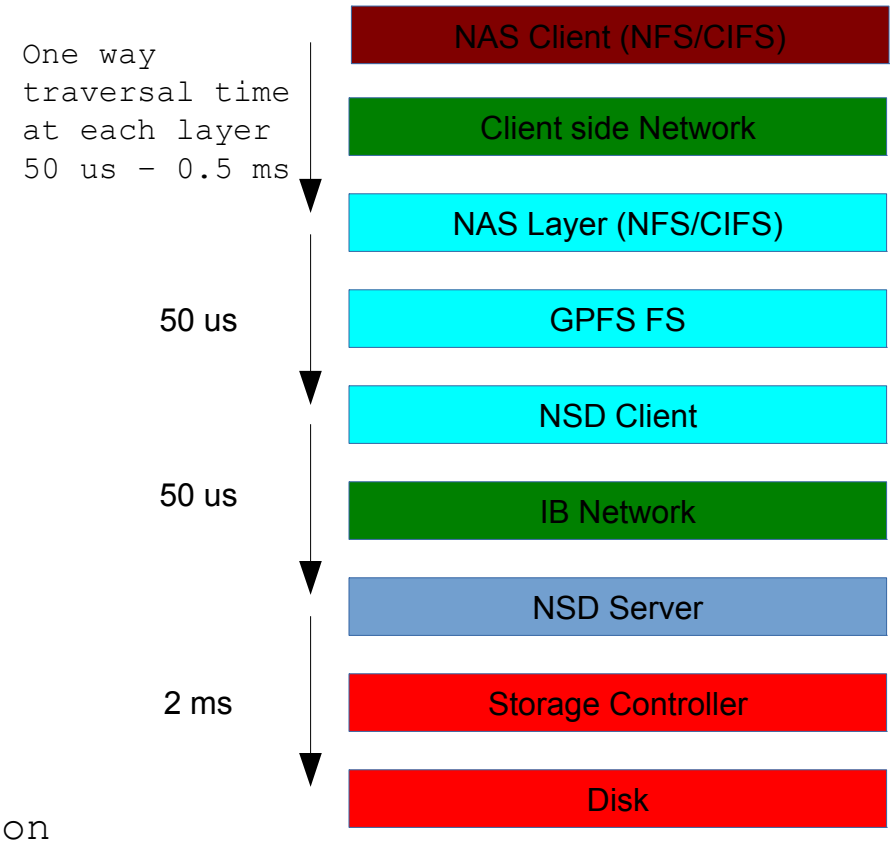
- Two consumer grade 200 GB SSDs cache a forty-eight 300 GB 10K SAS disk Spectrum Scale storage system



- Initially, with all data coming from the disk storage system, the client reads data from the SAS disks at ~ 5,000 IOPS
- As more data is cached in Flash, client performance increases to 33,000 IOPS while reducing the load on the disk subsystem by more than 95%

Pain Point: Small and Synchronous Write Performance

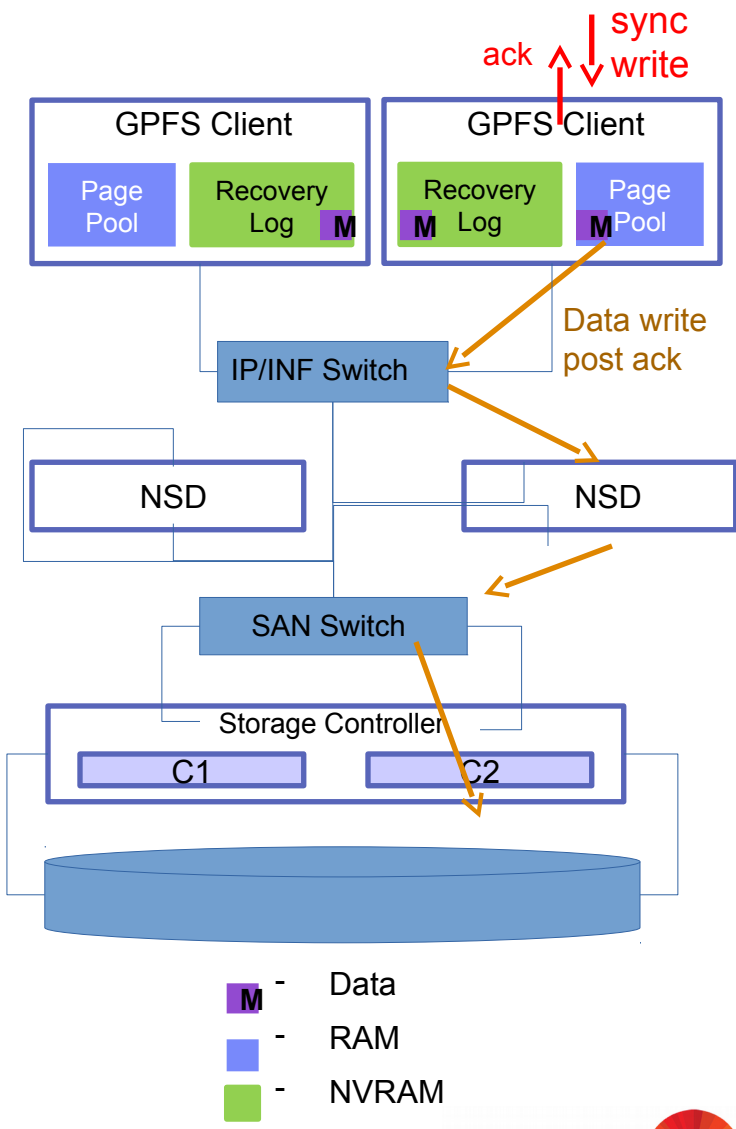
- Common issue in
  - Small and medium-sized workloads
  - EDA workload
  - Virtual Machine Solutions
- Issue across wide range of workloads
  - VMs
  - Databases
  - Windows home directory
  - Logging
  - ISSM (ECM, Websphere, etc)
- Require low-latency and non-volatile memory
  - Flash-backed DIMMs
  - Large batteries
  - Fast SSDs (Fusion-IO, etc)
  - FLASHSYSTEMS
- Cannot optimize data path in isolation
  - Recovery log updates occur on application writes to sparse files, e.g., VM disk images



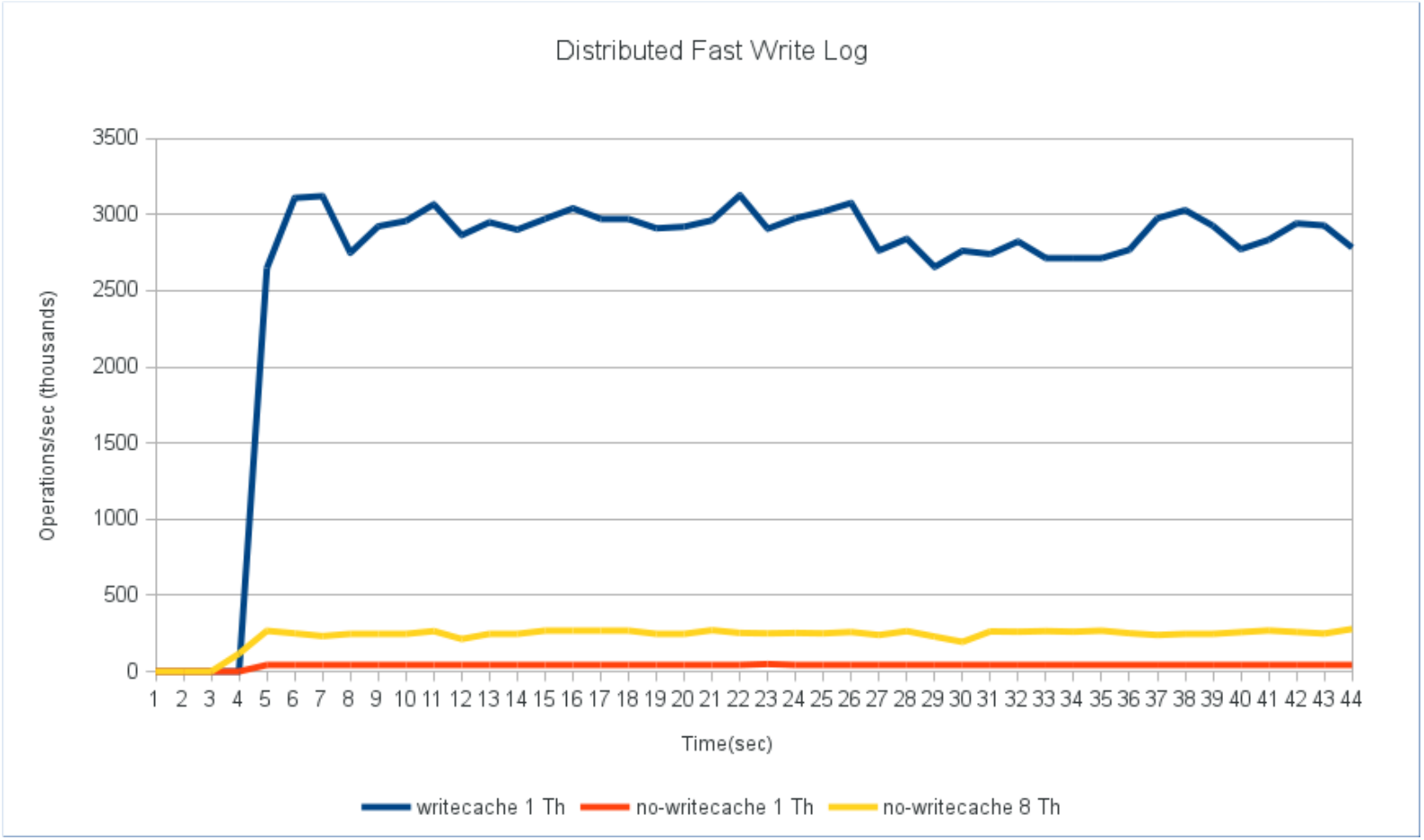
4KB Total Round Trip Time = ~5 ms

Solution : HAWC – Highly available Write Cache

- HAWC (Log writes)
  - Store recovery log in client NVRAM
  - Either replicate in pairs or store on shared storage
  - Log writes in recovery log
  - Log small writes and send large writes directly to disk
  - Logging data only hardens it
  - Data remains in pagepool and is sent to disk post-logging
    - Leverages write gathering
    - Fast read-cache performance
  - On node failure, run recovery log to place data on disk



Synthetic Benchmark with local attached JBOD disks



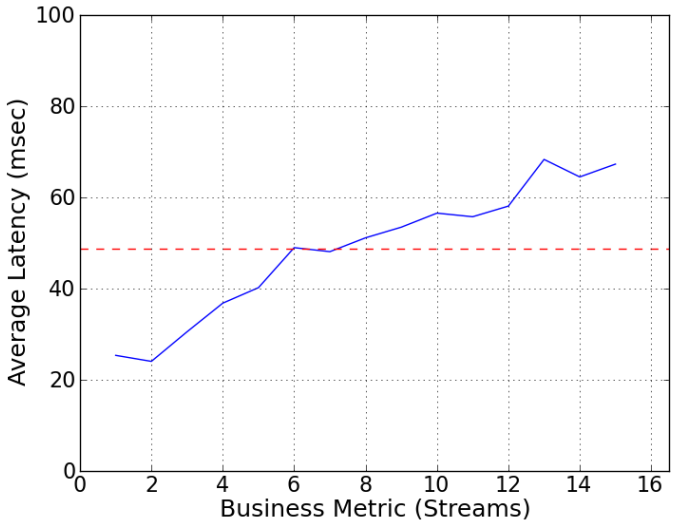
Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
Deepflash 150

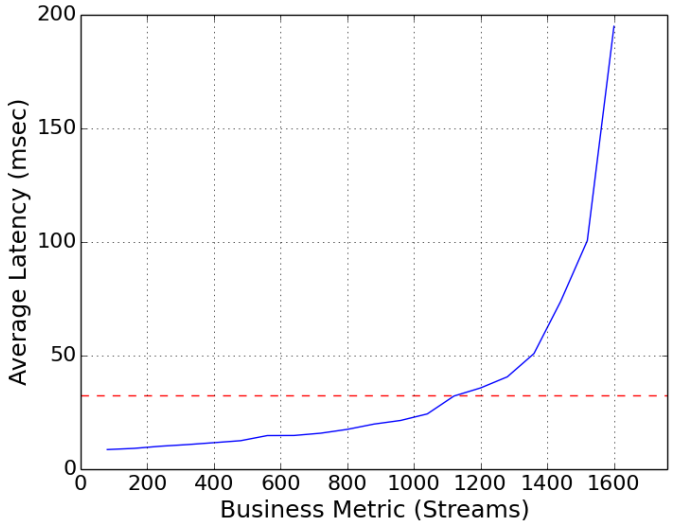
First SpecSFS 2014 VDA Publication

Factor ~30 faster per spindle

different scale in graphs !



SpecSFS2014 Reference Solution [1]
with 96 x 10k SAS drives
15 Streams @ 48.79 ms



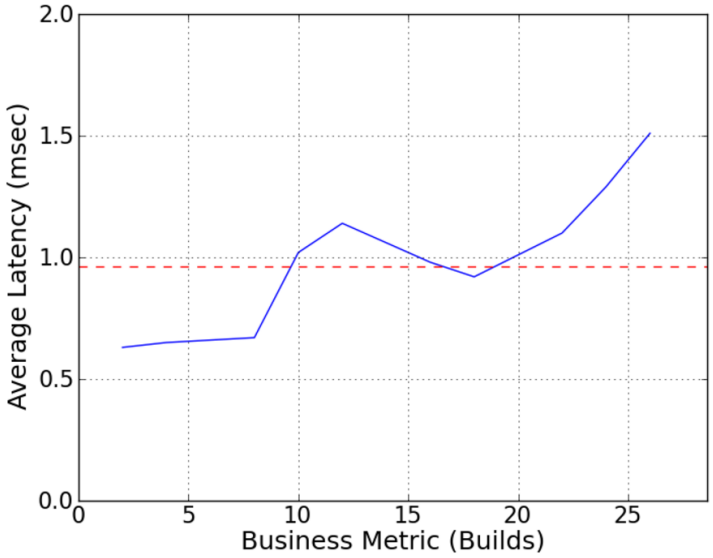
Single ESS – GL6 with 348 x 7.2k
NLSAS disks [2]
1600 Streams @ 33.98 ms

[1] https://www.spec.org/sfs2014/results/res2014q4/sfs2014-20141029-00003.html
[2] https://www.spec.org/sfs2014/results/res2016q2/sfs2014-20160411-00012.html

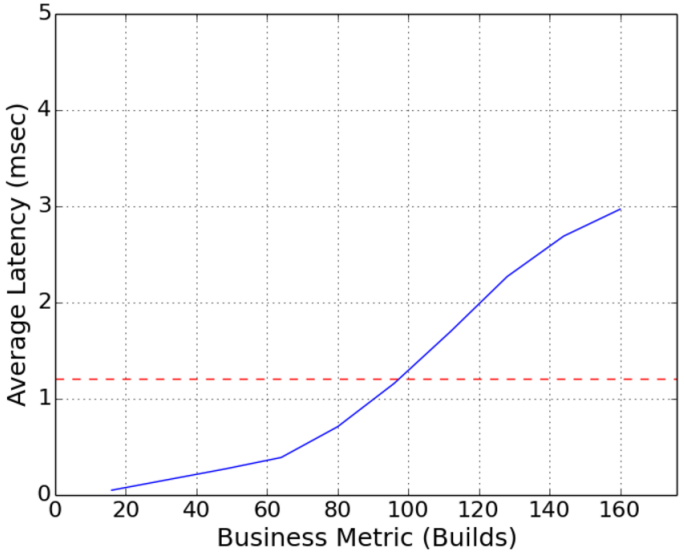
Factor ~2 faster per spindle

First SpecSFS 2014 SWBUILD Publication

different scale in graphs !



SpecSFS2014 Reference Solution [1]
with 96 x 10k SAS drives
26 Builds @ 0.96 ms



Single ESS – GL6 with 348 x 7.2k
NLSAS disks [2]
160 Builds @ 1.21 ms

[1] https://www.spec.org/sfs2014/results/res2014q4/sfs2014-20141029-00002.html
[2] https://www.spec.org/sfs2014/results/res2016q2/sfs2014-20160411-00013.html

Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
Deepflash 150

~25% speedup in write performance in 4.2.1

Spectrum Scale Raid large block random performance on GL6

```
Summary:
        api                = POSIX
        test filename      = /ibm/fs2-1m-07/shared/ior//iorfile
        access             = file-per-process
        pattern            = segmented (1 segment)
        ordering in a file = sequential offsets
        ordering inter file= no tasks offsets
        clients            = 12 (1 per node)
        repetitions        = 10
        xfersize           = 1 MiB
        blocksize          = 64 GiB
        aggregate filesize = 768 GiB
Using Time Stamp 1463398064 (0x5739aeb0) for Data Signature
delaying 10 seconds . . .
Commencing write performance test.
Mon May 16 04:27:54 2016
access    bw(MiB/s)  block(KiB) xfer(KiB)  open(s)    wr/rd(s)   close(s) total(s)  iter
------    ---------  ---------- ---------  --------   --------   --------  --------   ----
write     20547      67108864   1024.00    0.560932   38.27      0.065744   38.27      0    XXCEL
delaying 10 seconds . . .
[RANK 000] open for reading file /ibm/fs2-1m-07/shared/ior//iorfile.00000000 XXCEL
Commencing read performance test.
Mon May 16 04:28:42 2016
read      26813      67108864   1024.00    0.000217   29.33      0.355600   29.33      0    XXCEL
Using Time Stamp 1463398151 (0x5739af07) for Data Signature
delaying 10 seconds . . .

… removed redundant repetitions

read      24675      67108864   1024.00    0.000132   31.87      0.336031   31.87      1    XXCEL
Using Time Stamp 1463398241 (0x5739af61) for Data Signature
Operation  Max (MiB)  Min (MiB)  Mean (MiB)   Std Dev  Max (OPs)  Min (OPs)  Mean (OPs)   Std Dev  Mean (s)  Op grep #Tasks tPN reps  fPP firstF reord
reordoff reordrand seed segcnt blksiz xsize aggsize
---------  ---------  ---------  ----------   -------  ---------  ---------  ----------   -------  --------
write       21115.04   20227.35    20674.95    249.05   21115.04   20227.35    20674.95    249.05  38.04344   12 1 10 1 0 0 1 0 0 1 68719476736
1048576 824633720832 -1 POSIX EXCEL
read        26813.17   23646.23    25236.65    878.94   26813.17   23646.23    25236.65    878.94  31.20020   12 1 10 1 0 0 1 0 0 1 68719476736
1048576 824633720832 -1 POSIX EXCEL
Max Write: 21115.04 MiB/sec (22140.73 MB/sec)
Max Read:  26813.17 MiB/sec (28115.65 MB/sec)
Run finished: Mon May 16 04:42:36 2016
```

Spectrum Scale Raid rebuild performance on GL6-2T 8+2p

5:30 min for critical rebuild - 10x improvement

As one can see during the critical rebuild impact on workload was high, but as soon as we were back to a single parity protection the impact to the customers workload was <2%

Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
DeepFlash 150

Realtime Performance Monitoring – OpenTSDB bridge used by Grafana

Agenda

Easier Tuning in 4.2.1 – 'Auto scale' Performance Optimization
Communication Overhaul – lower latency, higher scale
Update on Non-Shared / Shared directory metadata performance
Flash Acceleration
Benchmark Publications
GNR Rebuild & Performance Improvements
Realtime Performance Monitoring – OpenTSDB bridge
DeepFlash 150

Deepflash 150 – GA August 2016

Data platform for Analytics

Example: Hadoop, Spark, SAP Hana

Unified data repository, support multiple analytics

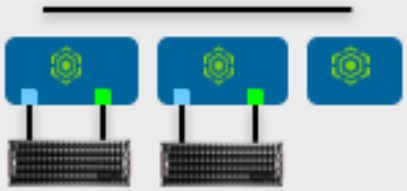
Key Advantage:

- Faster time to insights
- Load and off load data to and from memory faster
- Shared data platform for multiple instance and forms of analytics

High Bandwidth Data Tier

Example: Digital Media, Life Science

Move entire working data set to flash, high availability through Spectrum Scale



Key Advantage:

- High bandwidth, low latency data tier
- Complete set of enterprise storage services and enterprise availability

Burst Buffer

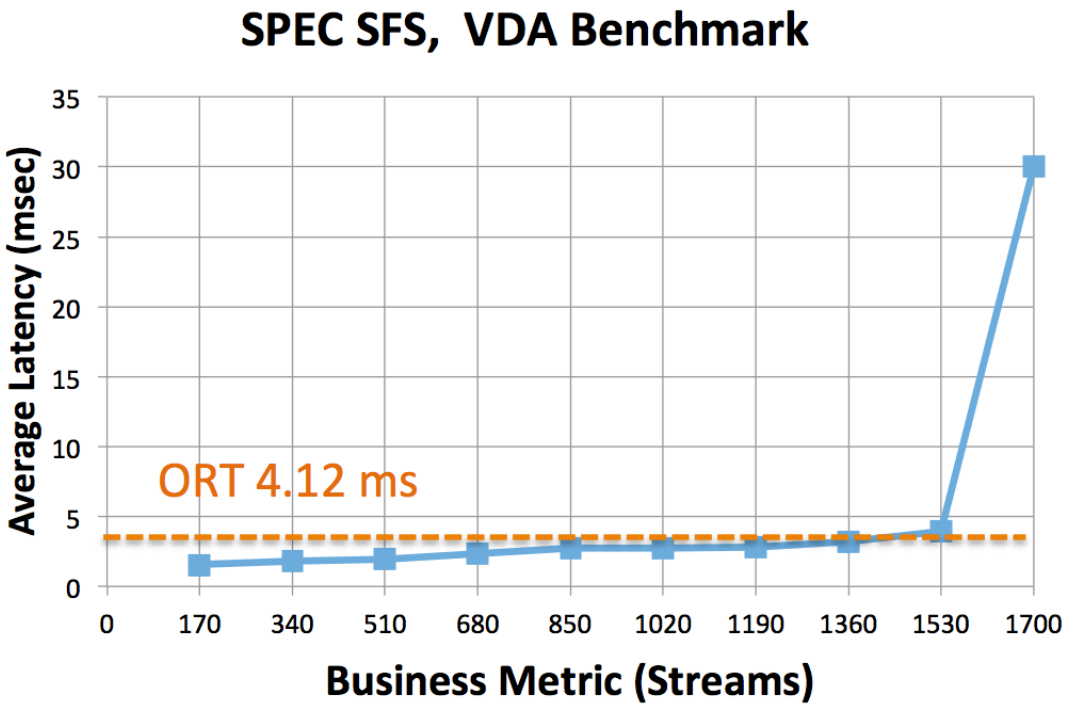
Example: R/W buffer for HPC

Large dedicated tier to speed up writes & especially reads

Key Advantage:

- Manage burst read/write patterns common to HPC applications
- Speed up MPI and check-pointing

SpecSFS 2014 - VDA



Business Metric (Streams)	Average Latency (ms)	Streams Ops/ sec	Streams MB/sec
170	1.57	1701	785
340	1.82	3401	1568
510	1.94	5102	2356
680	2.34	6803	3134
850	2.74	8504	3917
1020	2.73	10204	4714
1190	2.83	11905	5492
1360	3.21	13606	6282
1530	3.93	15306	7066
1700	30	16980	7821

1700 Streams, Overall Response Time (ORT) 4.12 ms

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, Bluemix, Blueworks Live, CICS, Clearcase, DOORS®, Enterprise Document Management System™, Global Business Services ®, Global Technology Services ®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, SoDA, SPSS, StoredIQ, Tivoli®, Trusteer®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.