Storage Benchmarking

Repeatable & Comparable

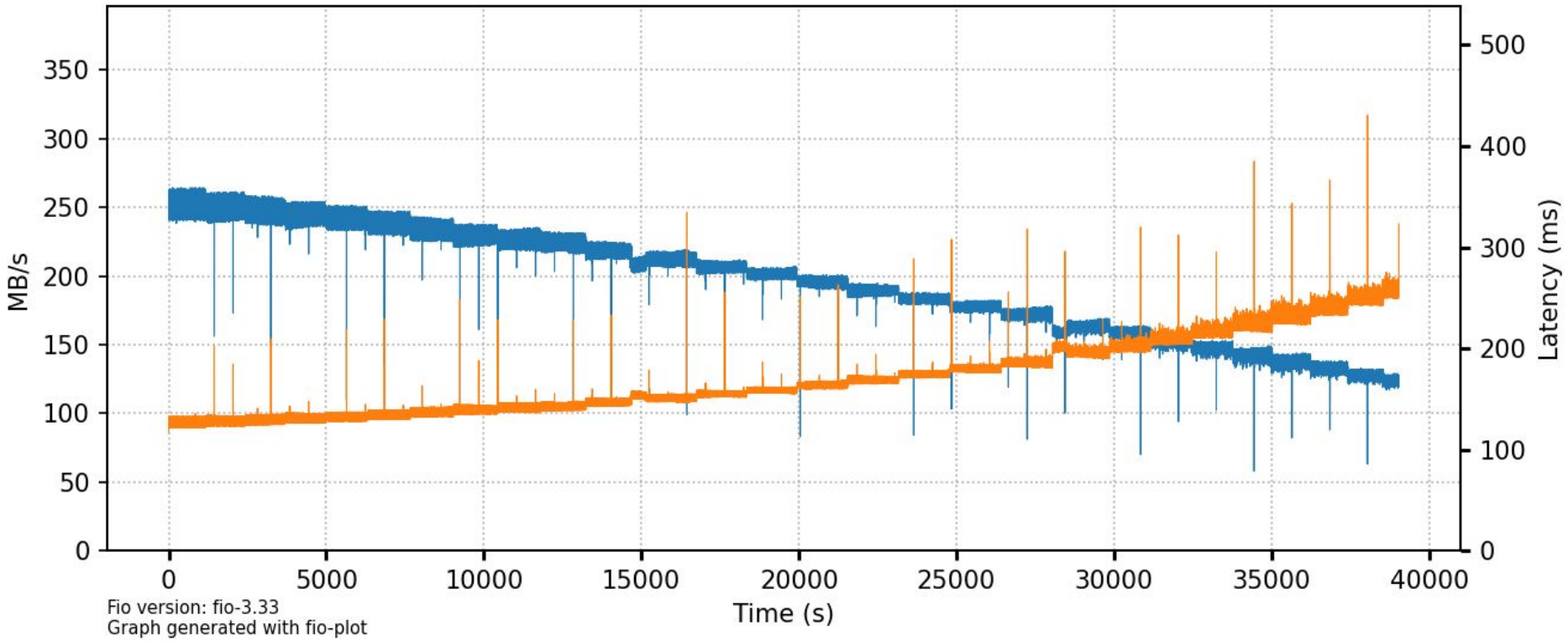
Trent Lloyd
Sustaining Engineering @ Canonical (Ubuntu)
@lathiat@fosstodon.org

https://fosstodon.org/@lathiat

Repeatable & Comparable

Sequential write performance of 8 TB HDD

| rw write | bs 1M | iodepth 32 | numjobs 1 | type bw, lat | filter read, write |



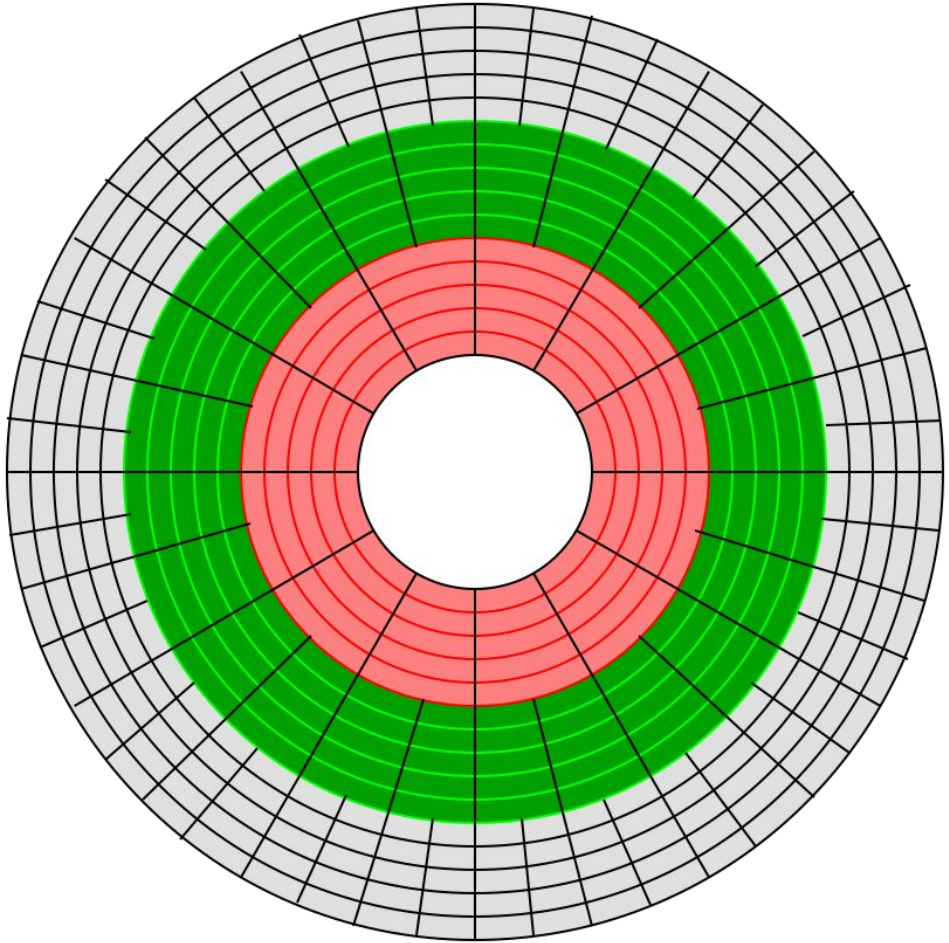
Fio version: fio-3.33
Graph generated with fio-plot

name	rw	type	qd	nj	mean	std%	P99.99
Toshiba MG08-D	write	bw	32	1	196	20.06	264
Toshiba MG08-D	write	lat	32	1	171	22.08	336

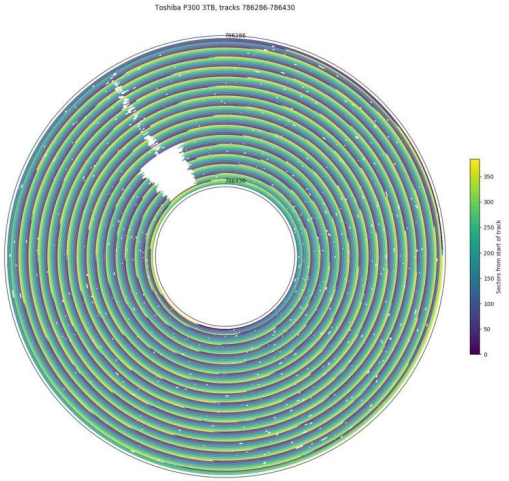
YouTube @bigbluebananabread https://www.youtube.com/shorts/mljraTiKYNY CC0

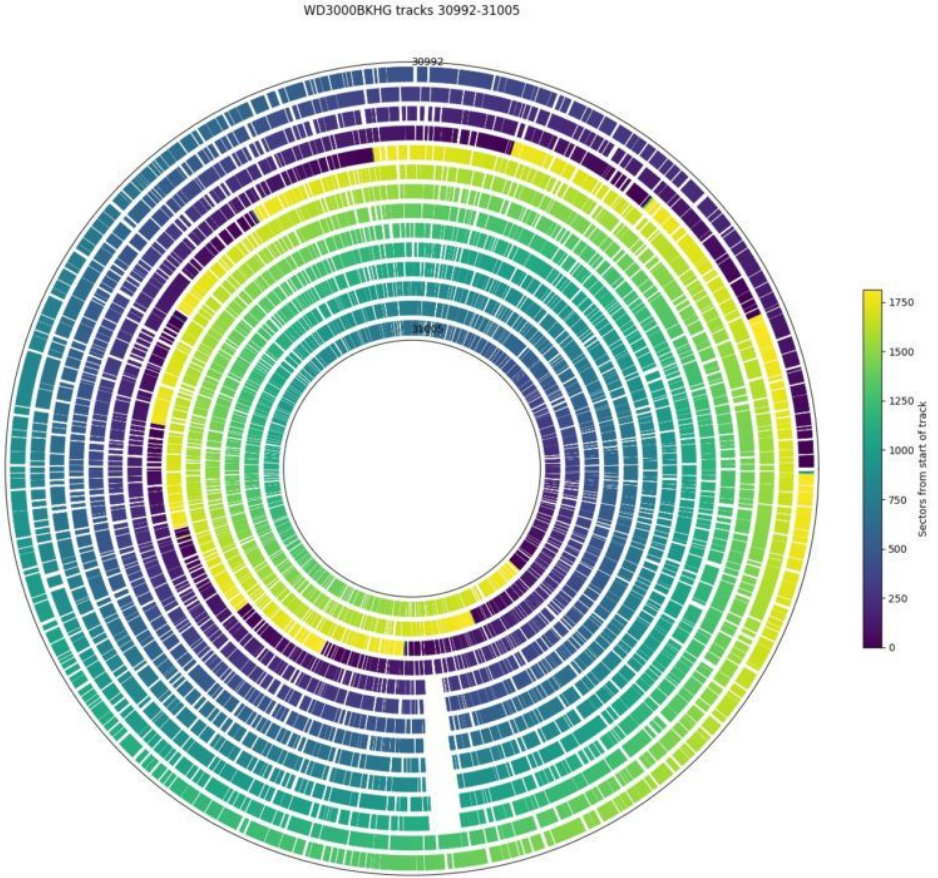
https://commons.wikimedia.org/wiki/File:Laserdisc_CAV.jpg CC-BY-SA-3.0

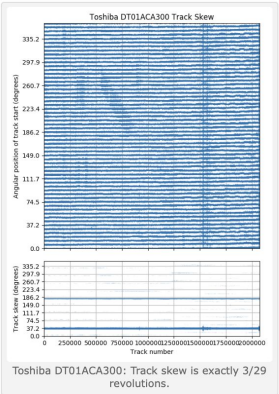
https://www.reddit.com/r/LaserDisc/comments/jrewab/i_like_how_you_can_see_the_skewed_vertical/



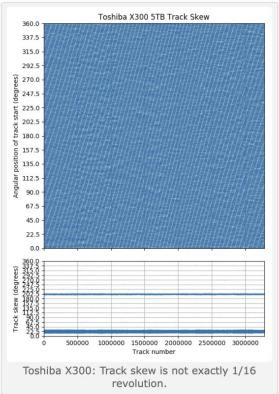
https://en.wikipedia.org/wiki/File:DiskStructure.svg



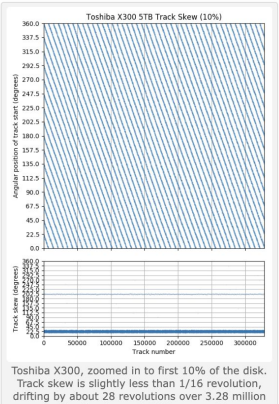




Toshiba DT01ACA300: Track skew is exactly 3/29 revolutions.



Toshiba X300: Track skew is not exactly 1/16 revolution.

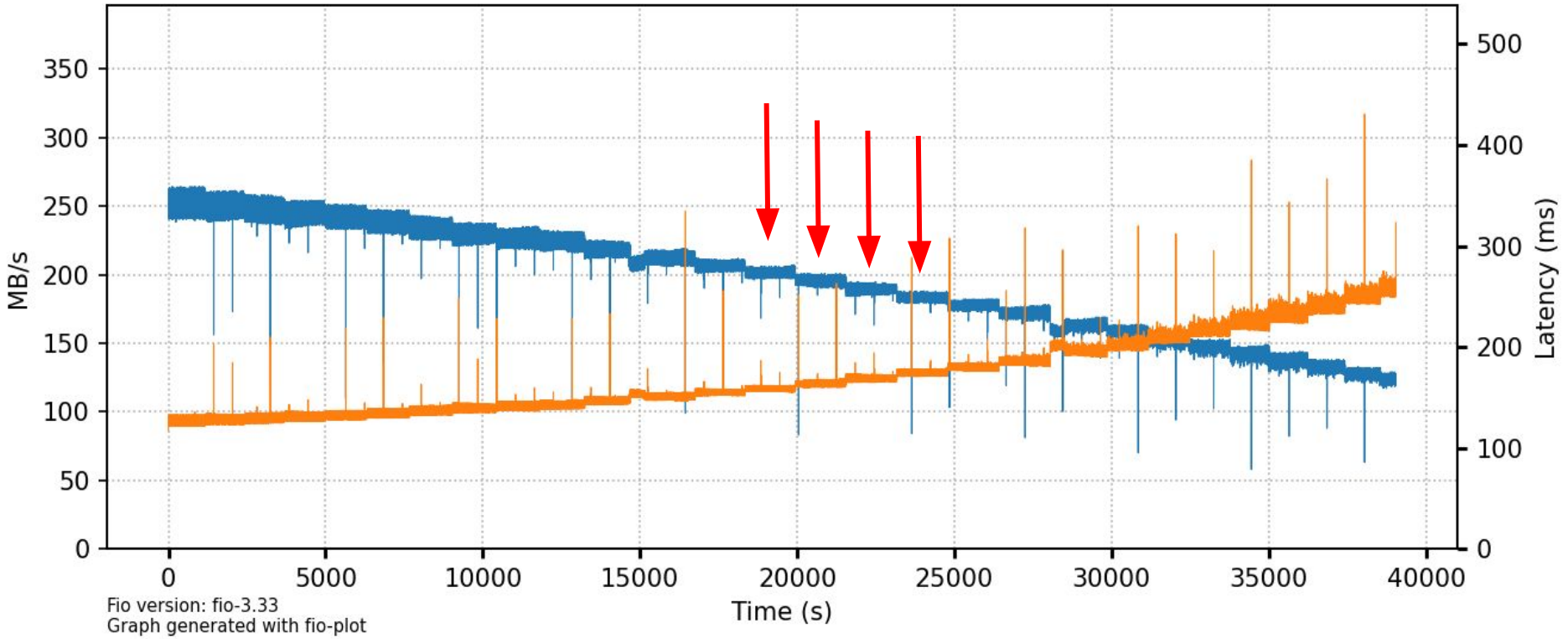


Toshiba X300, zoomed in to first 10% of the disk. Track skew is slightly less than 1/16 revolution, drifting by about 28 revolutions over 3.28 million tracks (or about -0.003° per track).

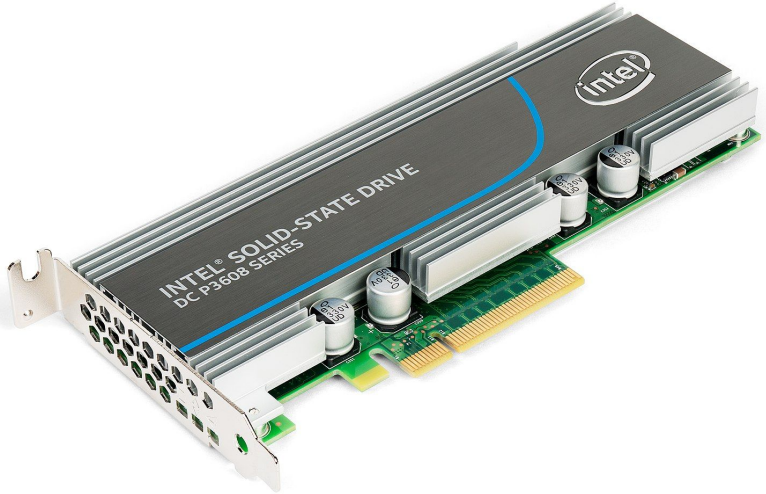
Discovering Hard Disk Physical Geometry through Microbenchmarking
https://blog.stuffedcow.net/2019/09/hard-disk-geometry-microbenchmarking/

Sequential write performance of 8 TB HDD

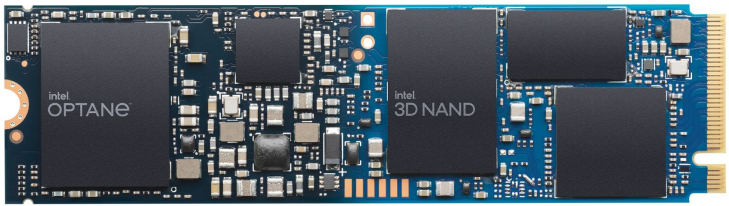
| rw write | bs 1M | iodepth 32 | numjobs 1 | type bw, lat | filter read, write |



name	rw	type	qd	nj	mean	std%	P99.99
Toshiba MG08-D	write	bw	32	1	196	20.06	264
Toshiba MG08-D	write	lat	32	1	171	22.08	336

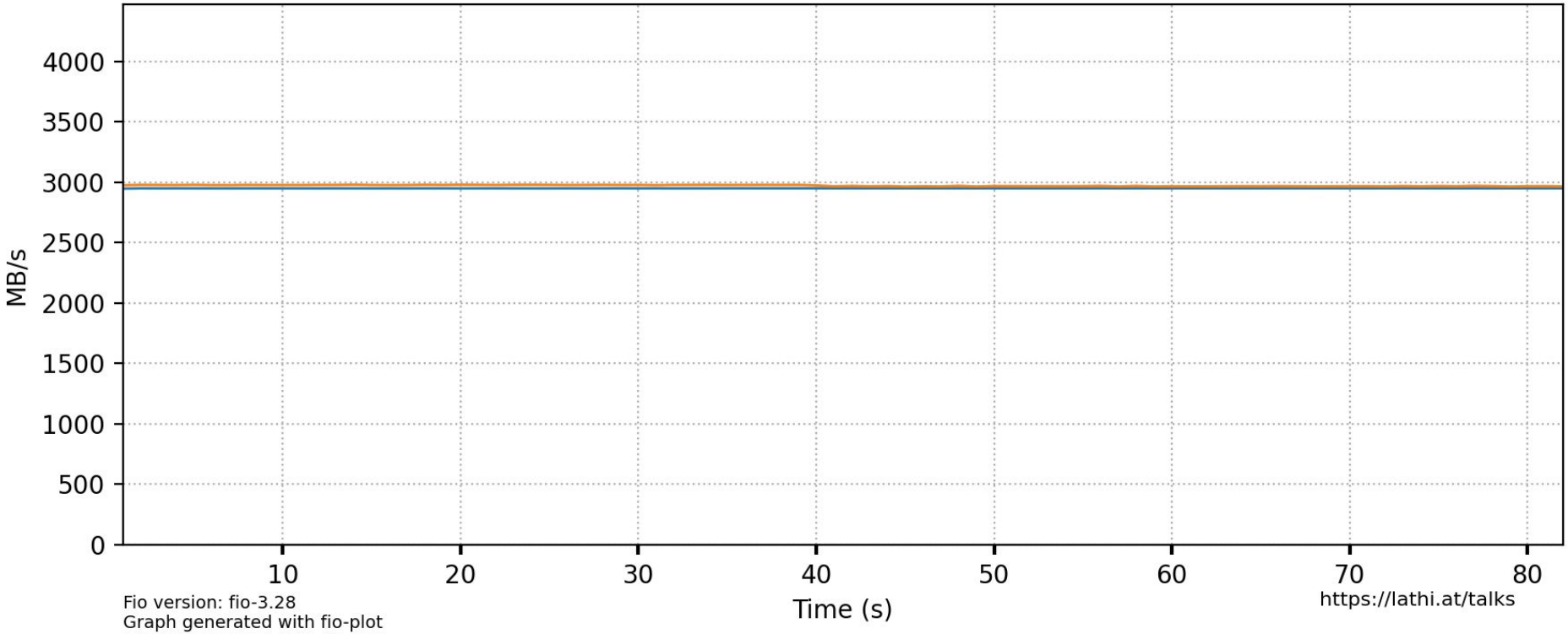

INTEL® SOLID-STATE DRIVE
DC P3608 SERIES
intel


SAMSUNG
PM1653


intel
OPTANE
intel
3D NAND

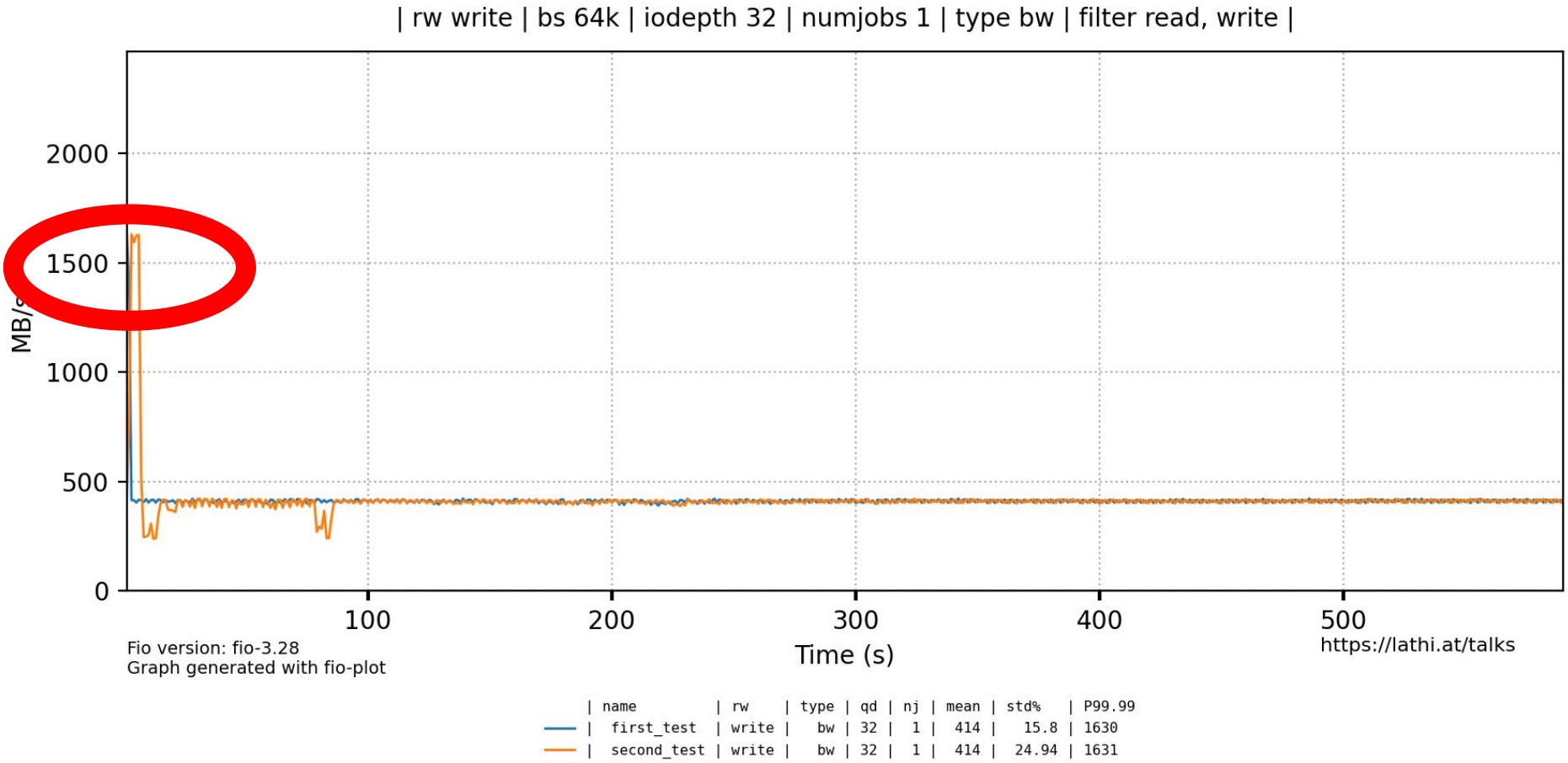
KXG60ZNV256G (after enhanced secure erase)

| rw read | bs 64k | iodepth 32 | numjobs 1 | type bw | filter read, write |



name	rw	type	qd	nj	mean	std%	P99.99
first_test	read	bw	32	1	2949	0.01	2950
second_test	read	bw	32	1	2973	0.18	2980

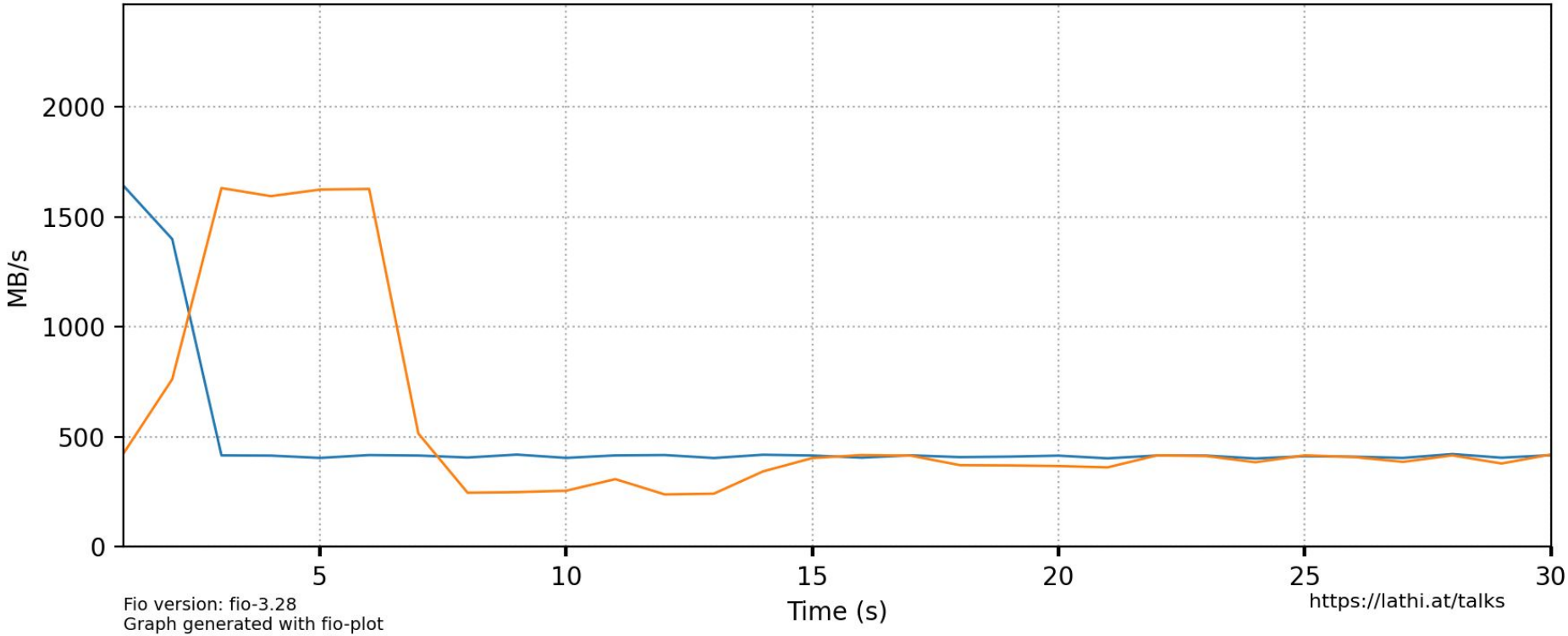
KXG60ZNV256G (after enhanced secure erase)



name	rw	type	qd	nj	mean	std%	P99.99
first_test	write	bw	32	1	414	15.8	1630
second_test	write	bw	32	1	414	24.94	1631

KXG60ZNV256G (after enhanced secure erase)

| rw write | bs 64k | iodepth 32 | numjobs 1 | type bw | filter read, write | Truncated x-axis |



name	rw	type	qd	nj	mean	std%	P99.99
first_test	write	bw	32	1	414	15.8	1630
second_test	write	bw	32	1	414	24.94	1631

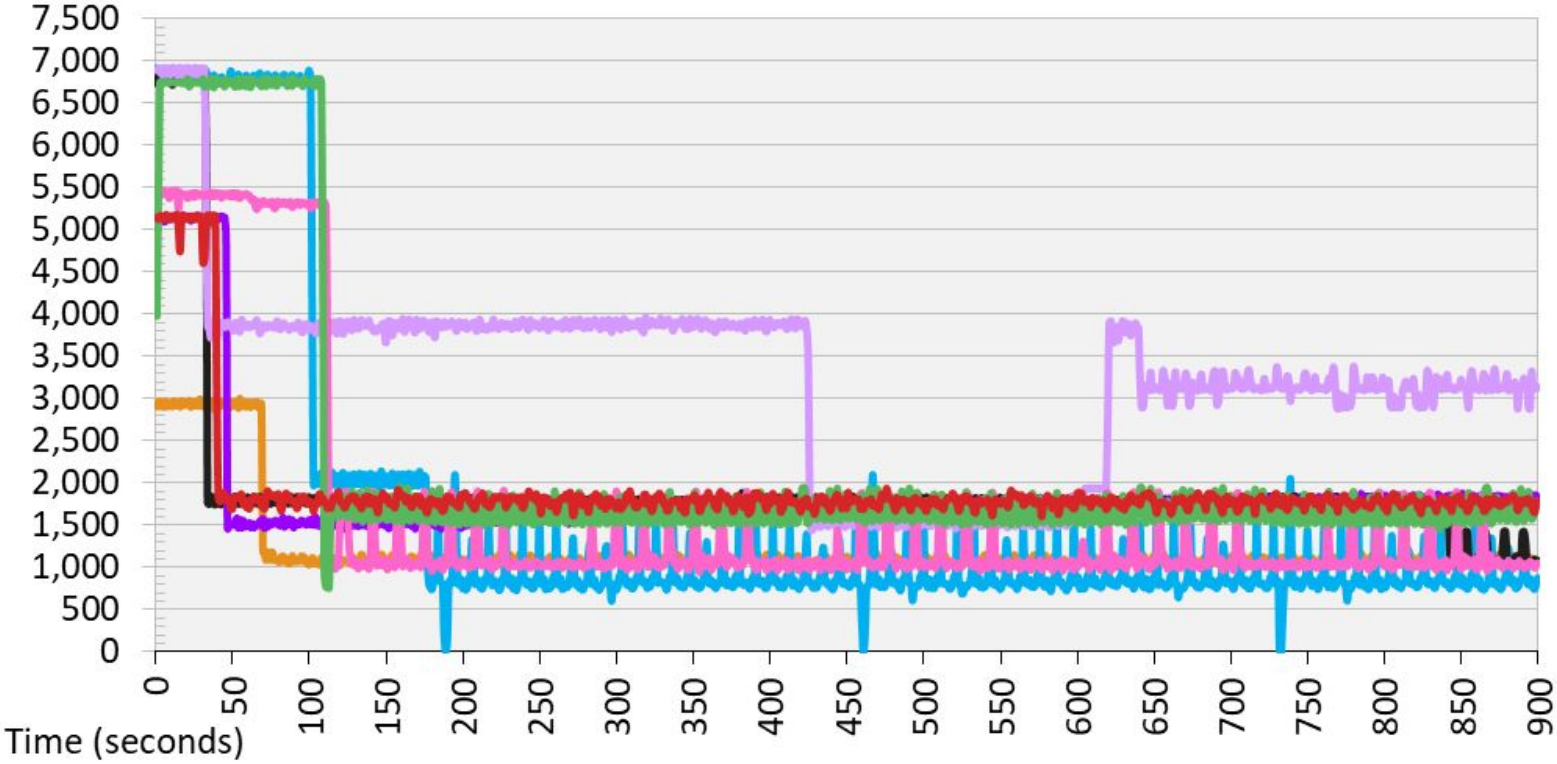
Base Model Number	KXG60ZNV1T02	KXG60ZNV512G	KXG60ZNV256G
SED Model Number	KXG6AZNV1T02	KXG6AZNV512G	KXG6AZNV256G
Capacity	1,024 GB	512 GB	256 GB
Basic Specifications			
Form Factor		M.2 2280-S2 Single-sided	
Interface		PCIe® 3.0, NVMe™ 1.3a	
Maximum Interface Speed		32 GT/s (PCIe® Gen3 x4)	
Flash Memory Type		BiCS FLASH™ TLC	
Performan (Up to)			
Sequential Read	3,180 MB/s	3,100 MB/s	3,050 MB/s
Sequential Write	2,960 MB/s	2,800 MB/s	1,550 MB/s
Random Read	355K IOPS	325K IOPS	270K IOPS
Random Write	365K IOPS	355K IOPS	335K IOPS

Source: https://apac.kioxia.com/en-apac/business/ssd/client-ssd/xg6.html

Sustained Sequential Write - 1MB QD 32

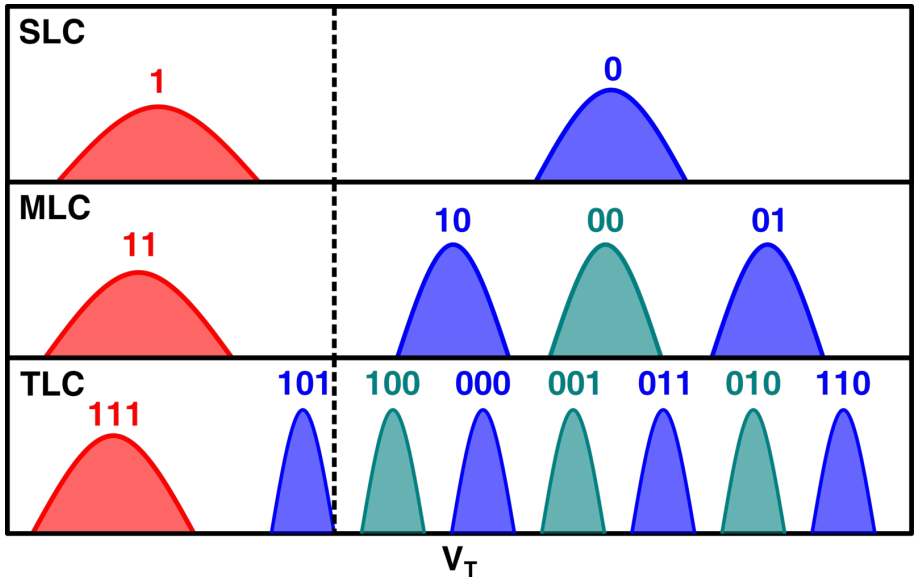
iometer - Write in MBps - Higher is Better

tom's HARDWARE

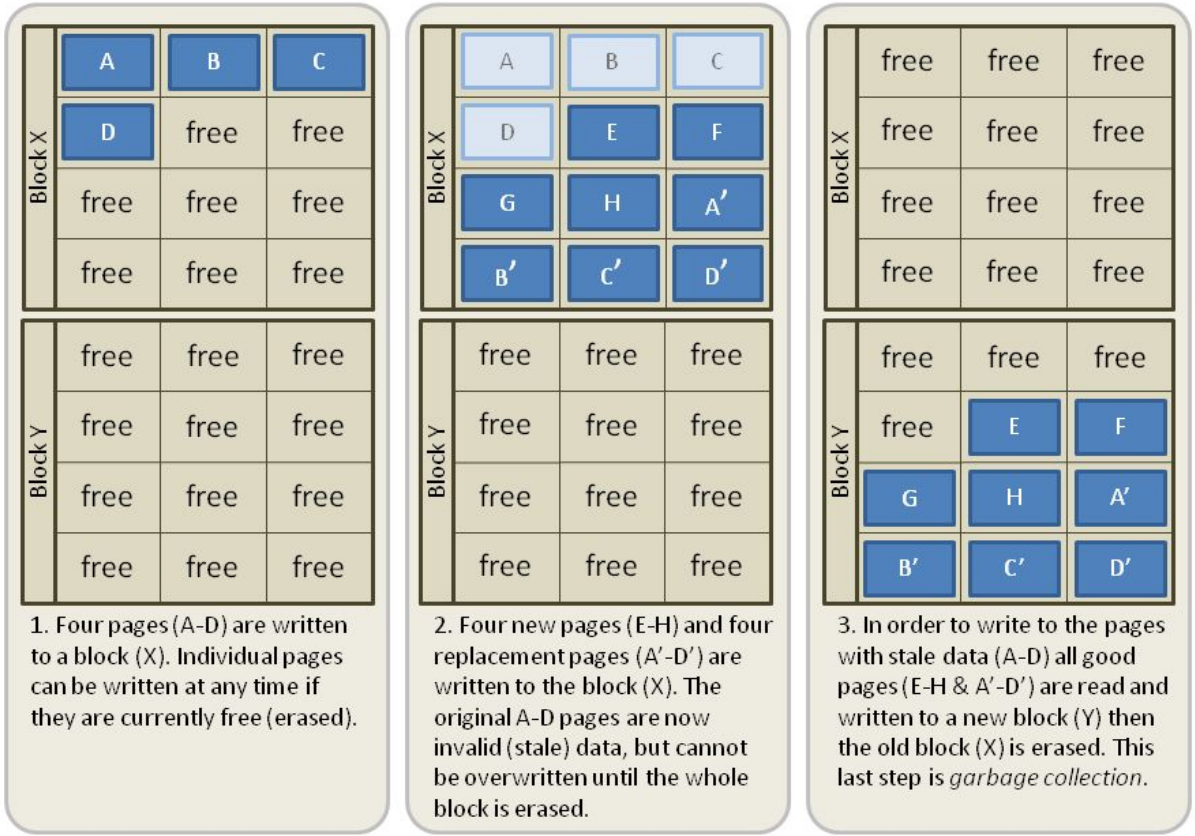


Bits-per-cell

	Bits-per-cell	States	Read Time
SLC	1	1	25us
MLC	2	4	50us
TLC	3	8	100us
MLC	4	16	200us



Garbage Collection



Trim / Discard

Attempting to Improve Discard Performance | Ceph Days NYC 2024

Wear Level Management

Recovery Time

USE Method

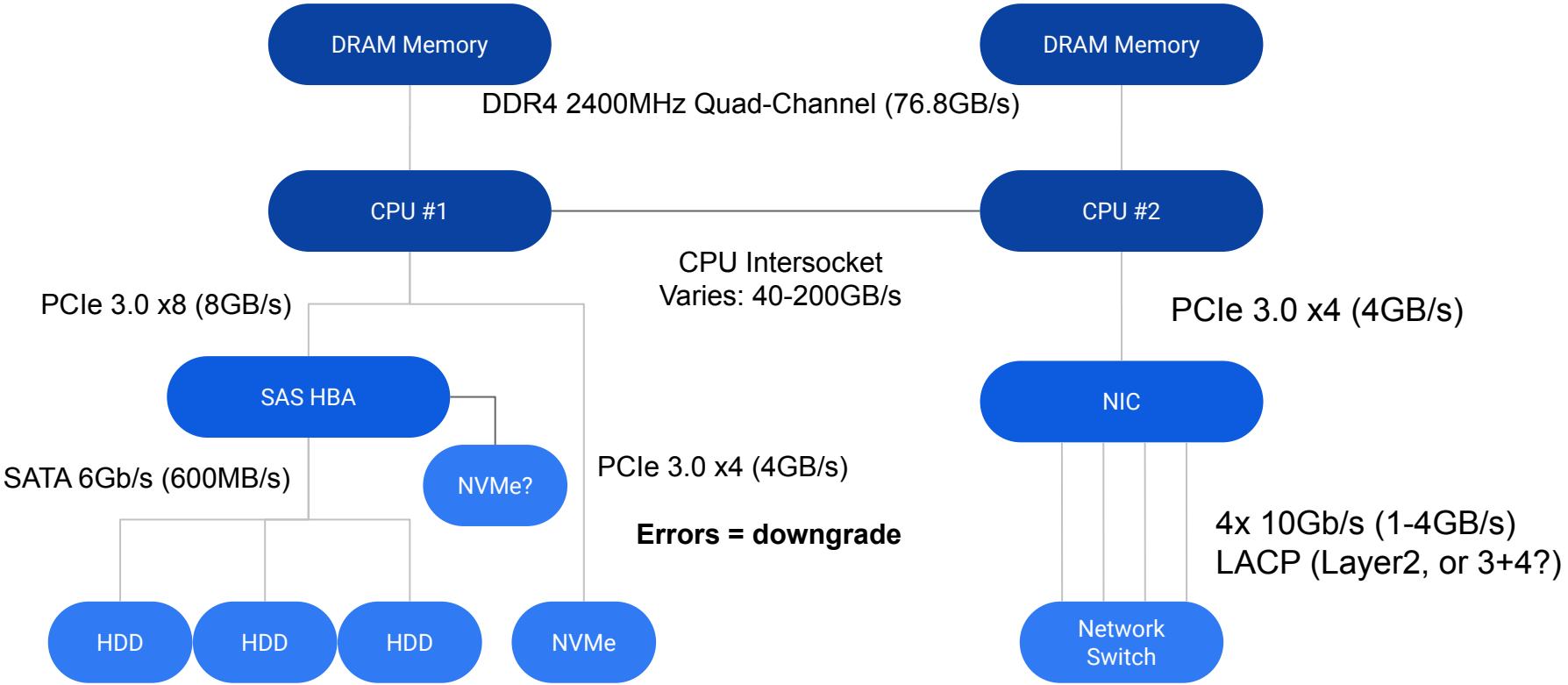
Utilisation

Saturation

Error Counters

More: https://www.brendangregg.com/usemethod.html

Hardware Bottlenecks



Less known USE Tools

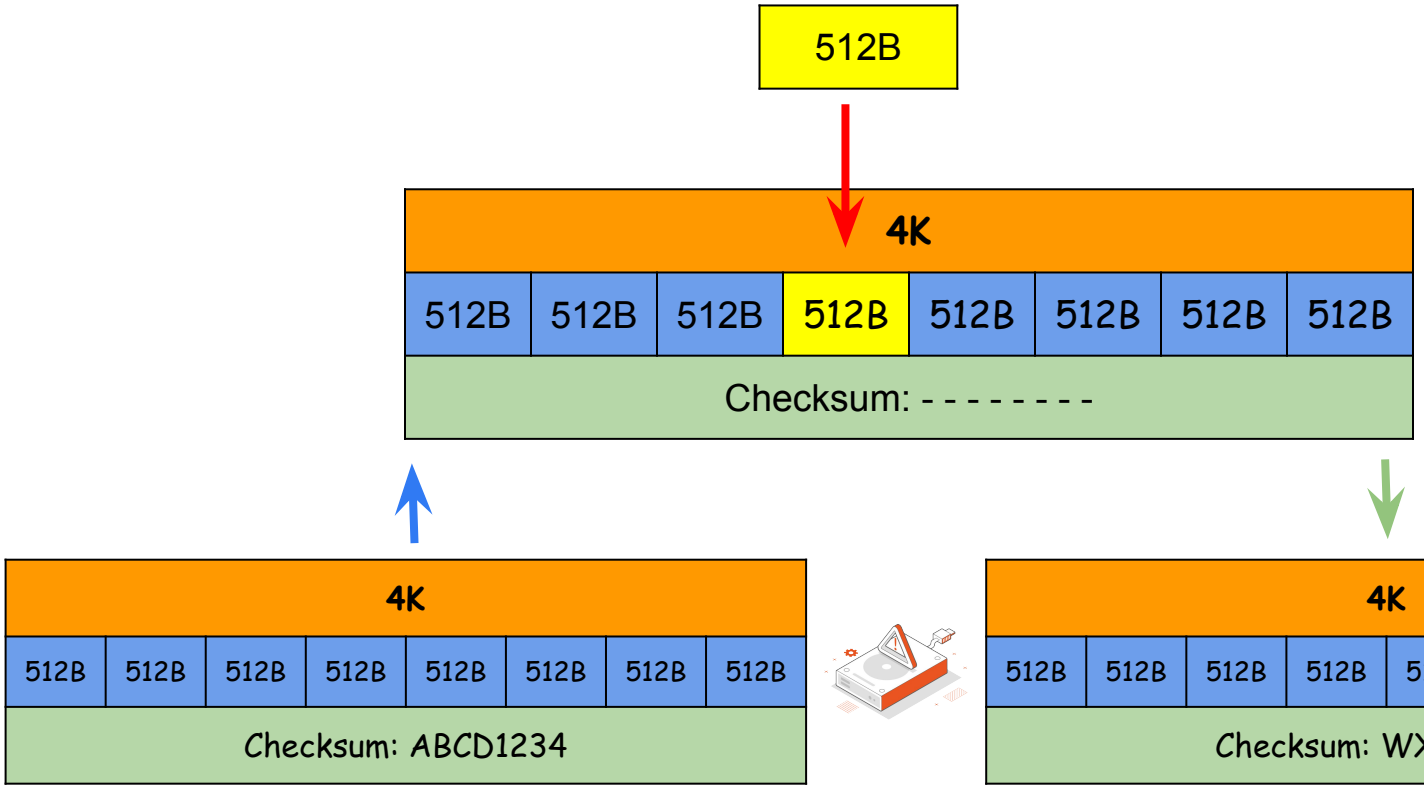
Find new tools
https://www.brendangregg.com/USEmethod/use-rosetta.html

sosreport

Physical Hardware Layout / Link Speeds
lstopo --of ascii
lspci -vv

Network Error Counters
netstat -s
ethtool -S

Read-Modify-Write



HDD
512b
512e
4Kn

ZFS
ashift: 4KiB
recordsize: 128KiB,
volblocksize: 16KiB

Ceph
checksum: 4K
compression: 16-64K

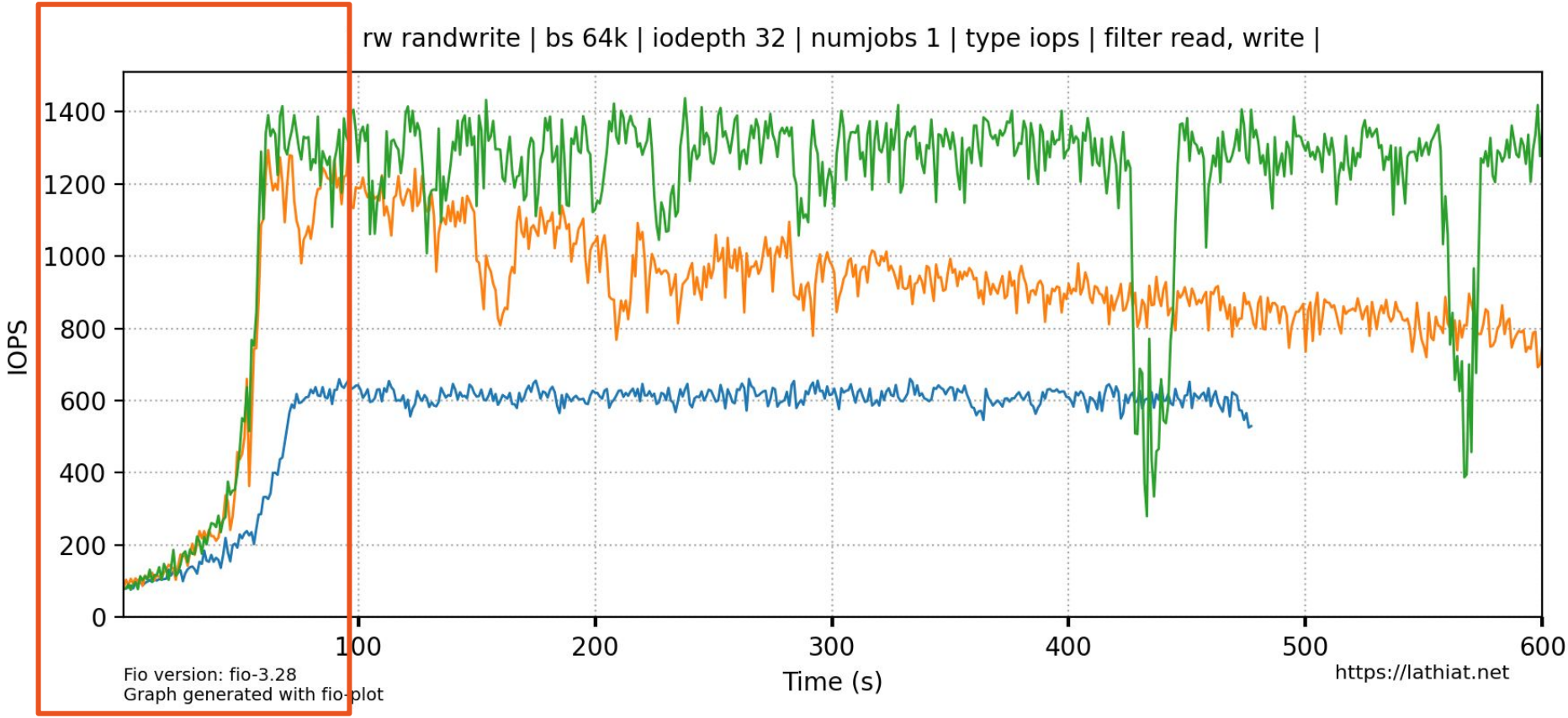
Read-Modify-Write: Non-4K aligned writes

- Non-4K aligned writes
  - Linux Guests: Almost always 4K-aligned
  - Windows Guests: Will write 512-byte aligned by default
  - Fixing I/O performance for Windows guests in OpenStack Ceph clouds: https://www.youtube.com/watch?v=_vfGcsvnn6U
- Mis-informed etcd fio test
  - https://www.ibm.com/cloud/blog/using-fio-to-tell-whether-your-storage-is-fast-enough-for-etcd (now removed)
  - `fio --rw=write --ioengine=sync --fdatasync=1 --size=22m` **`--bs=2300`** `--name=mytest1`

Sparse Allocation / Thin Provisioning

RBD	16MB			
	4MB	4MB	4MB	4MB
RADOS	disk1.1	disk1.2	disk1.3	disk1.4

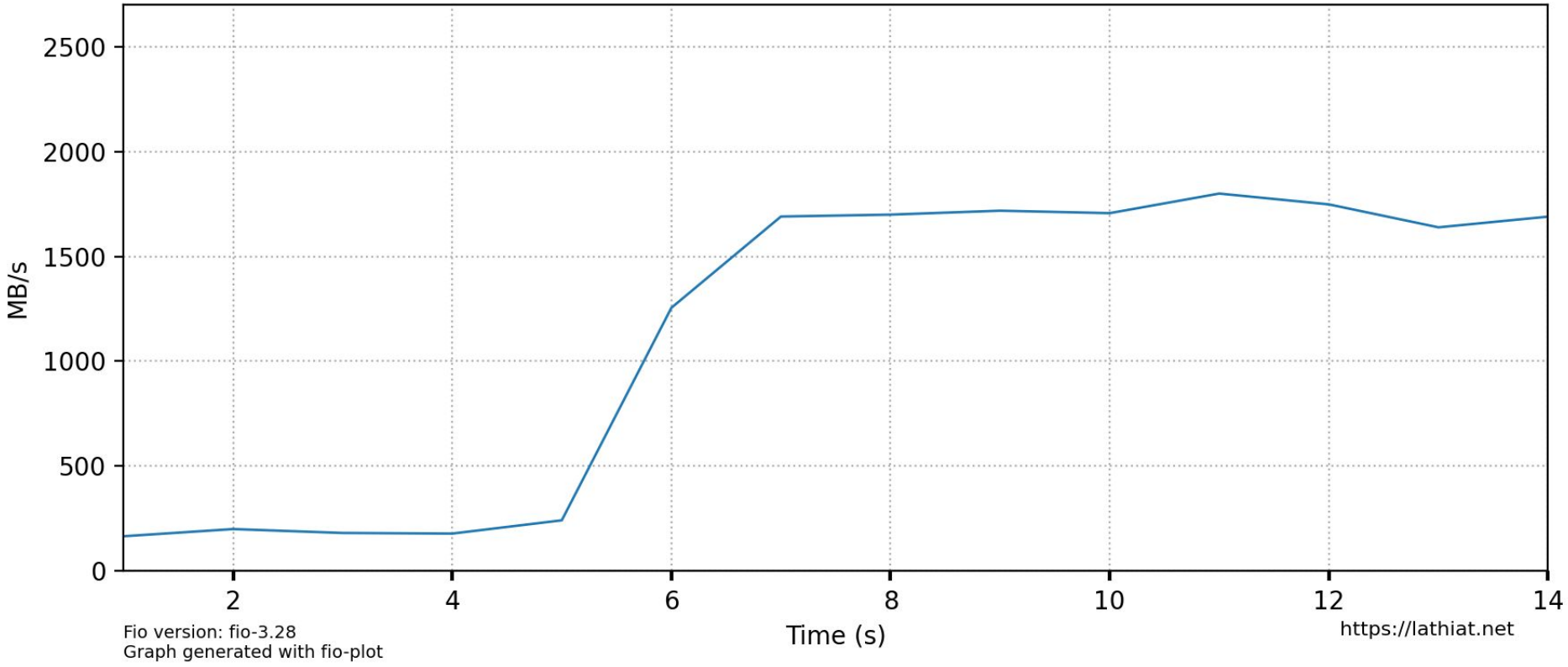
RBD Thin First Write, 12x HDD Ceph Cluster



name	rw	type	qd	nj	mean	std%	P99.99
test1/64k	write	iops	32	1	549	28.57	660
test2/3k	write	iops	32	1	880	28.07	1304
test2/4k	write	iops	32	1	1153	29.91	1438

RBD Thin Read, 12x HDD Ceph Cluster

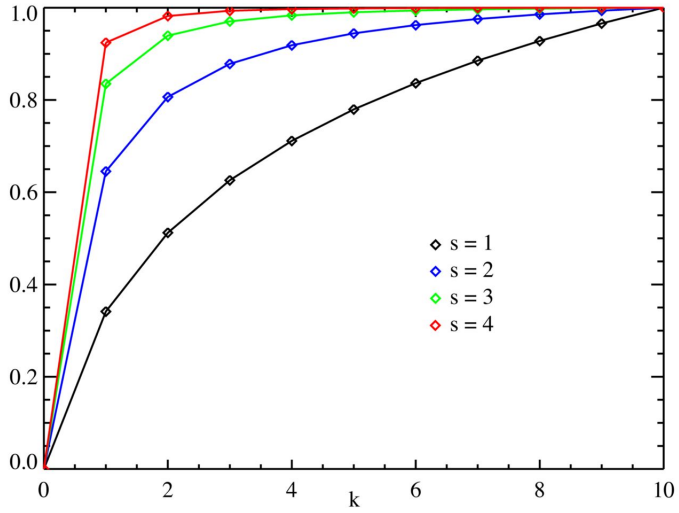
| rw read | bs 64k | iodepth 32 | numjobs 1 | type bw | filter read, write |



name	rw	type	qd	nj	mean	std%	P99.99
test1/64k	read	bw	32	1	1137	62.8	1800

Working Set Size

- Write a realistic total amount of data to each drive/the entire cluster
- Issue reads and writes to a realistic percentage of the total storage
  - fio --io_size
- ZIPF Distribution
  - fio --random-distribution=zipf:1.2



"Zipf distribution CMF" https://commons.wikimedia.org/wiki/File:Zipf_distribution_CMF.png CC-BY-SA-3.0

Why? Caching

- Memory caching
  - Metadata: indirect block allocation maps
  - Data itself

- SSD Caching (bcache, lvmcache, storage tiering, etc)
  - May not trigger writeback
  - May never read/write to the actual backing HDD

Why? Deep Scrub

- Default = Once a week
- 8TB / 1 week = 13MB/s
  - Significant background load

Benchmarking Workload

- Tool Selection
  - **Bad:** hdparm, dd, cp, rados bench
  - **Good**: fio (with the right config), use --direct
- Parallelism
  - Parallel I/O submission
  - Multiple VMs on Multiple Hosts
    - Multiple Client IPs, MACs
    - Multiple Network Links
- Filesystem vs Raw Block
  - ext4 inode table lazy initialization
  - Metadata Overheads
  - Lock Contention
  - fstrim runs once a week, undoes a thick provision

Ceph - Dynamic Resizing

- PG Autoscaler
- PG Balancer
- RadosGW Bucket Sharding

Tuning Guides

Almost always trade off **throughput**

for **latency** and **crash safety**

Tuning

Don't assume changing a config option **dynamically** actually works

net.ipv4.{tcp_rmem,tcp_wmem}
Requires TCP to reconnect

Simple guidelines

1. Benchmark raw block device instead of a filesystem
2. Pre-condition/pre-write the entire virtual disk size at the start
3. Fill the underlying storage to a reasonable percentage (60-80%)
4. Working Set Size
    a. Avoid writing to only a small portion of a virtual disk
    b. Write to the disk at random or use a distribution like zipf
    c. Benchmark a range of sizes and compare the effect
5. Benchmark duration
    a. Benchmark long enough to push any SSDs past the bi-modal speed cut-off
6. Switch back and forth between the two compared options
7. Research and determine the limiting factor (USE Method)
    a. Improve it, go again

Why do we need to do all of these things when we **don't** do them for the **production** workload?


Canonical
Ubuntu
SILVER
Canonical

Questions

lathi.at/talks
twitter.com/lathiat
@lathiat@fosstodon.org
linkedin.com/in/lathiat

trent.lloyd@canonical.com