Using Software to Reduce High Tail Latencies on SSDs

Kapil Karkra

Principal Engineer

Intel Non-volatile Memory Solutions Group (NSG)

Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

No computer system can be absolutely secure.

Performance results are based on testing as of July 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase.

Tail Latency Problem At Scale

Assume one in a 1000 queries to an SSD will result in a longer (tail) latency event.

At 1 SSD it is

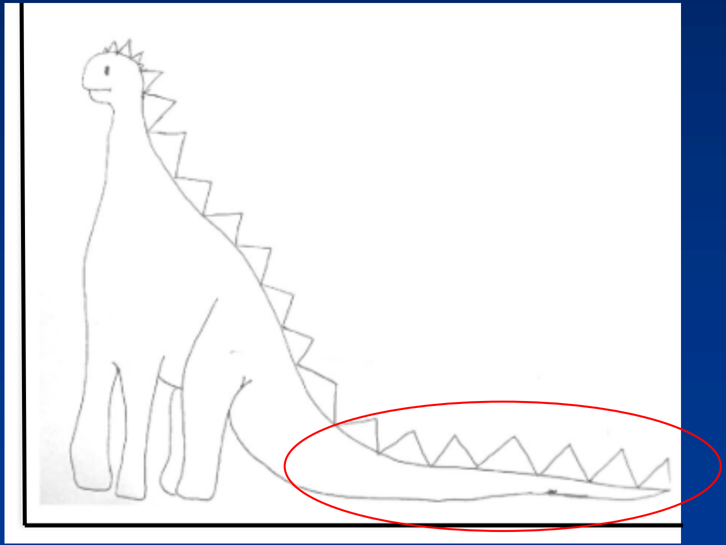
$1 - (999/1000)^1 = 0.1\%$

At 100 SSDs it is 10%

- $1 - (999/1000)^{100} = 10\%$

At 1000 SSDs it is 63%

- $1 - (999/1000)^{1000} =$ **63%**



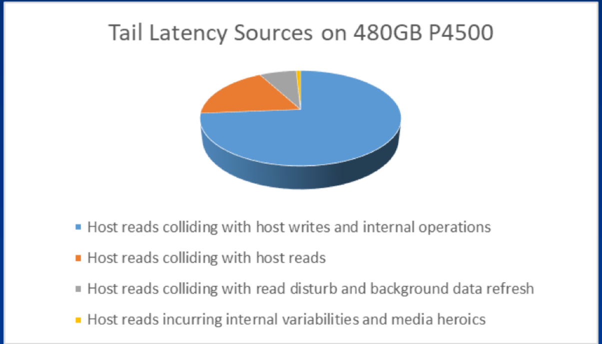
What causes tail latency problem in SSDs?

Read Tail Latency in SSDs

The main sources of high tail latency are:

1. Host reads colliding with host writes (includes garbage collection)
2. Host reads colliding with other host reads
3. Host reads colliding with asynchronous background operations (includes Read Disturb (RD) and Background Data Refresh (BDR))

	480 GB SSD, rd qd=64 wr qd=64			480GB SSD, rd qd=1, wr qd=1		
% of IOs	rw mix	rd only	rd only no BDR/RD	rw mix	rd only	rd only no BDR/RD
99	9.152	3.184	3.12	3.856	0.181	0.181
99.5	10.944	3.44	3.344	4.08	0.183	0.183
99.9	15.296	3.984	3.76	5.344	0.183	0.183
99.95	17.024	4.32	3.888	7.904	0.185	0.185
99.99	21.632	5.536	4.192	13.376	1.736	0.185



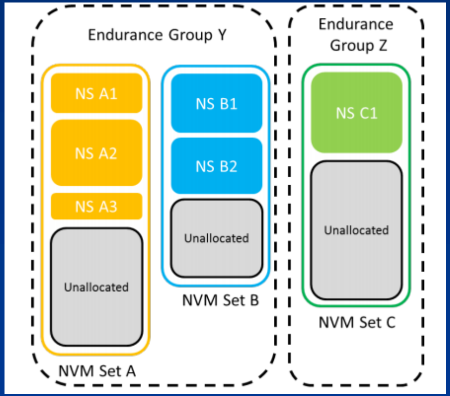
Configuration: Cliffdale P4500 480GB, custom firmware suppressing BDR and RD
Benchmark: FIO, 12h, 4k random reads, 4k random writes (a) qd=64 (b) qd=1

(1) Separate host reads from host writes to avoid read on write collisions
(2) Minimize garbage collection and other background activities
(3) Avoid high queue depth reads and writes to avoid too many collisions and resource utilization

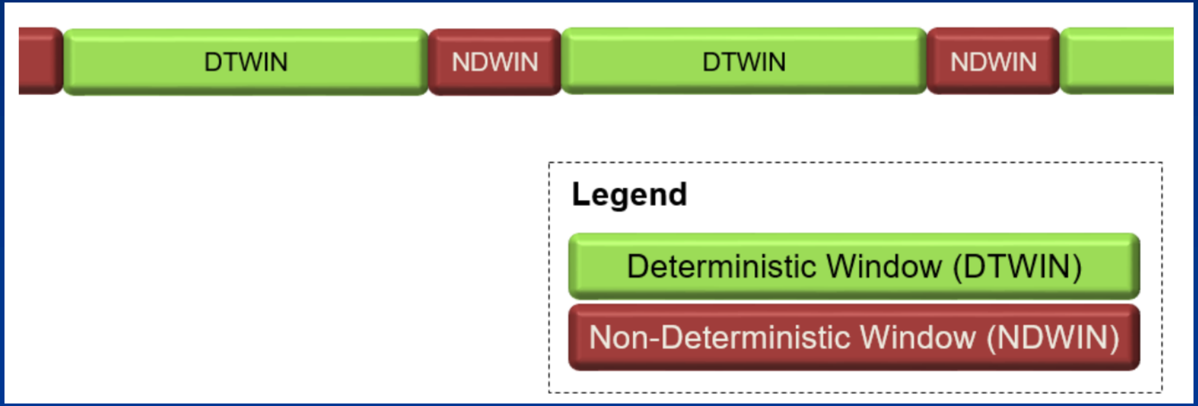
I/O Determinism SSD Capabilities Overview

Key IO determinism capabilities

1. NVM Sets and Endurance Groups
2. Deterministic/Non Deterministic (D/ND) Windows



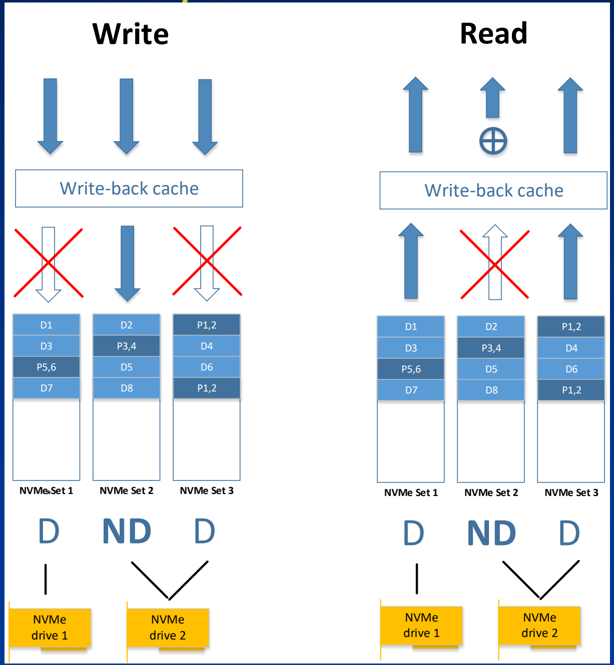
1. NVM Sets and Endurance Groups



2. D/ND Windows

How can software take advantage of these capabilities to achieve I/O Determinism?

Software Approach #1: Solving tail latency using data redundancy, a D/ND I/O scheduler, and a write-back cache



	KV/LSM workload, 2TB Volume (3 1TB Sets on P4600)			
	Desired read latency (ms)	Passthrough set (rw mix)	RAID5 style redundancy (rd only w/IO scheduling)	RAID5 style redundancy (rd only w/D/ND IO scheduling)
p99	3	4.490	2.442	0.181
p99.99	6.5	9.896	4.490	0.338
P99.9999	11	13.042	5.997	0.868
Rd IOPS (kIOPS)	8.75	27	27	27
Wr BW (MB/s)	72	250	250	250

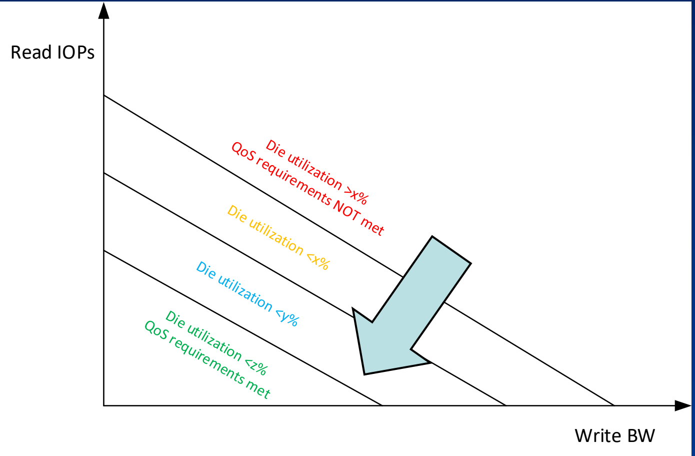
Configuration: Cliffdale P4600 1TB, 3 set RAID5 vs. P4600 passthrough, custom firmware supporting D/ND
Benchmark: FIO, 12h, 4k random reads, qd=10, two threads of 25% random write bursts; thread1: 512k writes with a thinktime of 2.56s, thread2: 512k writes with thinktime of 1.28s

The approach achieves (a) read write separation (b) reduces read on read collisions (c) eliminates the read-on-background-write collisions

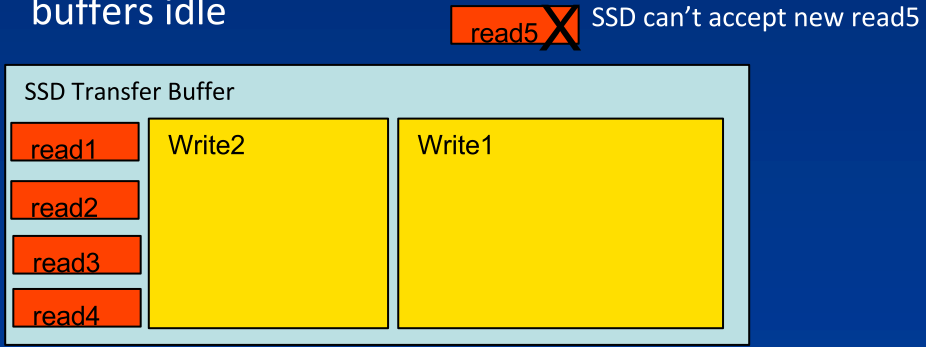
What if you don't like the capacity tradeoff or cost of an additional write buffer and don't have the I/O determinism capabilities on your SSD?

SSD Resource Utilization and Tail Latency

Keep the NAND dies idle



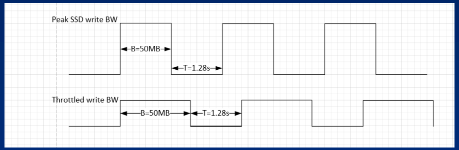
Keep writes small for the transfer buffers idle

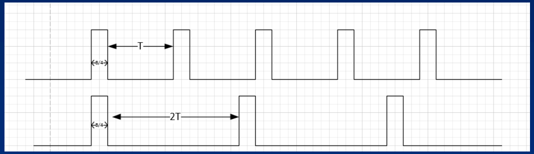


The more the resources are utilized, the higher the latency spike

Software Approach #2: I/O Shaping

1. Throttling/Trickling writes: Die Idle



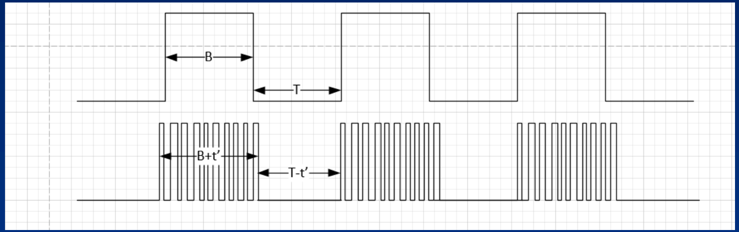


	Intel P4510, 1TB Sets, throttling				
	Desired	Sets	Sets	Sets	Sets
p99	3	5.088	4.896	3.376	1.4
p99.9	5	6.88	6.688	5.024	3.056
p99.99	7	14.144	12.608	7.584	4.576
Wr BW (MB/s)		67	57	41	20

Configuration: Cliffdale P4510 1TB Sets
Benchmark: FIO, 12h, 4k random reads, qd=10, two threads of 25% random write bursts; thread1: 512k writes with a thinktime of 2.56s, thread2: 512k writes with thinktime of 1.28s

This approach trades off write bandwidth for read determinism

2. Chopping writes: Transfer Buffer efficiency



	Intel P4510, Bursts, 2TB Sets, Chopping			
	Sets	Sets	Sets	Sets
	T=1.28, B=160, BS=512k	T=1.28, B=160, BS=128k	T=1.28, B=640, BS=32k	T=1.28, B=5120, BS=4k
p70	0.111	0.112	0.111	0.111
p99	2.544	1.544	1.464	1.48
p99.9	6.752	4.384	4.192	3.984
p99.99	17.536	11.072	10.56	5.28
Wr BW (MB/s)	15	15.43	15.38	14.42

Configuration: Cliffdale P4510 2TB Sets
Benchmark: FIO, 12h, 4k random reads, qd=10, two threads of 25% random write bursts; thread1: writes with a thinktime of 2.56s, thread2: writes with thinktime of 1.28s, varying block sizes from 512k down to 4k

Workload Characteristics and Tail Latency

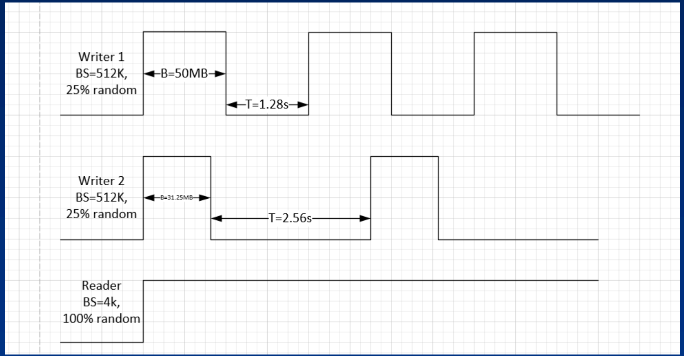
Flash Memory Summit

1. Workloads that mix different lifetime data (e.g., mixing frequently updated data with static data) increase garbage collection and thus tail latency

Workload	Different Data Lifetime Streams: 3 Sequential Streams and 1 Random Stream; Large velocity delta among sequential streams, random partition only 5%, uniform random
Data Lifetime classifier	LBA Ranges different for three sequential workers with 1x, 7x, and 17x velocity difference (QD=4) while a single random stream of 1x velocity (QD=16)
WAF improvement	60% (2.9→1.2)
Performance Improvement	3.5x
Read QoS (P9999)	34%
Device	P4500

NAND drive: P4500 Prototype 480GB, firmware modified to support 4 streams
Benchmark: FIO, 3 sequential write streams (different velocity), 1 random write stream, 12h

2. Write bursts in a workload also cause high tail latency



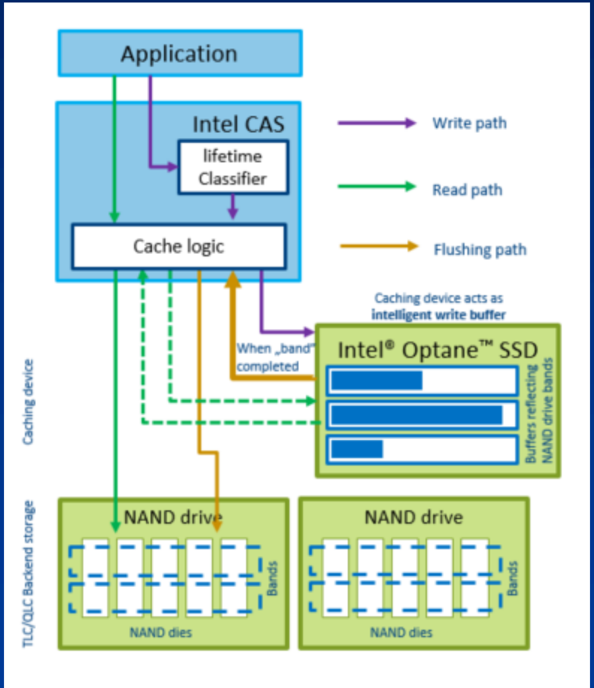
	Intel P4510 w/ 1TB Sets		
	Desired	bursts	no bursts
p70	0.8	0.173	0.197
p99	3	5.088	1.512
p99.9	5	6.88	2.544
p99.99	7	14.144	3.344
Rd IOPS (kIOPS)	7	7	7
Wr BW (MB/s)	77	67	82

Configuration: Cliffdale P4510 1TB Sets
Benchmark: FIO, 12h, 4k random reads, qd=10, two threads of 25% random write bursts; thread1: 512k writes with a thinktime of 2.56s, thread2: 512k writes with thinktime of 1.28s; no burst run is without the thinktime

Can we shape workload to (a) avoid mixing different lifetime data streams (b) eliminate bursts from the workload?

intel

Software Approach #3: Intel® Optane™ SSD Write Buffer



Solution Details:

- All writes from application go to Intel® Optane™ SSD buffer
- Data in buffer partitioned, based on lifetime classifier
- Flushing of data performed in buckets of size equal to NAND drive erase unit size
  - Only one bucket at a time
  - Erase unit filled with data with same stream (e.g. same velocity)
- Flushing throttle – TB algorithm to prevent write bursts on NAND drive

	Baseline	Intel® Optane™ SSD Buffer
Host writes	223,395	223,623
NAND writes	549,940	280,223
WAF	2.46	1.25

NAND drive: Cliffdale P4500 1TiB, Model: INTEL SSDPE2KX010T7
Optane drive: P4800X 280GiB, Model: INTEL SSDPED1D280GA
Benchmark: FIO, 3 sequential write streams (different velocity), 1 random write stream, 12h

		Baseline	Intel® Optane™ SSD Buffer
Latency (usec)	50	116	180
	70	265	562
	90	2,737	2,147
	99	11,600	4,228
	99.9	19,530	6,587
	99.99	25,560	8,356
	99.999	28,443	9,503
	99.9999	36,439	11,338
	99.99999	43,779	12,911
BW	RD (MiB/s)	27	27
	WR (MiB/s)	56	76

NAND drive: Cliffdale P4500 1TiB, Model: INTEL SSDPE2KX010T7
Optane drive: P4800X 280GiB, Model: INTEL SSDPED1D280GA
Benchmark: FIO, Facebook workload (4K_L2R6DWPD.FIO)
Optane buffer flushing throttle: 80MB/s

The approach achieves (a) write amp reduction through lifetime classification (b) eliminates bursts (c) Intel® Optane™ Technology can absorb multiple reads and writes to improve tail latency

Conclusion

- Software techniques (redundancy, IO shaping, and caching/buffering) built on top of hardware capabilities (Intel® Optane™ SSD and IO Determinism capable SSDs) are a powerful tool to cut the tail