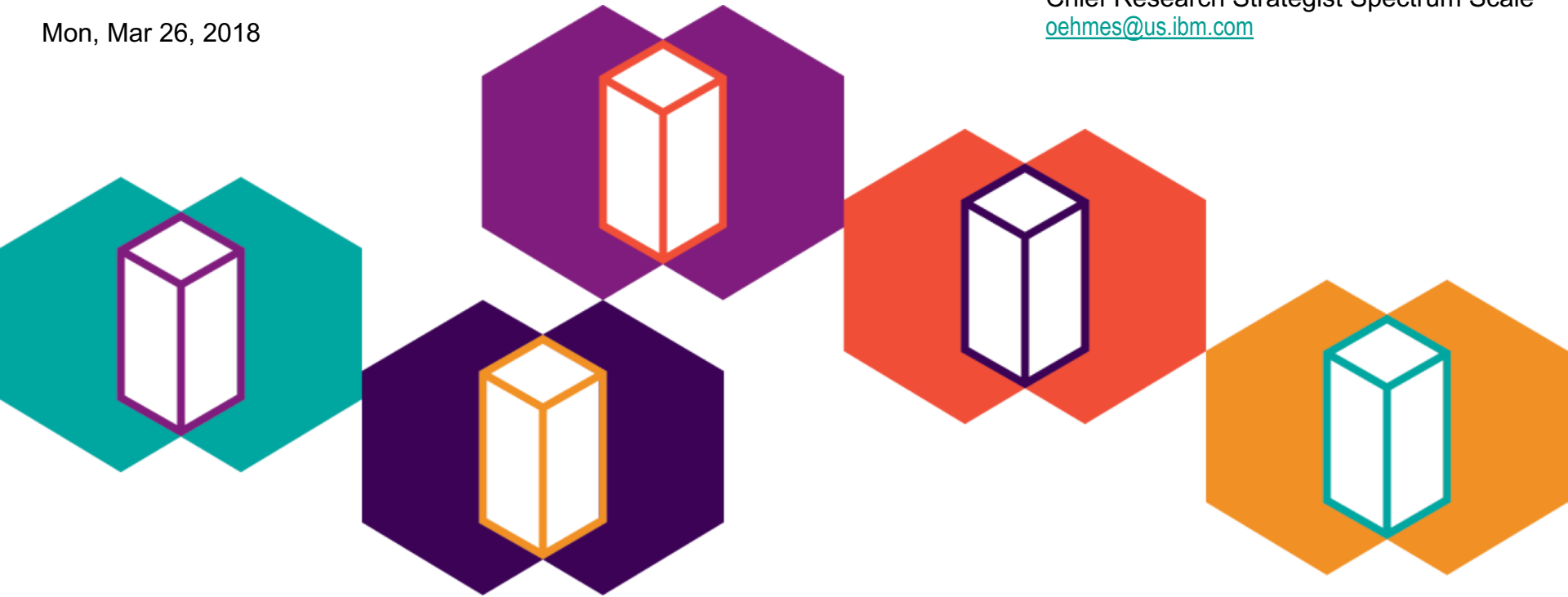
Spectrum Scale – CORAL Enhancements

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

Mon, Mar 26, 2018

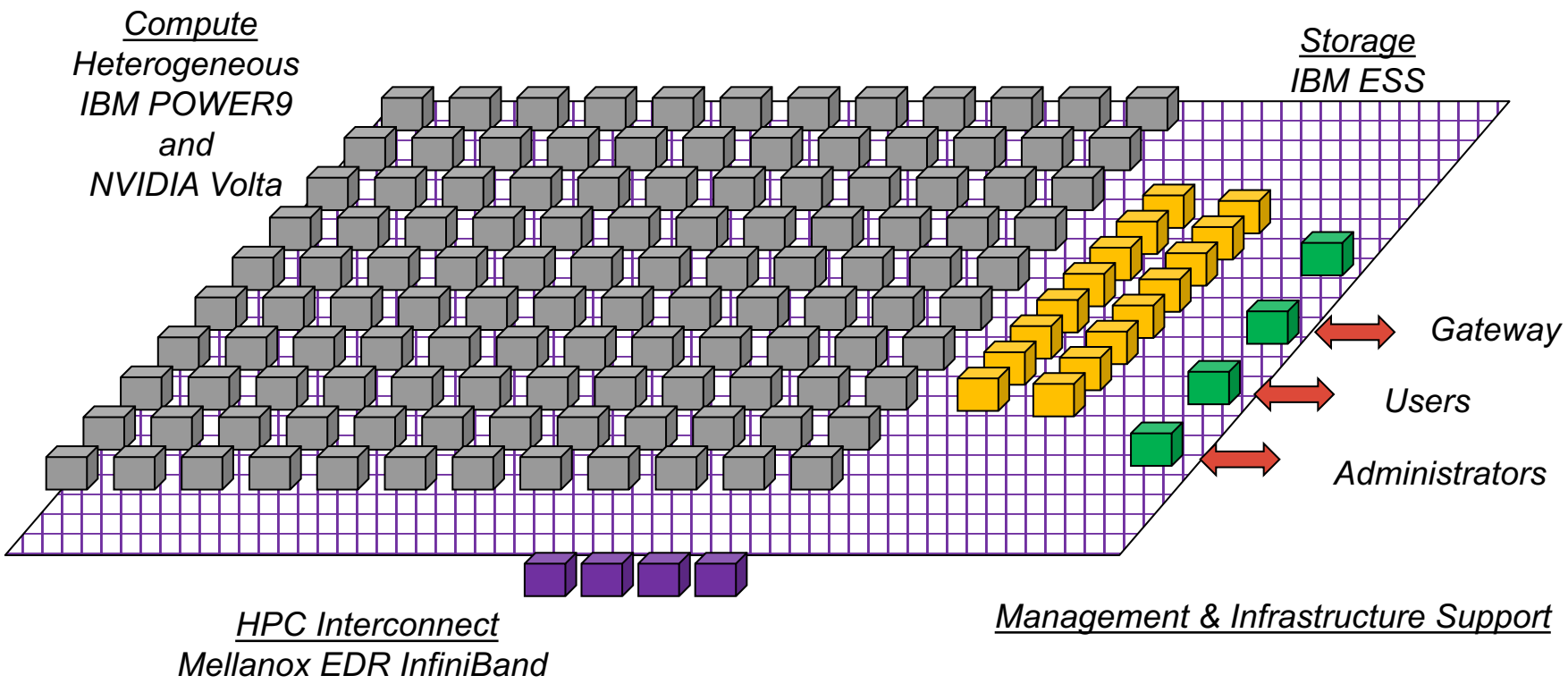


Spectrum Scale

WHAT is CORAL ?

The first rows of the CORAL Project

High Level Layout of the complete system



Serial Number 0 – First CORAL P9 Based ESS Prototype

1 Half – rack building Block – 20U ~2000 Pound

2x P9 based dual Socket Control Nodes
4x 4U 106 Drive enclosures , each 104 HDD

4 PB of HDD
1 TB Memory
NVMe attached DRAM for Write caching
90 GB/sec Network connectivity (4x EDR per node)

Scalability Targets for SPIDER 3 (Oak Ridge Leadership Computing Facility)

The single namespace CFS will meet the following

- Single name space supporting 250 PB capacity
- Total number of files supported is 100 B
- Maximum file size equal to aggregate system memory
- 10 M files per directory

Enhancements needed in Spectrum Scale

- Improvements in fsck – time to run (including nodes to use), progress reporting, …
- Parallel virtual disk creation
- Reduce contention to allow more concurrency

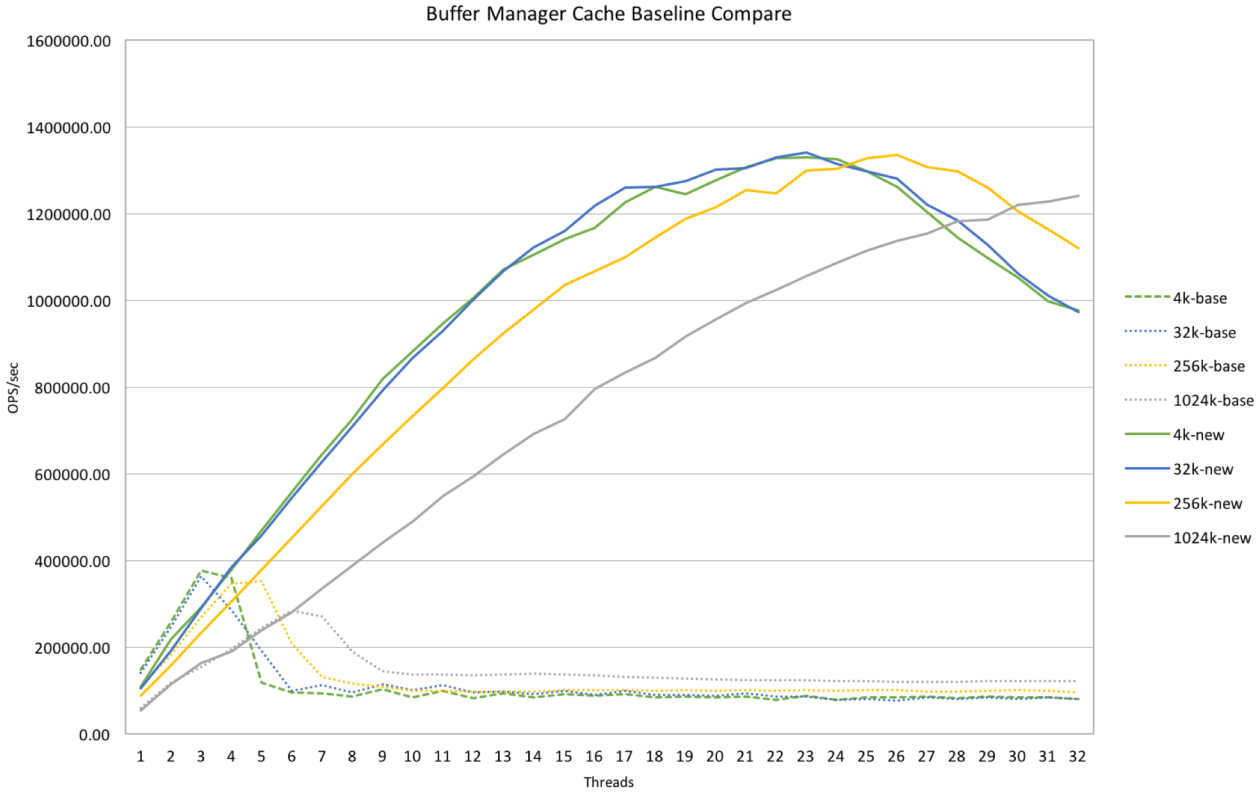
Specific performance targets

- Single Node 16 GB/sec sequential read/write as requested from ORNL
- Performs at an aggregate sequential peak read/write bandwidth of 2.5 TB/s
- Performs at an aggregate random peak read/write bandwidth of 2.2 TB/s
- Provides rich metadata performance - single directory parallel create rate of 50,000/s
- Provides rich interactive performance - @32 KiB I/O 2.6 million IOPs

Spectrum Scale

VBUF2 - GNR new buffer manager

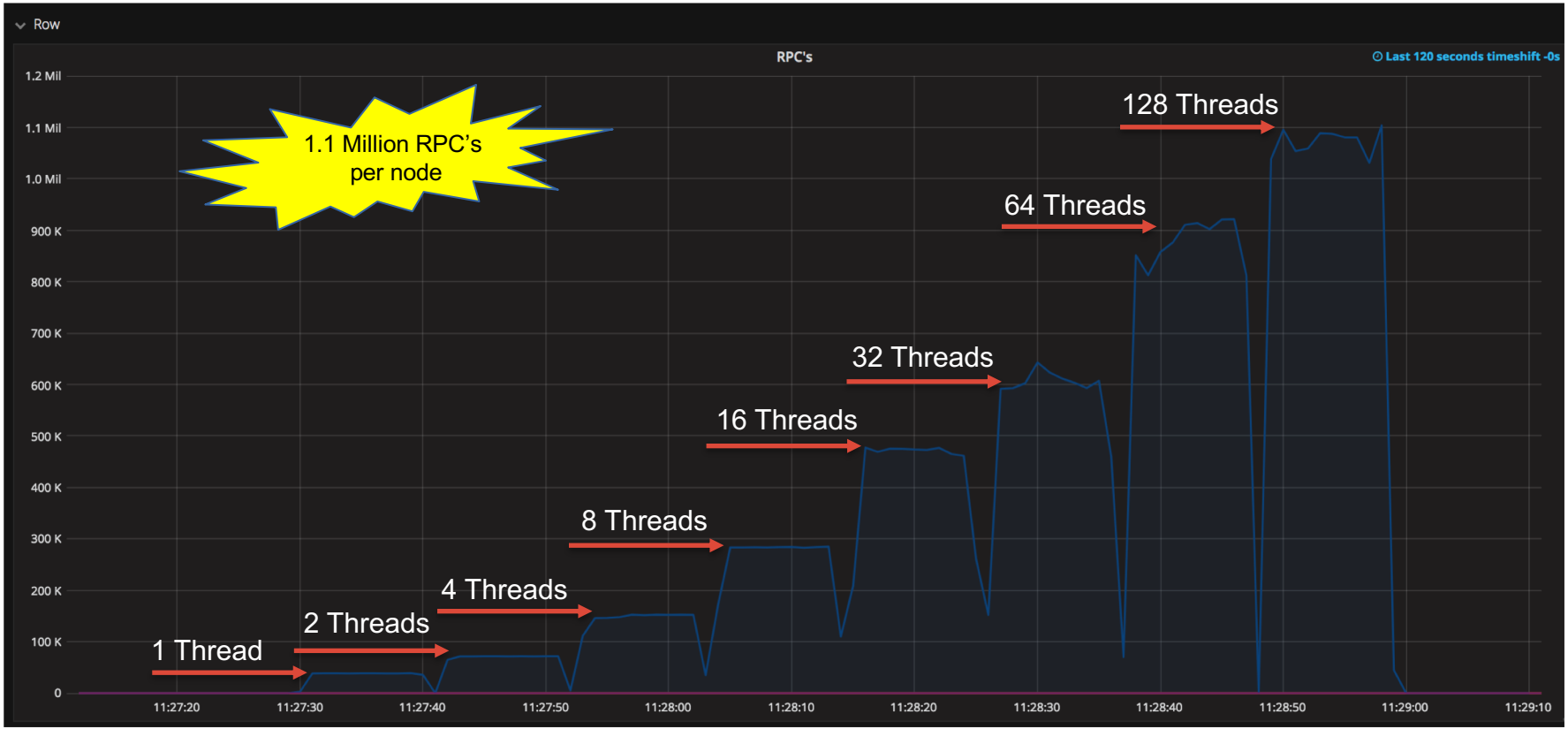
GNR level local out of cache Performance 4.2.3 vs 5.0.0 (VBUF2)



Spectrum Scale

Massive Communication Overhaul

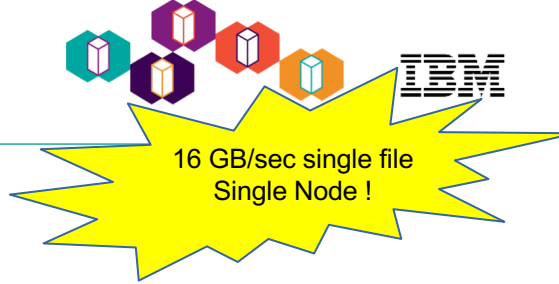
2 Node communication code scaling with 1 byte RPC's



Spectrum Scale

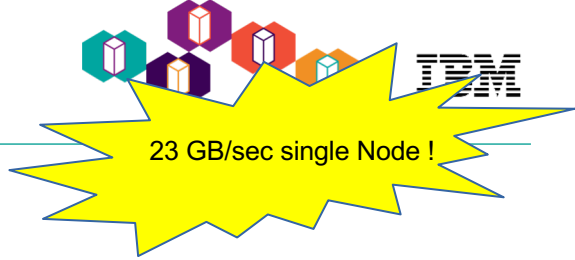
some proof points

Single client throughput enhancements – single file




```
[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 client throughput enhancements

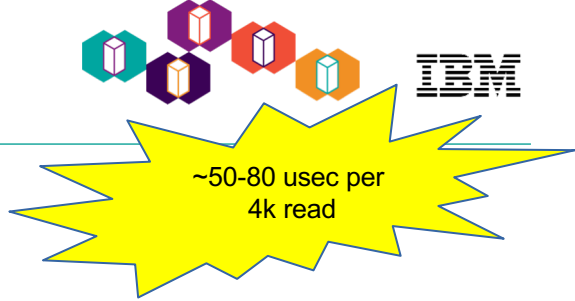



```
Began: Sat Nov 11 20:47:05 2017
Command line used: /perform/io-500-dev.ppc64le//bin/ior -w -C -Q 1 -g -G 27 -k -e -t 16m -b 128g -F -o /ibm/fs2-16m-10/shared/iorfile
Machine: Linux p8n19hyp

Test 0 started: Sat Nov 11 20:47:05 2017
Summary:
        api                = MPIIO (version=3, subversion=1)
        test filename      = /ibm/fs2-16m-10/shared/iorfile
        access             = file-per-process
        ordering in a file = sequential offsets
        ordering inter file= constant task offsets = 1
        clients            = 8 (8 per node)
        repetitions        = 1
        xfersize           = 16 MiB
        blocksize          = 128 GiB
        aggregate filesize = 1024 GiB

access    bw(MiB/s)  block(KiB) xfer(KiB)  open(s)    wr/rd(s)   close(s)   total(s)   iter
------    ---------  ---------- ---------  --------   --------   --------   --------   ----
write     22261      134217728  16384      0.021629   47.08      0.001264   47.10      0

Max Write: 22261.01 MiB/sec (23342.36 MB/sec)

Summary of all tests:
Operation   Max(MiB)   Min(MiB)  Mean(MiB)     StdDev    Mean(s) Test# #Tasks tPN reps fPP reord reordoff reordrand seed segcnt blksiz xsize aggsize API RefNum
write       22261.01   22261.01   22261.01       0.00   47.10371 0 8 8 1 1 1 1 0 0 1 137438953472 16777216 1099511627776 MPIIO 0

Finished: Sat Nov 11 20:47:52 2017
```

IOR with GL4c – ESS with CORAL Enclosure (reduced output) – 16M

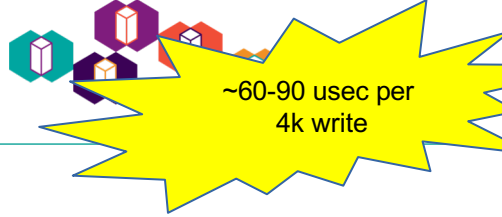

```
Began: Fri Oct 27 01:34:10 2017
Command line used: /tmp/ior-binary-dir/ior -F -i 3 -d 180 -w -r -e -t 16m -b 4064g -o /ibm/fs2-16m-10/ior-test-dir-1/iorfile -L
Machine: Linux fire01.sonasad.almaden.ibm.com

Test 0 started: Fri Oct 27 01:34:10 2017
Summary:
        api                = POSIX
        test filename      = /ibm/fs2-16m-10/ior-test-dir-1/iorfile
        access             = file-per-process
        ordering in a file = sequential offsets
        ordering inter file= no tasks offsets
        clients            = 12 (1 per node)
        repetitions        = 3
        xfersize           = 16 MiB
        blocksize          = 4064 GiB
        aggregate filesize = 48768 GiB

Max Write: 34507.52 MiB/sec (36183.76 MB/sec)
Max Read:  41420.56 MiB/sec (43432.61 MB/sec)

Summary of all tests:
Operation   Max(MiB)   Min(MiB)  Mean(MiB)     StdDev    Mean(s) Test# #Tasks tPN reps fPP reord reordoff reordrand seed segcnt blksiz xsize aggsize API
RefNum
write       34507.52   34321.65   34418.00      76.04 1450.94658 0 12 1 3 1 0 1 0 0 1 4363686772736 16777216 52364241272832 POSIX 0
read        41420.56   41210.13   41340.40      92.93 1207.98744 0 12 1 3 1 0 1 0 0 1 4363686772736 16777216 52364241272832 POSIX 0

Finished: Fri Oct 27 04:05:07 2017
```

Single thread 4k seq read i/o (client – server roundtrip) - NORAID




```
[root@client01 ~]#  tsqosperf read seq -r 4k /ibm/fs2-256k-08/shared/test –dio
tsqosperf read seq /ibm/fs2-256k-08/shared/test
  recSize 4K nBytes 128M fileSize 128M
  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 55111.52 Kbytes/sec, Op Rate was 13454.96 Ops/sec, Avg Latency was 0.074 milliseconds, thread utilization 1.000, bytesTransferred
134217728

 [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
--------------- -- ----------- ---------------- -----  ------- --------- --------- ------------------ --- --------------- --------- ----------
11:37:54.451846  R        data    4:192933224        8    0.055    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
11:37:54.451918  R        data    4:192933232        8    0.055    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
11:37:54.451990  R        data    4:192933240        8    0.054    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
11:37:54.452061  R        data    4:192933248        8    0.054    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
11:37:54.452132  R        data    4:192933256        8    0.055    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
11:37:54.452205  R        data    4:192933264        8    0.053    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
11:37:54.452275  R        data    4:192933272        8    0.057    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
11:37:54.452349  R        data    4:192933280        8    0.056    284160       504  C0A74D01:58BD6495 cli  192.167.20.127 MBHandler DioHandlerThread
```

Single thread 4k random write i/o (client – server roundtrip) - GNR



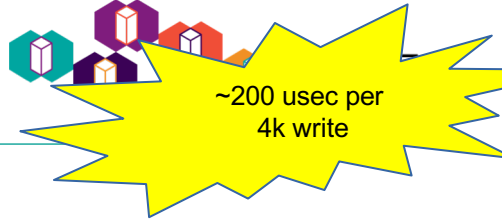
[root@client07 ~]# tsqosperf write rand -r 4k -n 1t -millis 10000 -dio -th 1 /gpfs/nvme/testfiles
tsqosperf write rand /gpfs/nvme/testfiles
  recSize 4K nBytes 1024G fileSize 32G
  nProcesses 1 nThreadsPerProcess 1
  file cache flushed before test
  using direct I/O
  offsets accessed will cycle through the same file segment
  not using shared memory buffer
  not releasing byte-range token after open
  no fsync at end of test
    Data rate was 47846.40 Kbytes/sec, Op Rate was 11681.25 Ops/sec, Avg Latency was **0.086 milliseconds**, thread utilization 1.000, bytesTransferred 478478336

as can be seen by the breakdown below where the time is spend most requests need ~60 usec in the lower i/o layer (time in ms below) with ~30 usec in Network and ~18 usec on the media

I/O history:

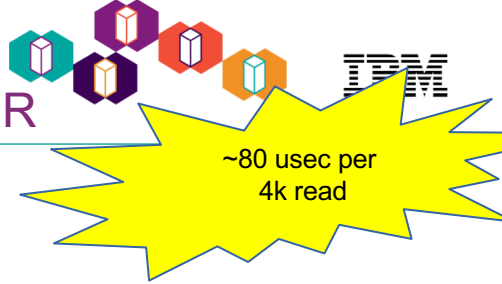
I/O start time	RW	Buf type	disk:sectorNum	nSec	time ms	tag1	tag2	Disk UID	typ	NSD node	context	thread	comment	qTime ms	RpcTimes ms	
09:40:21.767122	W	data	2:3000800	8	0.056	337920	974	C0A70D01:59726709	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.029	0.016
09:40:21.767201	W	data	1:596568	8	0.057	337920	595	C0A70D01:59726707	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.029	0.016
09:40:21.767281	W	data	2:4353872	8	0.055	337920	2222	C0A70D01:59726709	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.028	0.016
09:40:21.767359	W	data	3:6357784	8	0.067	337920	1824	C0A70D01:59726708	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.029	0.024
09:40:21.767452	W	data	2:12000944	8	0.062	337920	4013	C0A70D01:59726709	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.032	0.018
09:40:21.767538	W	data	3:5161904	8	0.058	337920	1638	C0A70D01:59726708	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.029	0.016
09:40:21.767618	W	data	3:17427776	8	0.060	337920	5874	C0A70D01:59726708	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.031	0.017
09:40:21.767701	W	data	1:10701688	8	0.063	337920	3286	C0A70D01:59726707	cli	192.167.13.8	MBHandler	DioHandlerThread		0.000	0.031	0.019

Single thread 4k seq write i/o (client – server roundtrip) - GNR



[root@client01 mpi]# gpfsperf write seq -n 200g -r 4k -millis 5000 -th 1 /gpfs/fs2-16m-09/shared/testfile –dio
gpfsperf write seq /gpfs/fs2-16m-09/shared/testfile
  recSize 4K nBytes 200G fileSize 200G
  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
  no fsync at end of test
   Data rate was 19840.34 Kbytes/sec, Op Rate was 4843.83 Ops/sec, Avg Latency was 0.206 milliseconds, thread utilization 1.000, bytesTransferred 99209216

Single thread 4k random read i/o (client – server roundtrip) - GNR



[root@client01 mpi]# gpfsperf read rand -n 200g -r 4k -millis 5000 -th 1 /gpfs/fs2-16m-09/shared/testfile –dio
gpfsperf read rand /gpfs/fs2-16m-09/shared/testfile
  recSize 4K nBytes 200G fileSize 200G
  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 52539.99 Kbytes/sec, Op Rate was 12827.15 Ops/sec, Avg Latency was 0.078 milliseconds, thread utilization 1.000, bytesTransferred 262717440

Spectrum Scale

shared file create performance

Shared directory file create – 50k target

```
-- started at 06/17/2017 11:14:09 --

mdtest-1.9.3 was launched with 23 total task(s) on 23 node(s)
Command line used: /tmp/mdtest-binary-dir/mdtest -d /ibm/fs2-16m-09/mdtest-dir-2 -i 1 -n 50000 -F -w 1024 -C -r
-T -p 15 -X
Path: /ibm/fs2-16m-09
FS: 64.0 TiB   Used FS: 0.0%   Inodes: 476.8 Mi   Used Inodes: 0.2%

23 tasks, 1150000 files

SUMMARY: (of 1 iterations)
   Operation                     Max            Min           Mean        Std Dev
   ---------                     ---            ---           ----        -------
   File creation     :      56935.959      56935.959      56935.959          0.000
   File stat         :    6305129.988    6305129.988    6305129.988          0.000
   File read         :          0.000          0.000          0.000          0.000
   File removal      :      63040.815      63040.815      63040.815          0.000
   Tree creation     :       7628.933       7628.933       7628.933          0.000
   Tree removal      :          0.359          0.359          0.359          0.000
```

New 5.0.0+ Allocation code optimizations

```
-- started at 02/09/2018 10:43:01 --

mdtest-1.9.3 was launched with 240 total task(s) on 12 node(s)
Command line used: /tmp/mdtest-binary-dir/mdtest -d /ibm/fs2-16m-10/mdtest-60 -i 1 -n 36864 -w 32768 -C -F -r -u -y
Path: /ibm/fs2-16m-10
FS: 96.3 TiB   Used FS: 3.7%   Inodes: 476.8 Mi   Used Inodes: 0.0%

240 tasks, 8847360 files

SUMMARY: (of 1 iterations)
   Operation                     Max            Min           Mean        Std Dev
   ---------                     ---            ---           ----        -------
   File creation     :      73870.935      73870.935      73870.935          0.000
   File stat         :          0.000          0.000          0.000          0.000
   File read         :          0.000          0.000          0.000          0.000
   File removal      :     249156.292     249156.292     249156.292          0.000
   Tree creation     :          0.492          0.492          0.492          0.000
   Tree removal      :          0.156          0.156          0.156          0.000

-- finished at 02/09/2018 10:45:45 --
```

Spectrum Scale

small file performance – non-shared create

More than 32 Sub blocks - why and what to expect ?

Why do we have Sub blocks ?

- Allow finer grained allocation – no space wasted
- Allows coalescing of small files in larger blocks – raid friendly

What Options do we have today ?

- We can store data in inode (default <4k)
- We can allocate a Sub block (1/32th of a Full block)
- We support 64 KB, 128 KB, 256 KB, 512 KB, 1 MB, 2 MB, 4 MB, 8 MB and 16 MB block size today

What's wrong with it ?

- You have to choose between waste space for small files (>4k and <1/32th of block size) or bandwidth
- You can never ever change it online, filesystem migration required
- It has a significant performance penalty for small files in large block size filesystems

So how do we fix it and what will it change ?

New 5.0 on-disk format changed defaults for new FS

```
New 5.0.0 fs created with default 4MB blockSize and
subblocksPerFullBlock will be derived from block size as follows:

block size  per full block subblock size
----------  -------------- -------------
64k          32           2k
128k         32           4k
256k         32           8k   <=== old file system default
384k         32          12k
512k         64           8k
1M          128           8k
2M          256           8k
4M          512           8k   <=== new file system default
8M          512          16k
16M        1024          16k
32M        2048          16k

relatime ais now the default !!
```

Best way to find out – measure it with mdtest (16M Blocksize)

IBM

4.2.1 base code - SUMMARY: (of 3 iterations)

Operation		Max	Min	Mean	Std Dev
File creation	:	2296.791	2197.553	2237.644	42.695
File stat	:	3402913.848	3383139.838	3390622.546	8759.559
File read	:	452144.282	383467.565	426670.673	30712.367
File removal	:	202219.699	88486.720	160499.542	51134.019
Tree creation	:	9425.078	3138.312	6945.652	2732.932
Tree removal	:	6710.394	3063.299	5196.237	1551.879

zero-end-of-file-padding (4.2.2 + ifdef for zero padding): SUMMARY: (of 3 iterations)

Operation		Max	Min	Mean	Std Dev
File creation	:	13053.701	12570.060	12866.842	212.194
File stat	:	4077992.847	3291830.765	3600173.039	342592.742
File read	:	450592.091	408552.363	424759.494	18462.970
File removal	:	105876.511	93884.369	99224.908	4982.772
Tree creation	:	8451.948	1936.832	4123.063	3061.035
Tree removal	:	535.050	154.181	363.642	157.800

more sub blocks per block (4.2.2 + morethan32subblock code):

Operation		Max	Min	Mean	Std Dev
File creation	:	51397.549	33005.542	40316.721	7967.608
File stat	:	3326016.821	3195765.701	3277674.290	58231.427
File read	:	616434.716	543430.803	568013.424	34240.371
File removal	:	134732.546	48867.351	86175.005	35945.588
Tree creation	:	7771.893	1039.578	3648.852	2949.535
Tree removal	:	2879.694	550.493	1859.348	972.530

Non shared directory mdtest 32k files with Scale 5.0


```
-- started at 10/16/2017 08:46:41 --

mdtest-1.9.3 was launched with 228 total task(s) on 12 node(s)
Command line used: /tmp/mdtest-binary-dir/mdtest -d /ibm/fs2-16m-10/mdtest-50000 -i 1 -n 65536 -w 32768 -C -F -r -u -W -U
Path: /ibm/fs2-16m-10
FS: 128.1 TiB   Used FS: 14.8%   Inodes: 476.8 Mi   Used Inodes: 0.0%

228 tasks, 14942208 files

SUMMARY: (of 1 iterations)
   Operation                     Max            Min           Mean        Std Dev
   ---------                     ---            ---           ----        -------
   File creation     :      56136.303      56136.303      56136.303          0.000
   File stat         :          0.000          0.000          0.000          0.000
   File read         :          0.000          0.000          0.000          0.000
   File removal      :     136952.900     136952.900     136952.900          0.000
   Tree creation     :          1.628          1.628          1.628          0.000
   Tree removal      :          0.054          0.054          0.054          0.000

-- finished at 10/16/2017 08:53:15 --
```

Non shared directory mdtest zero-length with Scale 5.0

```
-- started at 10/17/2017 22:29:16 --

mdtest-1.9.3 was launched with 88 total task(s) on 11 node(s)
Command line used: /ghome/oehmes/mpi/bin/mdtest-pcmpi9131-existingdir -d /ibm/fs2-16m-09/shared/mdtest-ec -i 1 -n 10000 -F -w 0 -Z -p 8 -N 11 -u
Path: /ibm/fs2-16m-09/shared
FS: 128.1 TiB   Used FS: 0.2%   Inodes: 476.8 Mi   Used Inodes: 0.0%

88 tasks, 880000 files

SUMMARY: (of 1 iterations)
   Operation                     Max            Min           Mean        Std Dev
   ---------                     ---            ---           ----        -------
   File creation     :     563808.751     563808.751     563808.751          0.000
   File stat         :   14393911.406   14393911.406   14393911.406          0.000
   File read         :    5126798.789    5126798.789    5126798.789          0.000
   File removal      :     575868.384     575868.384     575868.384          0.000
   Tree creation     :          7.634          7.634          7.634          0.000
   Tree removal      :          1.145          1.145          1.145          0.000

-- finished at 10/17/2017 22:29:28 --
```

Spectrum Scale

Simplification

Spectrum Scale Tuning Simplification

- 4.2.1
  - First introduction of WorkerThreads, eliminated almost 30 Parameters that had to be set on each cluster
  - ESS 5.0 ships scale tuned Profile
- 5.0
  - Elimination of 32 Subblock limit
  - changes to default Filesystem layout and blocksize 4MB
  - Significant reduction in communication parameter
  - Activation on relatime setting and increased default Log size
  - Elimination of 10 Communication Parameter
  - ESS 5.3 ships enhanced tuned profile and udev rules for proper network and device tuning
  - eliminating default separate metadata and data vdisks for GNR setups
- 5.0+
  - Attempt to simplify of NSD Server configuration, no queue, threads per disk, threads per queue, etc
  - framework to provide hints to users, what should be changed e.g. increase pagepool, increase maxfilestocache, other performance related tips

Trademarks and Disclaimers

ZSP03490-USEN-00