

Spectrum Scale Expert Talks

Episode 19 (Version 3):

Spectrum Scale Performance updates

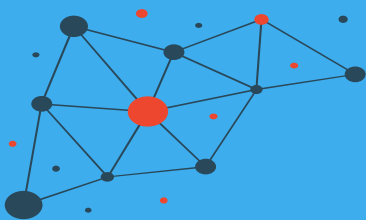


Show notes:

www.spectrumscaleug.org/experttalks

Join our conversation:

www.spectrumscaleug.org/join

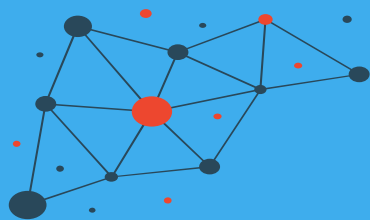


About the user group

- Independent, work with IBM to develop events
- Not a replacement for PMR!
- Email and Slack community
- <https://www.spectrumscaleug.org/join>

#SSUG





We are ...

Current User Group Leads

- Paul Tomlinson (UK)
- Kristy Kallback-Rose (USA)
- Bob Oesterlin (USA)

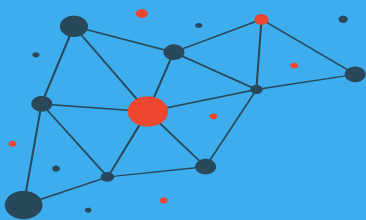
Former User Group Leads

- Simon Thompson (UK)
- Bill Anderson (USA)
- Chris Schlipalius (Australia)

#SSUG

IBM **CHAMPION**

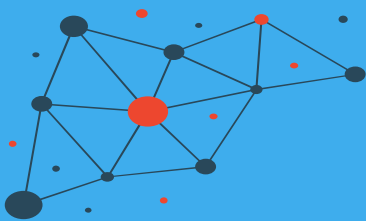




Check <https://www.spectrumscaleug.org/experttalks>
for charts, show notes and upcoming talks

- Past talks:
 - 001: What is new in Spectrum Scale 5.0.5?
 - 002: Best practices for building a stretched cluster
 - 003: Strategy update
 - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
 - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
 - 006: Persistent Storage for Kubernetes and OpenShift environments
 - 007: Manage the lifecycle of your files using the policy engine
 - 008: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
 - 009: Continental: Deep Thought – An AI Project for Autonomous Driving Development
 - 010: Data Accelerator for Analytics and AI (DAAA)
 - 011: What is new in Spectrum Scale 5.1.0?
 - 012: Lenovo - Spectrum Scale and NVMe Storage
 - 013: Event driven data management and security using Spectrum Scale Clustered Watch Folder and File Audit Logging
 - 014: What is new in Spectrum Scale 5.1.1?
 - 015: IBM Spectrum Scale Container Native Storage Access
 - 016: What is new in Spectrum Scale 5.1.2?
 - 017: Multiple Connections over TCP (MCOT)
 - 018: NVIDIA GPU Direct Storage with IBM Spectrum Scale
- This talk
 - 019: Spectrum Scale Performance Improvements





Speakers

- John Lewars
- Jay Vaddi
- Loads of thanks to Olaf Weiser and Pidad D'Souza!!!! (IBM)



Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.

Agenda

- **io500 Related Work**
- **Prefetch Performance Enhancements**
- **TRIM**
- **Inode Allocation Details**

IO500 Related Work

io500 Work and Plan

- The io500 benchmark suite has received an increasing amount of focus in recent years and now provides an important set of performance metrics that the Spectrum Scale Research and Development teams are working on.
- One of the goals of io500 is to measure ‘hard’ workloads to determine the worst possible performance that may be achieved across all possible I/O patterns.
- By improving the performance of the ‘hard’ io500 benchmarks, we expect to improve the performance of challenging modern workloads, following this plan:
 1. Focus on the lowest performing benchmarks, determine bottlenecks, and then apply existing tuning parameters to improve performance.
 2. Develop new tuning parameters/hints that allow us to target focus workloads.
 3. Improve heuristics so that we can automatically adapt to workloads without specific tuning parameters or hints so that future runs of the benchmark are able to achieve optimal performance without explicit hints/tuning.

SC21 io500 Submission on ESS3200 System (1/2) Spectrum Scale

IBM has submitted an SC21 Ten Node io500 submission with a **total score of 68.8**, which placed us at number 28 in the latest published ten node list (SC1 list: <https://io500.org/list/sc21/ten>)

Details of Ten Node io500 submission:

2x ESS 3200 Building Blocks, 2 servers/canisters per BB with 8MB Blocksize File System:

ESS 6.1.1 (RHEL 8.2) + Spectrum Scale upgraded to 5.1.2 on all canisters

Four HDR-IB links per canister

Single socket 48-core processor per canister

24x Samsung NVMe Drives (shared across both canisters in each BB)

10x Lenovo AMD clients:

Spectrum Scale 5.1.2 GA on all clients

RHEL 8.1 w/ 4.18.0-147.el8.x86_64 kernel on all clients

One HDR-Infiniband connection per client

Single socket - AMD EPYC 7302P 16-Core Processor per client

256GB Memory per client

Important mmchconfig Tuning on Clients Disables the Normal/Full Block Prefetch function:

prefetchAggressivenessRead=0

allowFullblockRead=0

SC21 io500 Submission on ESS3200 System (2/2)



Spectrum Scale

Benchmark	Normal Prefetch Disabled (SC21 Sub)	Default Prefetch Enabled (Baseline)	Latest Runs (Jan. 10 2022)
ior-easy-write	106.4	103.6	
mdtest-easy-write	195.6	187.9	
ior-hard-write	4.3	3.2	5.5
mdtest-hard-write	22.3	19.3	13.3
find	1185.2	2469.3	
ior-easy-read	88.1	149.6	149.1
mdtest-easy-stat	272.2	267.2	
ior-hard-read	29.3	1.9	28.2
mdtest-hard-stat	266.9	264.7	
mdtest-easy-delete	113.4	114.2	
mdtest-hard-read	205.4	251.3	
mdtest-hard-delete	20.5	22.3	42.8
BW Score	33.0	17.5	39.7
IOPS Score	143.5	158.9	150.6
Total Score	68.8	52.8	79.0

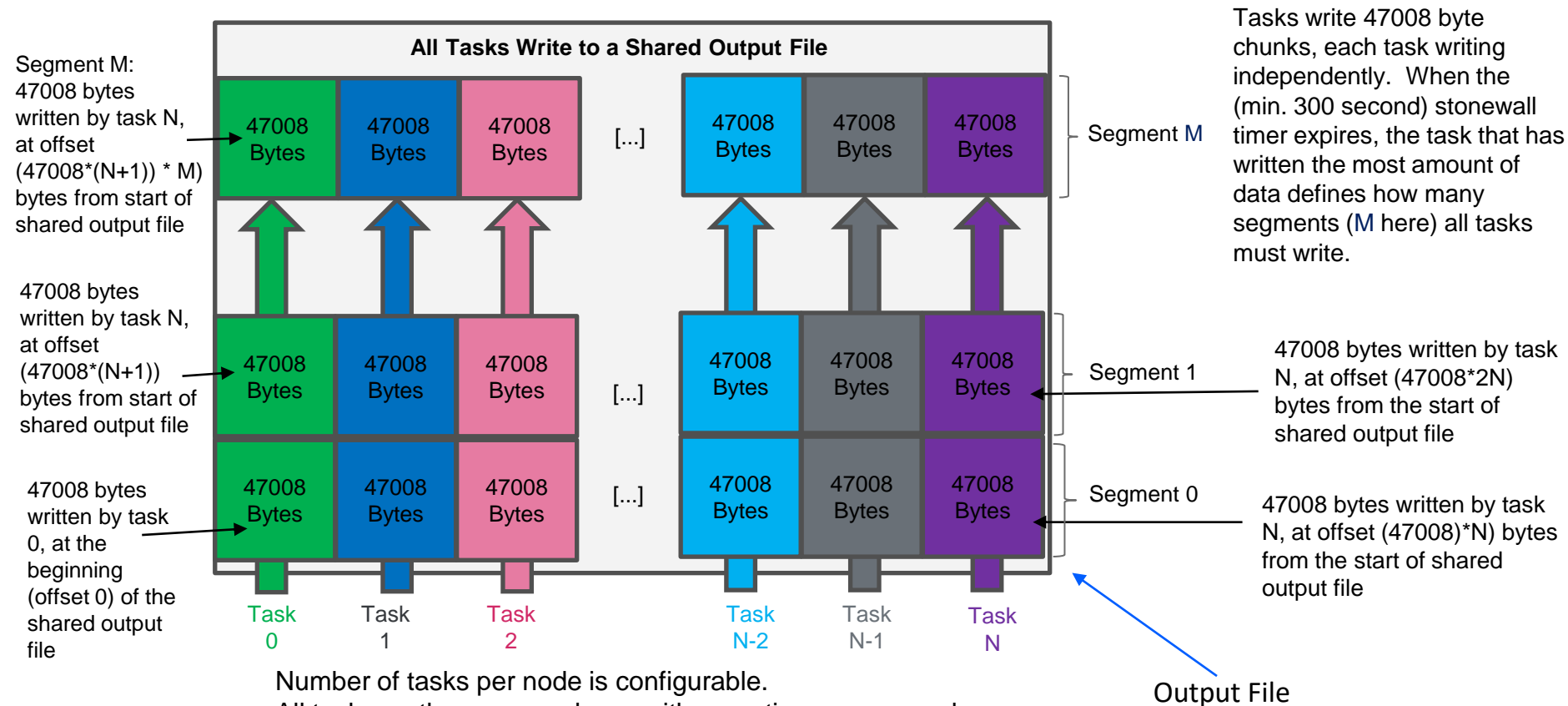
With the default tunings we prefetch too aggressively for optimal ior-hard-read performance

With adjusted tunings used for our io500 submission, ior-hard-read is improved but ior-easy-read performance degrades.

Changes have already been made to io500 benchmarks for ior-hard-write and ior-hard-read, to enable hints that optimize the performance of these benchmarks. The hint for ior-hard-read will give the best performance for this benchmark while the global prefetch settings are enabled.

The next targeted benchmarks are mdtest-hard-write and mdtest-hard-delete,

io500 Benchmark Focus : IOR Hard Write



IOR Hard Write Challenges

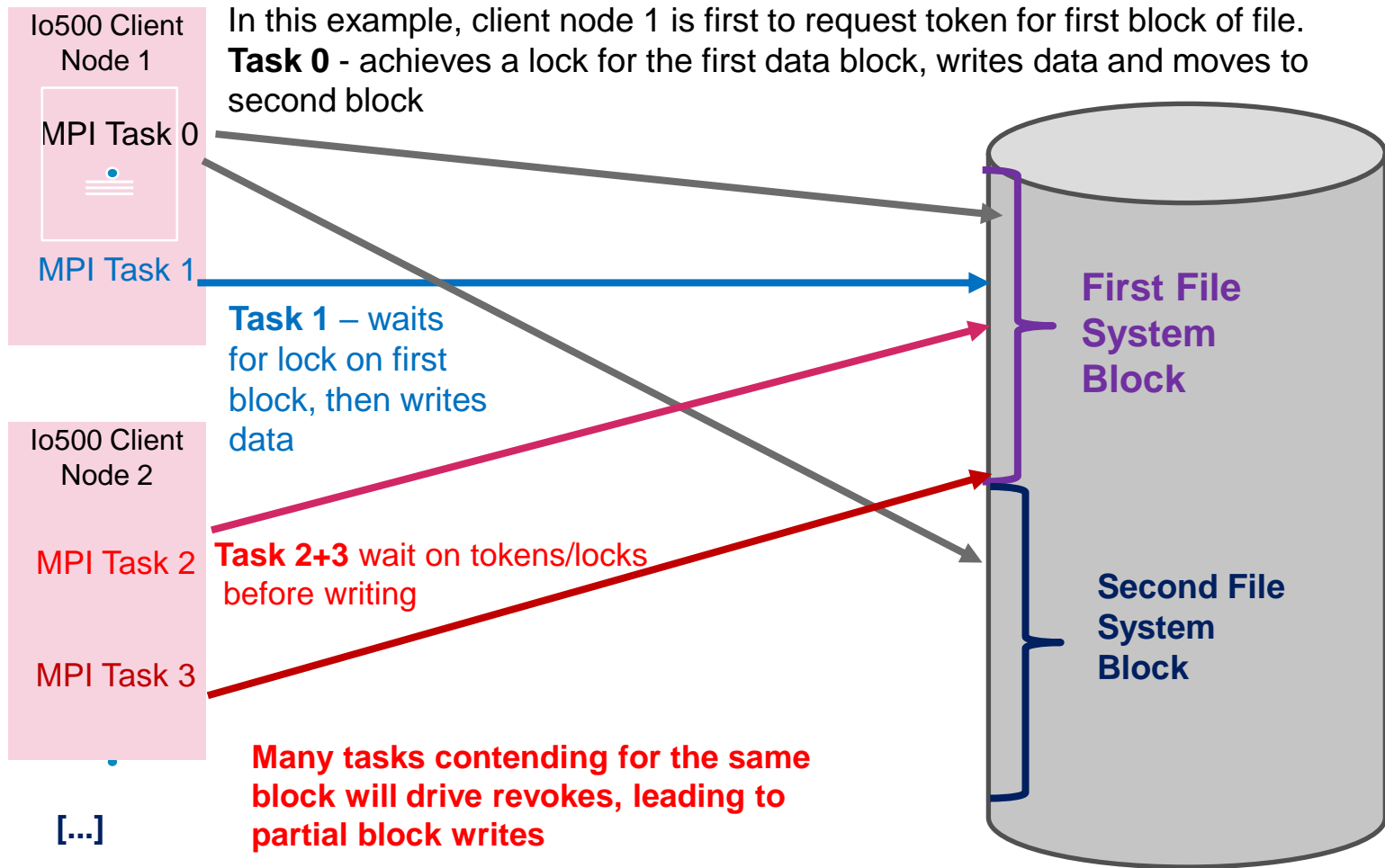
What makes ior hard write so hard?

1. Shared block contention. Tokens needed for writes are granted on a file system block basis (only one node can be writing to the same block at a time) and delays waiting on tokens/locks will slow down the clients due to false sharing.
2. Token contention will lead to token revokes which will drive the flushing of data on nodes that must release tokens, leading to less efficient (smaller than full block) I/Os to the drives, particularly given the small 47008 byte I/O request sizes from the application.
3. For databases we have a good solution - using Direct I/O will eliminate locking in GPFS, but the use of unaligned write request sizes in ior hard write prevents the use of direct I/O.
4. When using MPI-IO the MPI layer can solve these problem by coordinating the I/Os through “intermediate collector nodes” but that limits the total number of clients sending directly to servers (and maximum network throughput) and doesn’t provide a general POSIX solution.

Impact of Token/Lock Contention in ior Hard Write Spectrum Scale

Unless the O_SYNC option is used (currently implicit in Direct I/O), any write call will return after data is written to a buffer in page pool memory.

A sync, buffer clean, or token revoke will flush data to be written to disk.

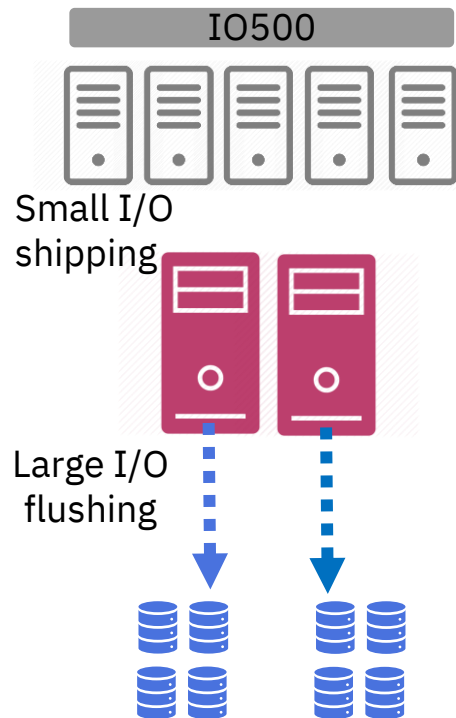


IOR Hard Write Action Plan

- To address the ior hard write challenges, new function was added in Spectrum Scale 5.1.2 which we intend to recommend when 5.1.3 is released

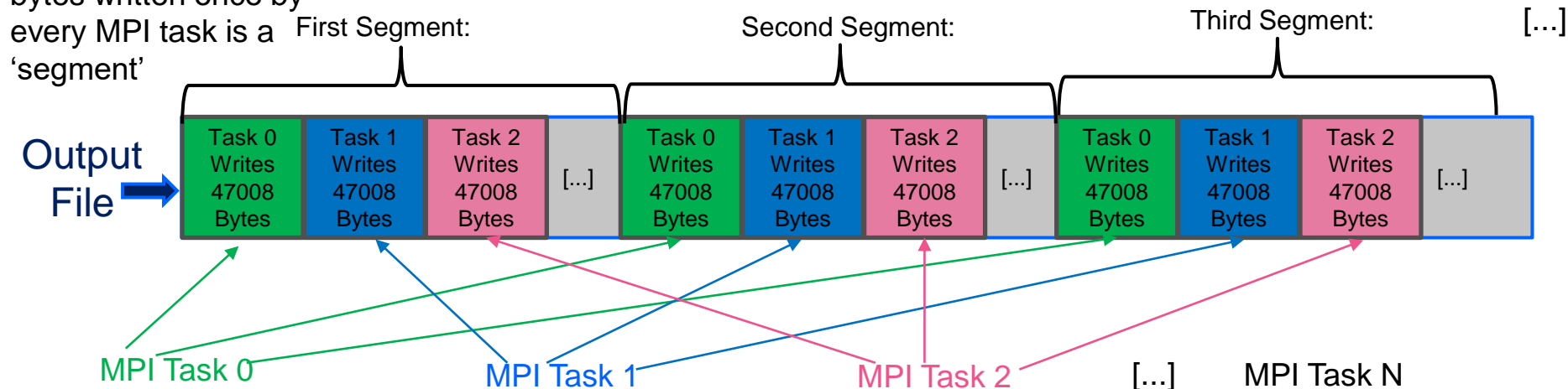
Data Shipping/FineGrainWriteSharing (FGWS) hint (enabled via fcntl call)

- This function is enabled in the ior benchmark via the following commit: <https://github.com/hpc/ior/issues/390>
- Clients ship their updates in a lockless manner to the I/O server nodes coalescing writes (coalescing also optimizes client use of the network)
- Use of Data Shipping requires that every client writes to a unique range of bytes in the shared output file (no two tasks write to the same byte) – clients ship the data to all available servers (this is scalable in that there's no specific manager of the coalescing – all servers participate)
- I/O Servers merge updates from the clients and, a using new token request batching function coming in 5.1.3, acquire appropriate locks
- Servers then write merged buffers via full file system block I/Os, avoiding Read-Modify-Write (RMW) operations when possible



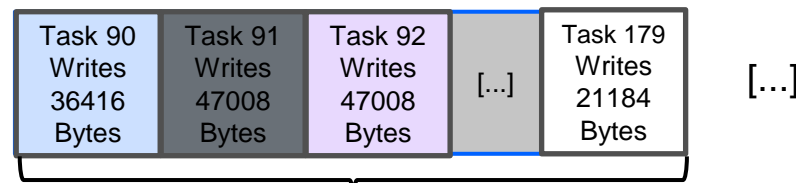
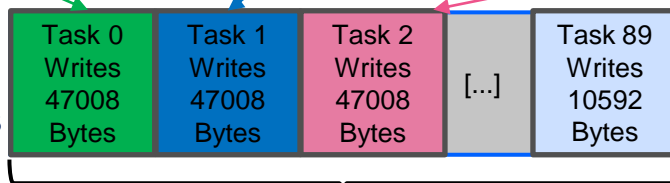
io500 Benchmark Focus : IOR Hard Write Coalescing

A grouping of 47008 bytes written once by every MPI task is a 'segment'



Servers Coalesce Client Writes
(Try to Avoid flushing less than full buffers.)

Example, for case of 4MB file system block size, of how writes are coalesced:



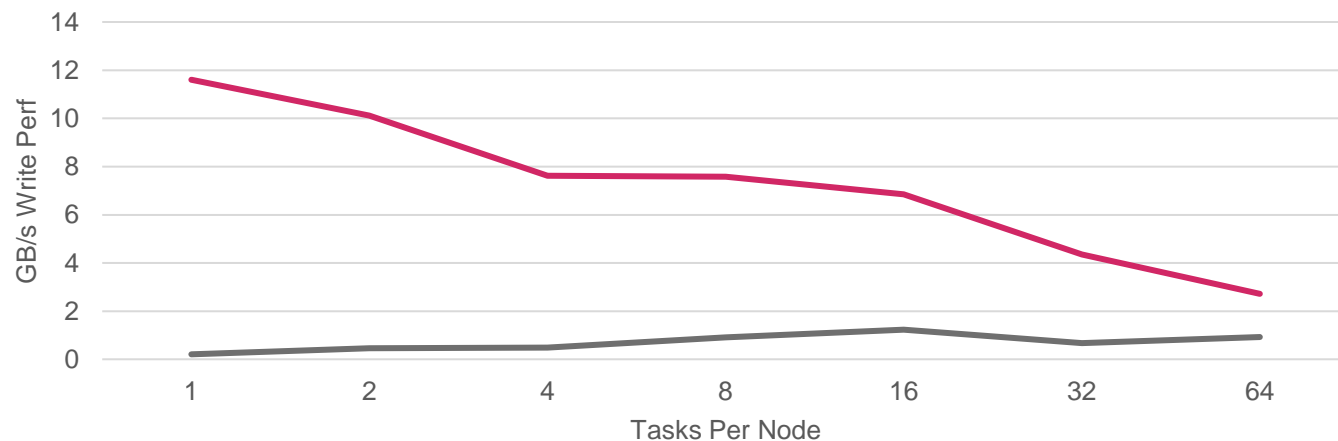
First Buffer on Server (4MB in this example) written by server using full block writes

Second Buffer on Server (4MB in this example) written by server using full block writes

Snapshot of IOR Hard Performance - Presented Week of SC'21

Comparison of IOR Hard Write performance of a recent 5.1.3 sandbox build of FineGrainWriteSharing (FGWS) dataShipping function vs G/A Spectrum Scale 5.1.1 code on a single ESS3000 building block (two 4X EDR links) – connected to single link FDR clients.

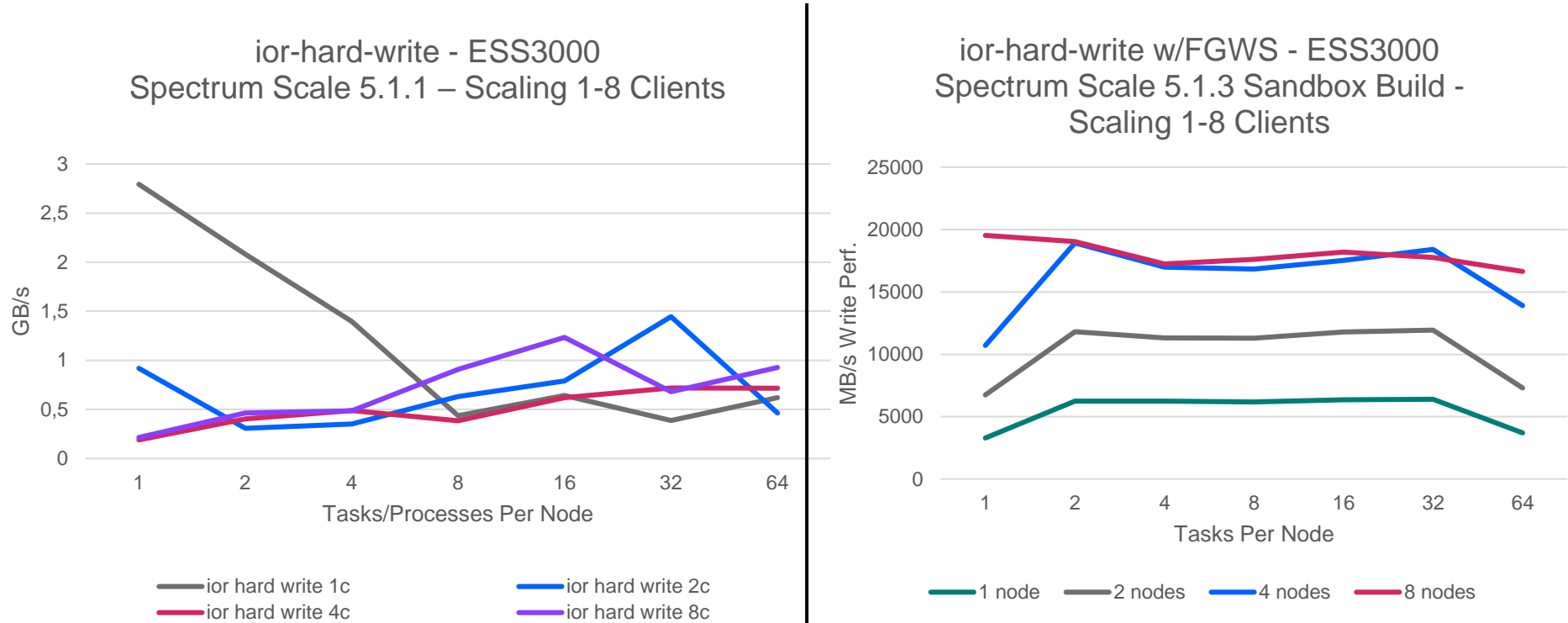
ior-hard-write w/FineGrainWriteSharing - ESS3000
8 Client Scaling-Spectrum Scale 5.1.1 vs 5.3.0 SB Build



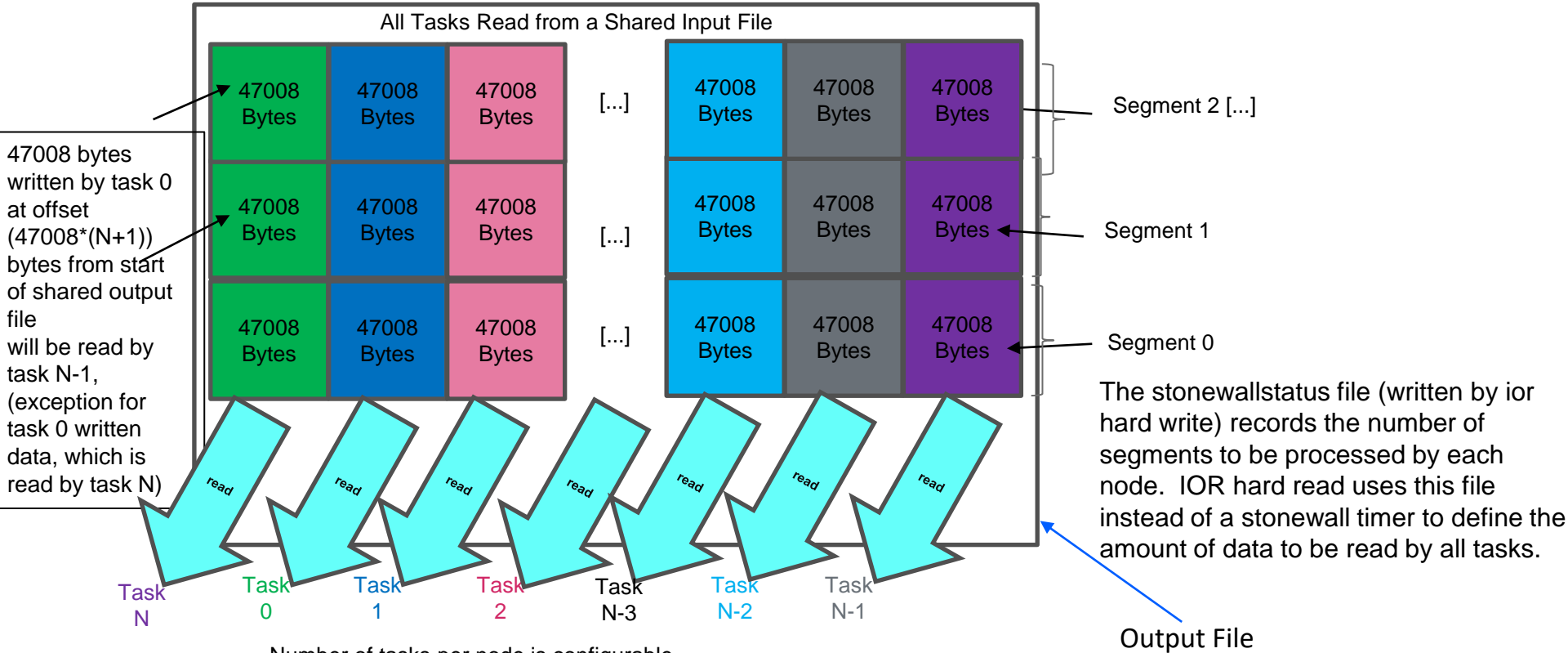
— 8 clients - Spectrum Scale 5.1.3 sb build with FGWS

— 8 nodes - Spectrum Scale 5.1.1

Current Snapshot of IOR Hard Performance (Jan. 19 2022)



io500 Benchmark Focus : IOR Hard Read



Number of tasks per node is configurable.

All tasks on the same node are contiguous (in terms of MPI task ID) or round-robin'ed (round-robin'ed meaning that, with 4 tasks per node and 8 nodes, the first node has tasks 0, 8, 16, and 24, etc.)

IO Hard Read Issues and Action Plan

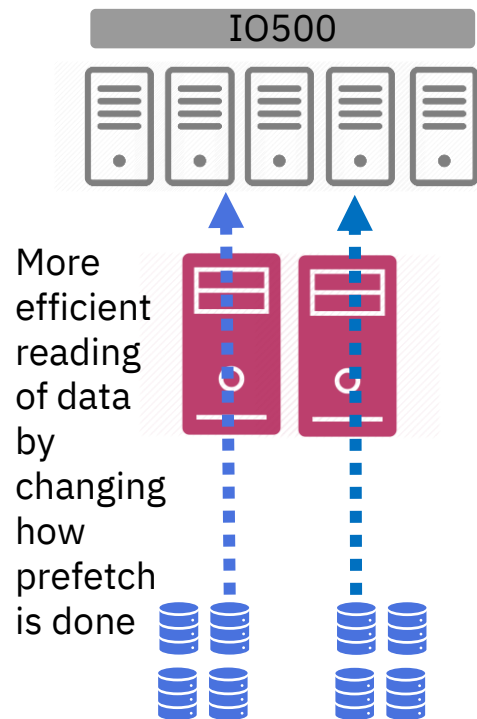
- **What makes ior hard read so hard?**

- Shared file reads currently causes aggressive prefetching leading to client nodes reading data that will not be consumed.
- Small read request size limits efficiencies of network transfers if prefetch isn't done correctly.
- The benchmark ensure that all tasks read data written by another task, which means there are token considerations (can be addressed by MPI hint to release tokens).
- Like ior hard write, a 47008 byte write request size is used, which prevents Direct IO from being used.

- **Action plan:**

- Multiple design changes are being made to prefetch with optimizations controlled via fcntl hint. A fineGrainReadSharing hint has been added to IOR via this commit:

<https://github.com/hpc/ior/issues/390>



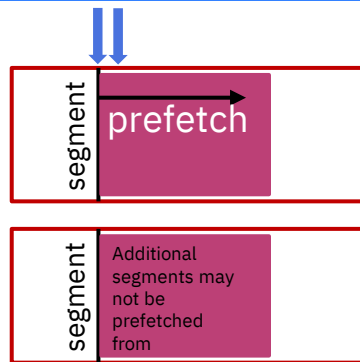


mdtest Hard Deletes - Prefetch Inode Lookups for Deletes (1/2)

- mdtest workload characterization
 - mdtest Hard consists of four phases: 1) create, 2) stat, 3) read, 4) **delete**
 - Both the mdtest hard delete and write.
- Analysis
 - Linux serializes updates on a directory. A single client node can only do one delete at a time for a given directory
 - Task mapping rotates between phases, files created on one node are deleted by another node => each file delete revokes inode lock token and reads inode
- Proposed approach – **Metadata (inode lookup) Prefetch for Deletes**
 - Observation: since files being deleted were all created on a different, but single node, inodes were allocated from the same (or small set of) segment(s) of the inode allocation map
 - Seeing the first few deletes allows to predict next files that will be deleted
 - This approach can be used to prefetch required tokens and read inodes in parallel (overcomes serialization of lookups)
- Implementation complete in 5.1.3 and lab measurements show a 2X improvement!

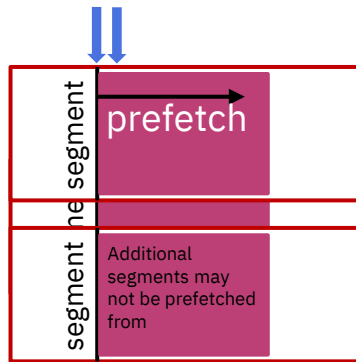
mdtest Hard Deletes - Prefetch Inode Lookups for Deletes (2/2)

Io500 running on node 1



[...]

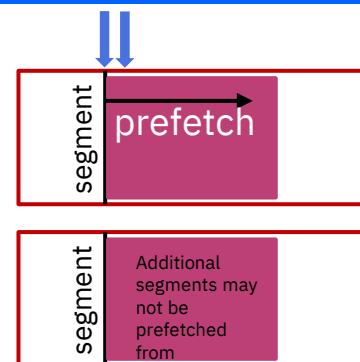
Io500 running on node 2



[...]

[... Additional nodes ...]

Io500 running on node X



[...]

Multiple segments may be dedicated to each node, but, since this benchmark operates on only a single directory and there's a mapping between directories and inode segments used, only one segment should be used at a time during creates (the total number of segments used will depend on how many files are created)

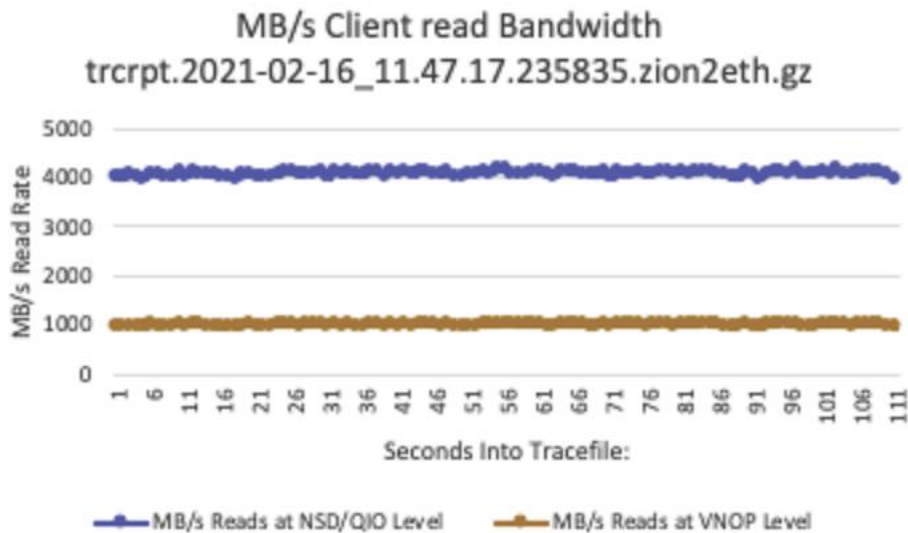
To get the 2X performance improvement we've observed in the lab, two things must be done:

1. There must be sufficient free inode segments so that the (mdtest-write-hard) create phase of the benchmark is able to allocate sufficient dedicated inode segments to all clients involved
2. Designated metanode function must be enabled: **echo 999 | mmchconfig preferDesignatedMnode=yes ***

* This may degrade the mdtest-hard-write benchmark, which we're currently debugging in the lab

Prefetch Related Performance Enhancements

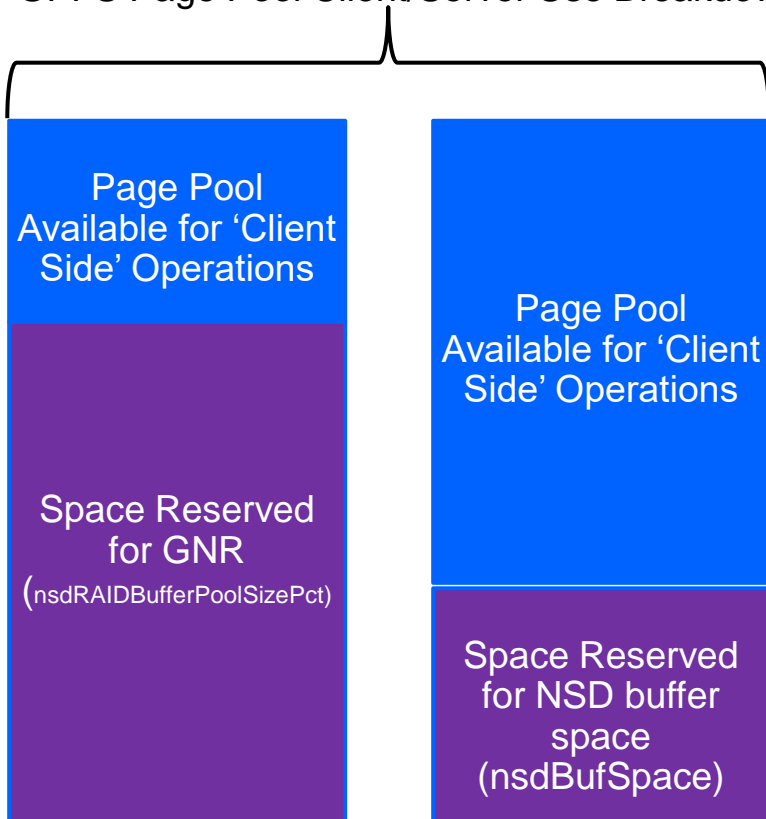
Fixed Percentage of Page Pool Used for Prefetch (5.1.1-2)



- In lab testing we found read variance on machines running a converged workload (a workload in which the clients also function as NSD servers)
- We observed that the rate of prefetching data was much higher than the rate at which the application consumed data, and this discrepancy did not correct itself over time.

Fixed Percentage of Page Pool Used for Prefetch (5.1.1-2)

GPFS Page Pool Client/Server Use Breakdown



- We observed variation with read workloads and that variation could be addressed by disabling read prefetch (**mmchconfig prefetchAggressiveness=0**) or by decreasing the number of prefetch threads (**mmchconfig prefetchThreads=(lower value)**)
- We found root cause: some read operations were done twice (the prefetch buffer was stolen, resulting in the need to repeat the read request)
- The intended design is to limit prefetch buffers to use a portion of the page pool, and, once the limit is hit, prefetch should stop until the buffers used for prefetching have been consumed by applications. This is addressed in 5.1.1-2.

VinfoLock Decoupled From Prefetch Flows (5.1.2)

In Spectrum Scale 5.1.2 prefetch has been made more efficient by eliminating the use of the VinfoLock.

We've seen the VinfoLock cause performance loss for a number of workloads.

Examples are mmap read workloads as described in the March 2020 Performance Update Presentation

Prefetch Related Code Changes in Spectrum Scale

In 5.0.4.3 via APAR IJ22412, we delivered a performance improvement for mmap workloads in which multiple threads/processes read the same file.

This patch set changes the mmapLock to a shared read lock so that all read faults are no longer serialized on the mmapLock and instead a much more efficient read shared lock (MML_READ_SHARED mmap lock) is implemented (serialization is moved to the write path).

Impact: The change dramatically improves performance when multiple threads/processes are reading from the same file.

Next Steps: improve how prefetching works with multiple threads accessing the same file. Currently prefetching can result in contention around VinfoLock.

Locking Issue with mmap read performance Demonstrating improvement with fio:

To demonstrate the best possible performance improvement resulting from the mmap-related changes in 5.0.4.3 with fio, there are a few steps required:

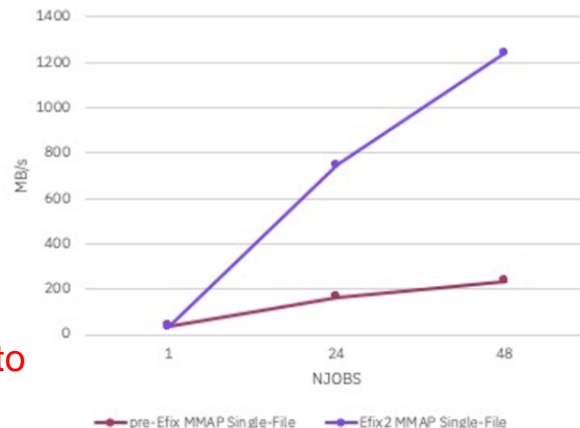
1. Disable pre-fetch:
`# mmchconfig prefetchAggressivenessRead=0 -i`

(This does not require a restart of GPFS to take effect)

2. There is additional locking associated with the mapping of the address space. It's best to map a single region once and then operate on it (via read/write operations). To minimize the overhead of gpfs_mmap calls in fio, set a large blocksize parameter.

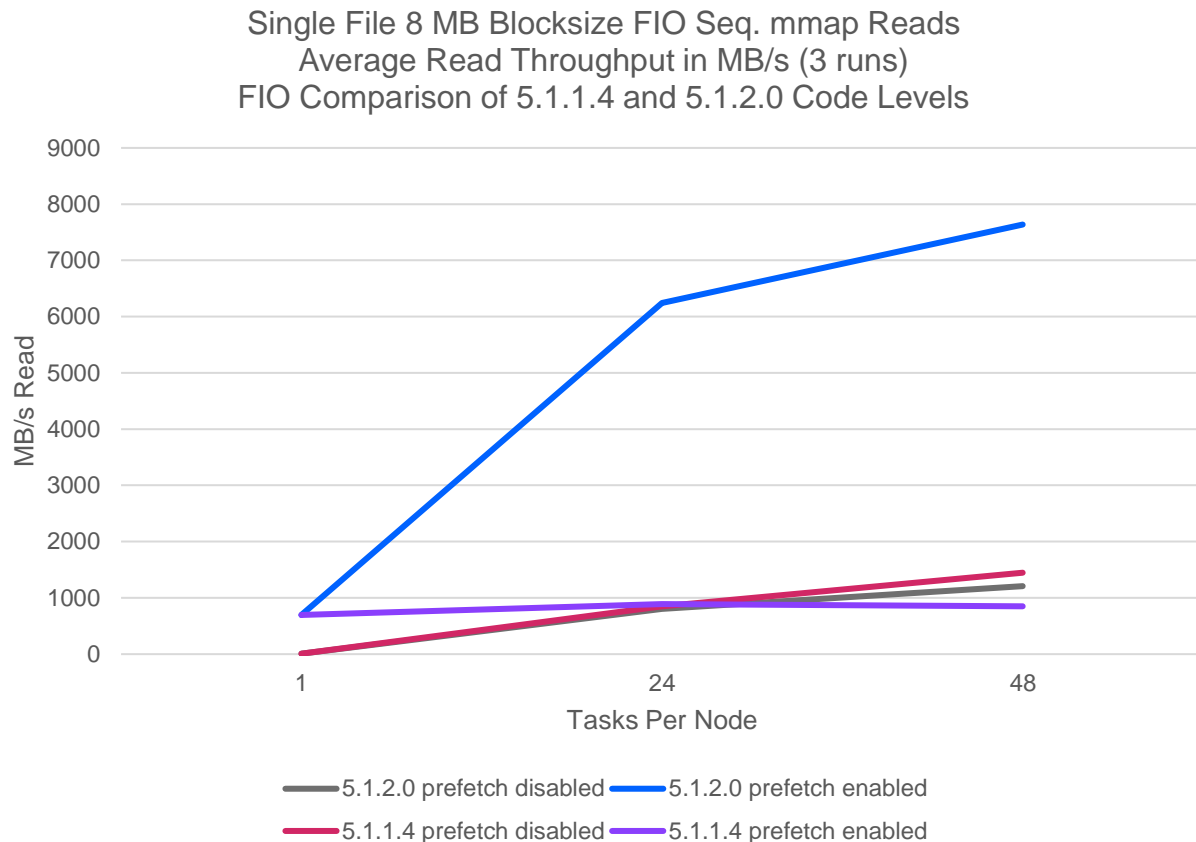
March 2020
Topic Related to
Prefetch

Single File FIO 8MB FIO blksize - Seq. Reads -
Average Read. BW in MB/s (average of 3 runs)
mmap FIO tests - comparing Baseline vs mmap
Locking Fix V2



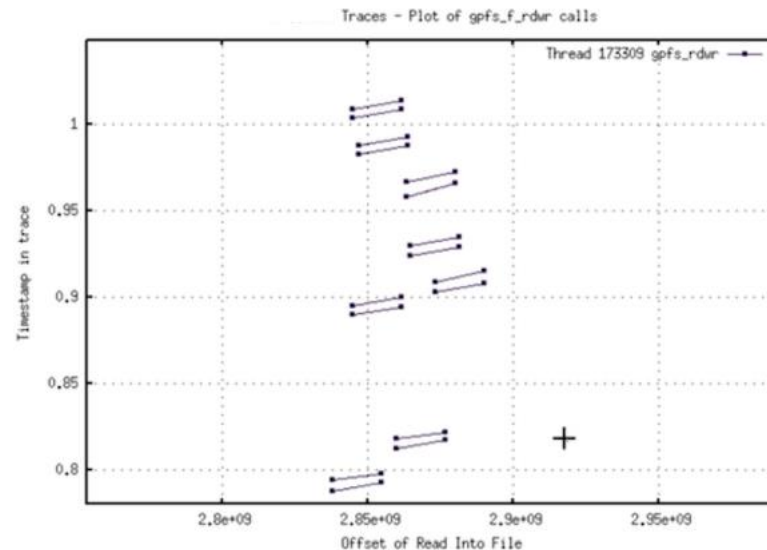
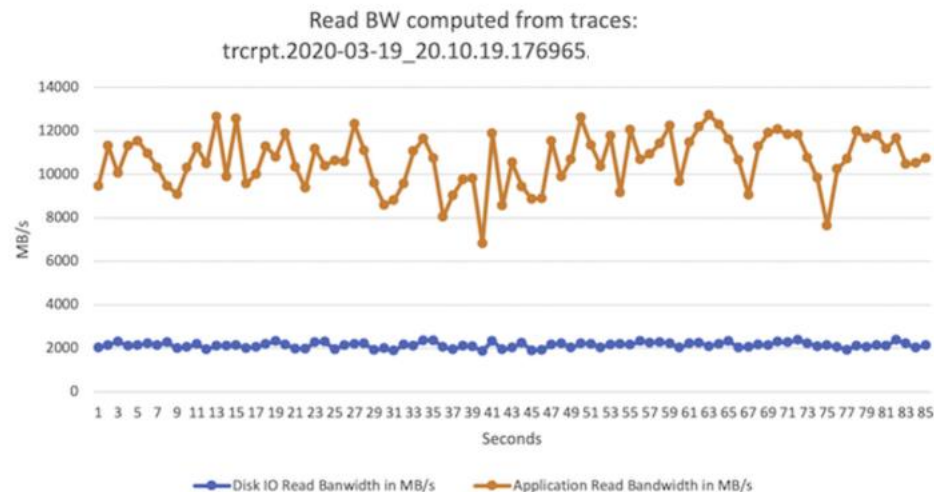
VinfoLock Decoupled From Prefetch Flows (5.1.2)

Now enabling prefetch for mmap allows multiple threads reading a shared file to leverage prefetch, without seeing performance degradation due to VinfoLock contention in Spectrum Scale.



Prefetch Activity Causing Application Interference (5.1.3 work item)

- A customer opened a case to investigate why one of their jobs was impacting other workloads on the system, and we believe that impact was related to network interference caused by the job.
- Traces revealed that there were windows of time in which the application was prefetching much more data than it consumed, and the access pattern was able to drive new prefetching for long enough to cause enough network activity to create network interference (so short spurts of sequential access continued to lead to prefetch using network bandwidth)



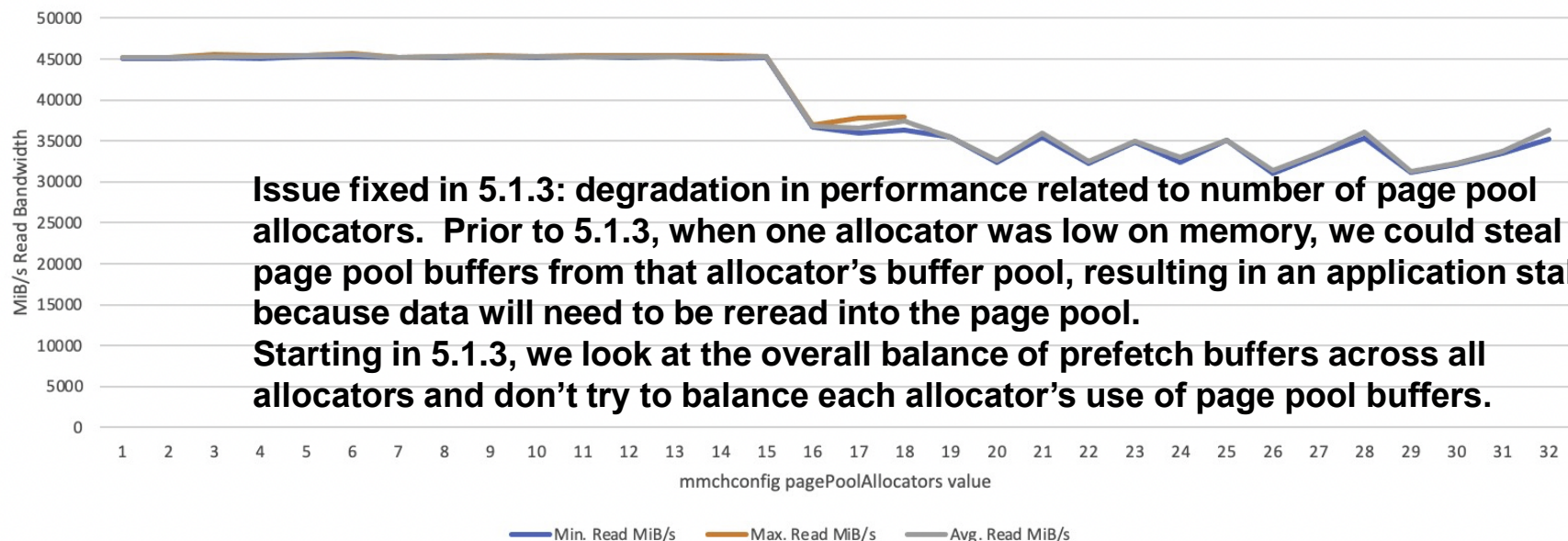
Prefetch Activity Causing Application Interference (5.1.3 work item)

- The 'normal' full block prefetching flows are tied to full block reads so increasing the block size of a file system will increase the rate at which prefetch may potentially occur
- We looked at the efficiency of prefetch in terms of how often a given buffer is consumed by the application before it is freed from the page pool, and we found that, for a customer application we focused on, the efficiency of prefetch decreased as we increased the block size
- In 5.1.3 the mmchconfig option **prefetchLargeBlockThreshold** allows for less aggressive prefetching (the rate at which prefetch ramps up is decreased).
- Setting **prefetchLargeBlockThreshold** equal or less than the size of the block size for a given file system will enable this feature for the file system(s) (for example, to enable this less aggressive prefetching flow for 4, 8, and 16 MB file systems set (prefetchLargeBlockThreshold=4194304))
- Though disabled by default, our plan is to work with customers to enable the **prefetchLargeBlockThreshold** option to address issues with prefetch efficiency, and to investigate additional heuristics (including exploring machine learning algorithms to improve prefetch)

Prefetch Improvements – Potential Variation Fixed in 5.1.3

We found another example of workloads that were showing variance that could be tuned away by reducing the mmchconfig option prefetchThreads and fixed this problem in Spectrum Scale 5.1.3.

IOR Read Performance w/16MB block size Varying pagePoolAllocators config option
(Runs on c202f06 ESS3K and 15 c202f08nXX x86 FDR clients)



Trim

Agenda

- High-level working of Solid-State Drives
- Resulting Issues - Performance Degradation Over Time and Wear of Flash Cells
- Trim and its Benefits
- Performance Measurements from IBM Lab

Firstly a few points to help understand TRIM better

- The Solid-State Drives (SSDs) store and manage data differently compared to traditional hard drives.
- The smallest unit of an SSD is a page (composed of several memory cells), and several pages form a block.
- Data can be read/written at the page level, but the deletion can only be done at block level.
- Unlike traditional hard drives, the data in NAND SSD can't be directly overwritten. Data can only be written to new or erased pages.
- An SSD knows data is invalid only when new data is written to that location.

1

1	2	3
4	5	6
7	8	9

a

b

2

1	2	3
4	5	6
7	8	9

a

1*	2*	3*

b

3

1	2	3
4	5	6
7	8	9

1*	2*	3*
4	5	6
7	8	9

4

1*	2*	3*
4	5	6
7	8	9



Valid data



Invalid data

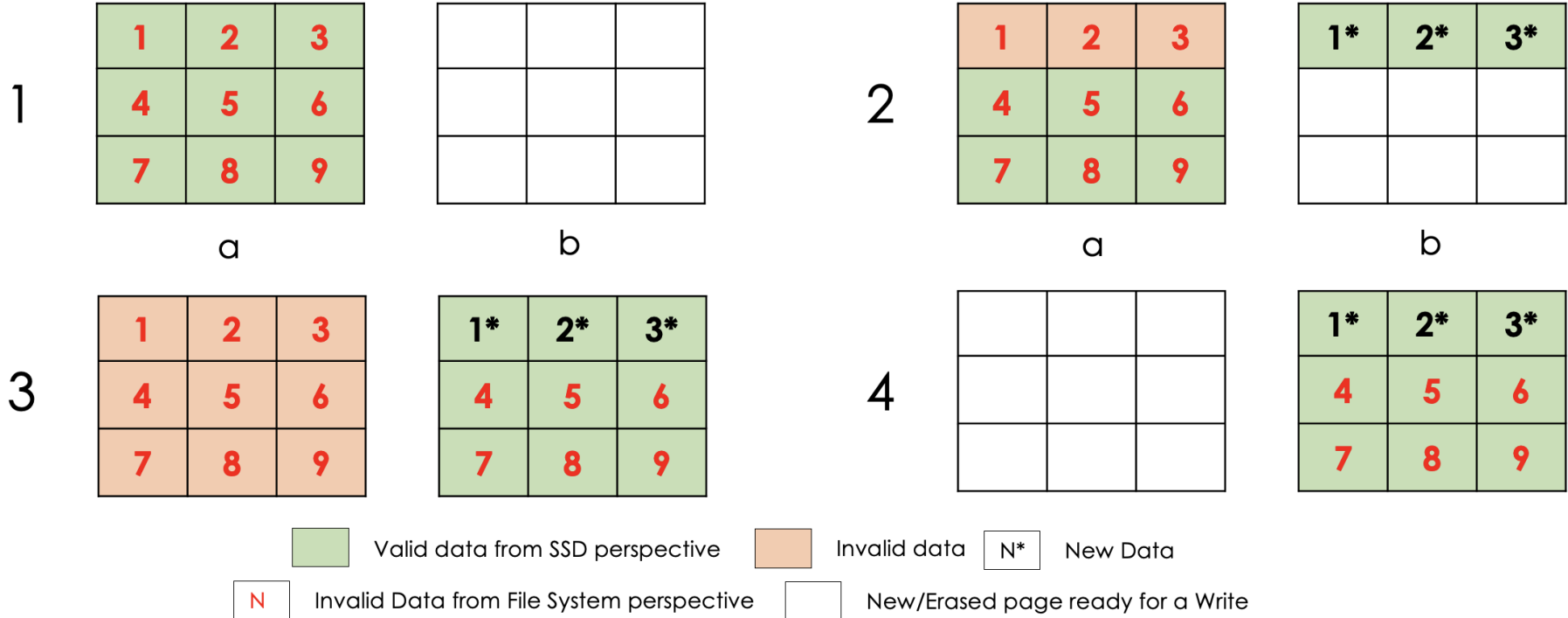


N* New Data



New/Erased page ready for a Write

- The storage media generally do not know which sectors/pages are in use, and which contain invalid data. When a file is deleted at the file system layer, the file system structures are updated, but the storage media is unaware the blocks have become available.



- When an SSD fills up (file system may not to be full), there are fewer fresh blocks to move the data into, and it gets to a point where the SSD slows down as it waits for new blocks to write data, thus impacting the performance.
- The flash memory cells have a finite working life and can fail after several thousand Program/Erase (P/E) cycles. The old/obsolete data, until over-written, can be on the SSD and get moved around as part of Garbage Collection (GC), thus increasing the write-amplification and wear of the flash cells.
- Trim command allows the software to inform the SSD which blocks of data are not in use and can be erased internally.
- Garbage Collection will erase the blocks and make them available in time for new data writes, and thus helping SSD perform consistently and better than the performance without Trim.
- Trim also helps with decreasing the write amplification, as the SSDs can reclaim the invalid/deleted pages and doesn't have to copy them unnecessarily during block erasure.
- `mmreclaimspace` is the Spectrum Scale command used to run Trim. The command is supported on Traditional Scale and Spectrum Scale RAID/GNR systems (Elastic Storage Systems, Erasure Code Edition).
- In case of Spectrum Scale RAID systems, Scale issues RPC to NSD/vdisk server, then GNR 1) performs clean up operations in GNR layer like purge buffer data and discard fast write log records for the vtrack and mark the vtrack unused (i.e. trim to vdisk), and 2) sends trim commands to the physical devices to reclaim space (i.e. trim to physical device).
Example Command: `mmreclaimspace foofs --reclaim-threshold 0`
- [mmreclaimspace in KC](#) has more details on the command and how to use it.

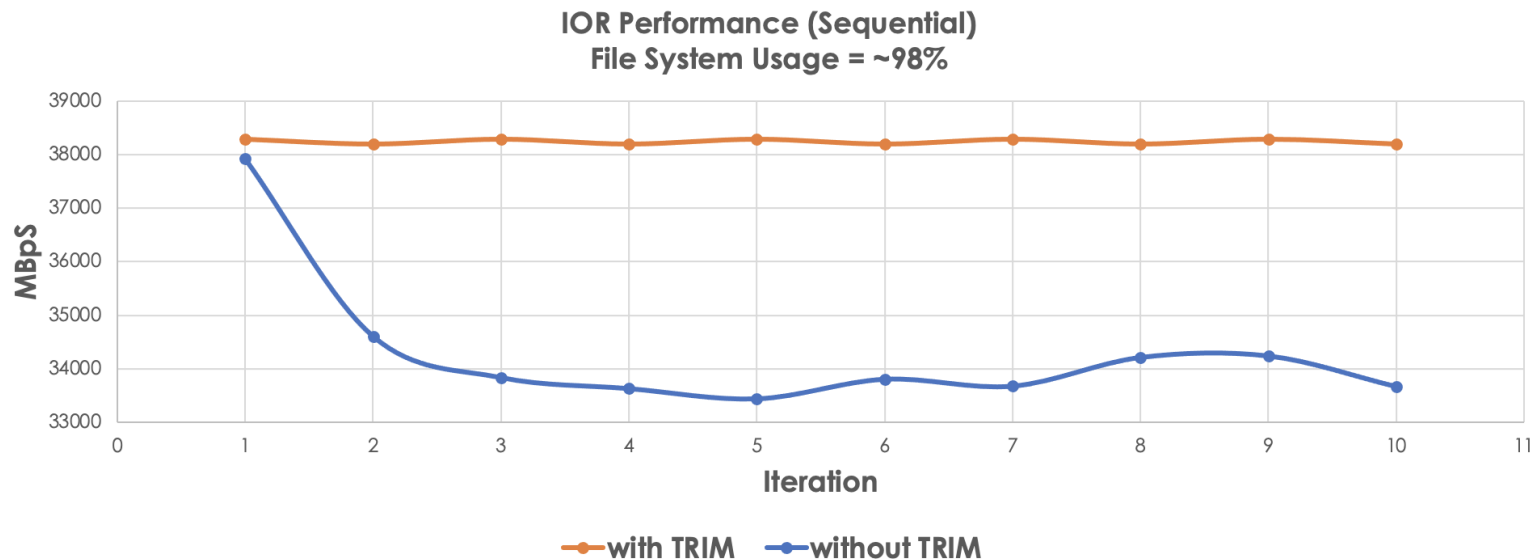
Please Note

- ❖ For the tests in this presentation,
 - A fully populated (24 x 3.84T NVMe) ESS 3000 was used.
 - 100% of drive capacity was used to create GNR Vdisks.
 - File system block size was 4M.

- ❖ The performance numbers shown in this presentation were produced in IBM lab under ideal conditions and are to help understand how Trim works. The actual performance could vary depending on several factors, like numbers of drives, drive type and capacity, server capability, etc.

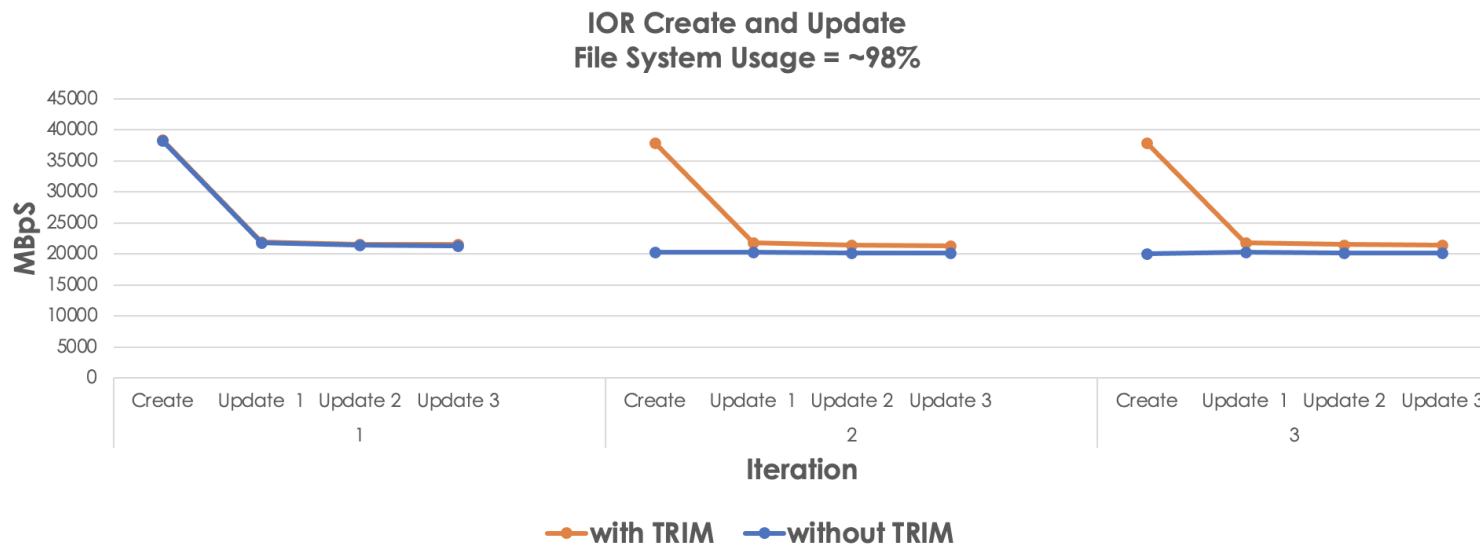
Create, Delete (and Reclaim)

0. Start with fresh file system.
1. Fill up file system (IOR Sequential - 24 x 2500GiB files).
2. Delete files and wait for mmdf to report 100% free.
3. Run "mmreclaimspace fs3k1 --reclaim-threshold 0". Wait for 10 mins for the drives to complete the async discard activity. (skip for without TRIM test)
4. Repeat 1, 2 & 3.

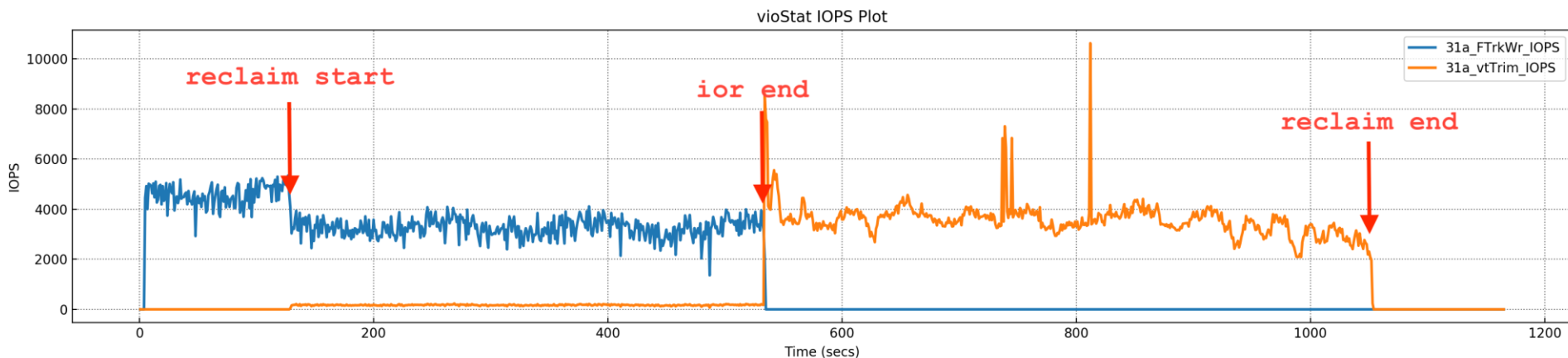
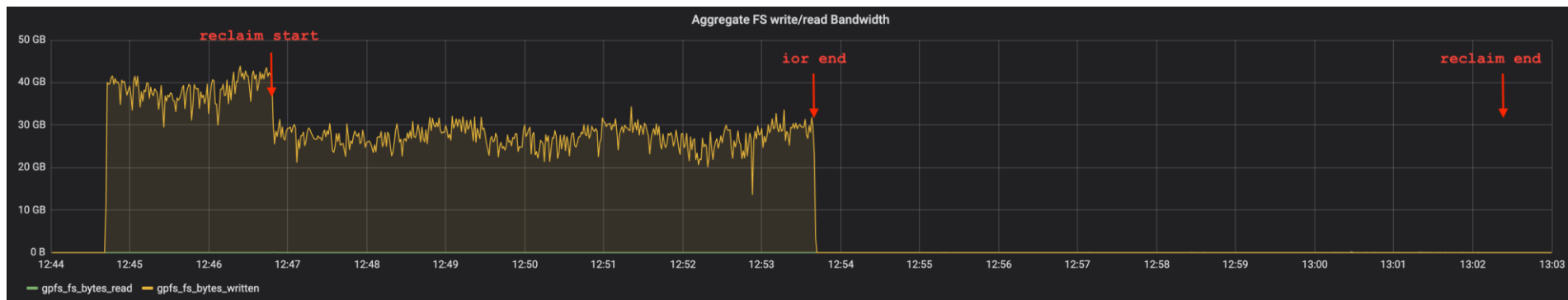


Create, Update, Delete (and Reclaim)

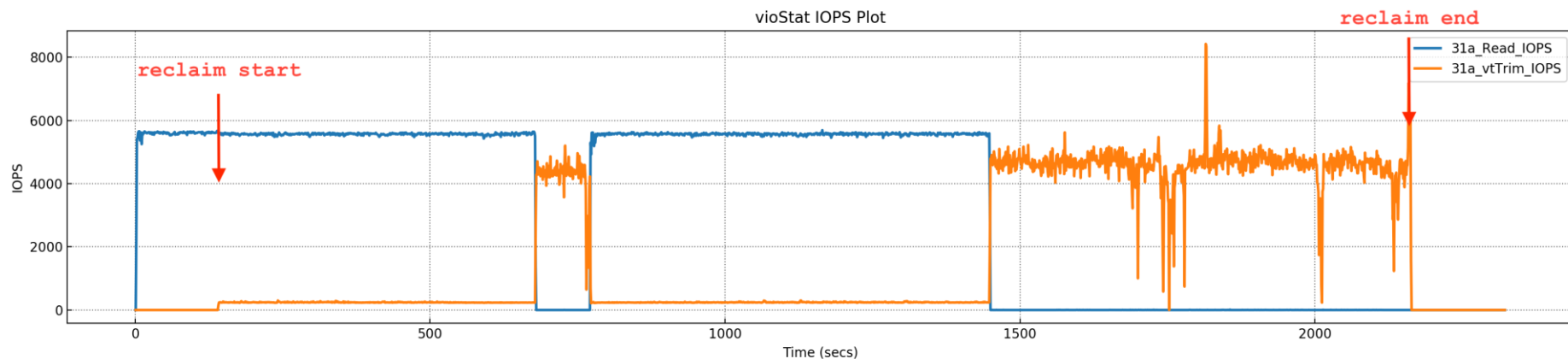
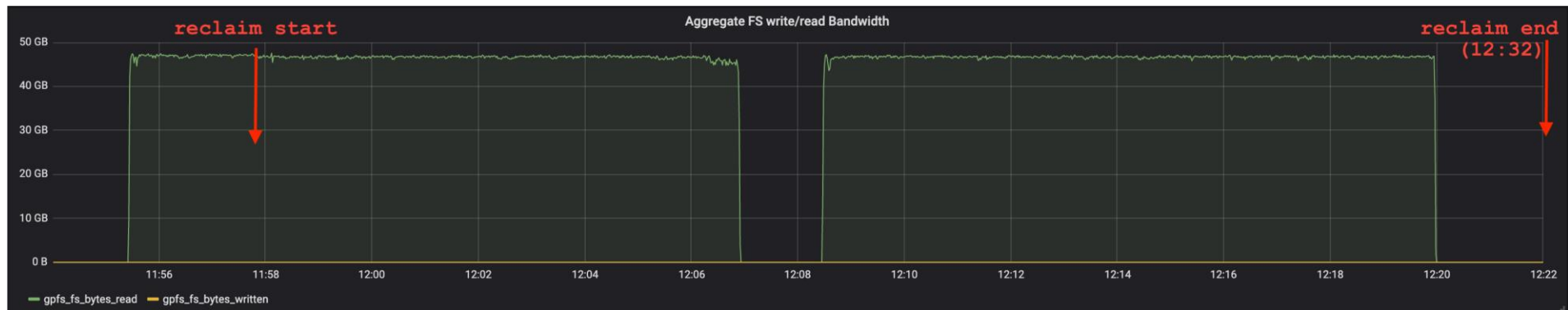
0. Start with fresh file system.
1. Fill up file system (IOR Sequential - 24 x 2500GiB files).
2. Re-write (IOR Random) all files without stonewalling. Repeat 3 times.
3. Delete files and wait for mmdf to report 100% free.
4. Run "mmreclaimspace fs3k1 --reclaim-threshold 0". Wait for 10 mins for the drives to complete the async discard activity. (skip for without TRIM test)
5. Repeat 1, 2, 3 & 4.



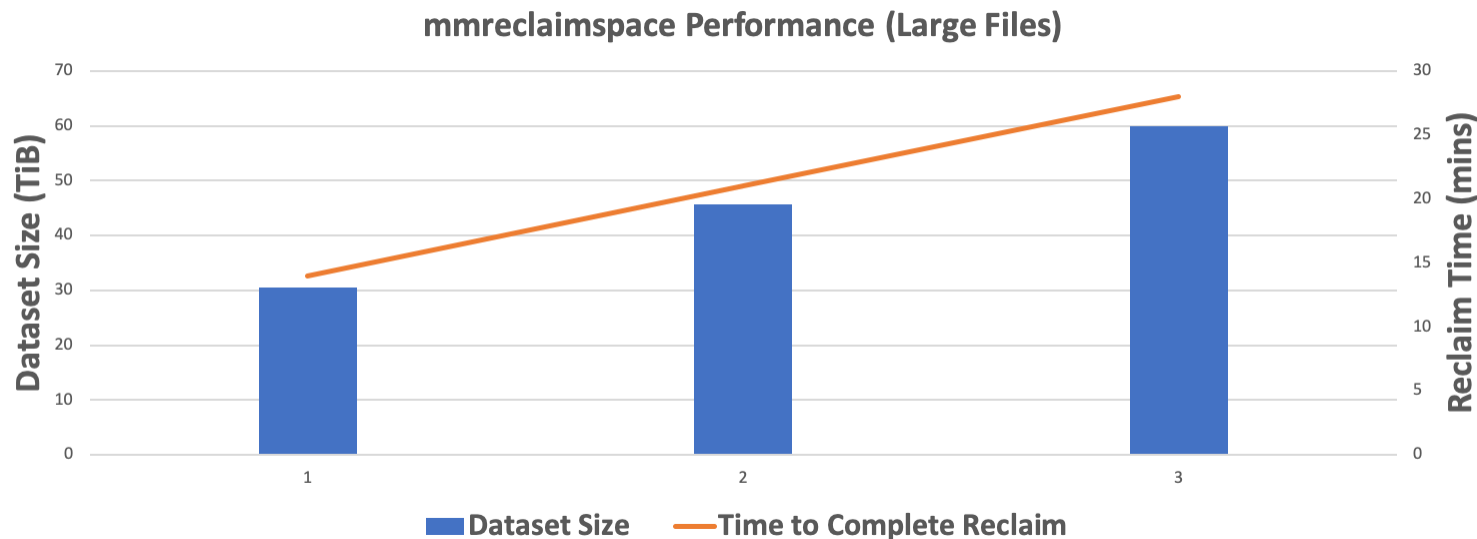
Write (Update Existing Files) Performance during Reclaim



Read Performance during Reclaim

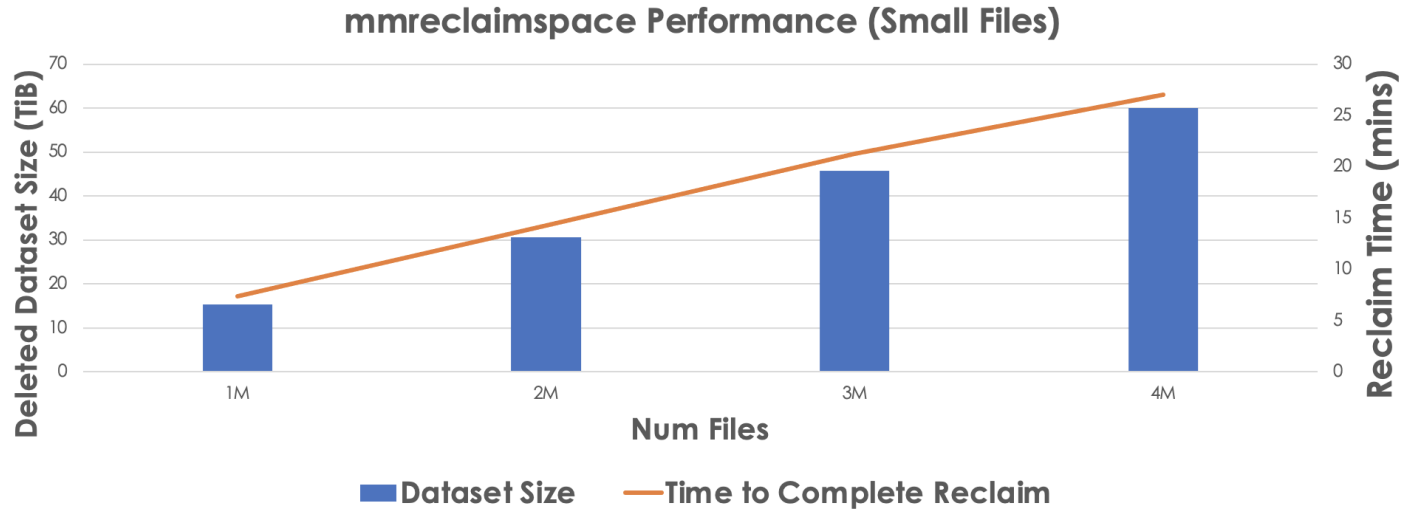


Reclaim Performance – Large Files



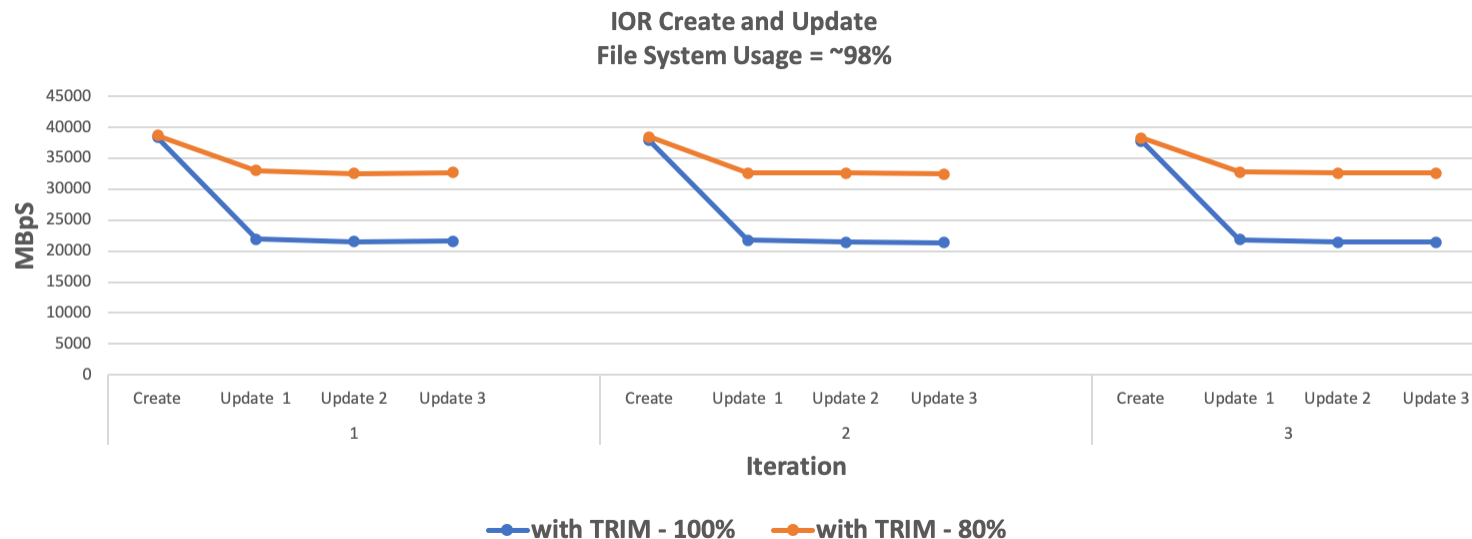
- Assuming the system is idle, the time to complete reclaim is expected to increase linearly with increase in space to be reclaimed. For example – it's expected to take ~56 mins to reclaim 120TiB space on a fully populated ESS 3000 building block.
- Time to complete reclaim also depends on number of drives doing Trim. For example – on a system with 12 drives and 60T of reclaimable space, it would take ~56 mins vs ~14 mins on a system with 48 drives and 60T of reclaimable space.

Reclaim Performance – Small Files



- The time to reclaim is dependent on the size of free space to reclaim. The number of files had very minimal (or no) impact.

Over-Provisioning

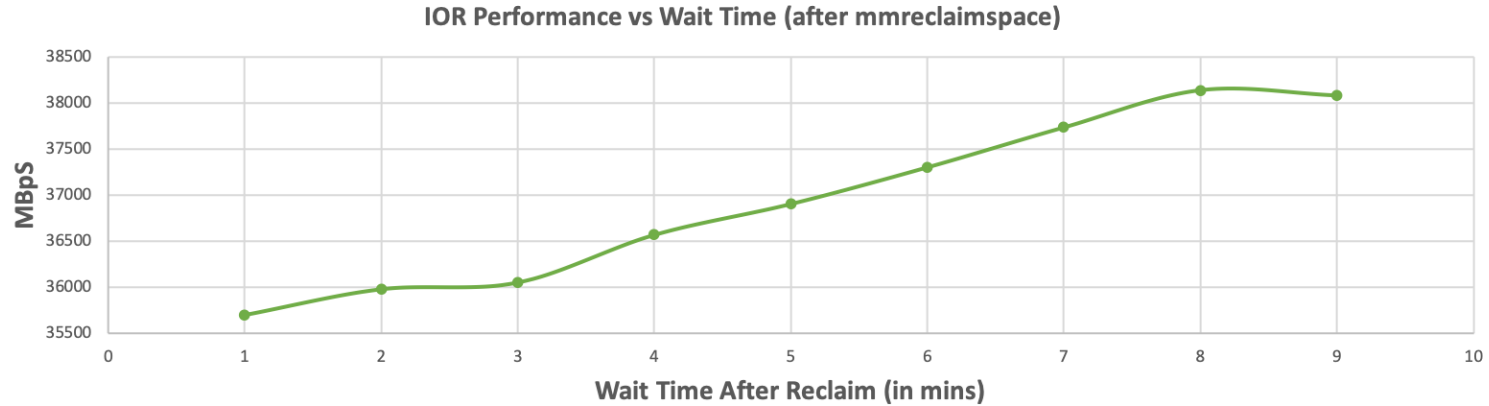


Blue Curve: Using 100% of available space for file system.

Orange Curve: Using 80% of available space for file system.

- The create is after a mmreclaimspace and the performance is equivalent to FOB performance. But the update performance in the following iterations is impacted by GC. The orange curve, which is using only 80% of available raw capacity, shows better update performance because of extra 20% over-provisioning.

The Actual Reclaim Happens in the Background



- Trim is asynchronous and the background activity, after mmreclaimspace completes, impacts performance.
- On ESS3000 with 3.84T drives, the performance was fully restored after 8mins.
- The wait time could vary on systems with different drive types and capacities.

Thanks!

Check out the FAQ!

<https://www.ibm.com/support/knowledgecenter/en/STXKQY/gpfsclustersfaq.html>

<https://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.pdf?view=kc>

<https://www.ibm.com/support/knowledgecenter/SSYSP8/gnrfaq.html>

HTML or PDF

Spectrum Scale version
compatibility with OS or
kernels

Updated regularly!



Log your RFE!

https://www.ibm.com/developerworks/rfe/execute?use_case=productsList

- [Spectrum Scale \(formerly known as GPFS\) - Private RFEs](#)
- [Spectrum Scale \(formerly known as GPFS\) - Public RFEs](#)

contact

Filter the page content by brand and product

Servers and Systems So... ▼

Spectrum Scale (formerly known as GPFS) - Pu... ▼



[Hot](#)

[Top](#)

[New](#)

Search



35
votes

eliminate lack of I/O on mmdelsnapshot start

When deletion of bunch of snapshots starts we a lack of I/O for about three minutes. NFS Clients see a huge delay of I/O. Related applications hanging for this time and user connections and run into t...

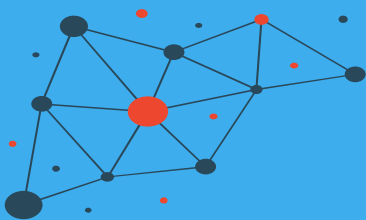
Under Consideration

21
votes

start services after gpfs (filesystems) is ready using systemd

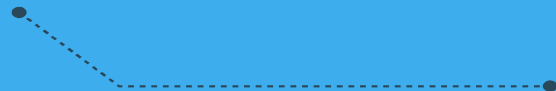
the current systemd units gpfs.service and <fs-mountpoint>.mount units can't be used to depend on (After/Required/.. systemd attributes) for other services. GPFS service is reporting itself as success...

Under Consideration



Check <https://www.spectrumscaleug.org/experttalks>
for charts, show notes and upcoming talks

- Past talks:
 - 001: What is new in Spectrum Scale 5.1.2?
 - 002: Best practices for building a stretched cluster
 - 003: Strategy update
 - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
 - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
 - 006: Persistent Storage for Kubernetes and OpenShift environments
 - 007: Manage the lifecycle of your files using the policy engine
 - 008: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
 - 009: Continental: Deep Thought – An AI Project for Autonomous Driving Development
 - 010: Data Accelerator for Analytics and AI (DAAA)
 - 011: What is new in Spectrum Scale 5.1.0?
 - 012: Lenovo - Spectrum Scale and NVMe Storage
 - 013: Event driven data management and security using Spectrum Scale Clustered Watch Folder and File Audit Logging
- Today:
 - May 19: What is new in Spectrum Scale 5.1.2?



Thank you!

Please help us to improve Spectrum Scale with your feedback

- If you get a survey in email or a popup from the GUI, please respond

- We read every single reply



Provide Feedback



Tell IBM What You Think

Let us know what you think about IBM Spectrum Scale. It takes only a couple of minutes for you to help us improve our service. [IBM Privacy Policy](#)

Not Now

 Provide Feedback



Spectrum Scale User Group

The Spectrum Scale (GPFS) User Group is free to join and open to all using, interested in using or integrating IBM Spectrum Scale.

The format of the group is as a web community with events held during the year, hosted by our members or by IBM.

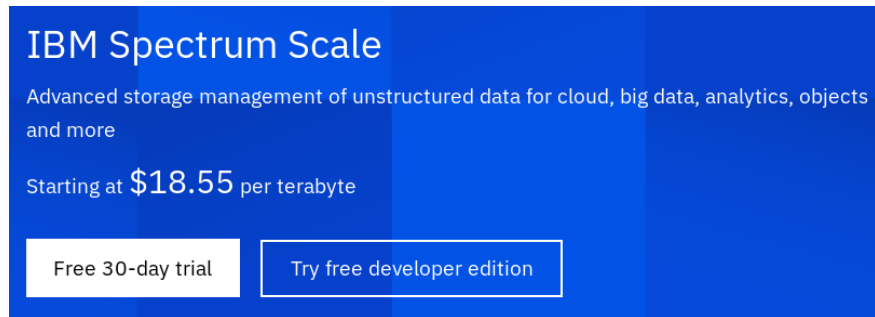
See our web page for upcoming events and presentations of past events. Join our conversation via mail and Slack.

www.spectrumscaleug.org

Spectrum Scale Developer Edition!

Fully functional!

- Based on first PTF of a release
- Derived from **Data Management Edition (DME)**
- Limited to 12 TBs:
enough for a small test cluster
- Available from the Scale "try and buy" page on ibm.com



IBM Spectrum Scale

Advanced storage management of unstructured data for cloud, big data, analytics, objects and more

Starting at **\$18.55** per terabyte

Free 30-day trial Try free developer edition

Free for non-production use, e.g. test, learning, upgrade prep...

- If you have to ask, it's probably not permitted

Not formally supported

Spectrum Scale Early Programs

Types of Programs:

Alpha

Influence the development of new technology by gaining before market access to product code. Alpha programs are typically confidential and the first opportunity for you to interact with a feature or function.

Beta

Try out a new offering with the team who owns the product and influence its usability and design. A Beta program gives you the ability to evaluate and provide feedback on IBM products before the products general availability. Beta programs are typically confidential and run prior to GA.

Early Support Program (ESP)

Be one of the few selected participants to validate new Software or Hardware and potentially give your enterprise an edge over the competition. The IBM early support programs give you and IBM the opportunity to develop, evaluate, and gain experience with a product or a set of products in your enterprise environment.



Customer Success

Talk to your IBM contact or Partner to be nominated!

- ☐ Evaluate new IBM HW or SW in your environment.
- ☐ Validate procedures and interoperability with other products in your enterprise.
- ☐ Opportunity to Influence Product Design
- ☐ Early Enablement and education
- ☐ Strengthen Partnership with IBM

Spectrum Scale on GitHub!

<https://github.com/IBM/SpectrumScaleTools>

- IBM Spectrum Scale Bridge for Grafana
- IBM Spectrum Scale cloud install
- IBM Spectrum Scale Container Storage Interface driver
- IBM Spectrum Scale install infra
- IBM Spectrum Scale Security Posture
- Oracle Cloud Infrastructure IBM Spectrum Scale terraform template
- SpectrumScale_ECE_CAPACITY_ESTIMATOR
- SpectrumScale_ECE_OS_OVERVIEW
- SpectrumScale_ECE_OS_READINESS
- SpectrumScale_ECE_STORAGE_READINESS
- SpectrumScale_ECE_tuned_profile
- SpectrumScale_NETWORK_READINESS

Find open source tools that are related with IBM Spectrum Scale.

Unless stated otherwise, the tools compiled in this list come with no warranty of any kind from IBM.

Check out the FAQ!

<https://www.ibm.com/support/knowledgecenter/en/STXKQY/gpfsclustersfaq.html>

<https://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.pdf?view=kc>

<https://www.ibm.com/support/knowledgecenter/SSYSP8/gnrfaq.html>

HTML or PDF

Spectrum Scale version compatibility with OS or kernels

Updated regularly!

