The Effects of Fragmentation and Capacity on Lustre File System Performance

John Kaitschuck, Senior Staff Engineer/Technologist CSSG, June, 2017

Seagate LUG 2017 Presentation

AGENDA

- History/Background
- Focus
- Test Environment
- Nomenclature & Methodology
- PaF Choices
- Series 1 & Series 2 Tests & Test Results
- Lab vs. Production & Instrumentation
- Observations
- Conclusion

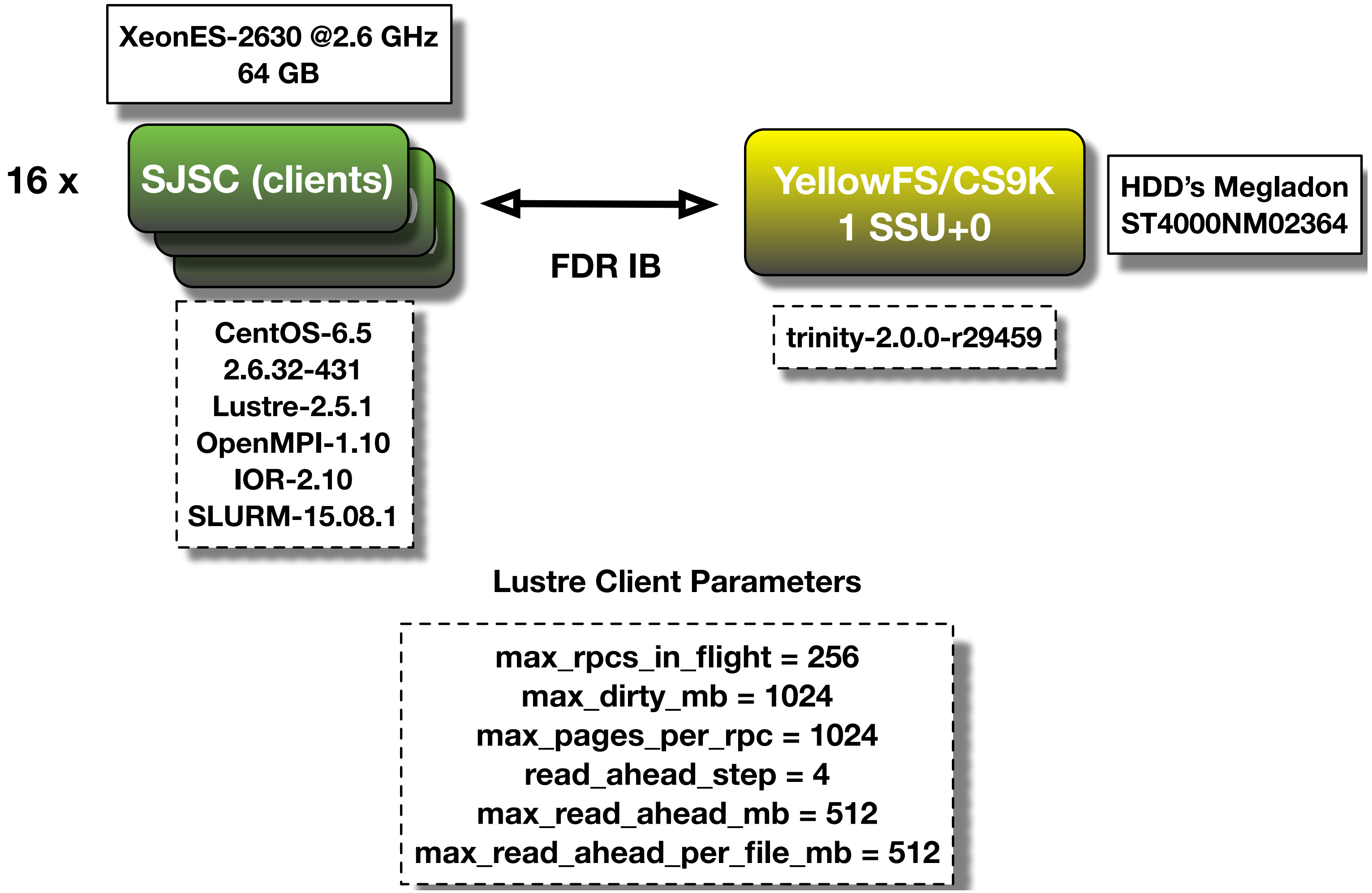
HISTORY/BACKGROUND

- General fragmentation as an issue goes back years in the system literature, and the system software space
- A standardized way to generate/test file system fragmentation doesn’t generally exist
- Some performance work, in general, has been done previously by others, on Lustre & fragmentation, but this was incomplete
- First Instance with ClusterStor of reported impactful fragmentation encountered at scale, dealing with a metadata performance decrease
- Dedicated resources allocated in Fremont lab to provide data collection.

FOCUS…

- Given limited resources/time….
    - Formalize the methodology to study capacity and fragmentation related to bandwidth impact
    - Gather as much data as possible related to performance impact of fragmentation on bandwidth, a very rough "baseline"
    - Produce information, to inform/educate customers and Seagate staff on the possible impacts of capacity and fragmentation related to the current ClusterStor product
- If possible….
    - Evaluate secondary approaches WRT: instrumentality
    - Determine, if data proves a high degree of impact with regard to this issue, possible points of remediation for longer term consideration

TEST ENVIRONMENT, FREMONT LAB



NOMENTCLATURE, “PaF” (POPULATE and FRAGMENT)

PAF: Populate and Fragment

24253639848 K blocks free

PAF: 70.30@264M<24253639848

70: 70% of Capacity

264M: File Size[s] in Structure

30: 30% Probabilistic Frag.

Simple

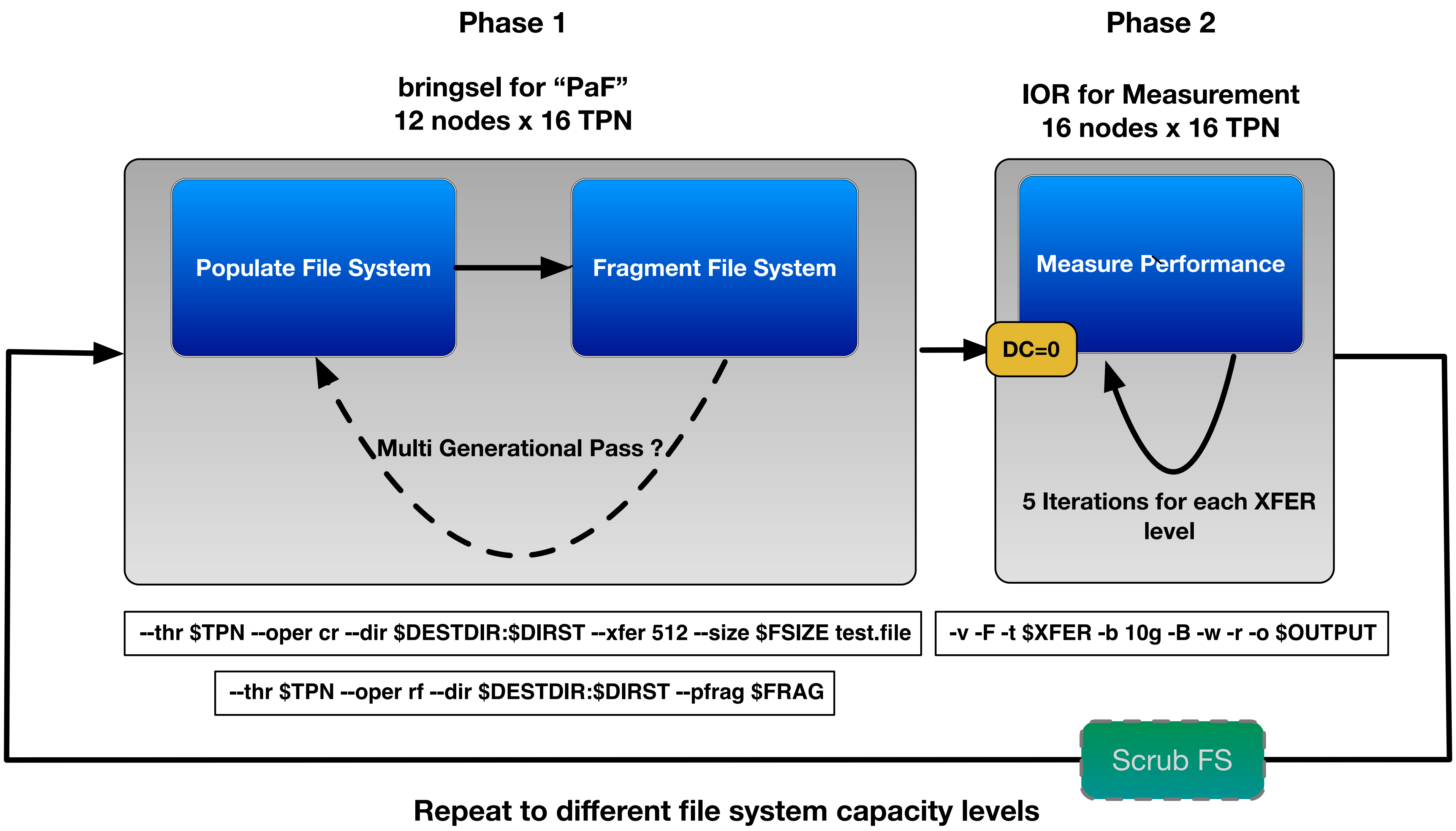
PAF:70.30@264M<24253639848	**Gen-1, Non-Deterministic Homogenous**
PAF:70.30,20.40@264M<24253639848	**Gen-N, Non-Deterministic Homogenous**
PAF:70.30,20.40:67005@264M,40M<24253639848	**Gen-N, Deterministic Heterogeneous**

Complex

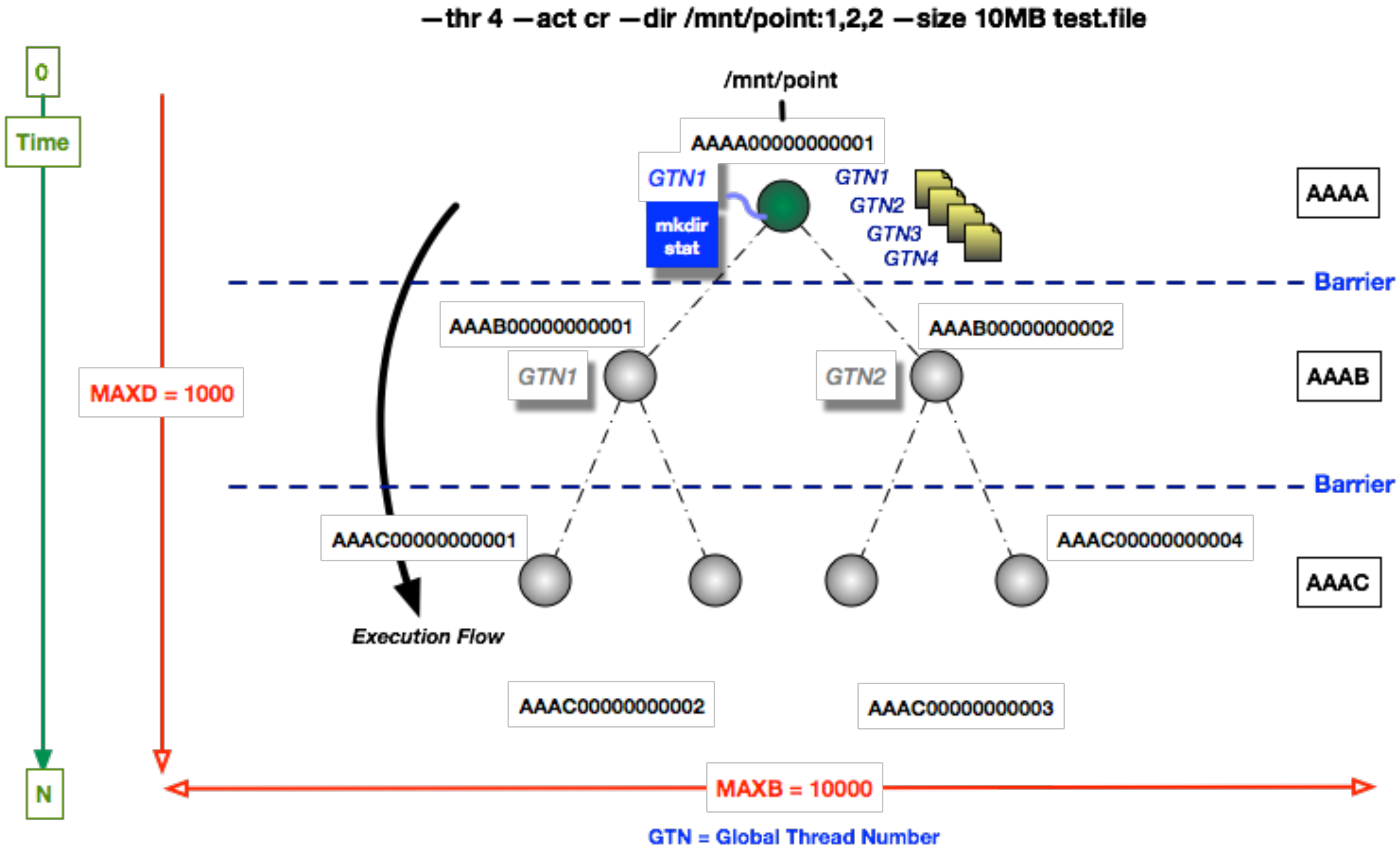
LIMITATIONS of GEN-1 STRUCTURES/TESTING

- **Actual usable capacity within FS higher at higher fragmentation levels**
- **While allowing broader data collection, Gen-1 testing doesn’t possibly represent worse case outcomes**
- **More difficult to attempt mapping from real world to in lab environment**
- **Doesn’t represent imbalance in “target” utilization, true of all Generated PaF structures**
- **Fragmentation is likely a contributing factor to performance degradation, in production environments, but <u>not the only factor</u>**

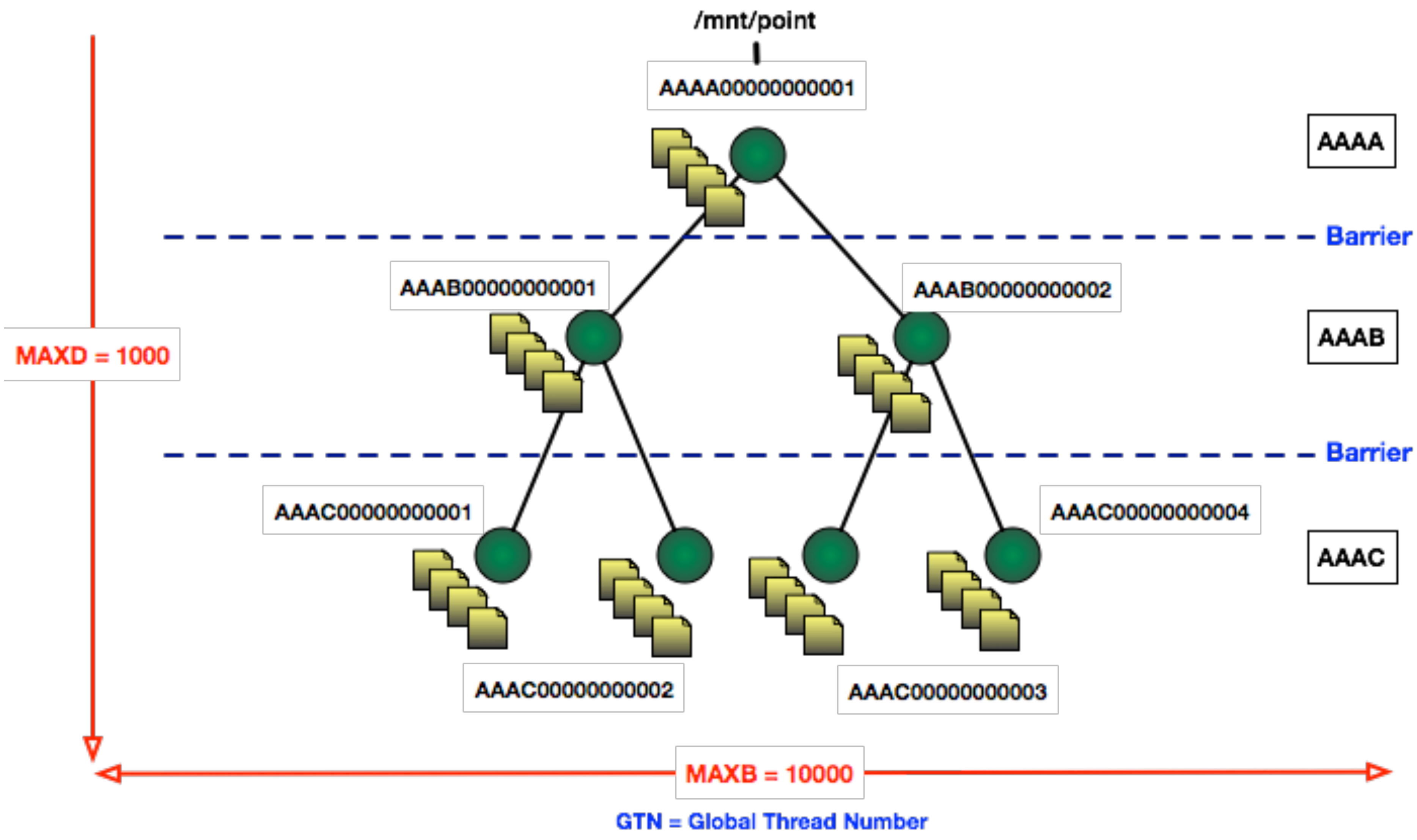
Methodology



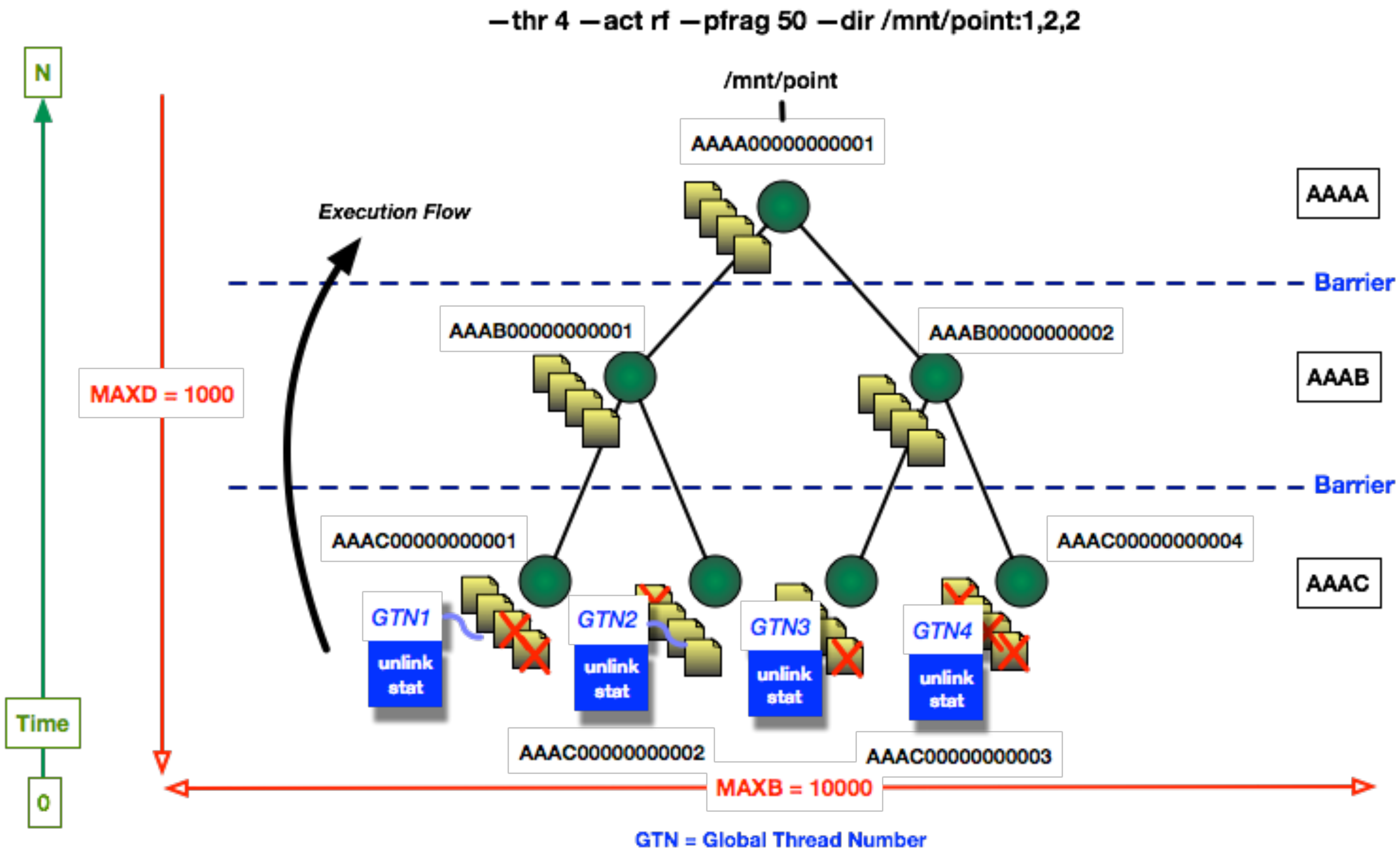
Methodology, Populating a File System, a Simple Example



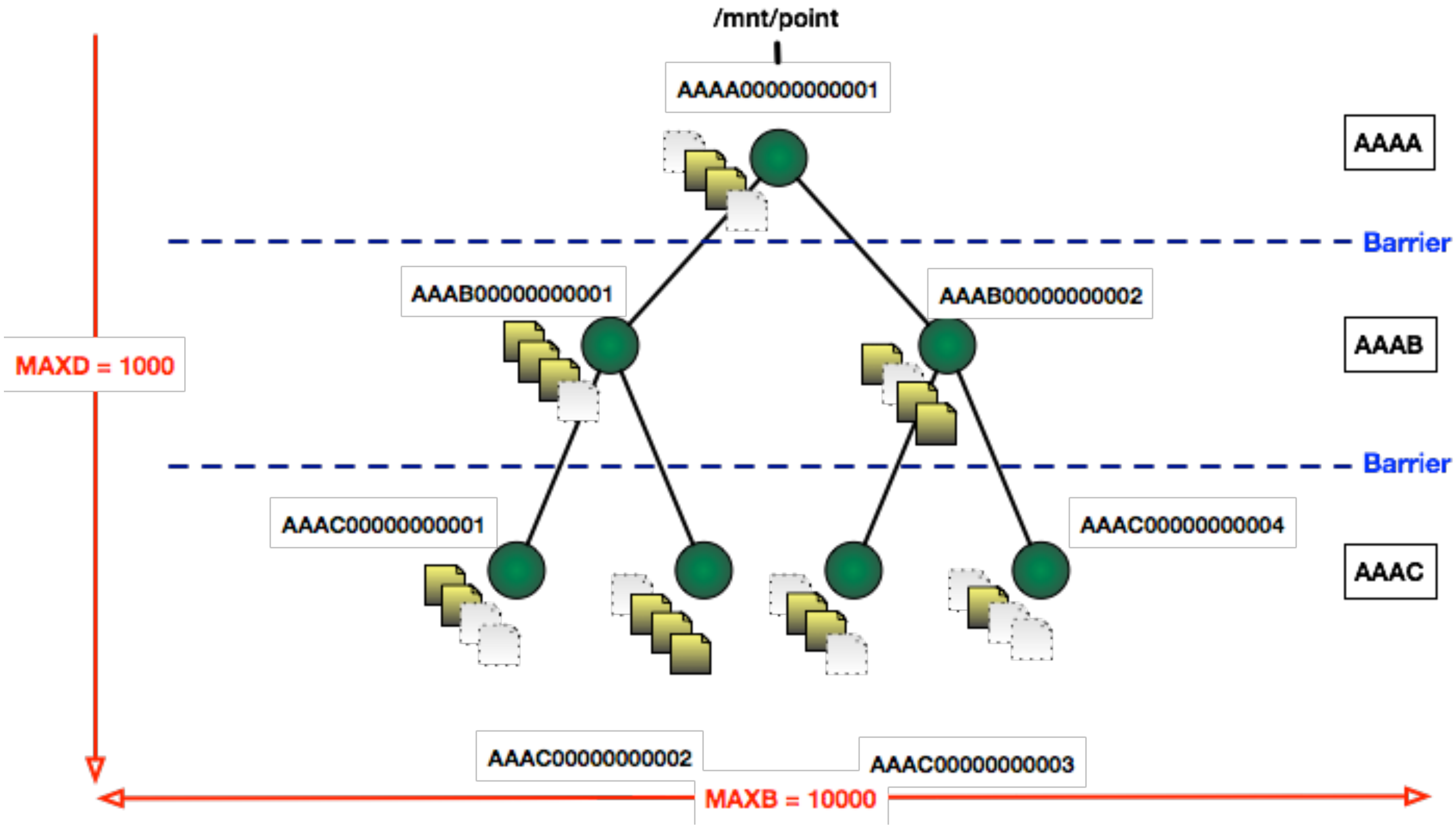
A Populated File System



Methodology, Fragmenting the File System, a Simple Example



A Fragmented File System



POPULATION and FRAGMENTATION, CHOICES….

- Non-deterministic? Provide a wider range of coverage for “baseline”
- Gen1/Homogenous? Quicker to generate structure, given time constraints
- 10% Fragmentation Spread? Baseline vs. time
- 10% Initial Capacity Increments? Again baseline vs. time
- Bringsel? Threaded code that scales to number of available clients
- Again, all of this equates to -> PAF:AA.BB@264M<24253639848

SERIES 1, PERFORMANCE TESTS…

- 8 different PaF capacities, 10% to 80%
- x 8 degrees of fragmentation, 0% to 70%
- x 7 xfer sizes from 1MB to 64MB
- x 5 iterations for each XFER size

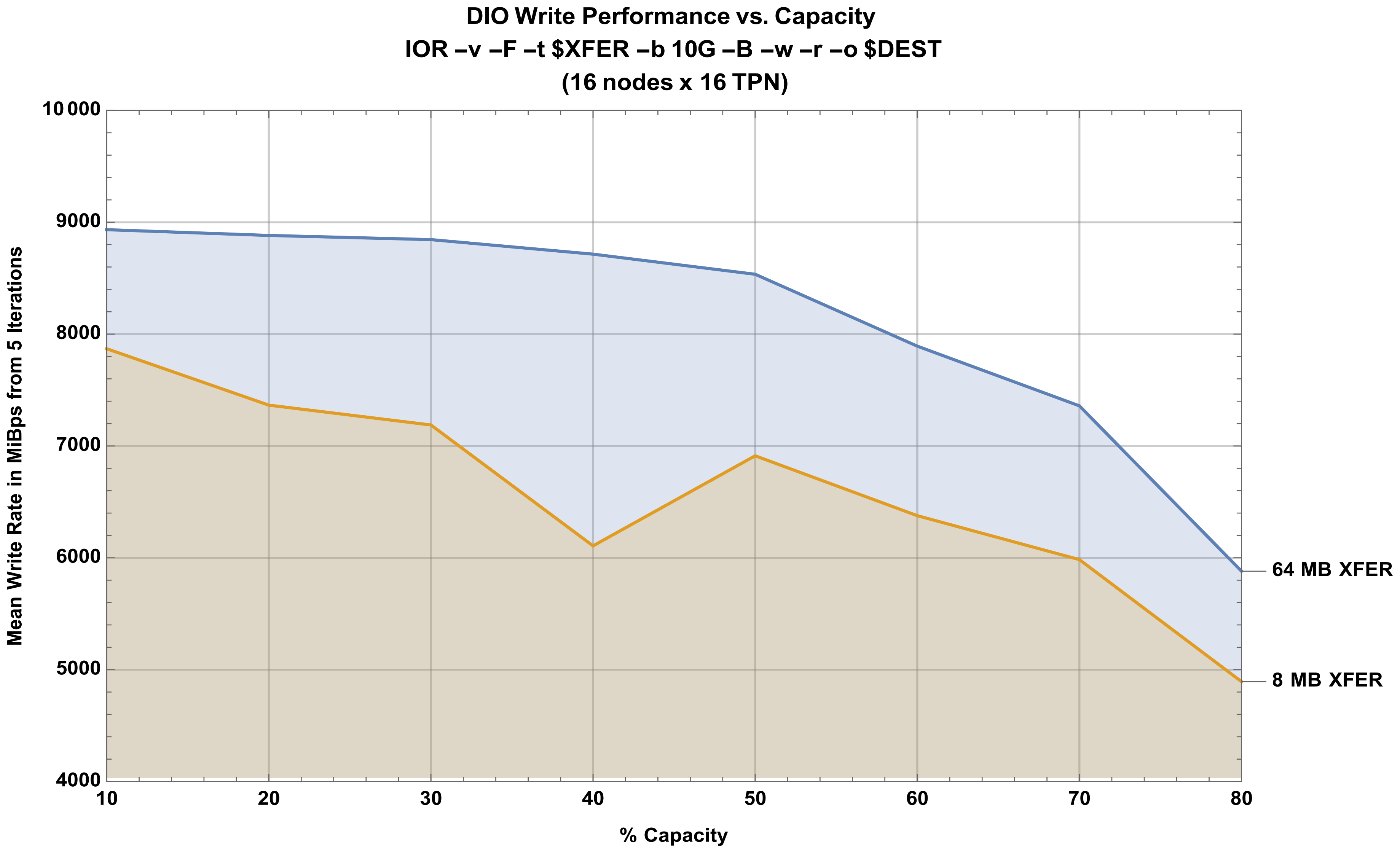
- = 2240 result pairs, composed of DIO write and reads.

PaF:00.00			PaF:10.00			PaF:10.10		
Blck Size	Write	Read	Blck Size	Write	Read	Blck Size	Write	Read
1024	1876.44	3247.40	1024	1892.41	3698.35	1024	1732.08	2741.20
1024	1868.17	3236.41	1024	1906.88	3668.86	1024	1861.66	3480.52
1024	1861.75	3268.75	1024	1909.14	3683.75	1024	1881.65	3590.47
1024	1860.45	3289.47	1024	1906.34	3689.90	1024	1880.23	3598.46
1024	1857.75	3264.11	1024	1908.22	3661.63	1024	1876.71	3618.22
2048	3102.12	3994.84	2048	3023.34	4437.04	2048	2944.24	3616.35
2048	3076.79	3984.72	2048	3007.68	4462.56	2048	2976.11	3975.91
2048	3068.11	4013.94	2048	3021.21	4451.30	2048	2972.22	4069.85
2048	3081.90	4001.21	2048	3022.25	4440.09	2048	2982.17	4066.15
2048	3086.71	3980.96	2048	3027.65	4447.00	2048	2960.63	4124.84
4096	4835.71	5020.15	4096	4857.95	5634.46	4096	4522.98	5031.87
4096	4846.52	5002.37	4096	4851.30	5644.79	4096	4696.30	5111.76
4096	4855.30	5047.38	4096	4871.66	5616.79	4096	4736.90	5151.99
4096	4855.22	4995.85	4096	4871.36	5637.33	4096	4760.94	5123.16
4096	4799.63	5034.75	4096	4869.68	5666.65	4096	4734.36	5123.50
8192	7637.70	7041.72	8192	7948.29	7870.26	8192	6649.10	7626.76
8192	7673.32	7036.38	8192	7940.25	7919.26	8192	7368.61	7461.29
8192	7501.03	7115.09	8192	7373.77	7847.78	8192	7680.36	7560.13
8192	7793.69	7068.29	8192	8005.58	7942.40	8192	7689.41	7390.62
8192	7841.45	7125.70	8192	8079.10	7931.40	8192	7640.98	7434.37
16384	8782.72	8197.94	16384	8853.96	8358.57	16384	6841.39	8019.07
16384	8786.37	8204.51	16384	8860.60	8367.81	16384	8412.51	8036.41
16384	8861.11	8172.41	16384	8790.30	8284.89	16384	8761.46	8164.84
16384	8869.50	8079.82	16384	8845.12	8366.97	16384	8853.44	8100.25
16384	8547.80	8216.02	16384	8890.05	8330.92	16384	8777.65	8168.40
32768	8865.64	8534.55	32768	8951.26	8617.33	32768	6953.11	8274.85
32768	8957.30	8647.72	32768	8945.93	8563.64	32768	8542.13	8474.71
32768	8926.09	8555.03	32768	8967.78	8437.41	32768	8919.46	8420.08
32768	8878.50	8703.94	32768	8747.66	8521.00	32768	8918.47	8581.68
32768	8897.52	8718.34	32768	8852.33	8470.67	32768	8887.88	8414.77
65536	8839.55	8894.92	65536	8907.97	8777.53	65536	6934.56	8361.71
65536	8811.55	8870.79	65536	8964.01	8798.98	65536	8408.03	8698.05
65536	8894.13	8848.84	65536	9000.10	8794.21	65536	8881.54	8751.73
65536	8910.26	8754.34	65536	8878.19	8677.60	65536	8913.28	8775.96
65536	8861.54	8869.67	65536	8918.37	8781.91	65536	8876.77	8736.61

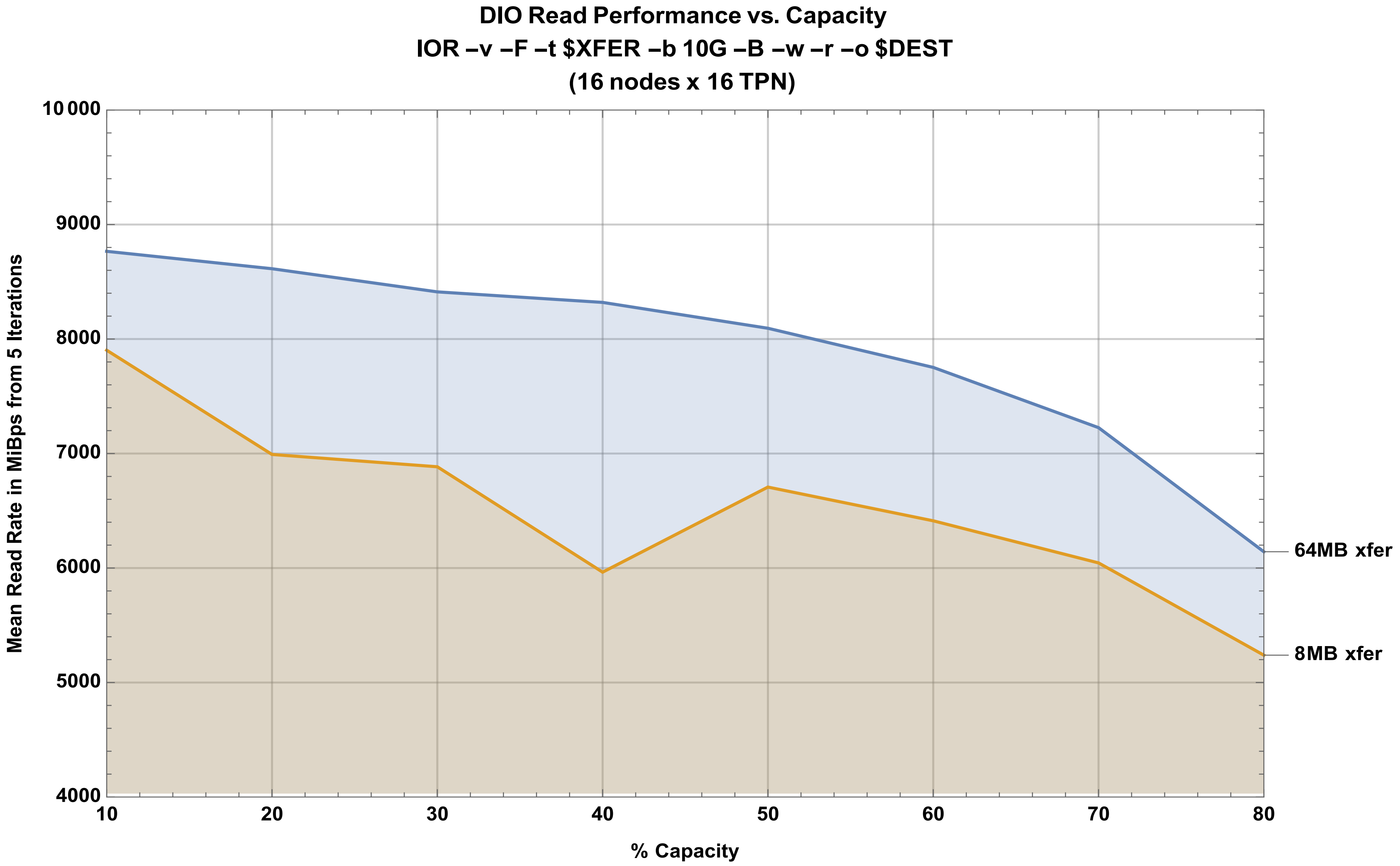
Series 1: PAF: AA.BB@264M<24253639848

Where AA = 10 to 80, and BB = 0 to 70

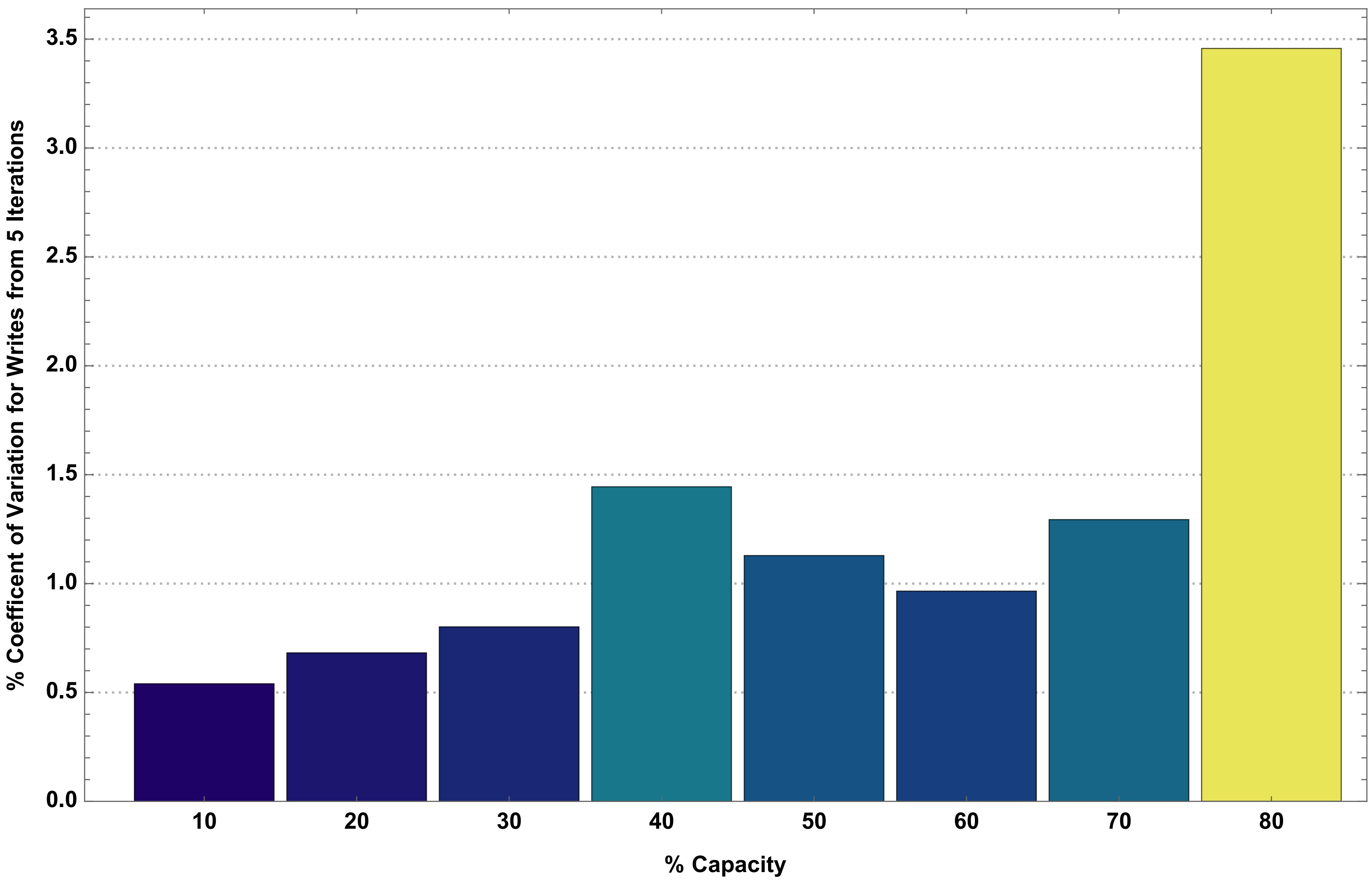
OD/ID PERFORMANCE IMPACTS, WRITES, PAF:10.00 TO PAF:80.00



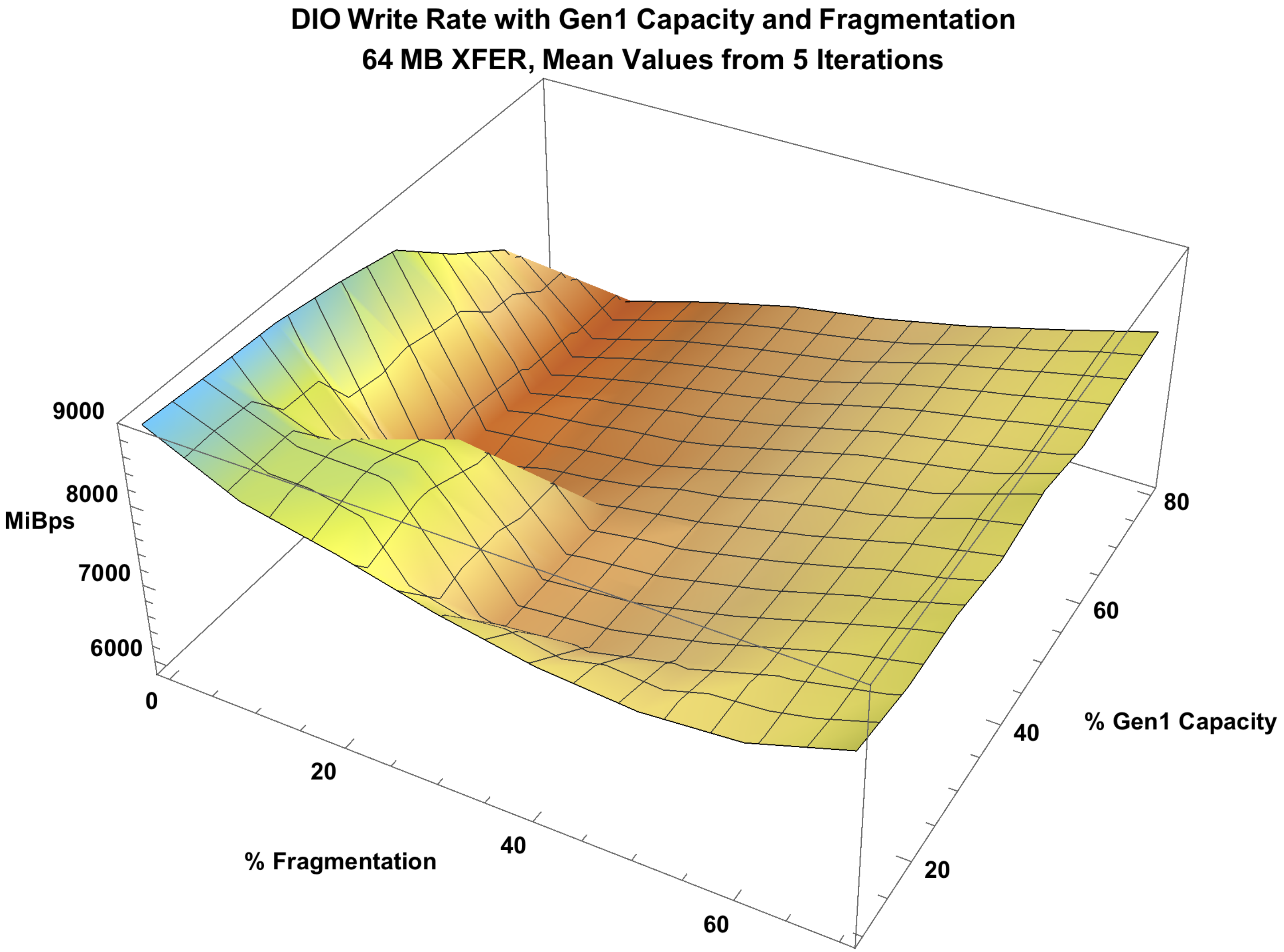
OD/ID PERFORAMANCE IMPACTS, READS, PAF:10.00 TO PAF:80.00



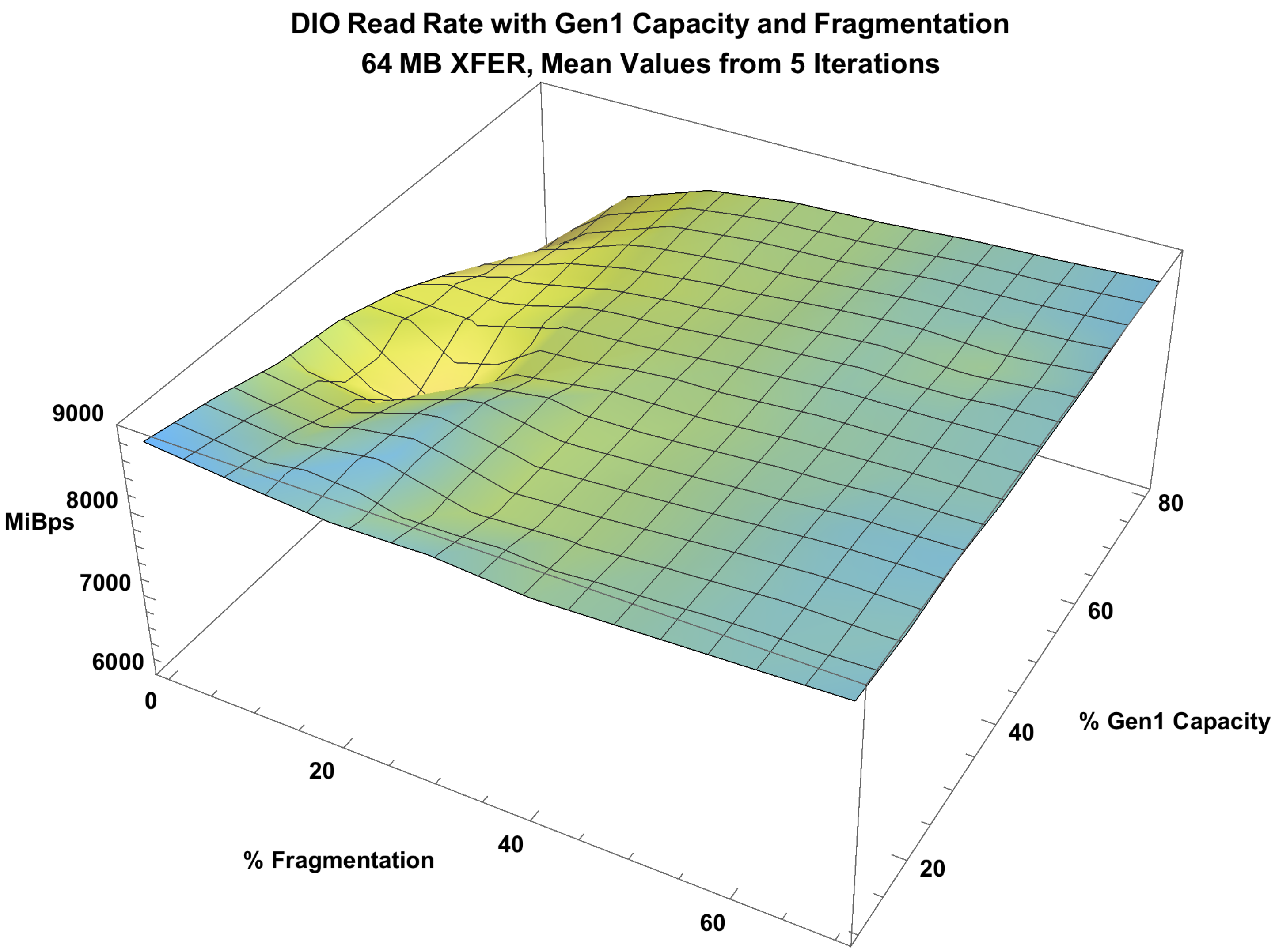
VARIATION % FOR DIO WRITES, 64 MB XFERs @ PaF:XX.00



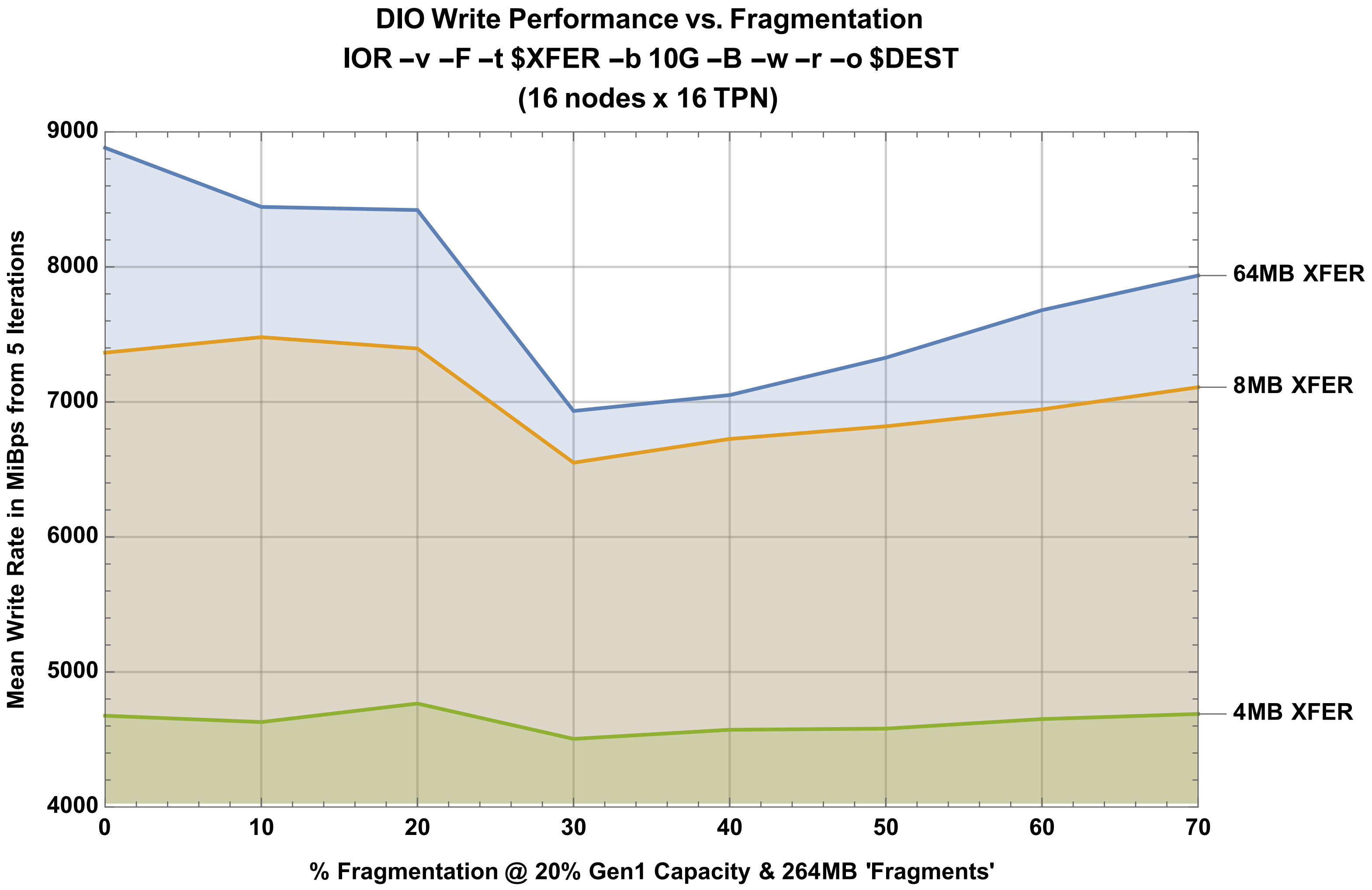
64 MB WRITE RESULTS, PaF:AA.BB (AA:10->80, BB:00->70)



64 MB READ RESULTS, PaF:AA.BB (AA:10->80, BB:00->70)



PERFORMANCE OF LARGE TRANSFERS, WRITES, WITH FRAGMENTATION



SERIES 2, PERFORMANCE TESTS….

- 1 PaF capacity, 20%
- x 7 degrees of fragmentation, 10% to 70%
- x 2 xfer sizes, 8MB to 64MB
- x 5 iterations for each XFER size (WR only)

- = 70 result values, composed of DIO writes

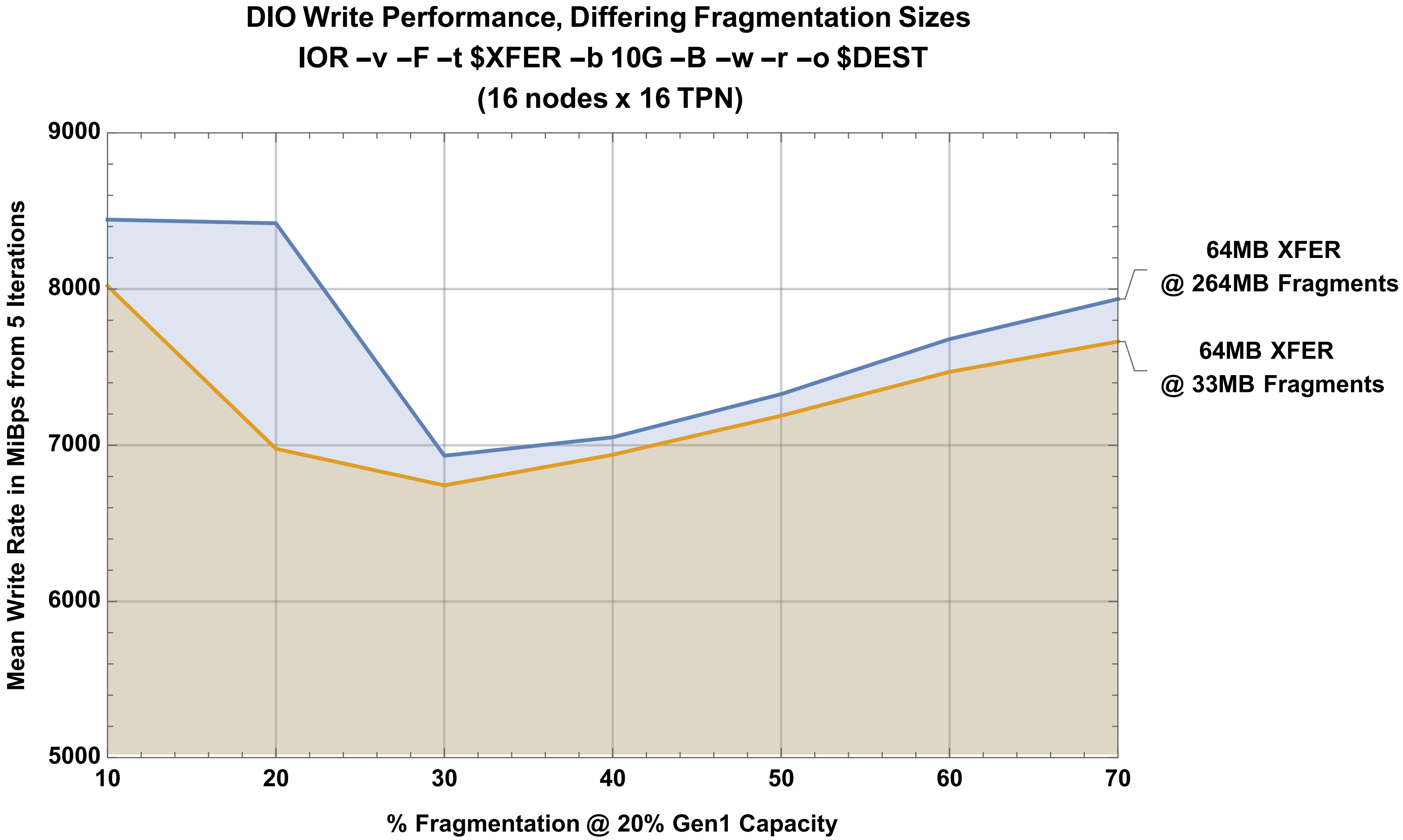
PAF: 20.AA@33M<24253639848

Where AA = 10 to 70

to compare to PAF:20.AA@264M<24253639848

PERFORMANCE IMPACTS OF DIFFERING FRAGMENTATION SIZES

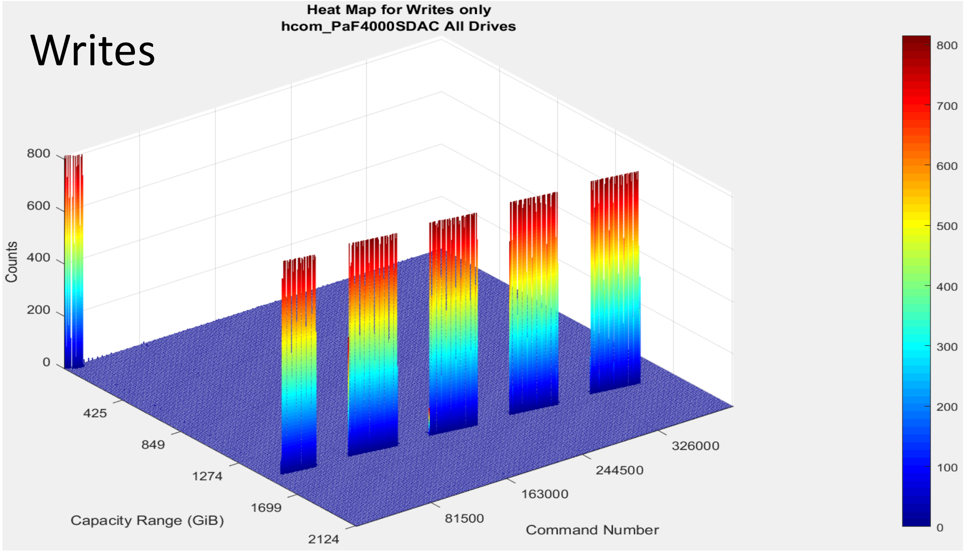
Homogenous Structures

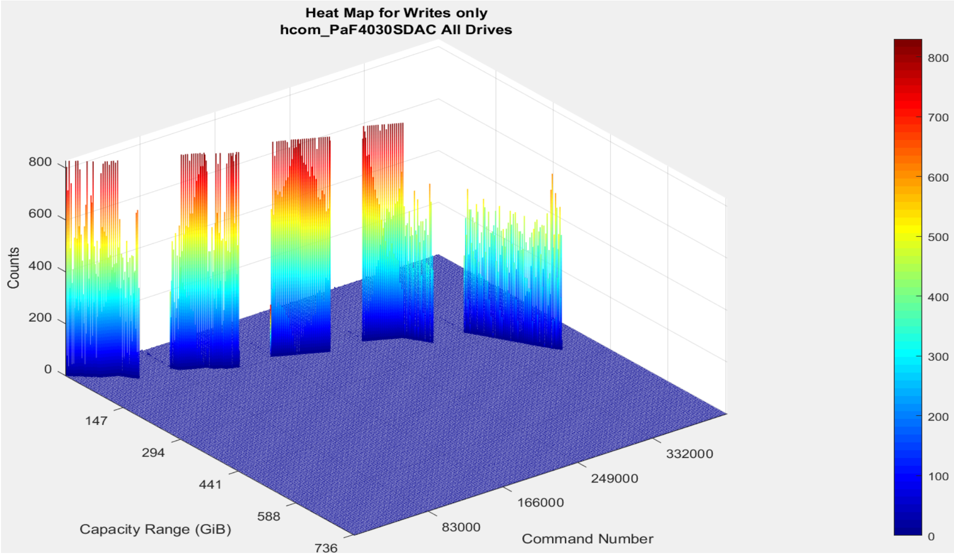


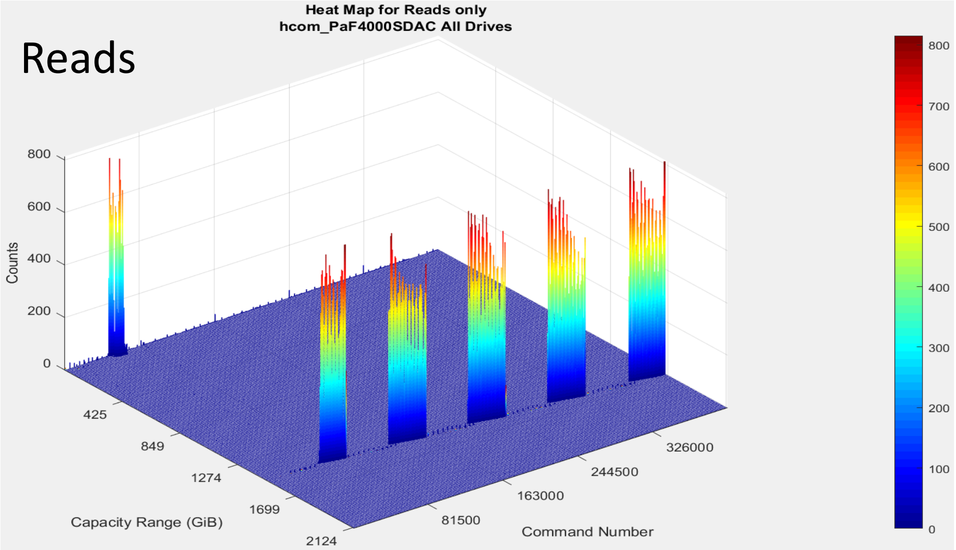
LAB VS. PRODUCTION FACILITY DATA COLLECTION….

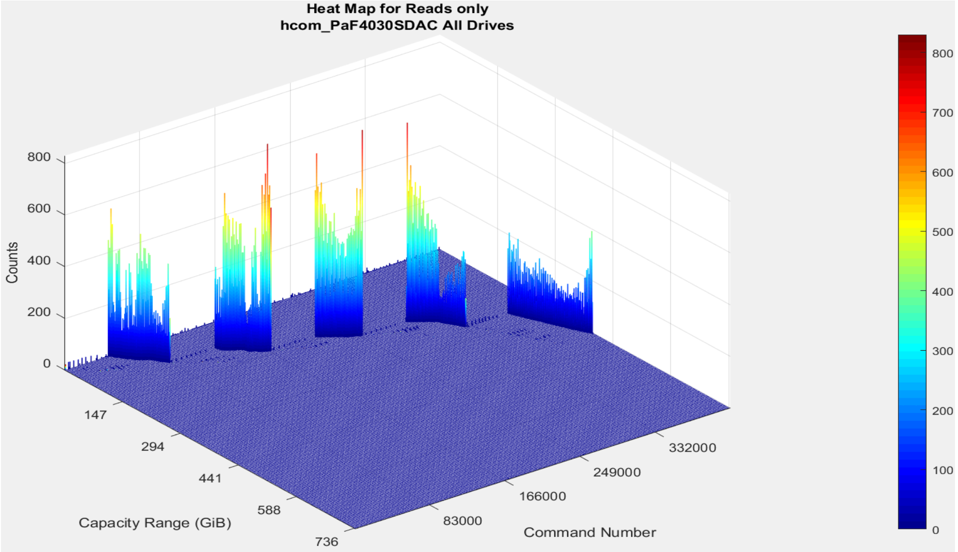
- **The problem, estimating the degree of impact of fragmentation to production sites**
- **Instrumentality problematic in production facilities >**
- **Two possible approaches, outside diagnostic sweeps:**
  - **Block traces on running file systems**
  - **dumpe2fs on OST targets**
- **Assuming all else, on site, WRT: the above, is optimal**
- **Block trace examples from the lab….**

BLOCK TRACES, LAB, NO FRAGMENTATION VS. FRAGMENTATION









BLOCK TRACES AS PRODUCTION INSTRUMENTATION

- **Can provide circumstantial evidence of fragmentation aligned with in lab observations**
- **Difficult to collect data for larger OST counts**
- **Time intensive analysis**
- **Assumes “implied”” knowledge of entire job stream for duration of collection**
- **Non-optimal to large scale production sites**

DUMPE2FS AS PRODUCTION INSTRUMENTATION

- Provides basic group allocation information on a per target basis. (OST)
- Invoked on each OSS, run duration of several minutes
- Can be run while FS is active
- Header and Group Allocation Information
- Sometimes used in combination with other utilities, filefrag, eval_dumpe2fs, etc
- Produces copious amounts of data, ~390+ MB, per OST

DUMPE2FS AS PRODUCTION INSTRUMENTATION, CONTINUED…

eval_dump2fs summary…..

Summary: 30316376064 blocks (25224631557 free), 118423424 inodes (118346357 free)925183 block groups with 32768 blocks and 128 inodes each:

924514 block groups with many (> 50%) free inodes,
306 block groups with enough (< 1/2) free inodes,
76 block groups with few (< 1/4) free inodes,
159 block groups with very few (< 1/8) free inodes,
35 block groups with *really* few (< 1/16) free inodes,

etc….

Needs much further refinement in line with data collection efforts

OBSERVATIONS

- A simple OST fragmentation model, in the lab, can cause up to ~35% loss of BW
- A simple OST fragmentation model, in the lab, can also impact metadata performance
- Impacts can occur at low % of utilization, based on initial starting point
- Most modeled fragmentation highly impactful on writes, less so on reads. Also confirmed in the field
- Fragmentation can contribute to the loss of performance, as Lustre in ClusterStor is utilized, "age" the file system
- AFA/SSS, back ends will not show any effects on write performance
- Larger installations/OST counts more likely to exhibit the symptoms of fragmentation impacted performance <N:N Issue>
- Unloaded buffer IO helps shield the fragmentation issue

OBSERVATIONS, CONTINUED…

- **Impact on performance more noticeable at higher xfer ranges**
- **Block traces provide circumstantial evidence, i.g. the deflection curve, but general usefulness in production environments is highly limited**
- **Impact of OST imbalance, and/or marginal HDD's in system along with fragmentation is an open question**
- **Remediation of fragmentation highly desirable, for use in production facilities**

CONCLUSION:

- We can easily model several different forms of fragmentation in laboratory conditions, to study various impacts around data and metadata
- Study results using a simple model yield BW impacts from 2% to ~35% overhead
- Previous efforts have shown a wide degree of variation to metadata performance with OST fragmentation
- Fragmentation can contribute to performance issues, but it is not the only factor
- Performance impacts vary, depending on utilization patterns
- Seagate core engineering team made aware of findings and possible, longer term, remediation options
- Instrumentality, is a longer development task, especially at scale

THANK YOU, QUESTIONS?

john <dot> **w** <dot> **kaitschuck** <at> **seagate** <dot> **com**
[or]
jkaitsch <at> **acm** <dot> **org**