Understanding Intrinsic Characteristics and System Implications of Flash memory based Solid State Drives

[SIGMETRICS 2009]
Feng Chen, David A. Koufaty, Xiaodong Zhang
Dept. of Computer Sci. & Eng., The Ohio State Univ.
System Tec. Lab., Intel Co.

Presented by Sangwhan Moon

Introduction: Flash Memory

- Flash Memory
  - Electronic Device: Fast and Reliable

Mica sensor

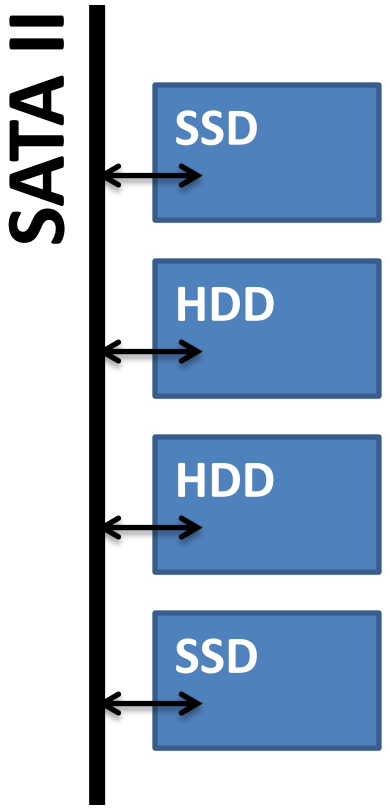
Intel SSD

EMC Server

Solid State Drives

- Interface compatibility
  - SSDs are accessed like HDD
  - → Hide SSD's characteristics from HDD
- SSD is not a just Flash memory
  - RAM Buffer
  - Controller
  - Firmware
    - For each vendor, model

Need to understand SSD



Understanding the SSD

1. Read Performance
2. Write Performance
3. Disk Cache
4. R/W Interference
5. Background ops. Interference
6. Workload Randomness
7. Fragmentation

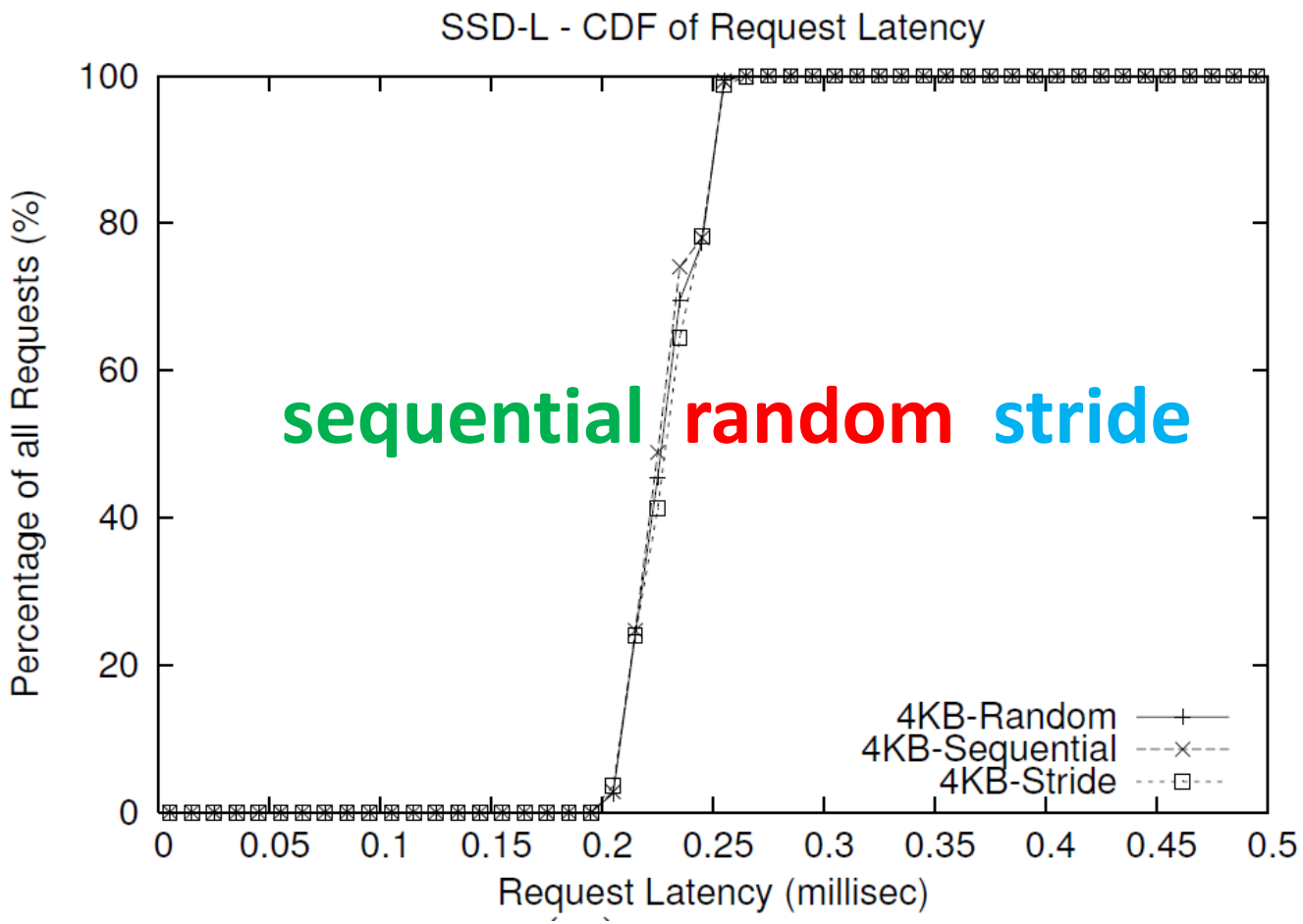
Evaluate Real SSD?

Experiment Environment

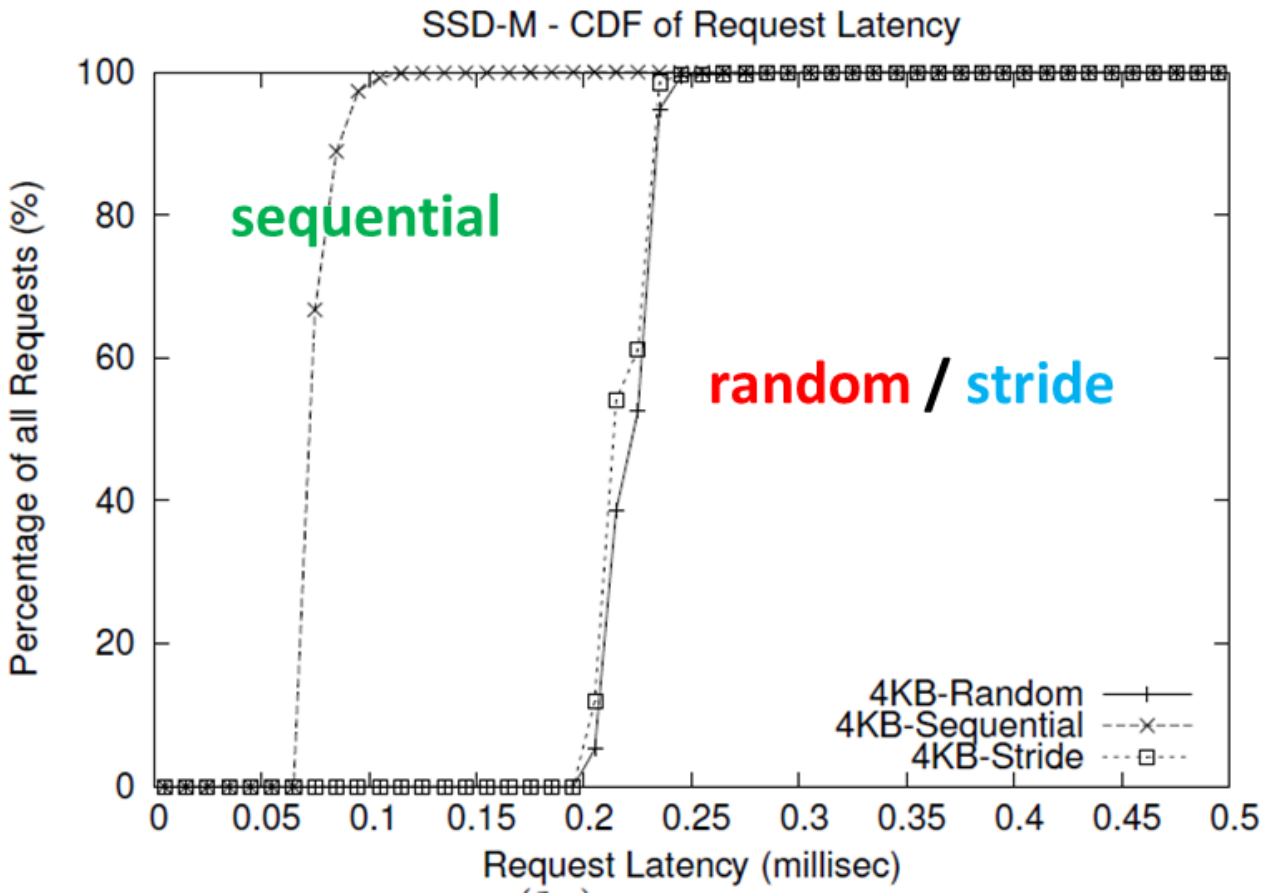
- 3 state-of-the-art SSDs
- Workload: Intel Open Storage Toolkit

	SSD-L (low-end)	SSD-M (middle-level)	SSD-H (high-end)
Capacity (GB)	30	80	32
Price ($/GB)	$5	$10	$25
Flash Memory	MLC	MLC	SLC
Page/Block Size (KB)	4/512	4/512	4/256
Read Latency (us)	60	50	25
Write Latency (us)	800	900	250
Erase Latency (us)	1500	3500	700

Experiment I: Read (1/2)



Experiment I: Read (2/2)



Questions

1. Does **read** on SSD have a uniform latency?
2. Would **random writes** be the worst case?
3. Is a **disk cache** effective for SSDs?
4. Do reads and writes **interfere** with each other?
5. Do **background operations** affect performance?
6. Would increasing **workload randomness** degrade performance?
7. How does **fragmentation** affect performance?

Q&A

Understanding Intrinsic Characteristics and System Implications of Flash memory based Solid State Drives

[SIGMETRICS 2009]
Feng Chen, David A. Koufaty, Xiaodong Zhang
Dept. of Computer Sci. & Eng., The Ohio State Univ.
System Tec. Lab., Intel Co.

Presented by Sangwhan Moon

Outline

- Introduction
- Motivation
- Questions
- Experiments
- Conclusion

Introduction: Flash Memory

- Flash Memory
  - Electronic Device: Fast and Reliable

Mica sensor

Intel SSD

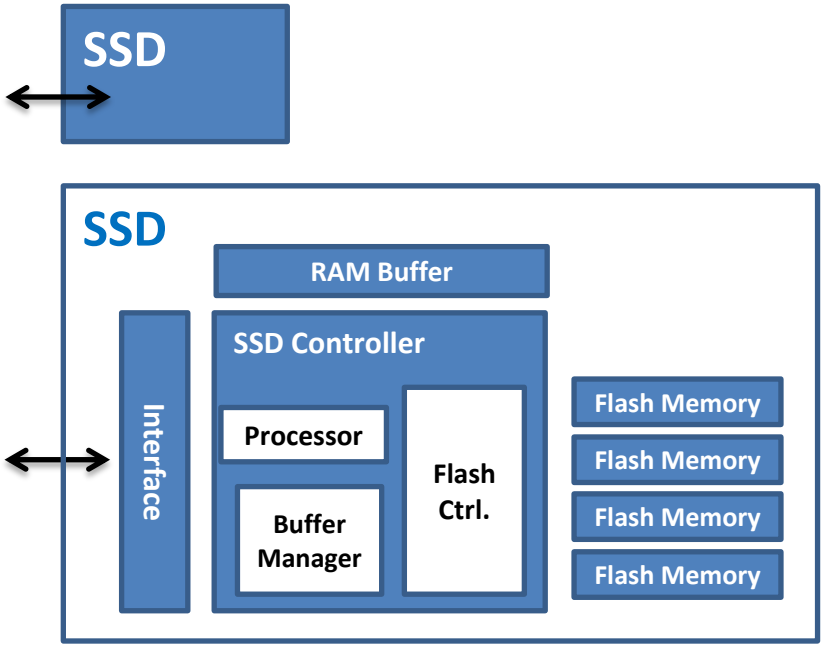
EMC Server

Introduction: Solid State Drive

- Interface
  - SATA, IDE, SCSI, etc

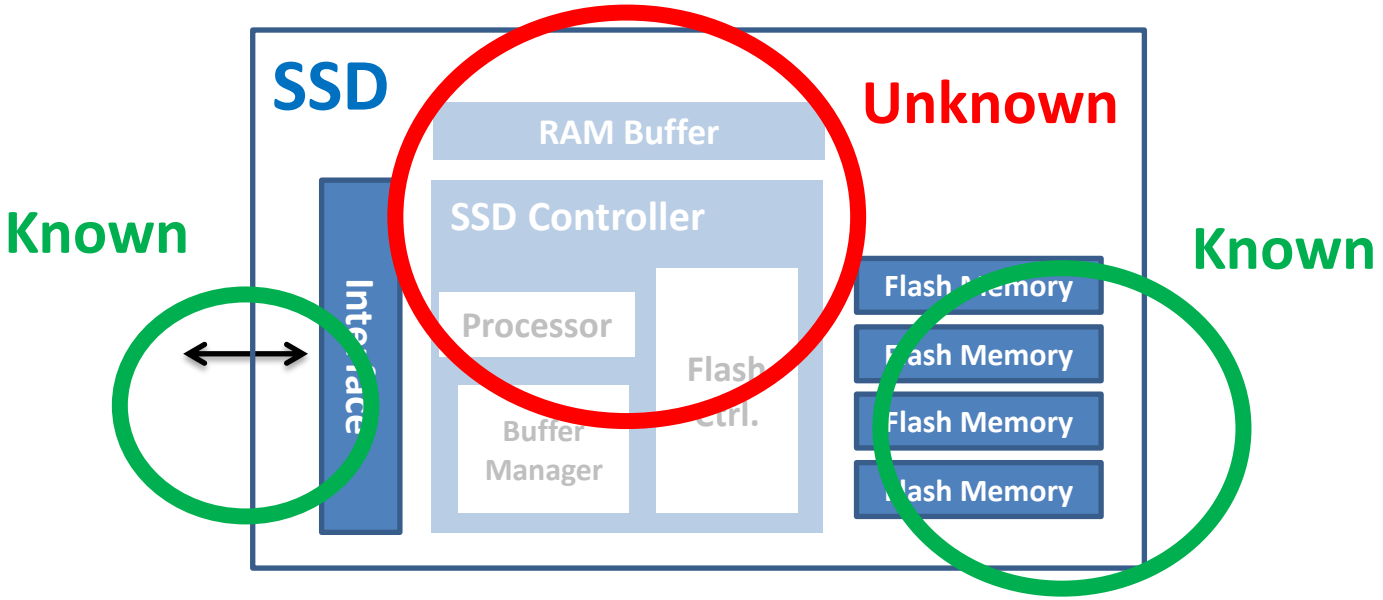
    Backward compatibility!

- Organization
  - Processor
  - Buffer Manager
  - Flash Controller
  - RAM Buffer
  - Flash Array



Motivation

- Common Interface w/ HDD
  - (+) backward compatibility
  - (–) Hidden internal characteristics



Motivation

- Some performance data are
  - not open
  - not match in practice
- Most studies are simulation based
- Only standard data from vendor are available
  - Bandwidth, Latency.

Performance evaluation is desirable to understand SSD

Questions

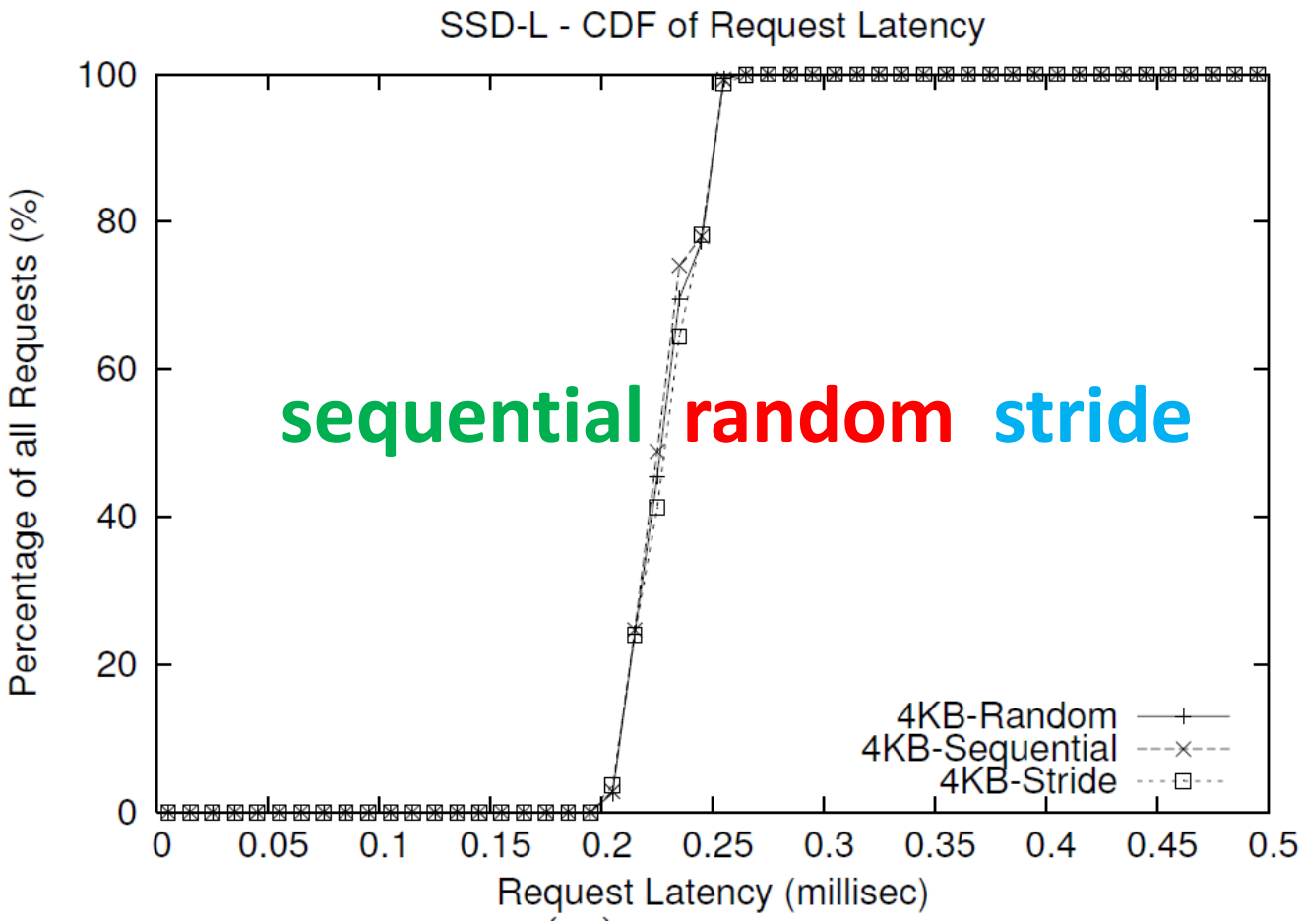
1. Does **read** on SSD have a uniform latency?
2. Would **random writes** be the worst case?
3. Is a **disk cache** effective for SSDs?
4. Do reads and writes **interfere** with each other?
5. Do **background operations** affect performance?
6. Would increasing **workload randomness** degrade performance?
7. How does **fragmentation** affect performance?

Experiment Environment

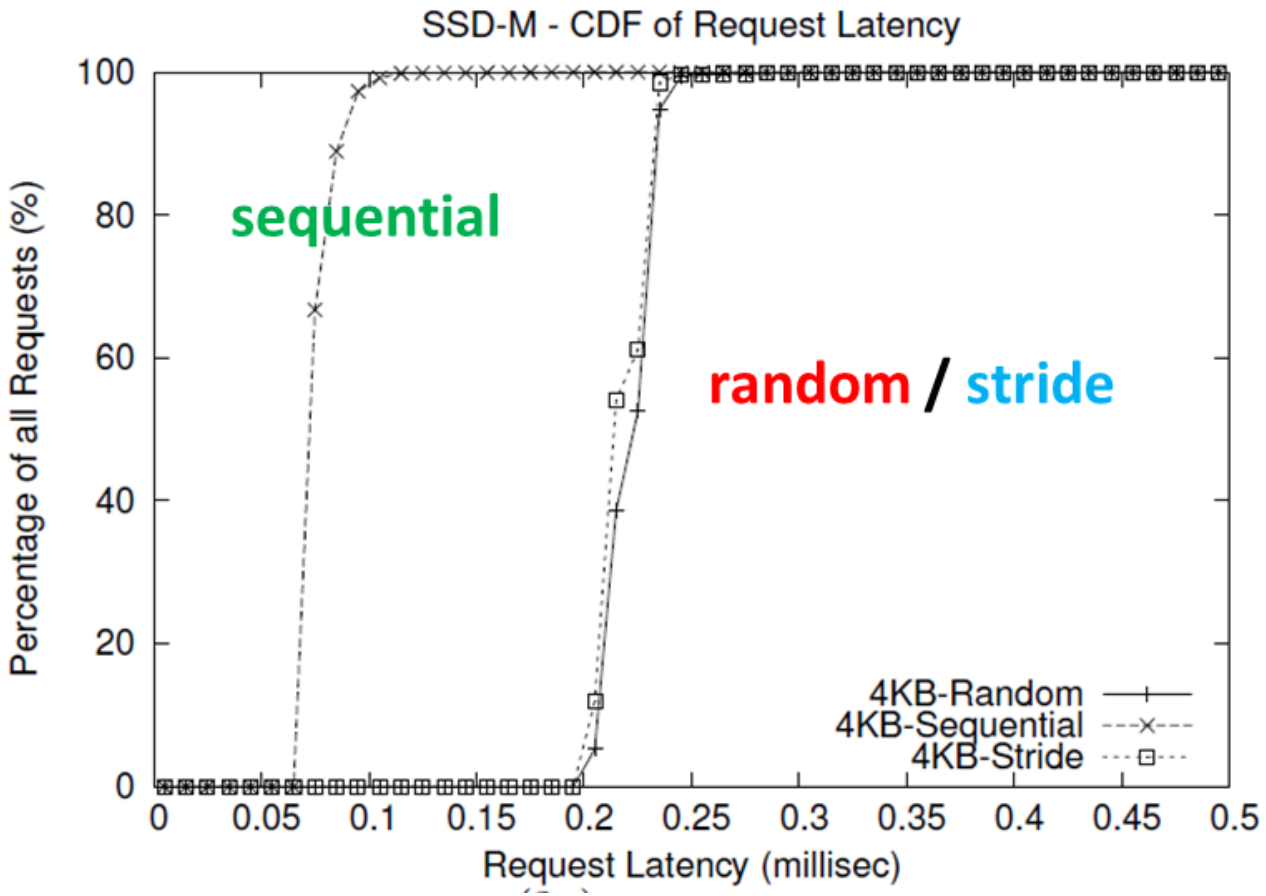
- 3 state-of-the-art SSDs
- Workload: Intel Open Storage Toolkit
  - Test various types of workloads
  - Directly accesses SSD as a row block device

	SSD-L (low-end)	SSD-M (middle-level)	SSD-H (high-end)
Capacity (GB)	30	80	32
Price ($/GB)	$5	$10	$25
Flash Memory	MLC	MLC	SLC
Page/Block Size (KB)	4/512	4/512	4/256
Read Latency (us)	60	50	25
Write Latency (us)	800	900	250
Erase Latency (us)	1500	3500	700

Experiment I: Read (1/3)



Experiment I: Read (2/3)

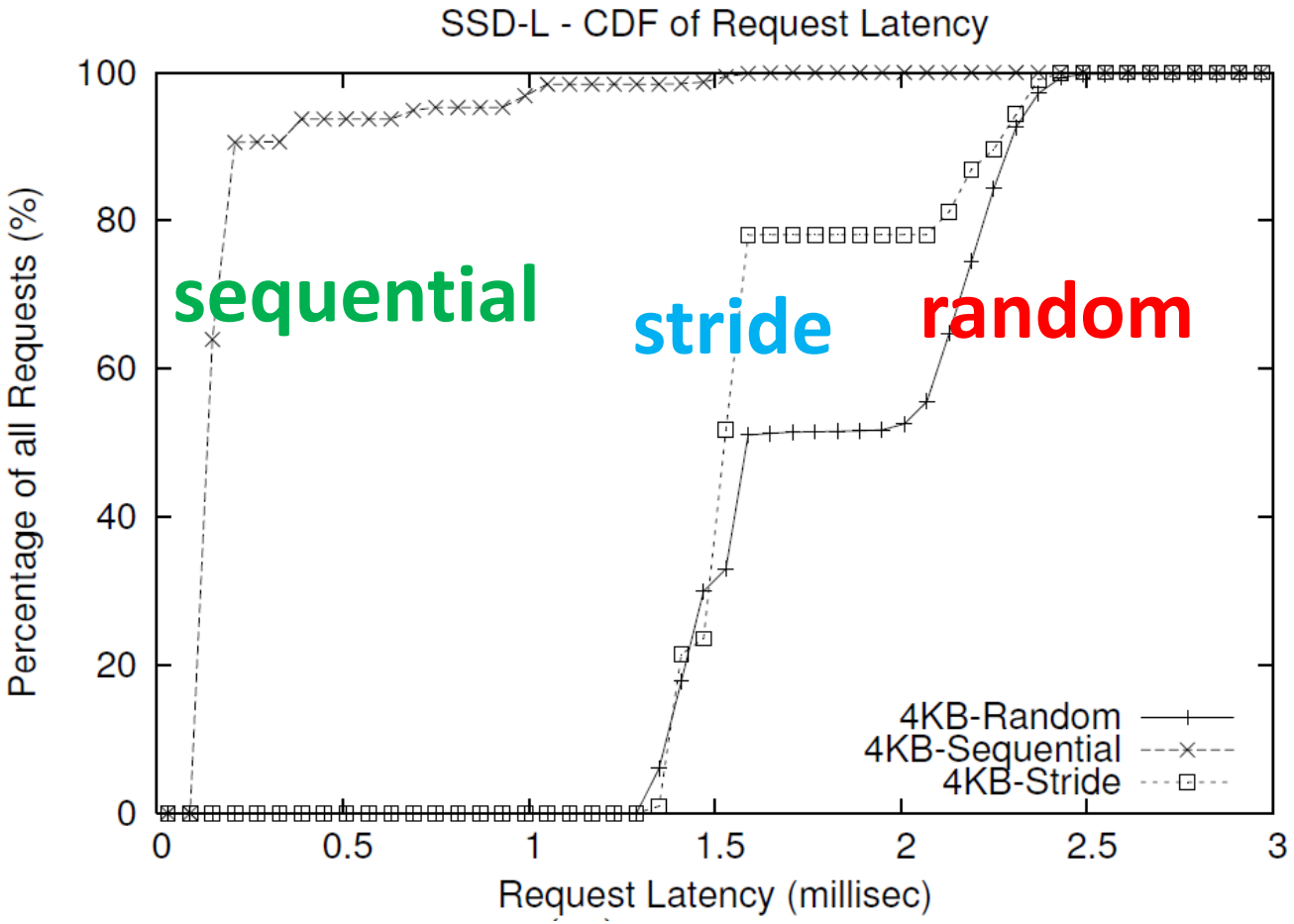


Experiment I: Read (3/3)

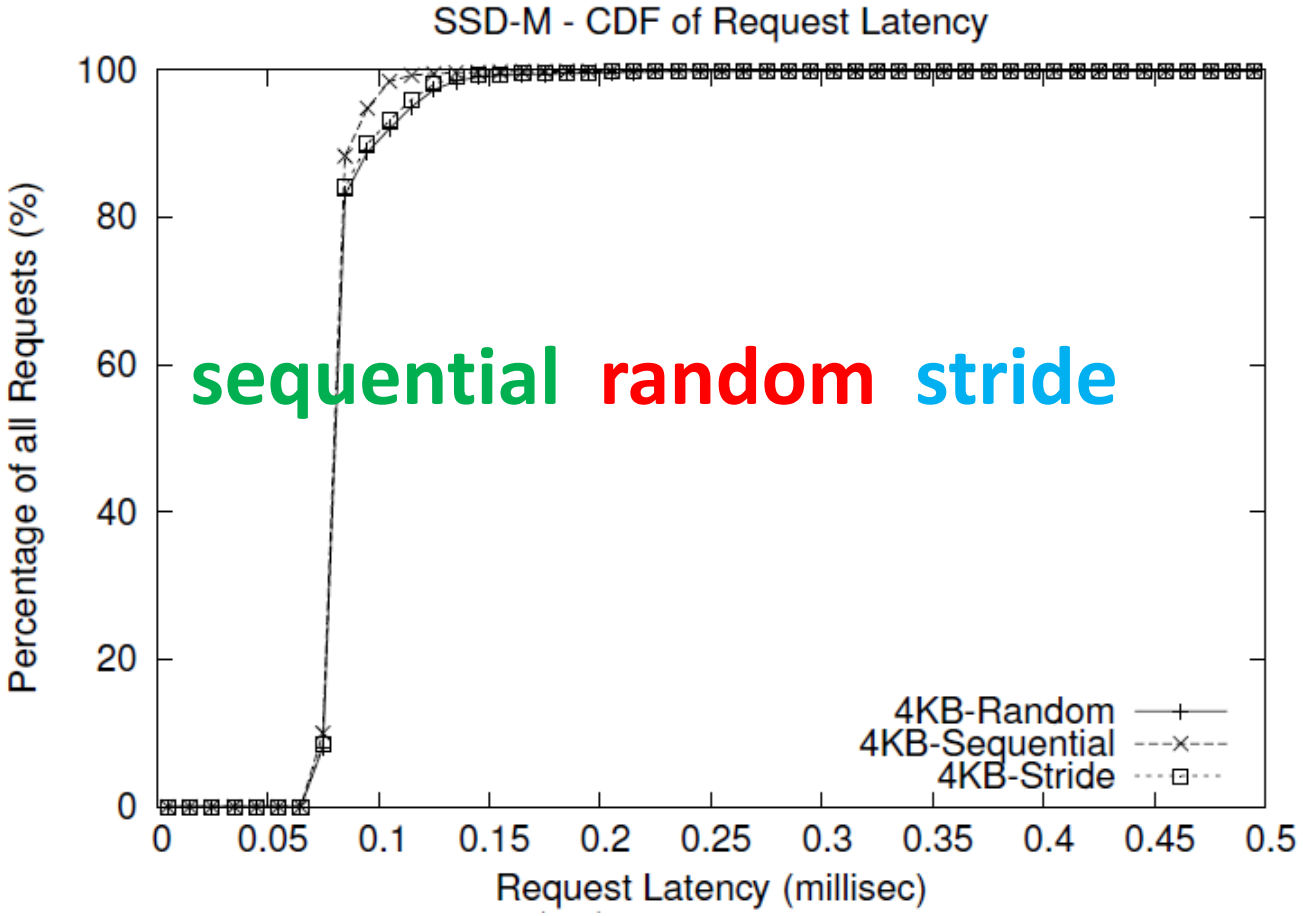
- ***Prefetching*** mechanism in SSD firmware

- The first 500 requests in sequential read
  - First 4 requests – 200~240us
    - A testing period (like in HDD)
  - After prefetching is initiated
    - 75-90us

Experiment II: Write (1/3)



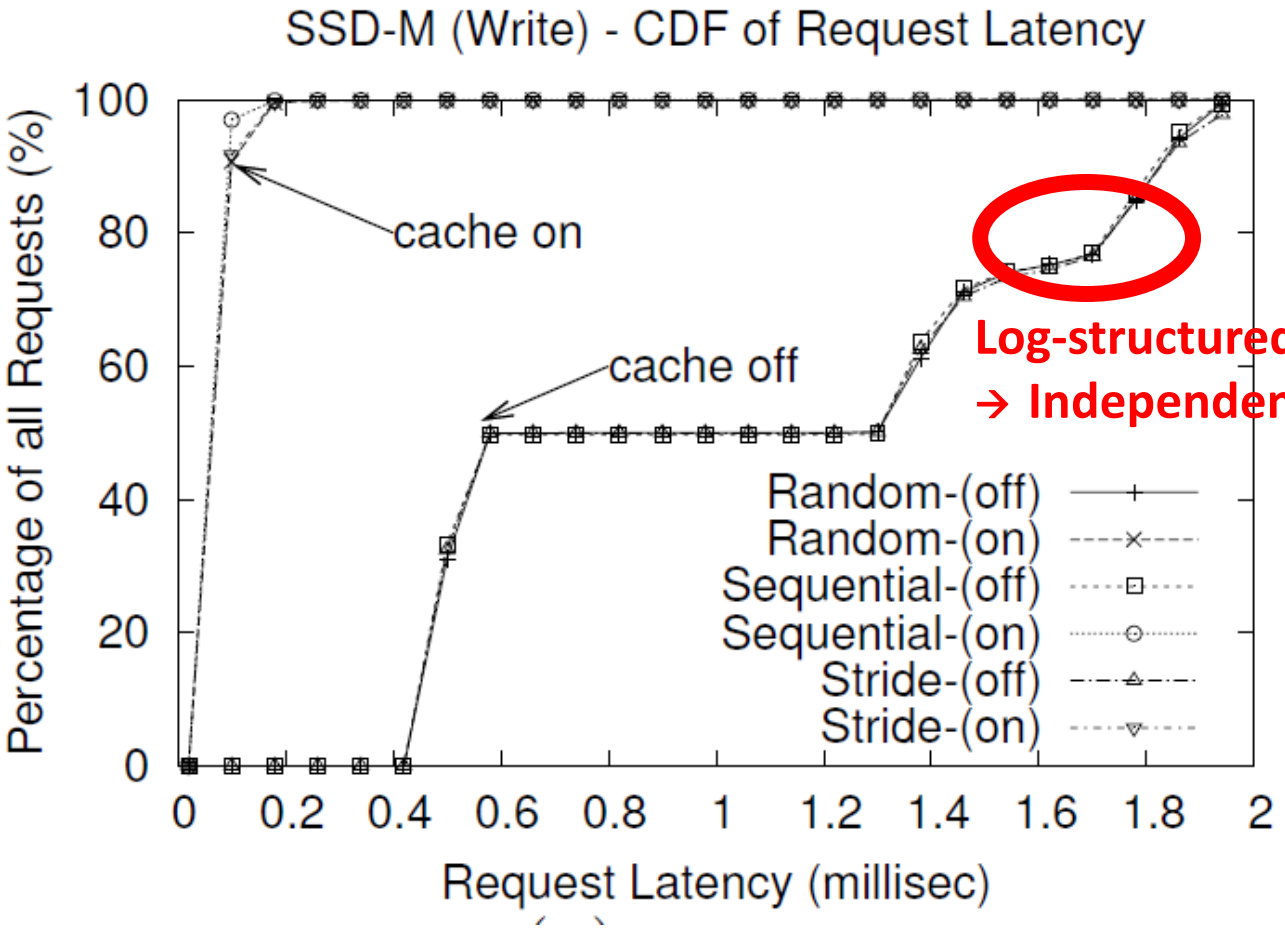
Experiment II: Write (2/3)



Experiment II: Write (3/3)

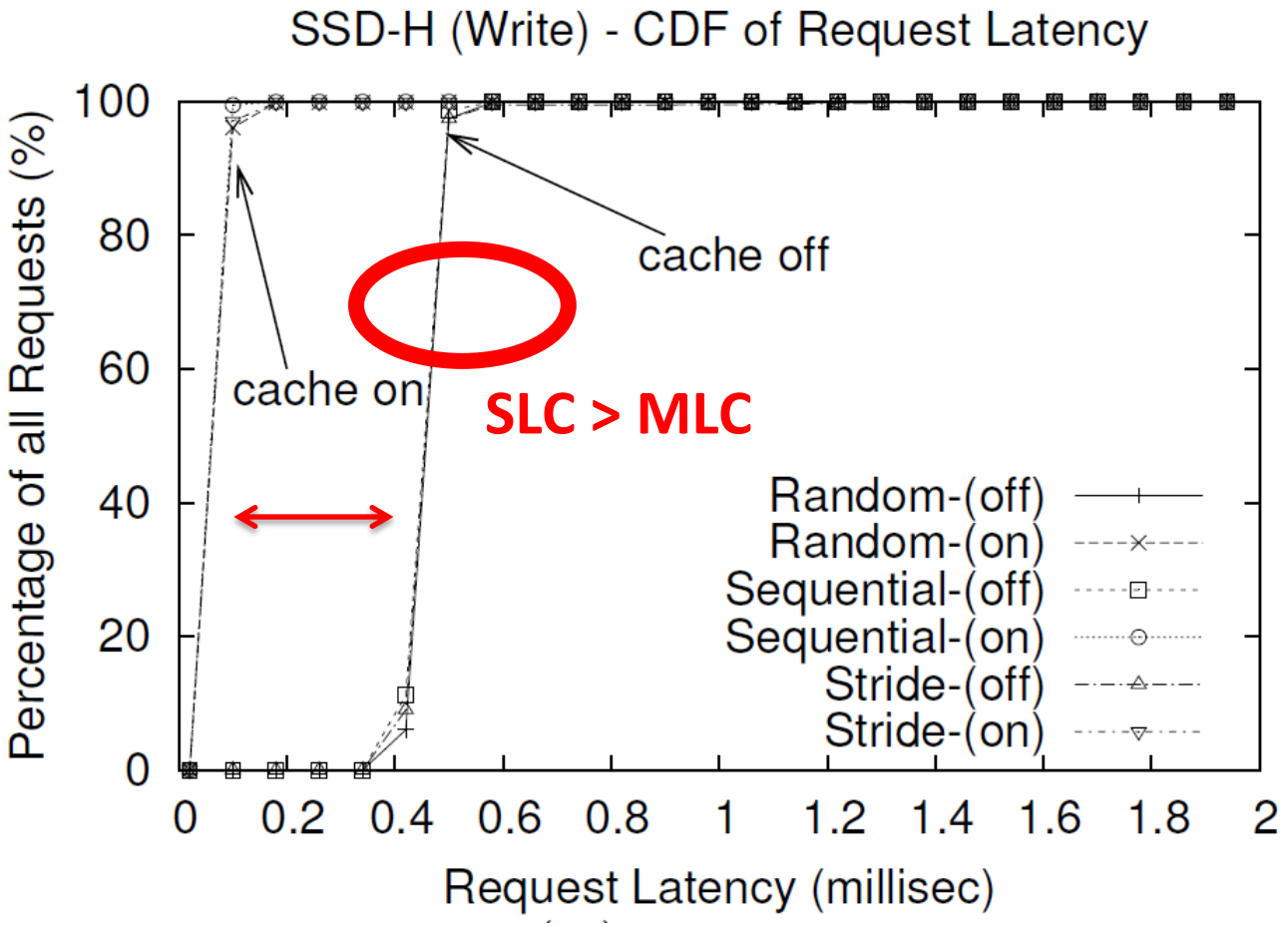
- SSD-L: On-device small buffer holds data.
  - **Sequential writes can be striped over multiple flash memory chips.**
  - Random writes cannot be striped.
- SSD-M,H: Buffer absolves most of requests.

Experiment III: Disk Cache (1/3)



Log-structured
→ Independent on pattern

Experiment III: Disk Cache (2/3)



Experiment III: Disk Cache (3/3)

- SSD-L's 16KB buffer cannot be disabled.
- SSD-M,H's performance severely degraded w/o disk cache.

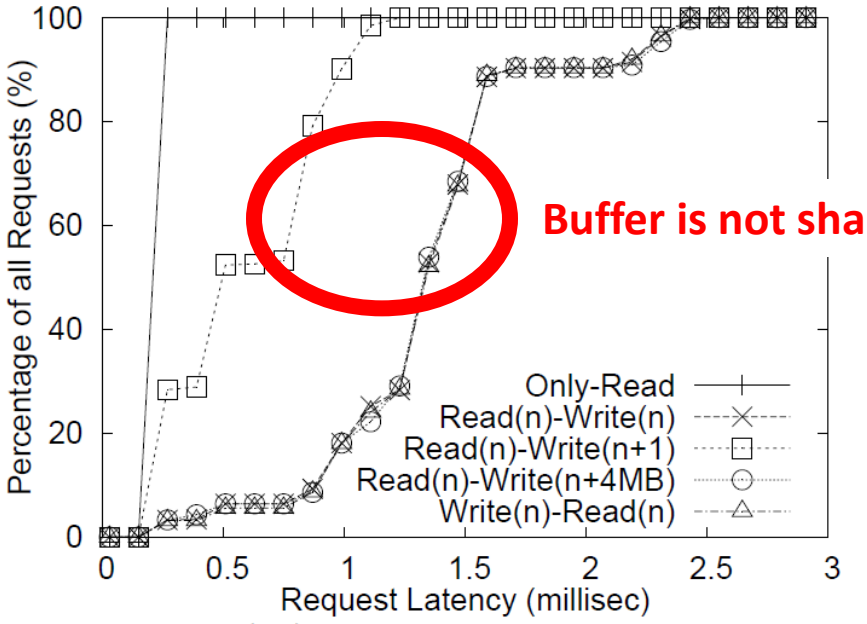
Disk cache take an important role in SSD's latency.

Experiment IV: Interactive Effect (1/3)

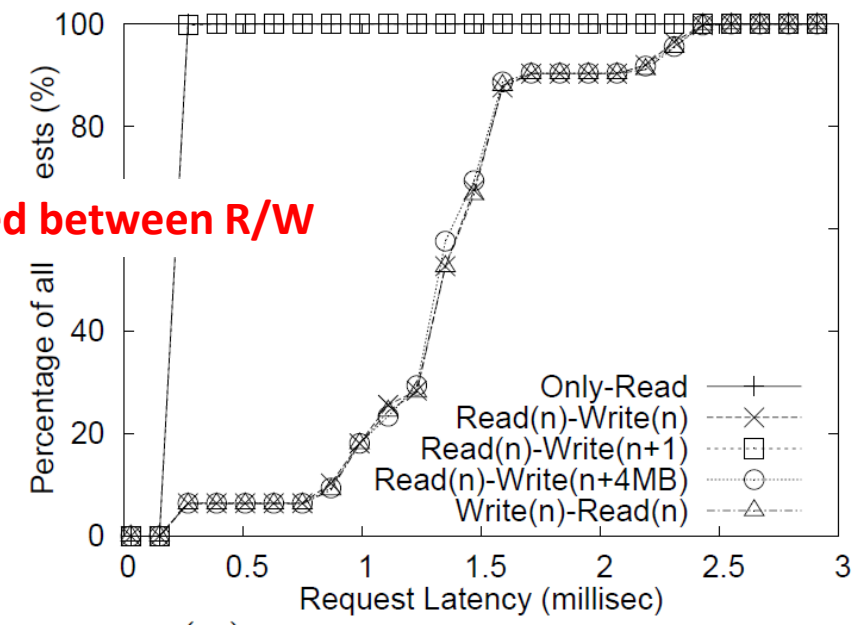
- Workload Patterns
  - Read(n) + Write(n)
    - R(0), W(0), R(1), W(1), R(2), W(2), …
  - Write(n) + Read(n)
    - W(0), R(0), W(1), R(1), W(2), R(2), …
  - Read(n) + Write(n+1)
    - R(0), W(1), R(2), W(3), R(4), W(5)
  - Read(n) + Write(n+4MB)
    - Re0), W(16000), R(1), W(16001), R(2), W(16002), …

Experiment IV: Interactive Effect (2/3)

- SSD-L
  - Read(n)-Write(n+1)
    - Partially Sequential Write
    - Read Interfered by Write

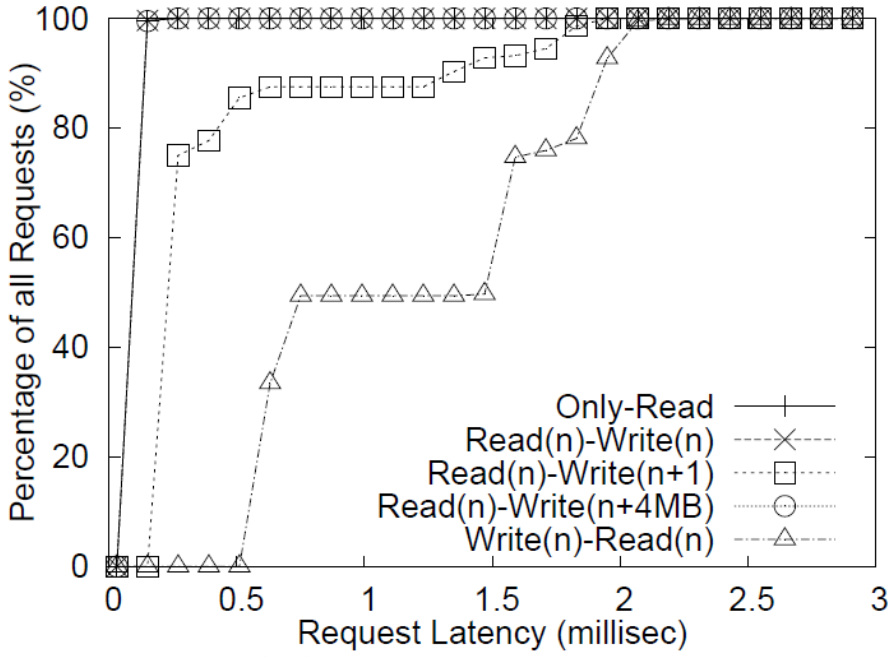


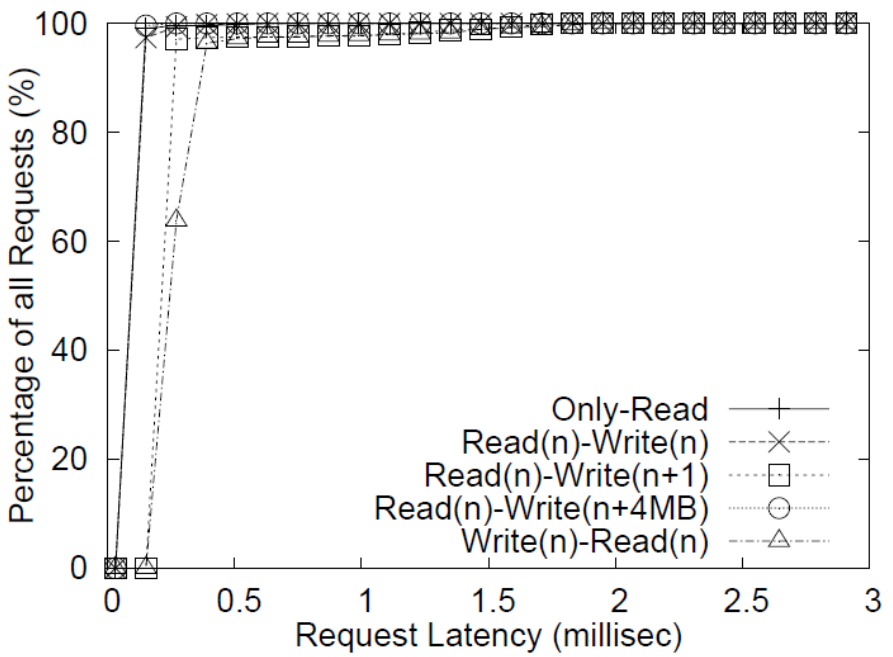
Buffer is not shared between R/W



Experiment IV: Interactive Effect (3/3)

- SSD-M
  - Read is delayed by write
    - Disk cache flushes dirty data





Experiment V: Background Operation (1/4)

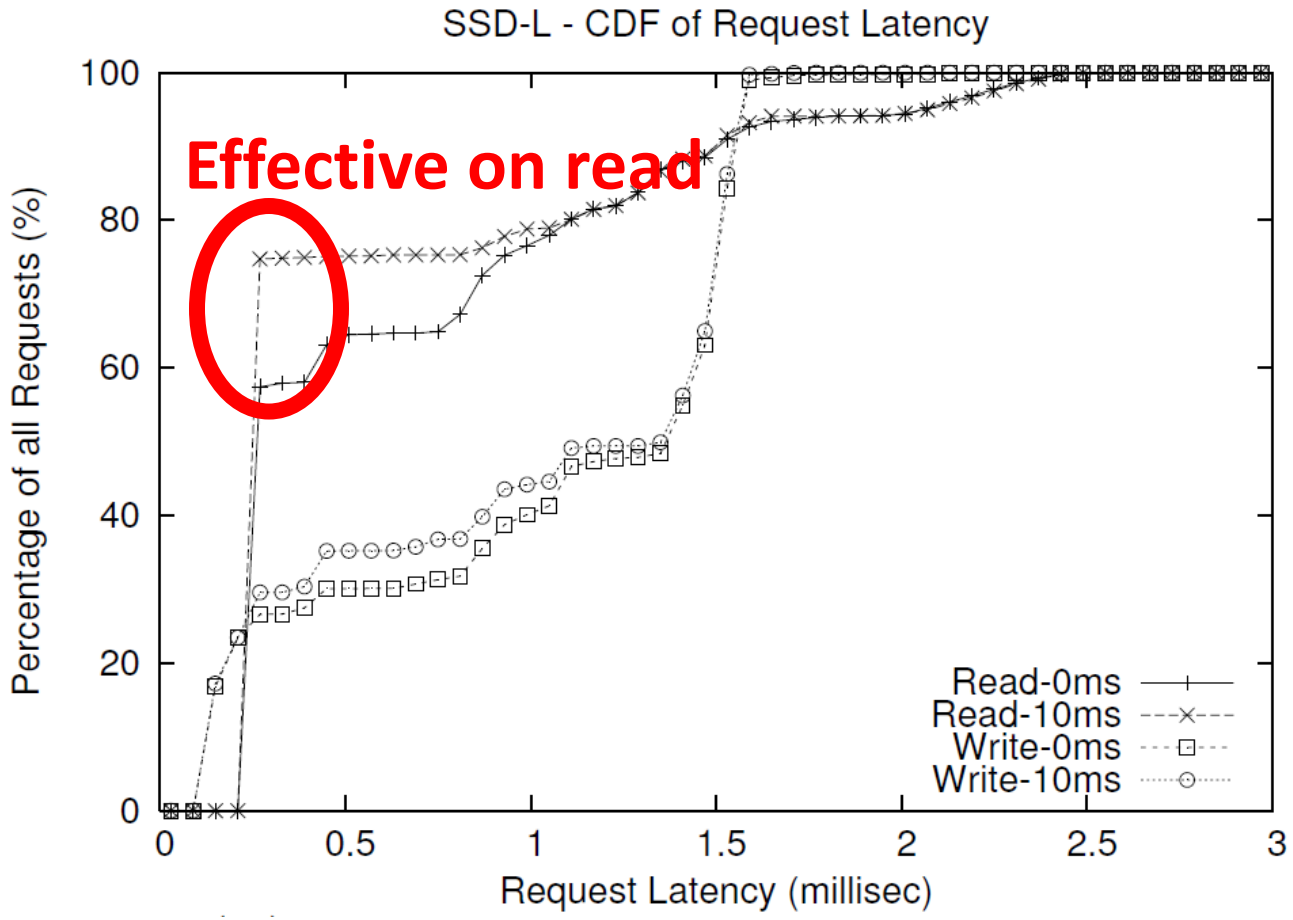
- Background ops.

  e.g. Delayed writes and cleaning.

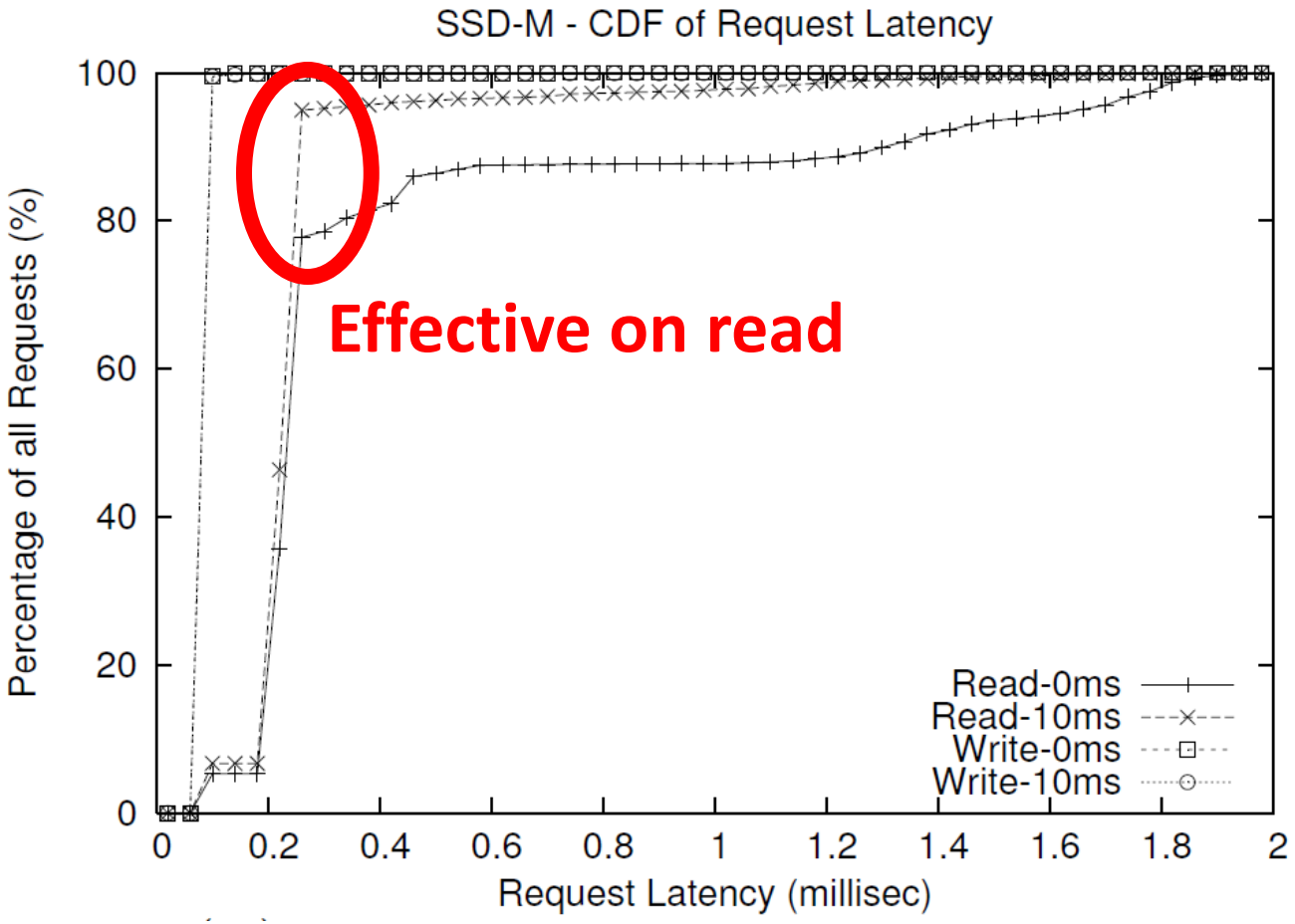
- Sequential workload
  - Read/write is randomly determined.
  - 50% of requests are writes.

- Intentionally insert 10ms interval
  - The no-op interval absorbs background ops.

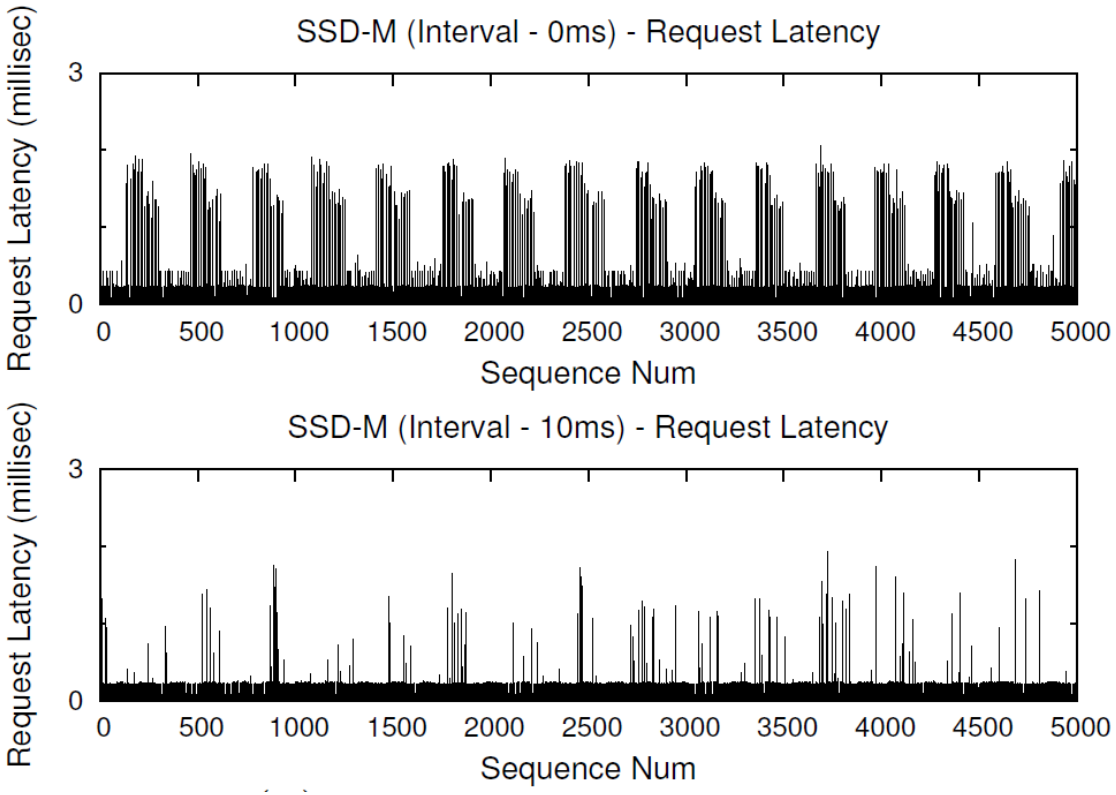
Experiment V: Background Operation (2/4)



Experiment V: Background Operation (3/4)



Experiment V: Background Operation (4/4)

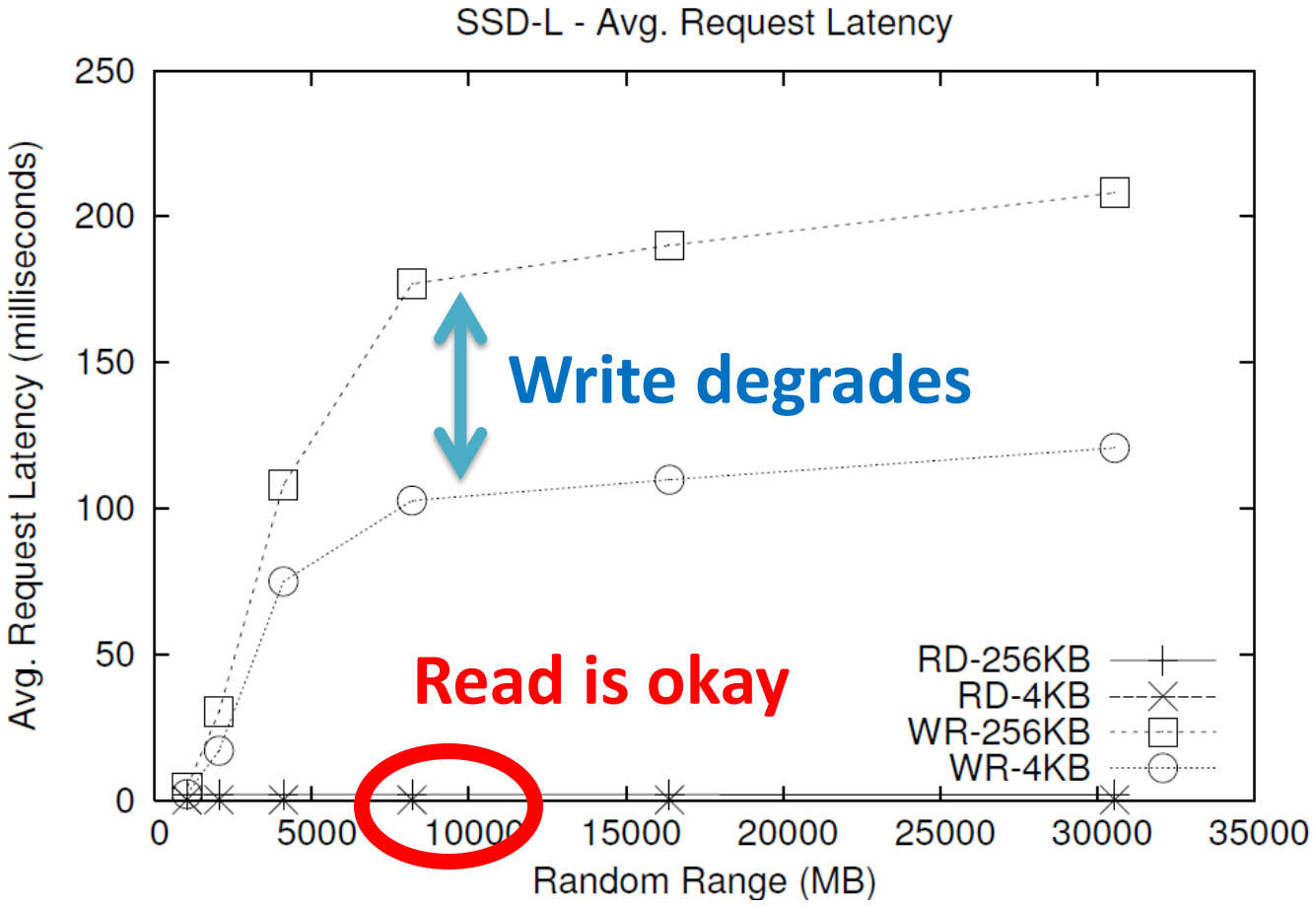


w/ background ops.

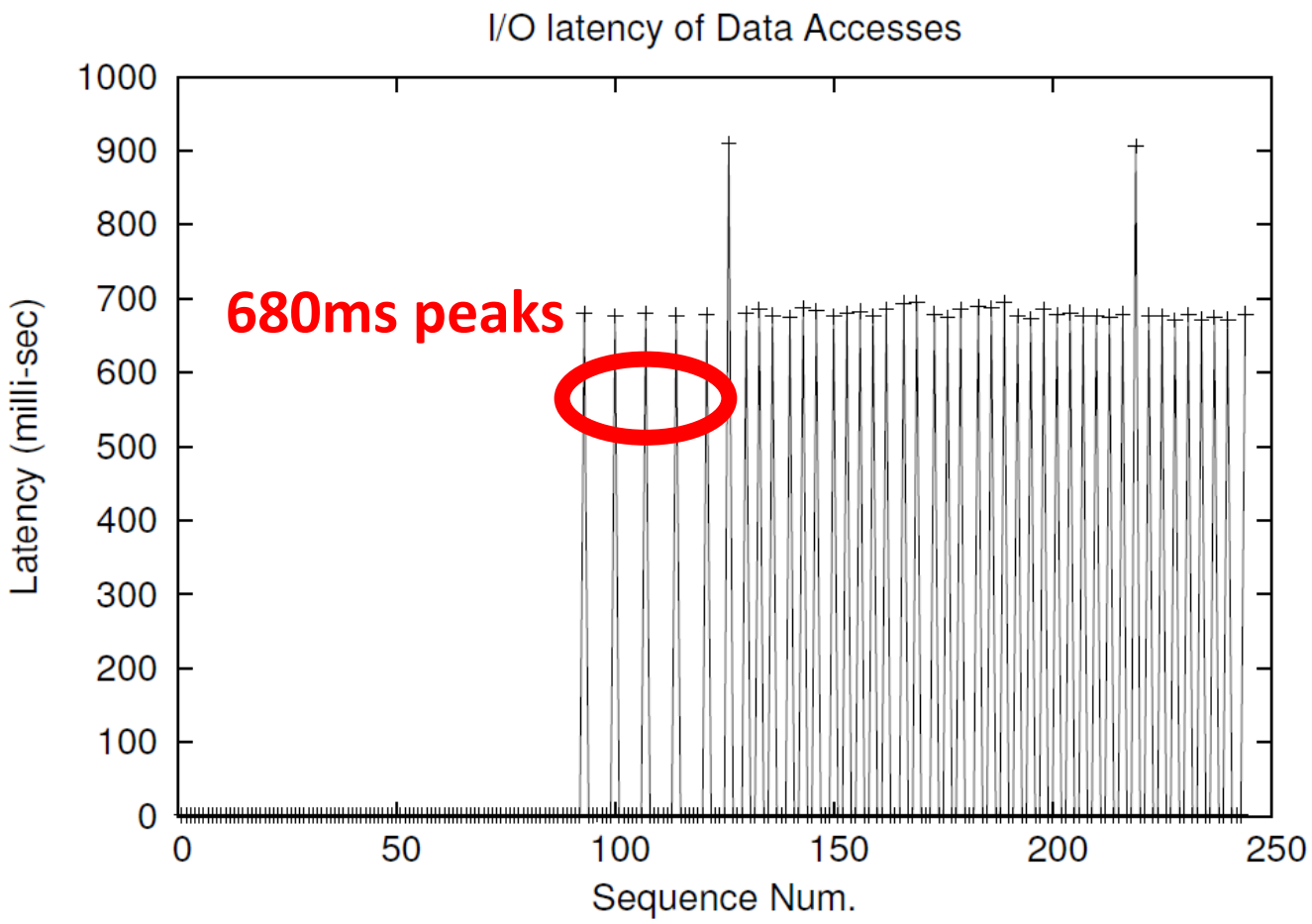
w/o background ops.

Background operations exist and can affect foreground jobs

Experiment VI: Workload Randomness (1/3)



Experiment VI: Workload Randomness (2/3)



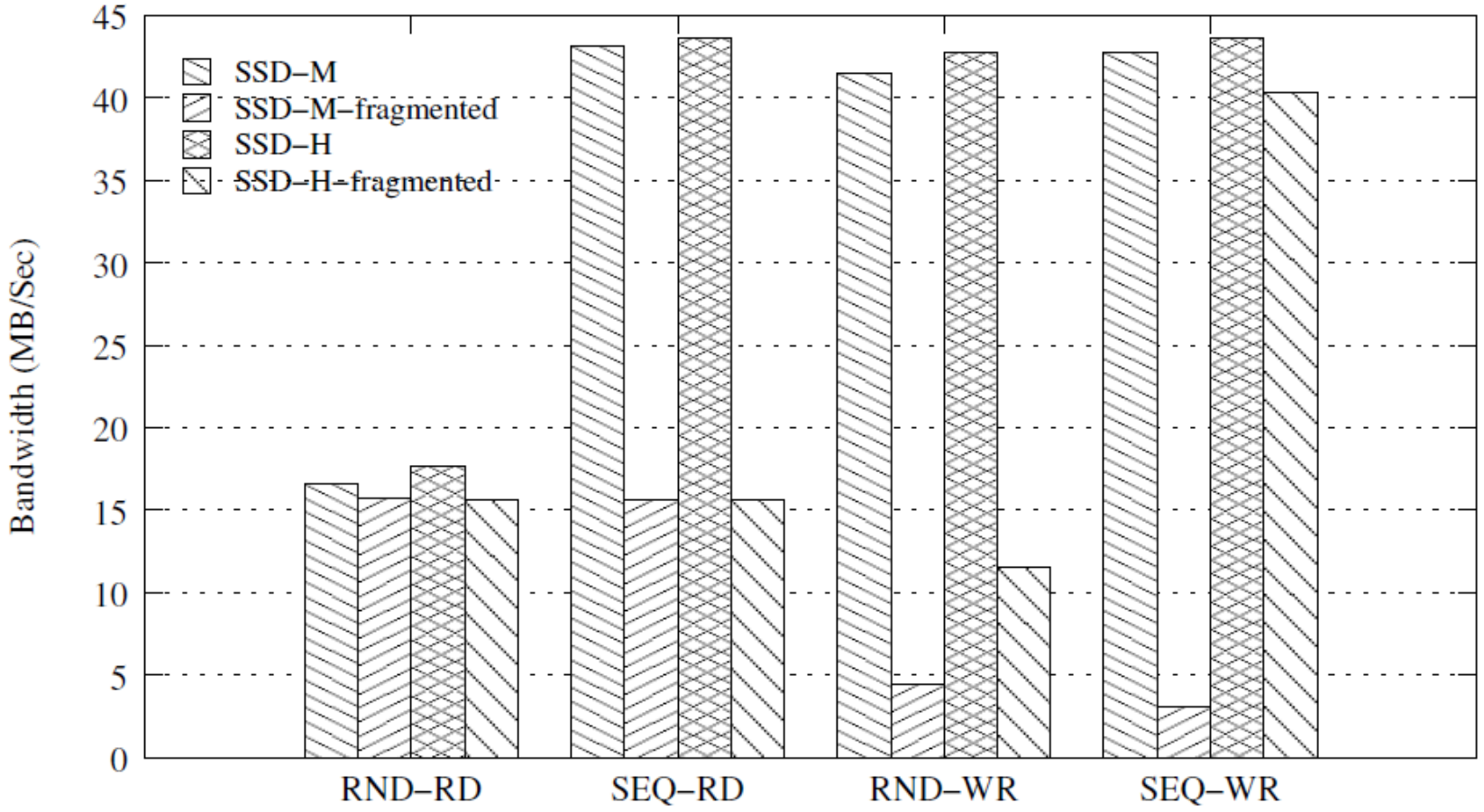
Random 4KB writes with 30GB range

Experiment VI: Workload Randomness (3/3)

- SSD-M, H didn't increase latency.
- Performance drop only applies to writes.
- Increasing request size to 256KB does not help to mitigate such a problem

- Why?
  - **Metadata synchronization**
  - **Log block merging**

Experiment VII: Fragmentation



Conclusion: Right understanding of SSD (1/2)

- Does **read** on SSD have a uniform latency?
  - No, it depends on reading pattern and prefetching.
- Would **random writes** be the worst case?
  - Yes, but same as sequential write w/ prefetching.
- Is a **disk cache** effective for SSDs?
  - Yes

Conclusion: Right understanding of SSD (2/2)

- Do reads and writes **interfere** with each other?
- Do **background operations** affect performance?
  - Yes, write's background ops. compete foreground ops.
- Would increasing **workload randomness** degrade performance?
  - Yes, it cause significant impact on performance.
- How does **fragmentation** affect performance?
  - Fragmentation degrades performance in some cases.

Q&A