Smart SSD Controller Design for Improving Random Read Performance

Yongsoo Joo

Embedded Software Research Center

Ewha Womans University

Supported by WCU (World Class University) program through National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R33-10085).

Overview

- Low random read performance of SSDs
  - For workloads having low concurrency
- Intelligence functions
  - Overlapping CPU computation & SSD accesses
  - Increasing queue depth
- Smart SSD controller
  - Integrating the proposed methods into the SSD controller

SSD Performance Optimization

- Single chip performance of MLC NAND flash
  - Page load: **30**MB/s
  - Page program: **7**MB/s
  - Data from Micron (http://onfi.org/presentations/)
    - Page size: 4KB x 2 (planes)
    - Page load/program: 50us/900us, data I/O: 211us
- HDD performance
  - Sequential read/write: about **100**MB/s
  - Random 4KB read: less than **1**MB/s
- How to increase SSD performance?
  - By exploiting SSD parallelism (8-16 chips per SSD)

SSD Performance Optimization

- Sequential I/O throughput
  - Increasing the number of channels & dies per channels
  - Data interleaving, improving an I/O interface, etc.
- Write performance
  - Efficient garbage collection and wear leveling
  - Over-provisioning for securing free blocks
  - I/O ordering & merging
  - Trim command
- Multiple concurrent I/O requests
  - Multi-channel architecture & controller support
  - Native command queueing (NCQ)
    - Queue depth: up to 32

SSD Performance Metrics

- The four corners
  - Sequential read (**SR**)
  - Sequential write (**SW**)
  - Random read (**RR**)
  - Random write (**RW**)

SSD Performance Metrics

- The six corners
  - Sequential read (**SR**)
  - Sequential write (**SW**)
  - Random read with QD=1 (**RR1**) and QD=32 (**RR32**)
  - Random write with QD=1 (**RW1**) and QD=32 (**RW32**)

- Queue depth (QD)
  - The number of outstanding I/O requests being processed in the SSD at a given time

SSD Performance Optimization

- Sequential I/O throughput: **SR/SW**
  - Increasing the number of channels & dies per channels
  - Data interleaving, improving an I/O interface, etc.
- Write performance: **SW/RW1/RW32**
  - Efficient garbage collection and wear leveling
  - Over-provisioning for securing free blocks
  - I/O ordering & merging
  - Trim command
- Multiple concurrent I/O requests: **RR32/RW32**
  - Multi-channel architecture & controller support
  - Native command queueing (NCQ)
    - Queue depth: up to 32

Performance Improvement

Device	Kingspec	X25-M G2	OCZ Vertex 3
Interface	PATA	SATA 2	SATA 3
SR	75 MB/s	263 MB/s	492 MB/s
SW	17 MB/s	112 MB/s	299 MB/s
RR1	16 MB/s	25 MB/s	35 MB/s
RR32	18 MB/s	165 MB/s	192 MB/s
RW1	2.5 MB/s	78 MB/s	95 MB/s
RW32	2.5 MB/s	99 MB/s	246 MB/s

2008.12: Kingspec 2.5” 64GB PATA IDE SSD
2009. 7: Intel X25-M G2 80GB
2011. 3: OCZ Vertex 3 240GB SATA 3 SandForce SF-2281

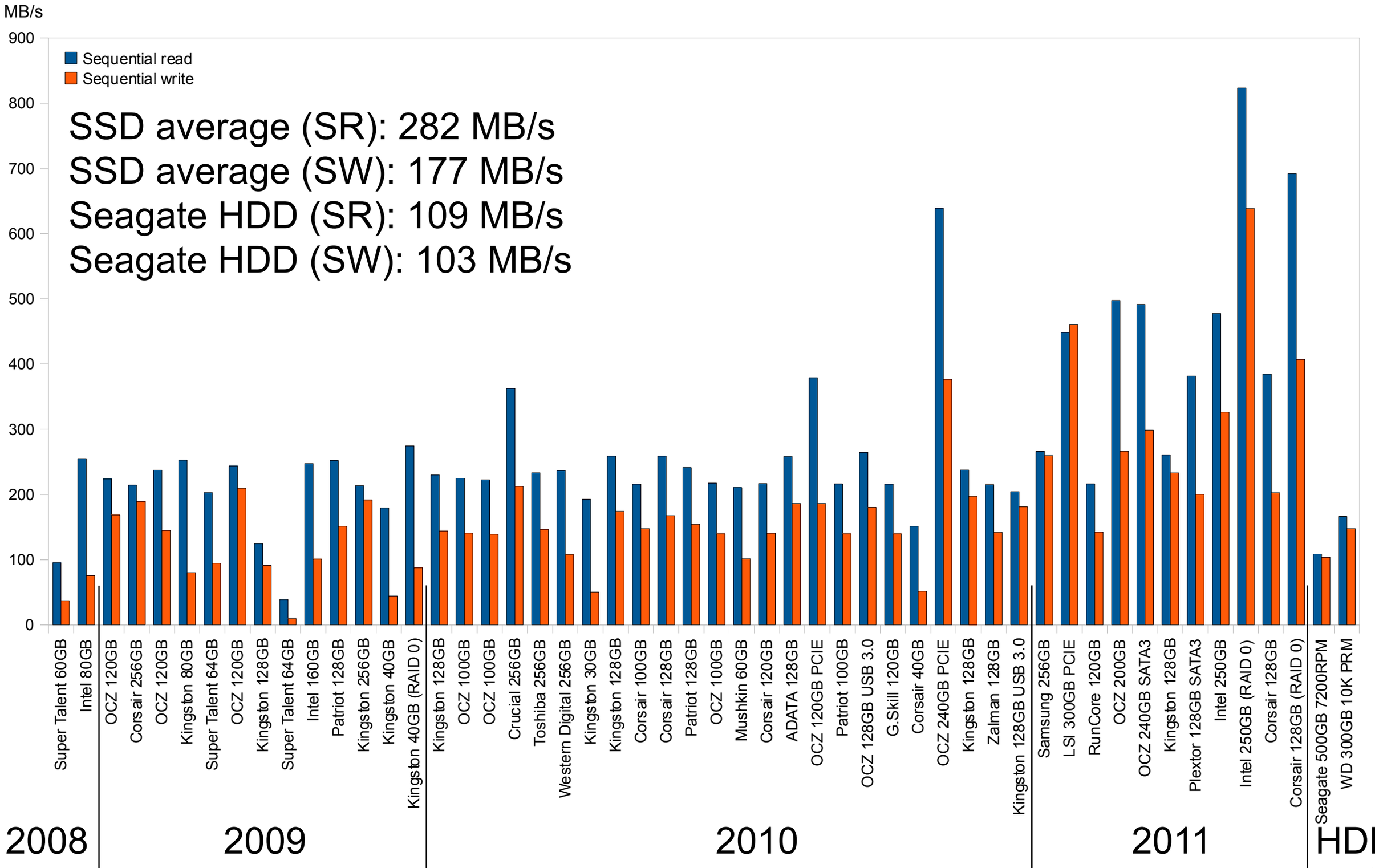
Performance Improvement

Device	Kingspec	X25-M G2	OCZ Vertex 3
Interface	PATA	SATA 2	SATA 3
SR	75 MB/s	263 MB/s	492 MB/s
SW	17 MB/s	112 MB/s	299 MB/s
RR1	16 MB/s	25 MB/s	35 MB/s
RR32	18 MB/s	192 MB/s	
RW1	2.5 MB/s	78 MB/s	95 MB/s
RW32	2.5 MB/s	99 MB/s	246 MB/s

Not much improved!

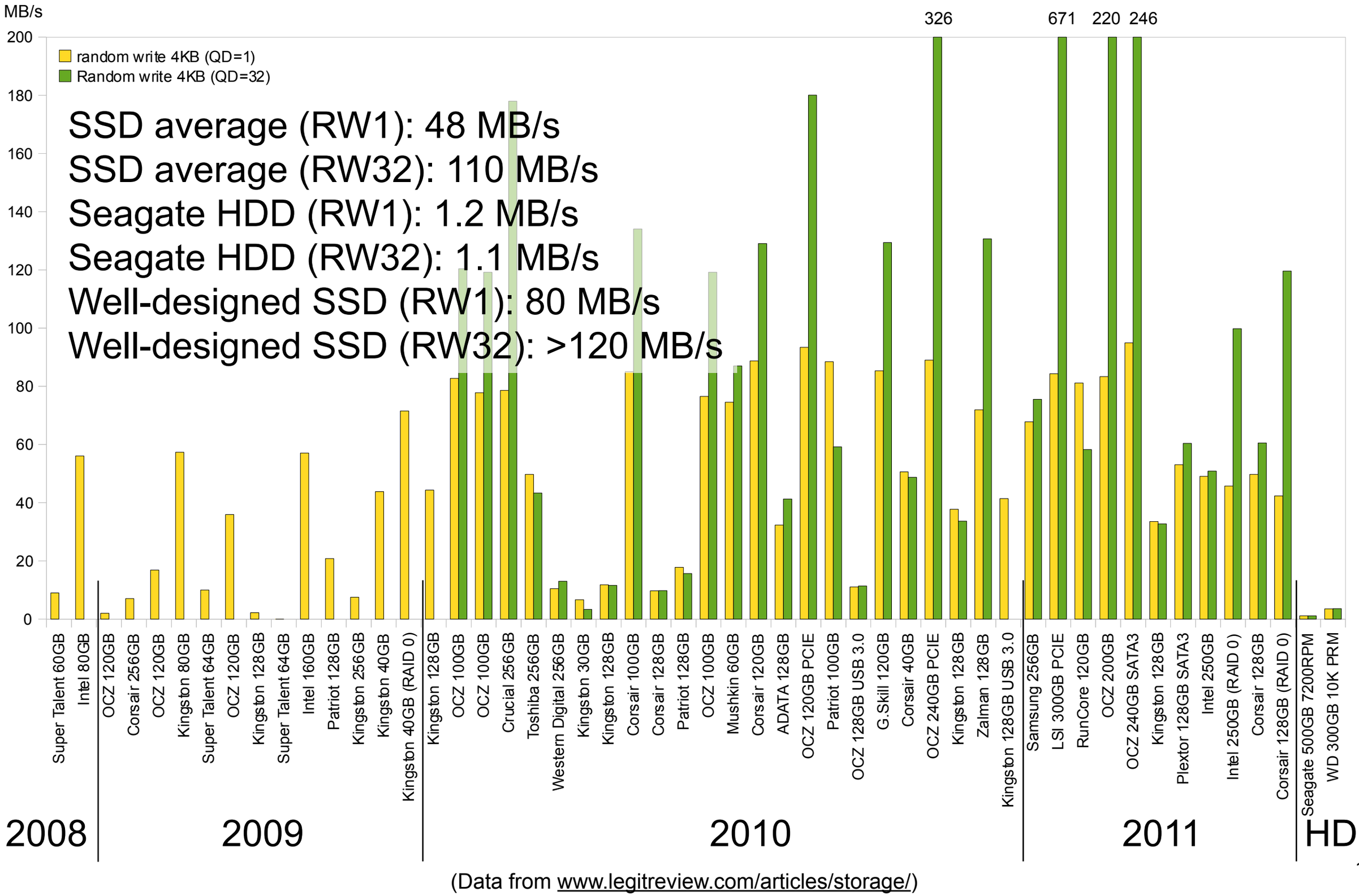
2008.12: Kingspec 2.5” 64GB PATA SSD
2009. 7: Intel X25-M G2 80GB
2011. 3: OCZ Vertex 3 240GB SATA 3 SandForce SF-2281

Sequential Read (SR) & Write (SW)



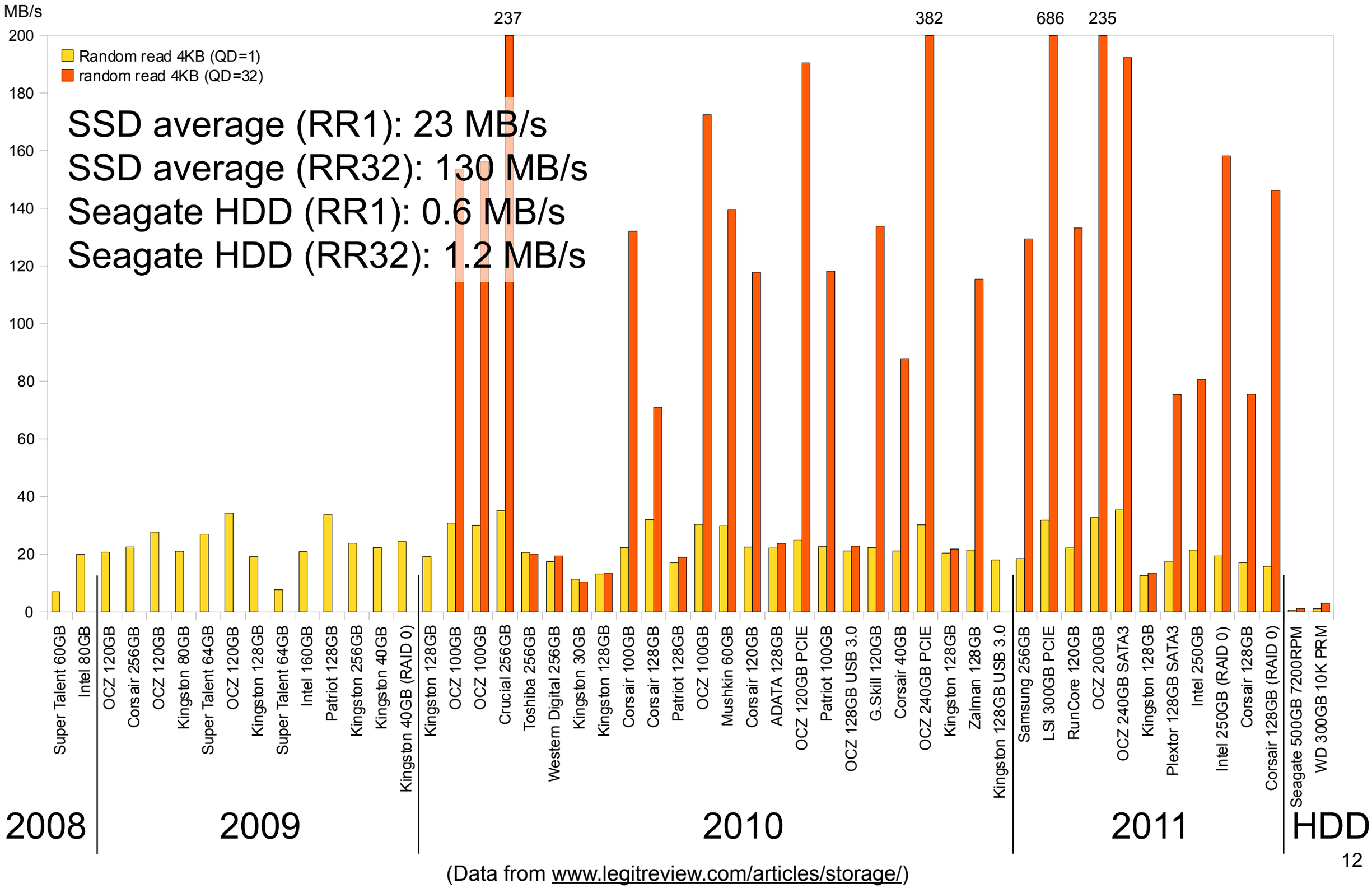
(Data from www.legitreview.com/articles/storage/)

Random Write (RW1 & RW32)



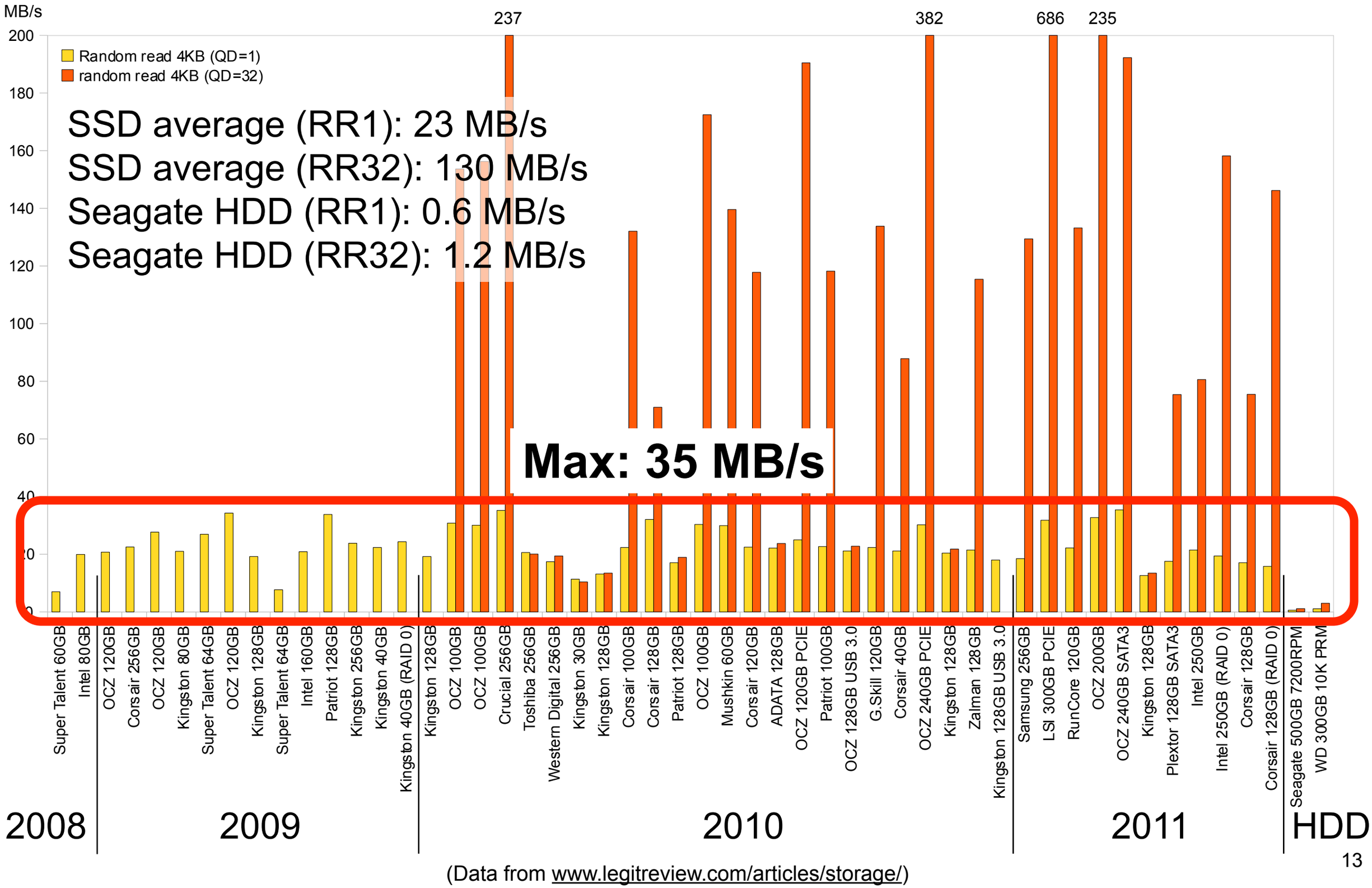
(Data from www.legitreview.com/articles/storage/)

Random Read (RR1 & RR32)



(Data from www.legitreview.com/articles/storage/)

Random Read (RR1 & RR32)



(Data from www.legitreview.com/articles/storage/)

RR1 Performance

- Much faster than HDDs
  - Seagate 7200RPM HDD: 0.6MB/s
  - Intel X25-M G2: 25MB/s (40x)

- Depend on single-chip performance
  - Improving I/O interface speed
    - ONFI 1.0 -> ONFI 2.0 -> ONFI 3.0
    - SATA2 -> SATA3 -> PCIE -> ??
  - Improving page load time
    - Speed: 2bit/cell -> 1bit/cell
    - Capacity: 2bit/cell -> 3bit/cell

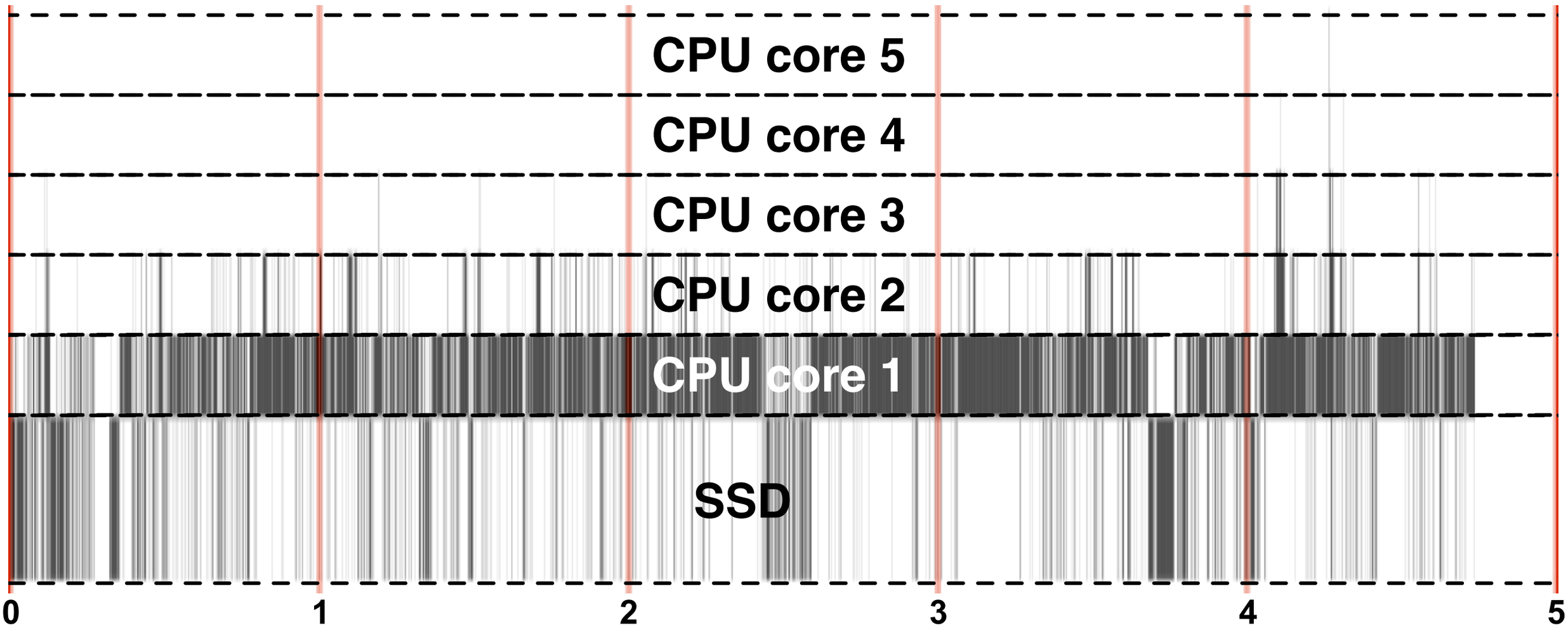
Random Read Workloads

- High I/O concurrency (RR32)
  - Multiple independent I/O streams
  - Multi-user systems
  - Web server workload, database applications, etc.

- Low I/O concurrency (RR1)
  - Single user systems
  - OS booting, application launch, game loading, etc.
    - Demand paging
  - Important metric for personal computing systems

Example: Application Launch

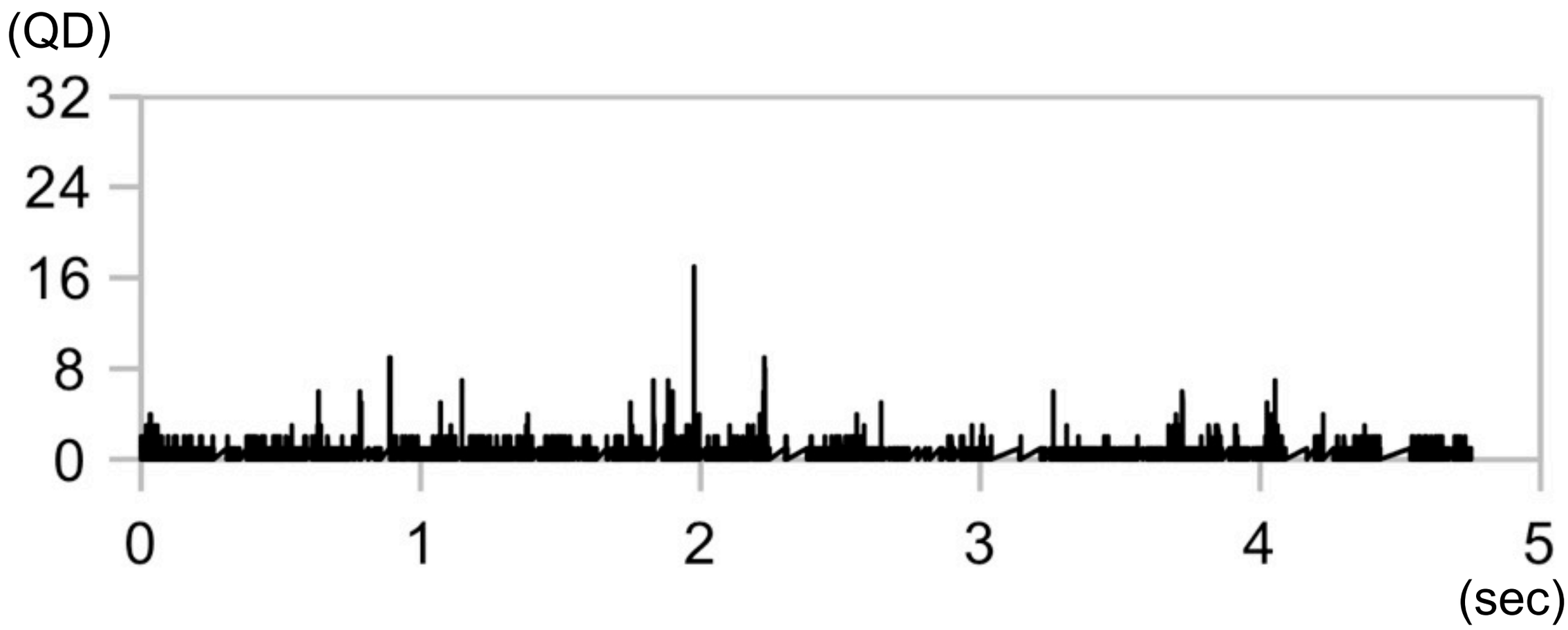
- CPU & SSD usage
  - Only one core active during the most time periods
  - SSD mostly idle when one or more CPU cores are active



(Application: Eclipse, OS: Linux)

Example: Application Launch

- SSD queue depth
  - Average QD: 0.3



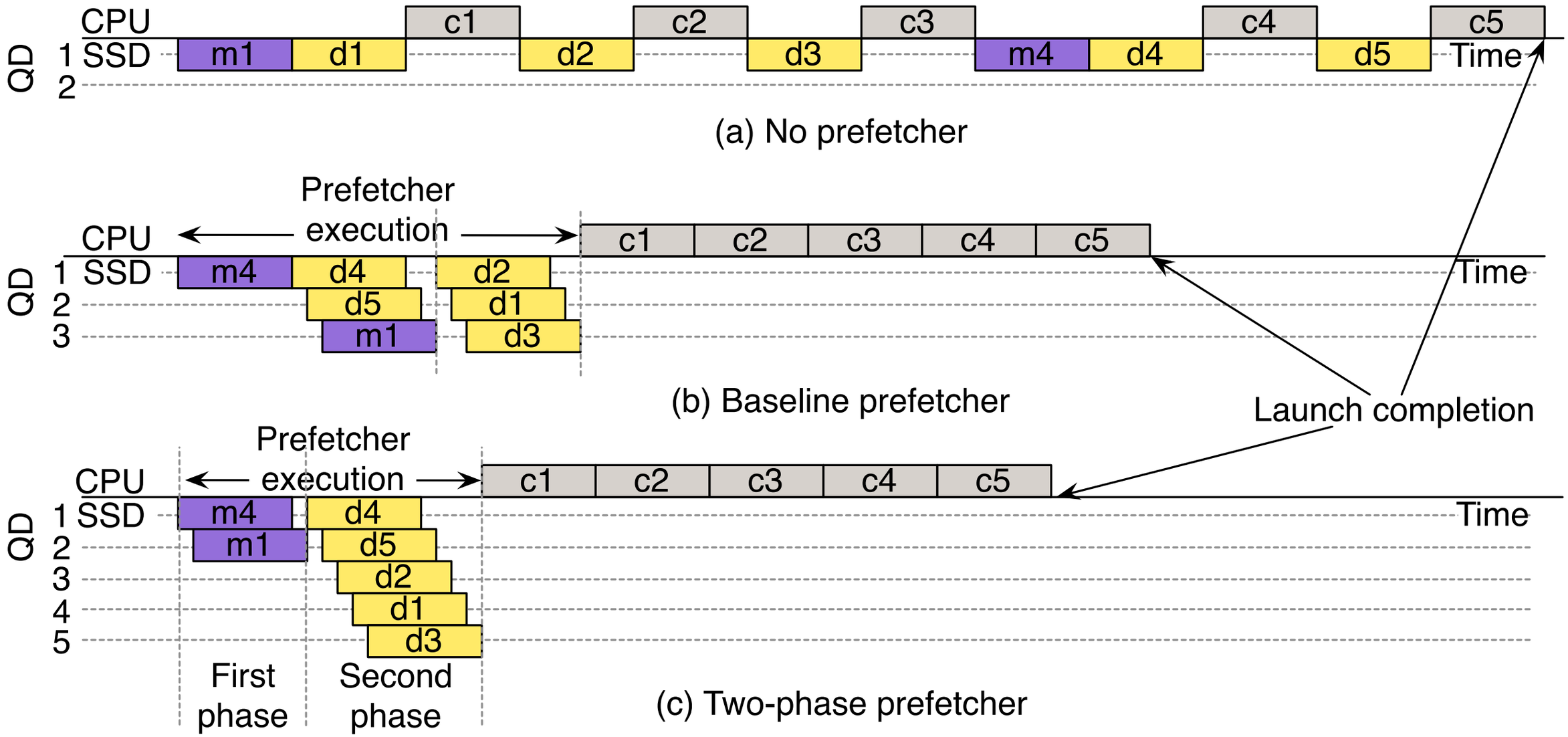
(Application: Eclipse, OS: Linux)

Improving RR1 Performance

- Two-phase prefetcher
  - Resolving metadata dependency for higher QD
  - "Exploiting SSD parallelism to accelerate application launch on SSDs," IET Electronics Letters, 2011.
- FAST: **F**ast **A**pplication **ST**arter
  - Overlapping CPU computation and SSD access time
  - "FAST: Quick Application Launch on Solid-State Drives," in Proc. USENIX FAST, 2011.

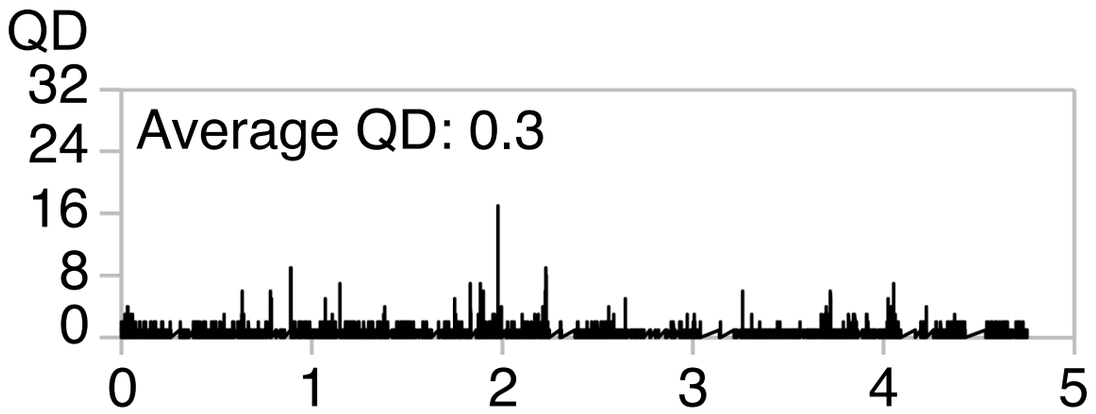
Two-Phase Prefetcher

- Resolving metadata dependency

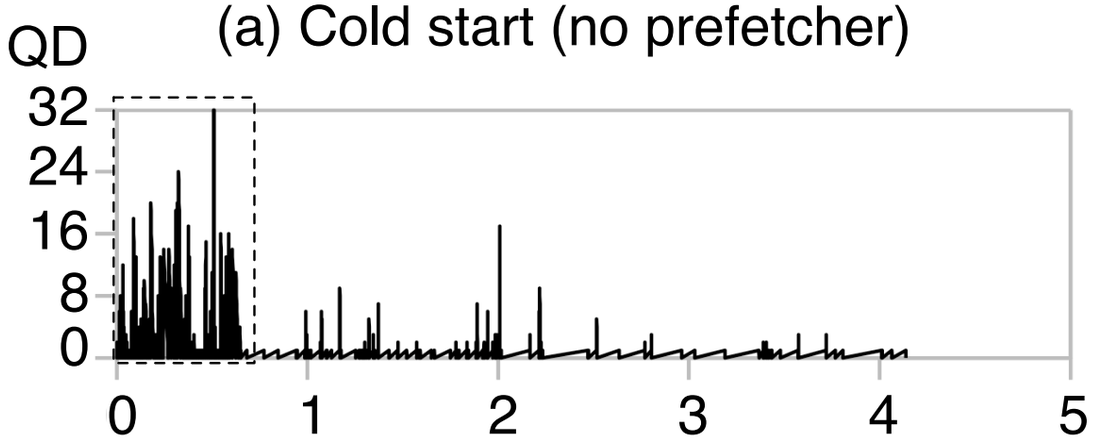


mi: metadata block, di: normal data block

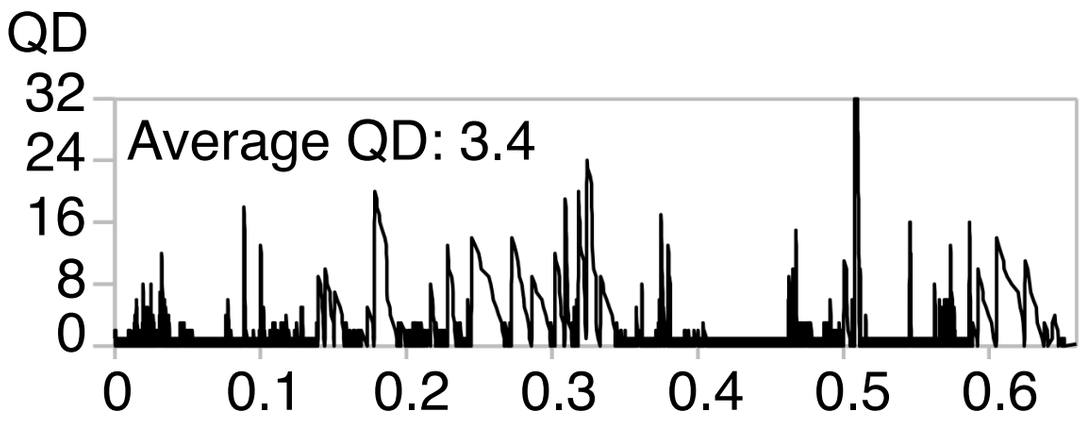
Two-Phase Prefetcher



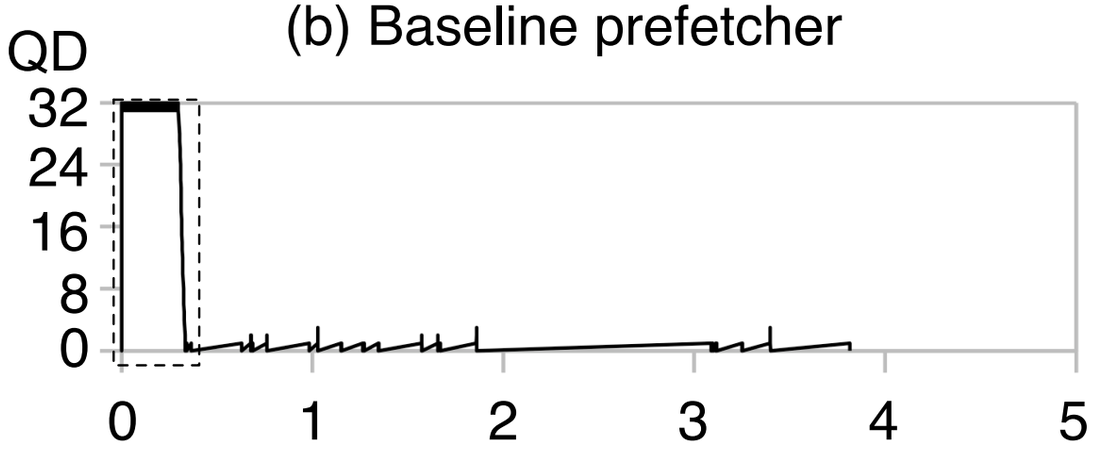
(a) Cold start (no prefetcher)



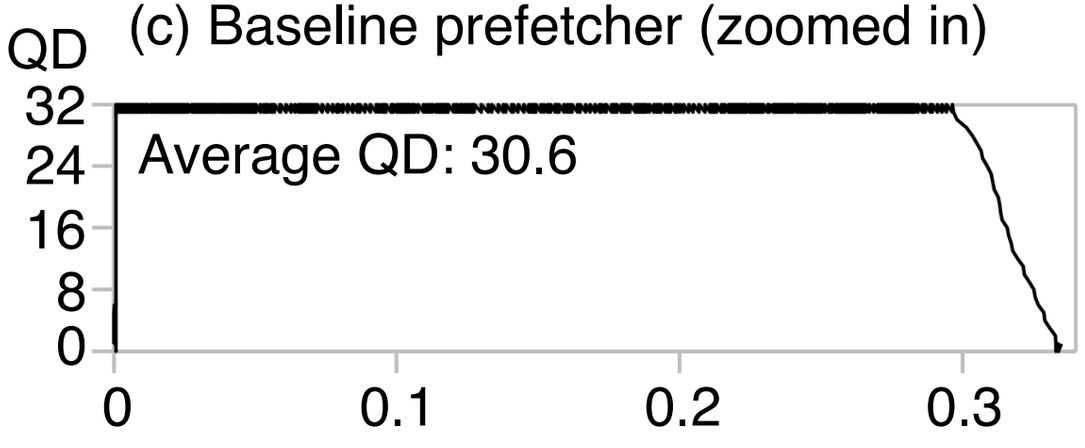
(b) Baseline prefetcher



(c) Baseline prefetcher (zoomed in)



(d) Two-phase prefetcher

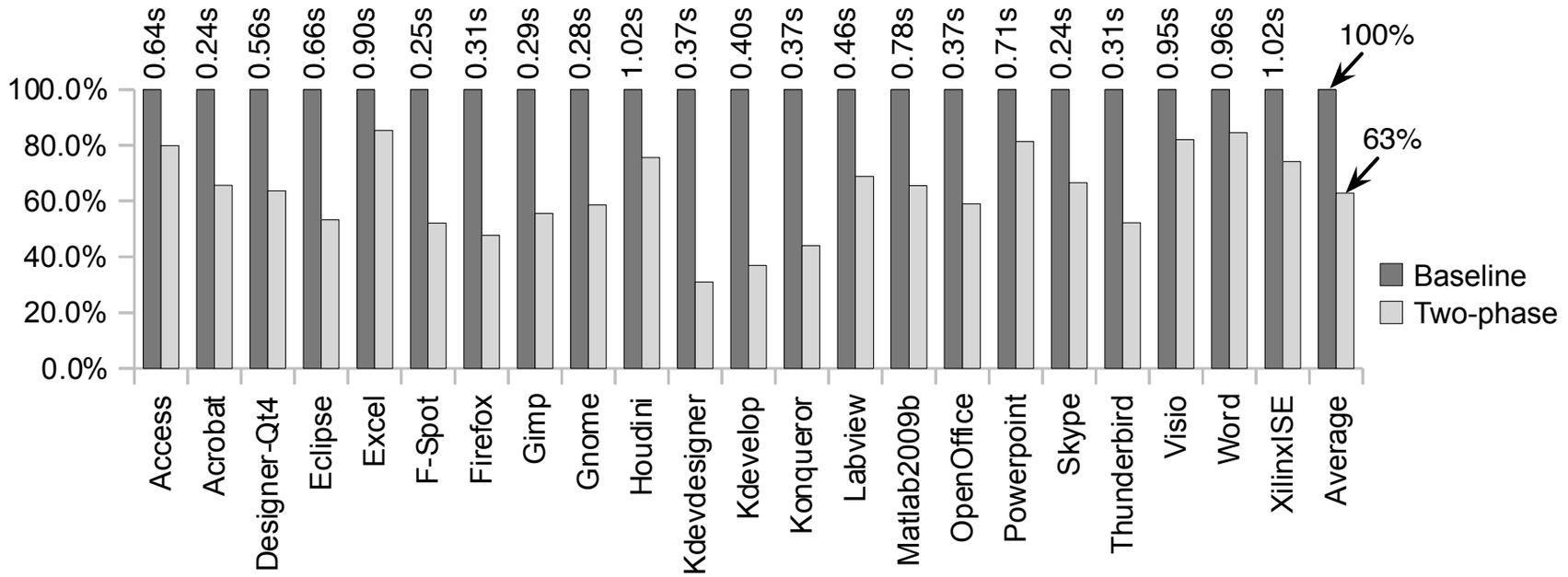


(e) Two-phase prefetcher (zoomed in)

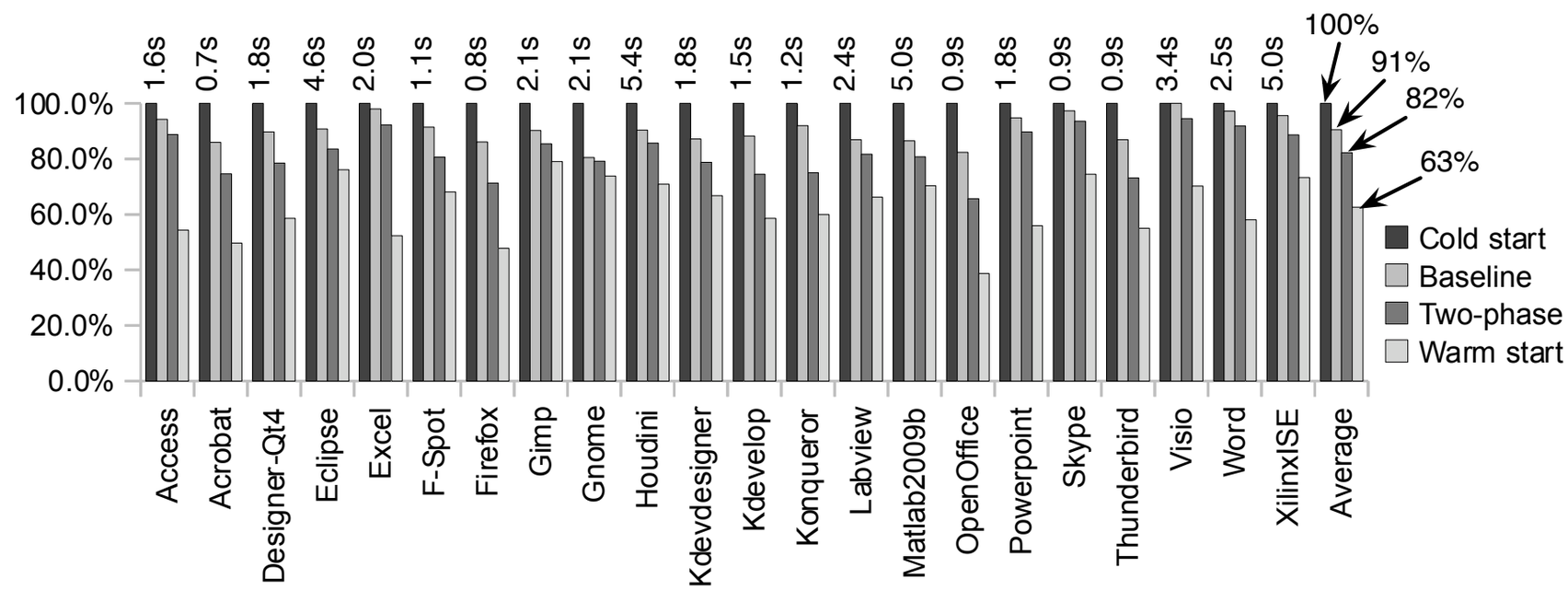
(Application: Eclipse, OS: Linux)

Two-Phase Prefetcher

- Prefetcher time
  - 37% reduction
- Launch time
  - 18% reduction



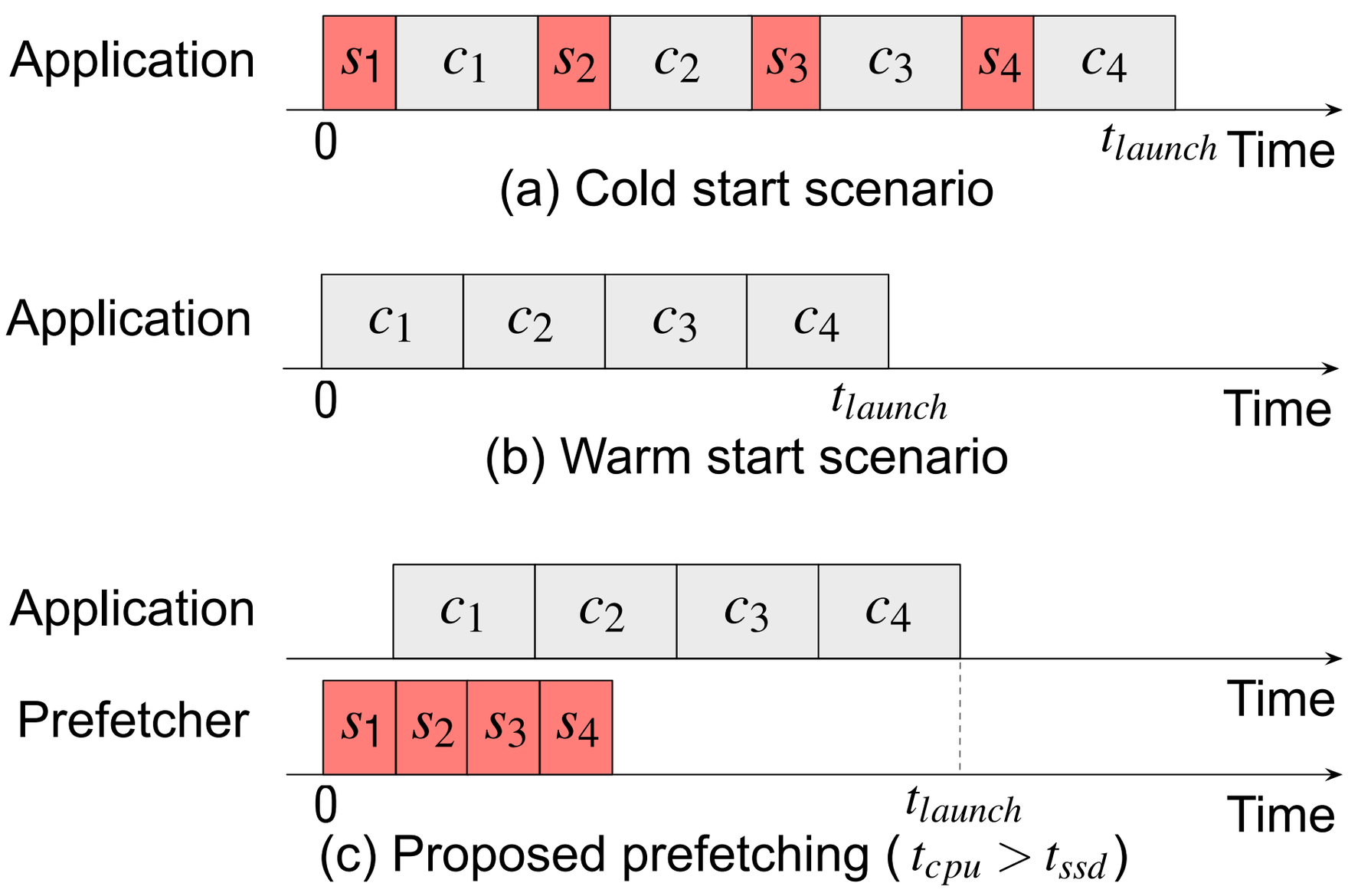
(a) Prefetcher execution time



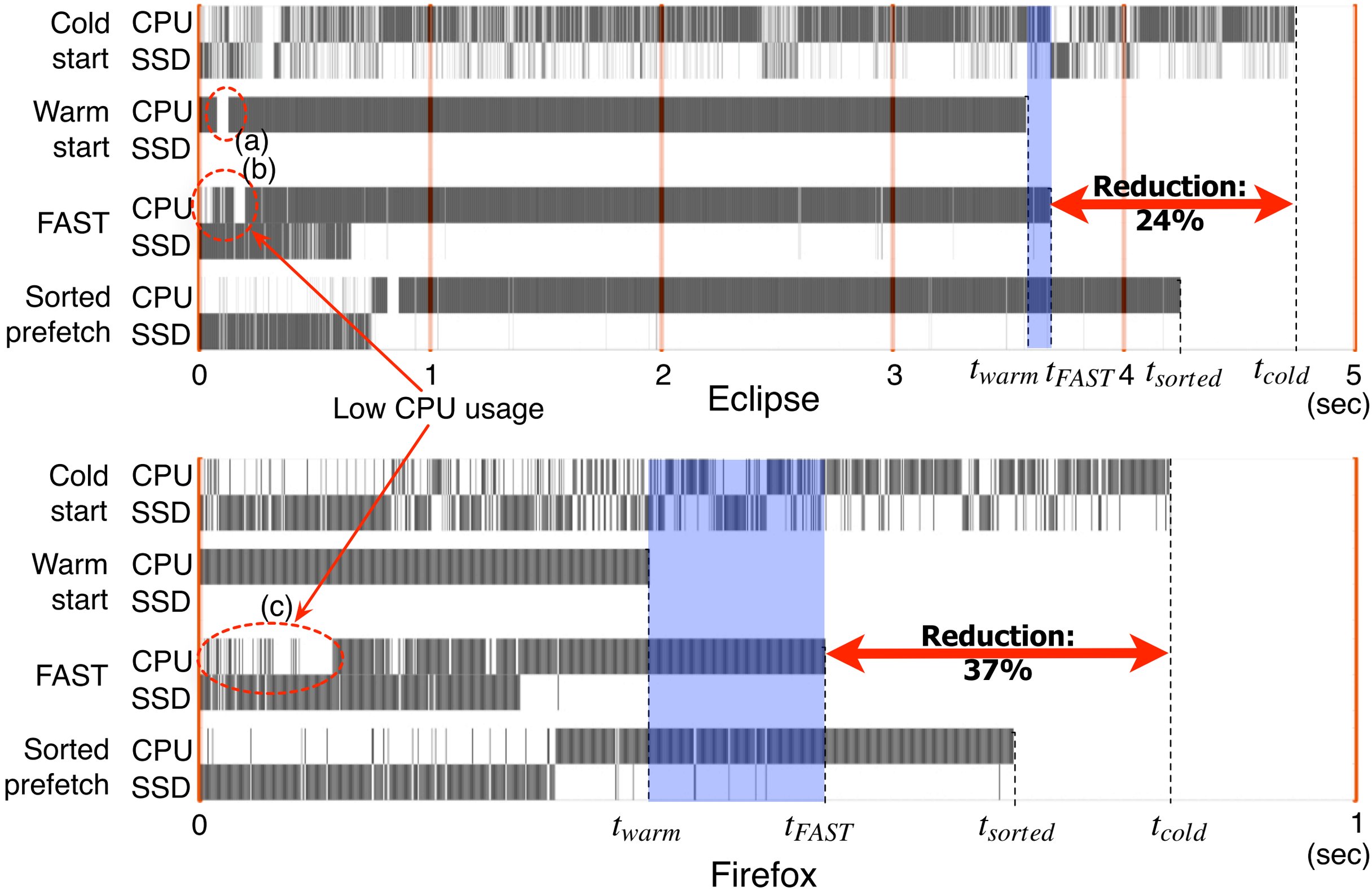
(b) Application launch time

FAST: Fast Application STarter

- Overlap CPU computation with SSD accesses

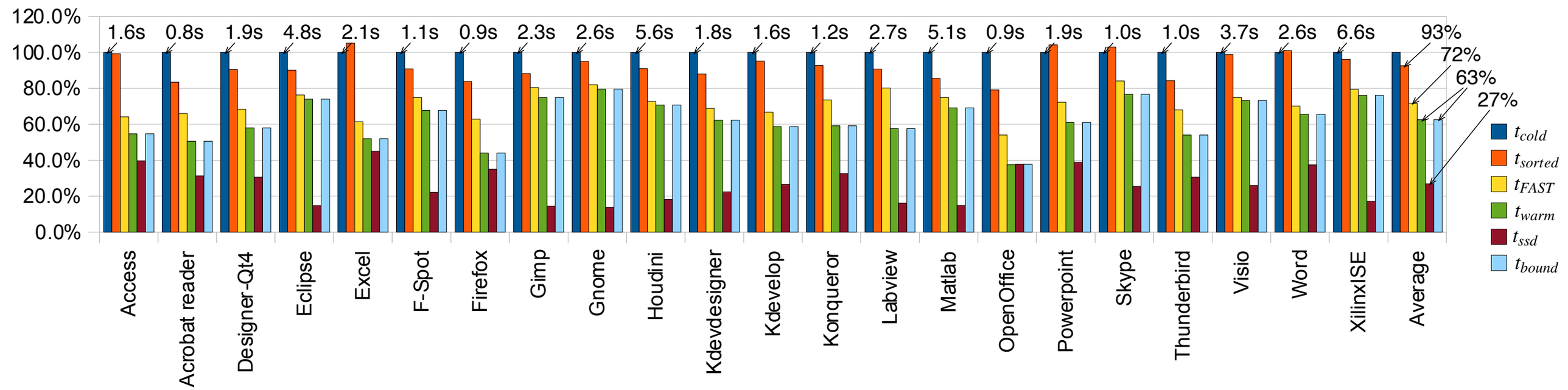


CPU and SSD Usage



Measured Application Launch Time

- Launch time reduction
  - Warm start: 37% (upper bound)
  - FAST: 28% (min: 16%, max: 46%)



(Normalized to the cold start time.)

I/O Sequence Determinism

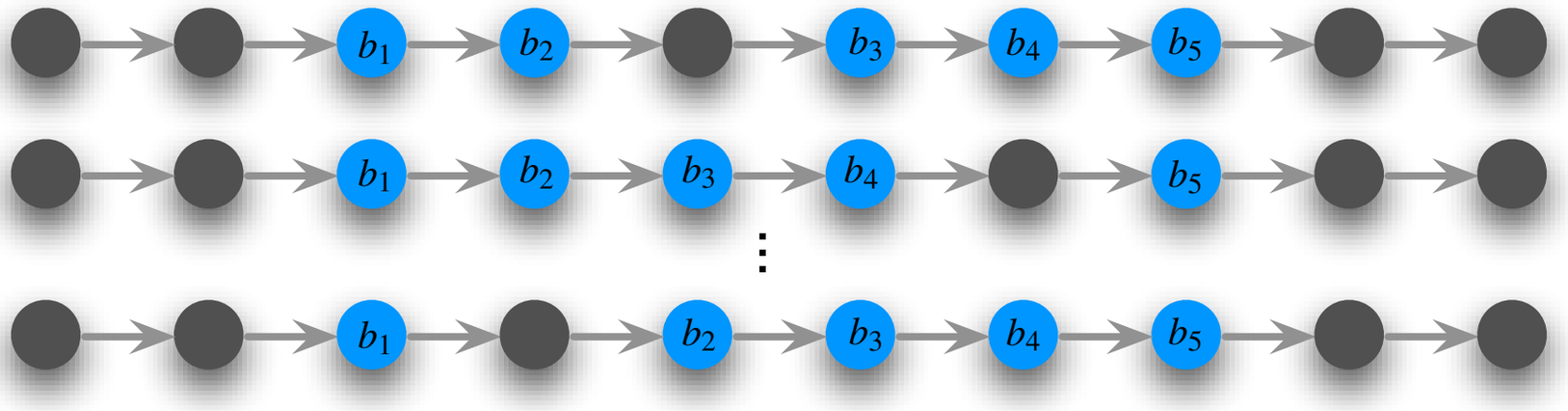
- Non-sequential I/O streams repeatedly occur
  - M. Bhadkamkar et al., "BORG: Block-reORGanization for Self-optimizing Storage Systems," in Proc. USENIX FAST, 2009.

Workload type	File System size [GB]	Memory size [GB]	Reads [GB]		Writes [GB]		File System accessed	Top 20% data access	Partial determinism
			Total	Unique	Total	Unique			
office	8.29	1.5	6.49	1.63	0.32	0.22	22.22 %	51.40 %	65.42 %
developer	45.59	2.0	3.82	2.57	10.46	3.96	14.32 %	60.27 %	61.56 %
SVN server	2.39	0.5	0.29	0.17	0.62	0.18	14.60 %	45.79 %	50.73 %
web server	169.54	0.5	21.07	7.32	2.24	0.33	4.51 %	59.50 %	15.55 %

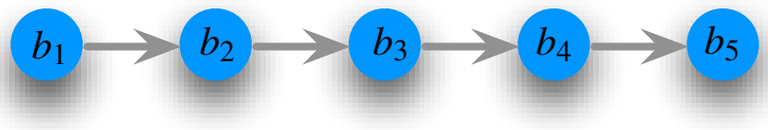
Table 1: **Summary statistics of week-long traces obtained from four different systems.**

Application Launch Sequence

- Deterministic block requests over repeated launches
- Raw block request traces



- Application launch sequence



Block requests irrelevant to the application launch

Smart SSD Controller

- Motivation
  - Integrate the optimization schemes into the SSD
- Smart SSD controller
  - Doing **something** more than just processing the received I/O requests
  - **Something**=intelligence functions
    - ex) Two-phase prefetcher, FAST
- Necessary components
  - Microprocessor and buffer memory
  - Already available in SSDs

Physical View

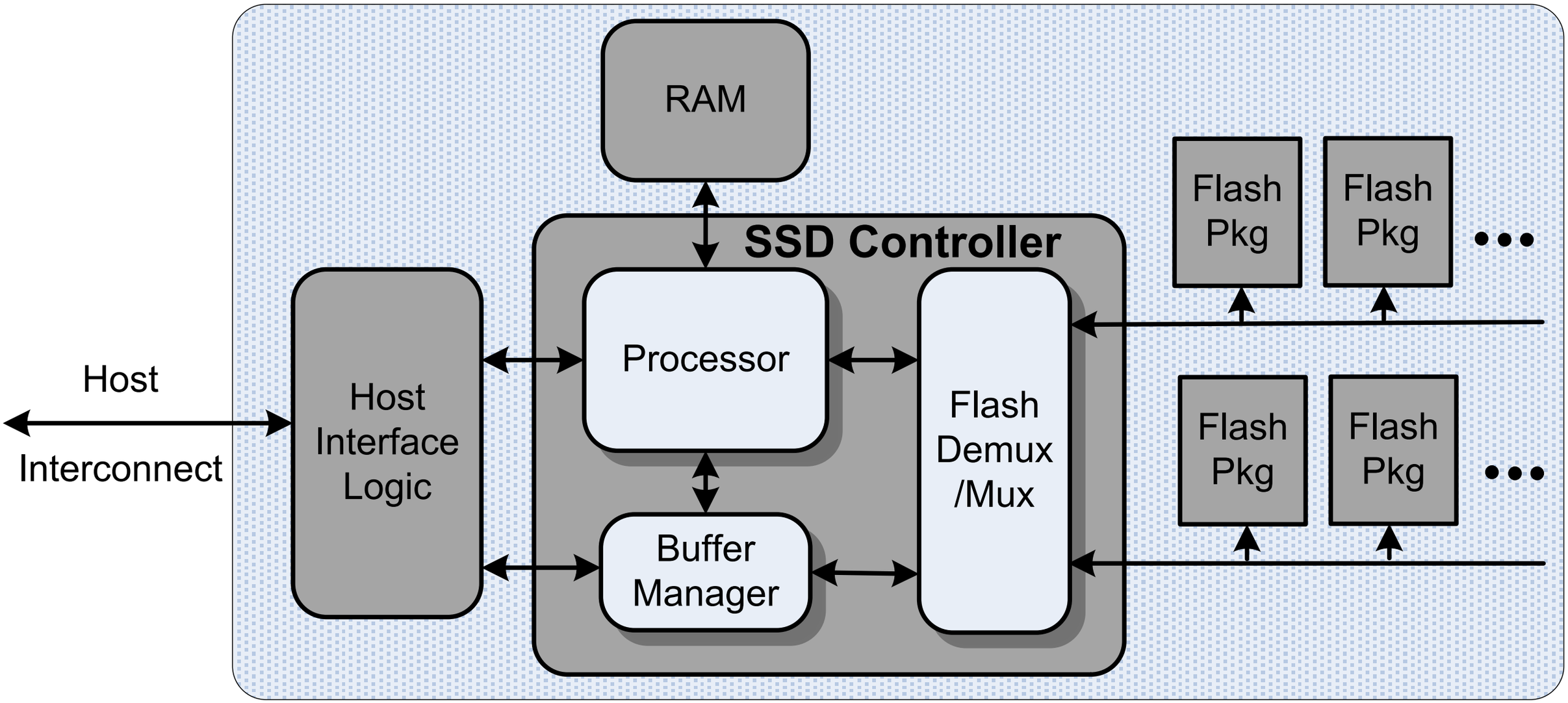


Figure source: Nitin Agrawal et al., “Design Tradeoffs for SSD Performance,” in Proc. USENIX ATC, 2008.

Physical View

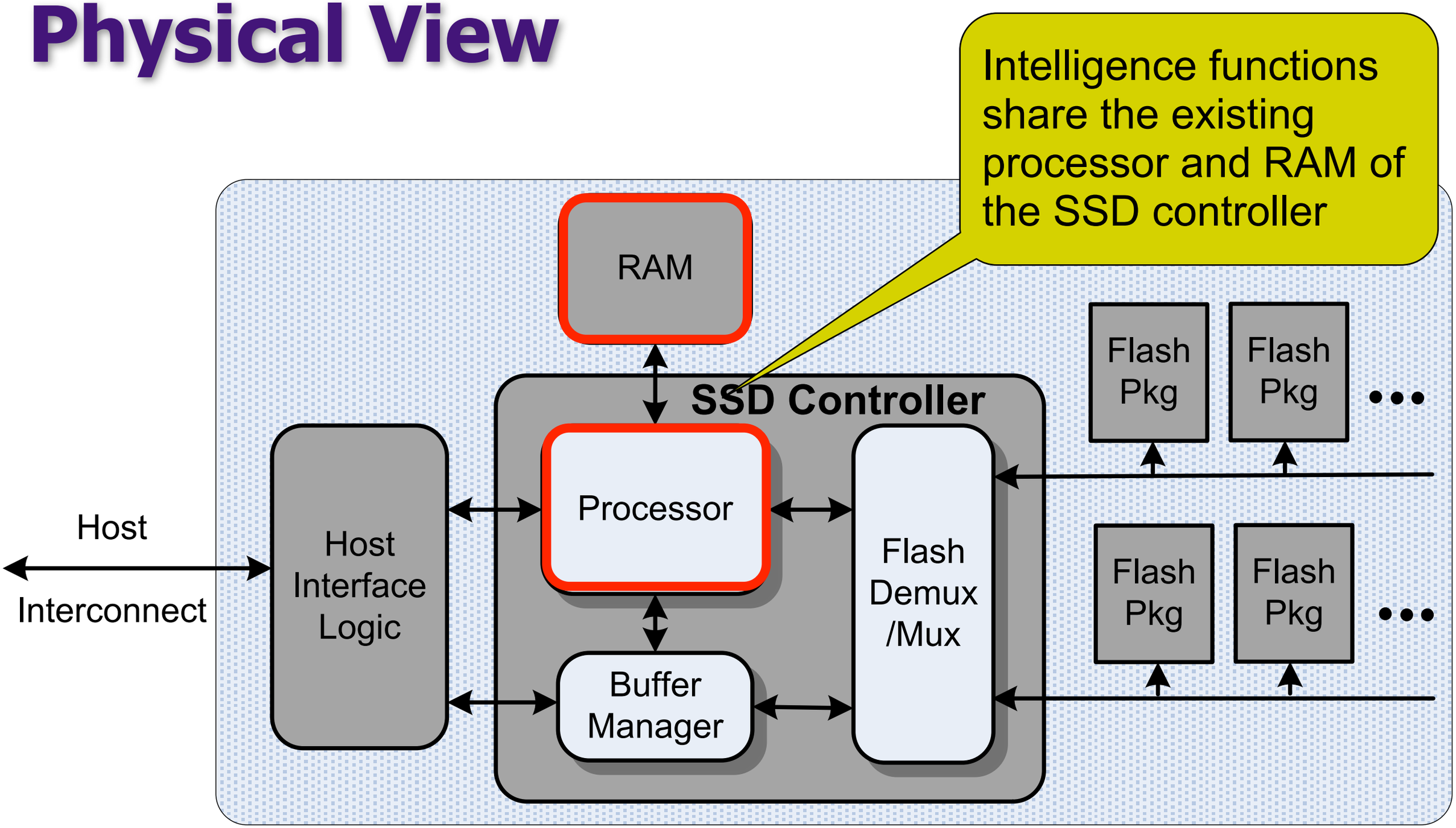
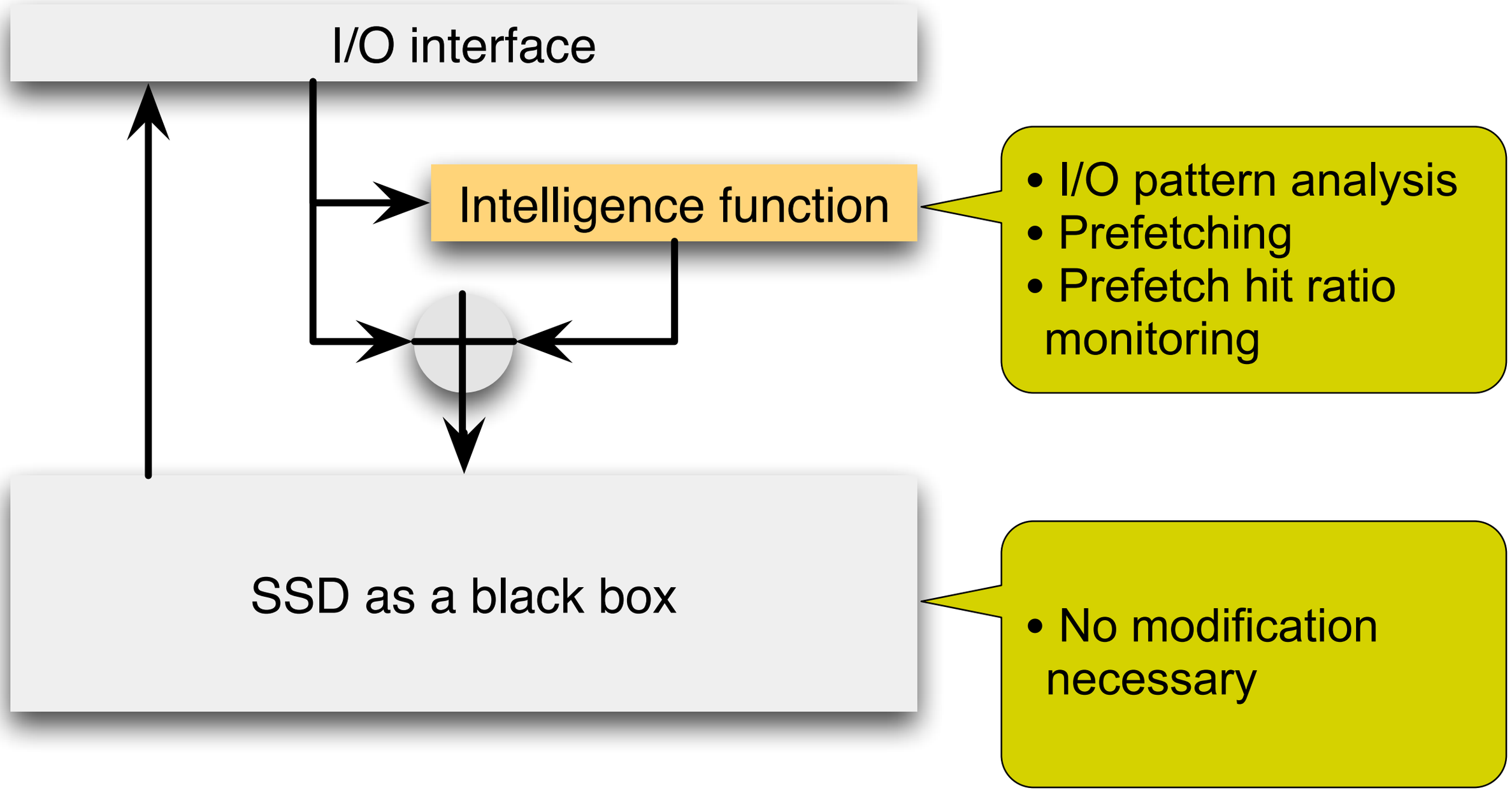


Figure source: Nitin Agrawal et al., “Design Tradeoffs for SSD Performance,” in Proc. USENIX ATC, 2008.

Logical View



Advantages

- Easiness in monitoring block I/Os
- OS independence
- No metadata dependency
- Immediate deployment
  - cf. Seagate hybrid HDD (Momentus XT)

Design Consideration

- Buffer management
  - Limited buffer memory capacity
  - Caching vs. prefetching

- Queue depth control
  - Maximum queue depth can be harmful

- Data transfer delay
  - From the SSD buffer to the main memory page cache

Q&A