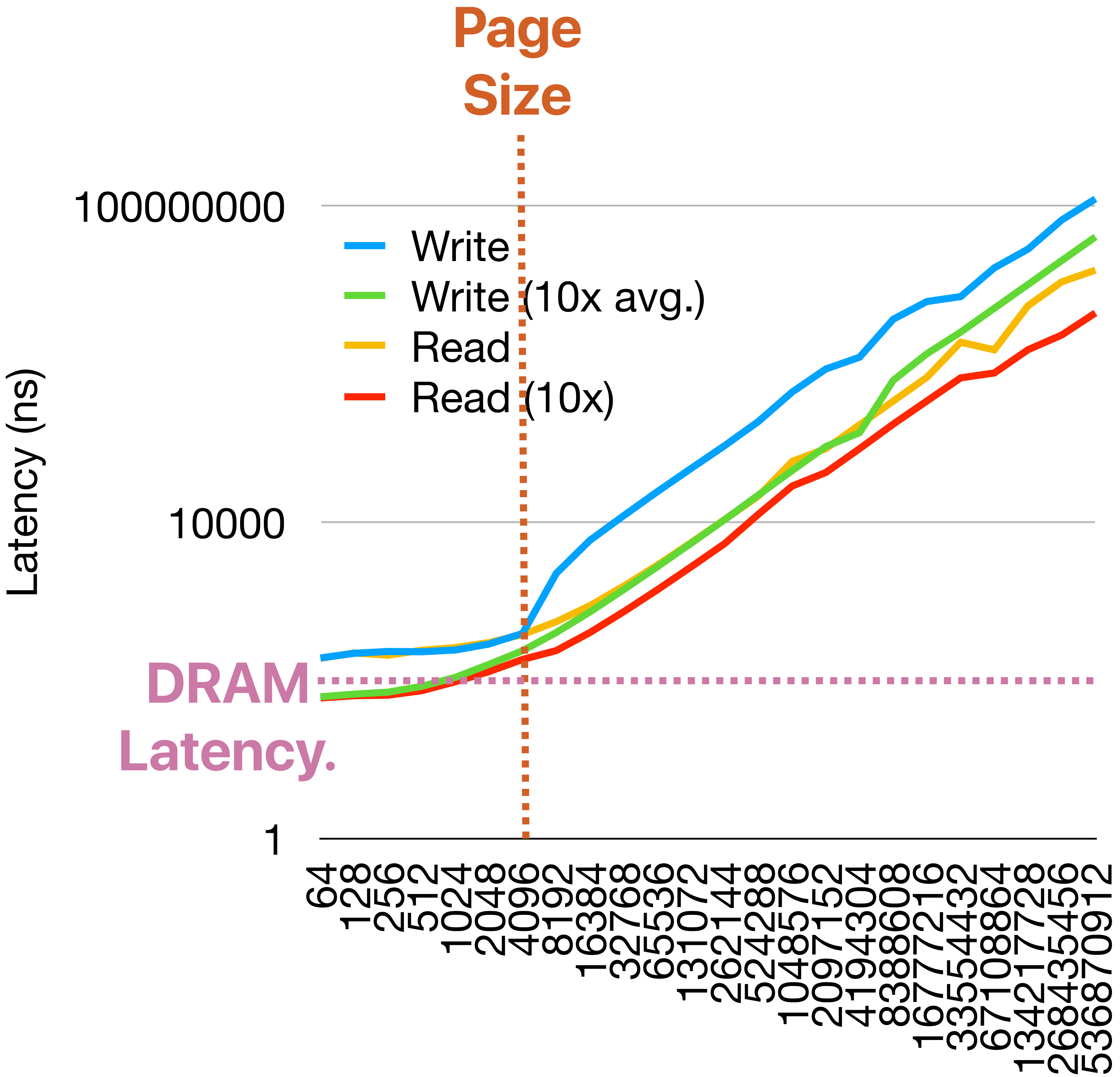
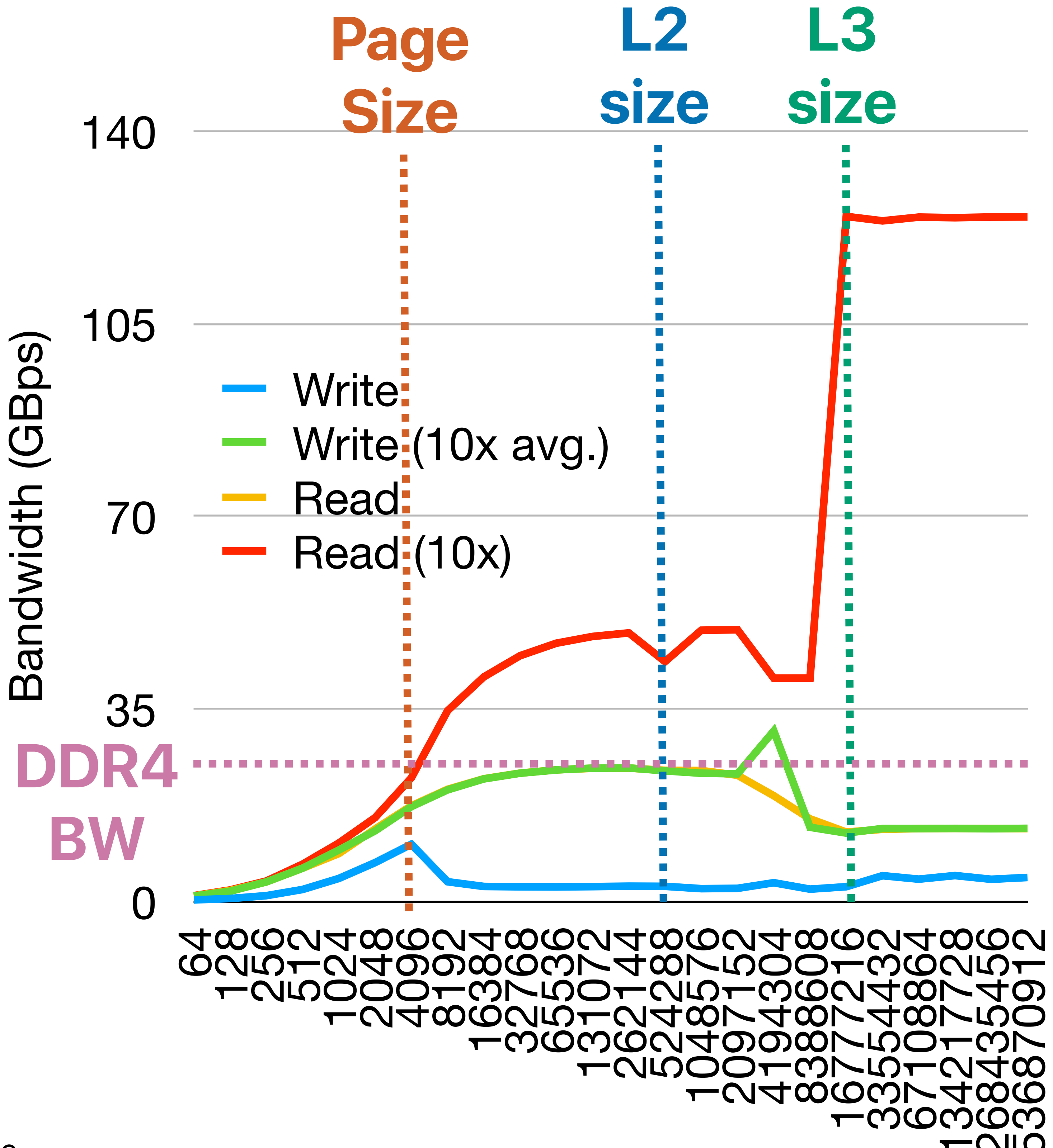
Modern Heterogeneous Computers: (6) Data Storage Systems

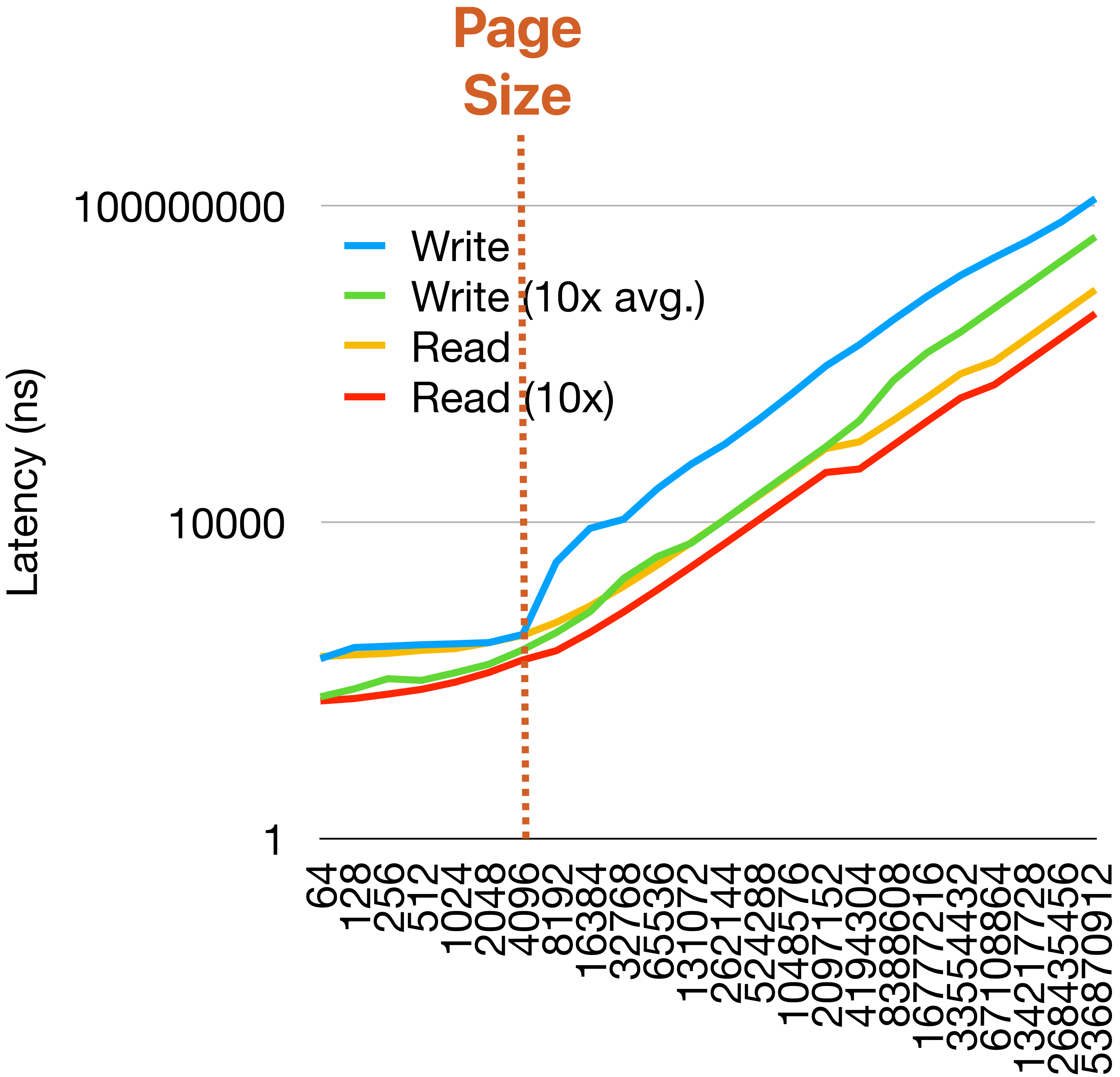
Hung-Wei Tseng

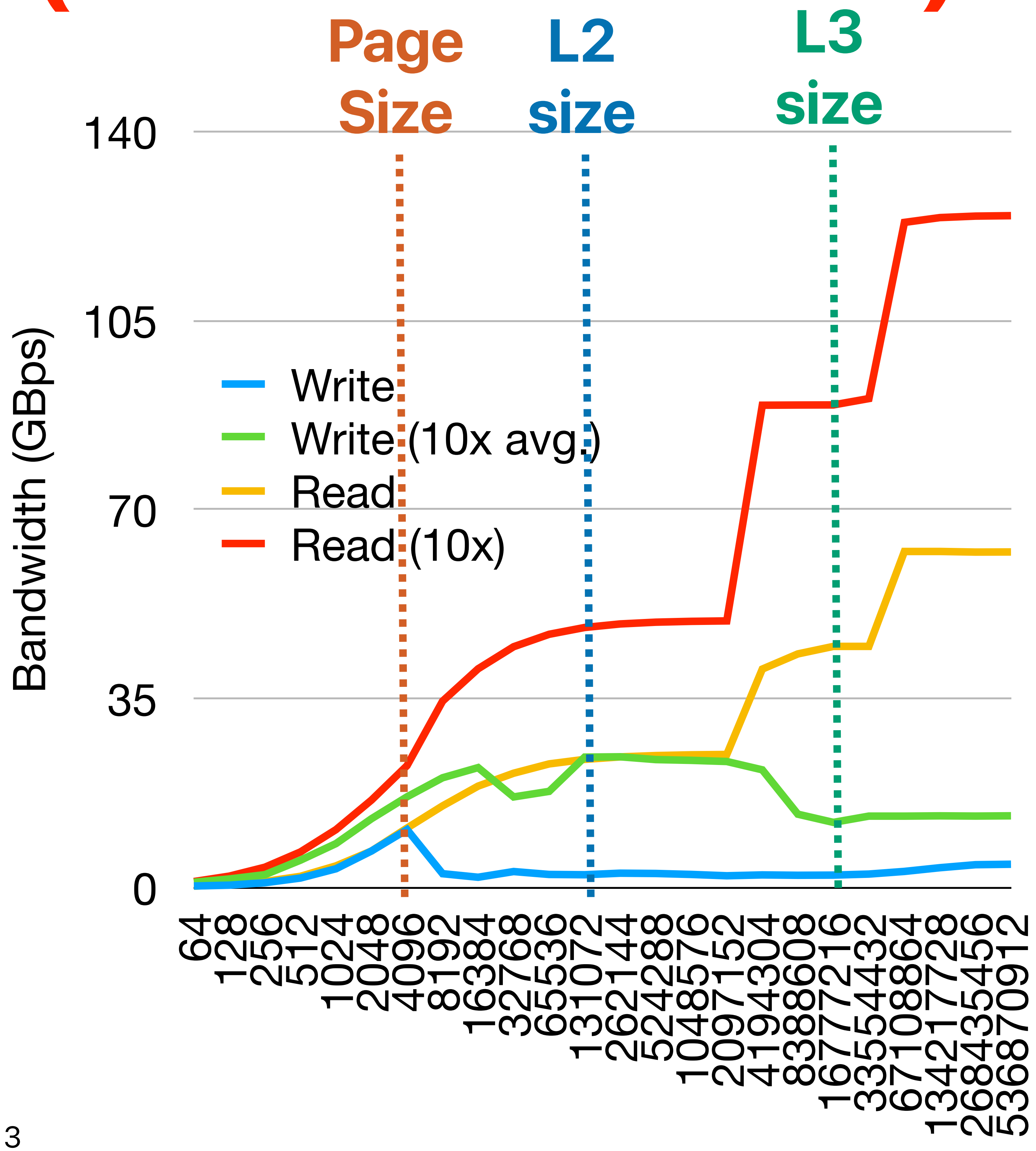
Corrected Performance Chart (on AMD RyZen 5 2600)



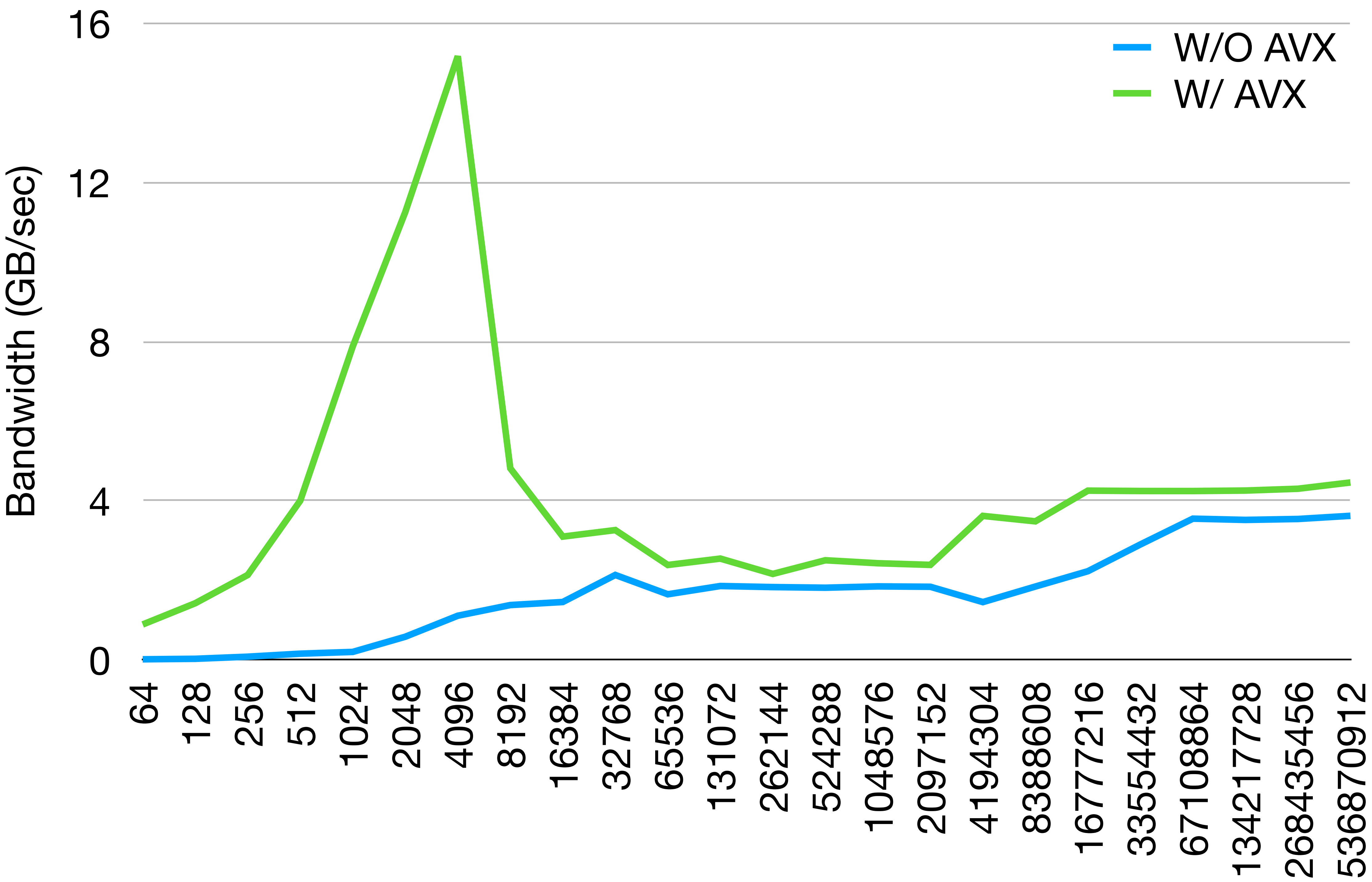


Performance Chart (on Core i5 12500)

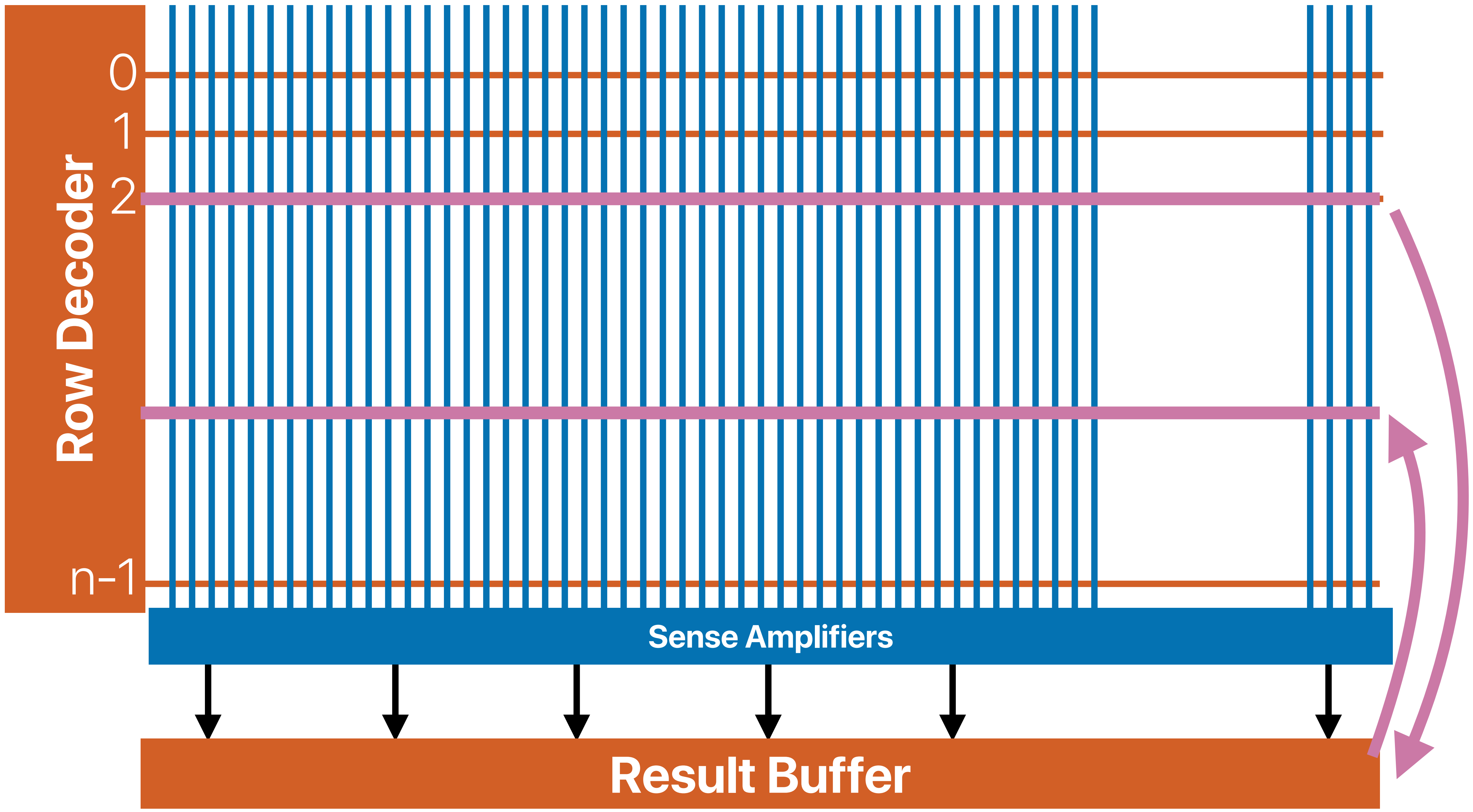




Memory copy bandwidth W/ AVX



Or we just clone/move?



memmove & memcpy: 5% cycles in Google’s datacenter

Svilen Kanev, Juan Pablo Darago, Kim Hazelwood, Parthasarathy Ranganathan, Tipp Moseley, Gu-Yeon Wei, and David Brooks.Profiling a warehouse-scale computer ISCA '15

Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. RowClone: fast and energy-efficient in-DRAM bulk data copy and initialization. In MICRO-46.

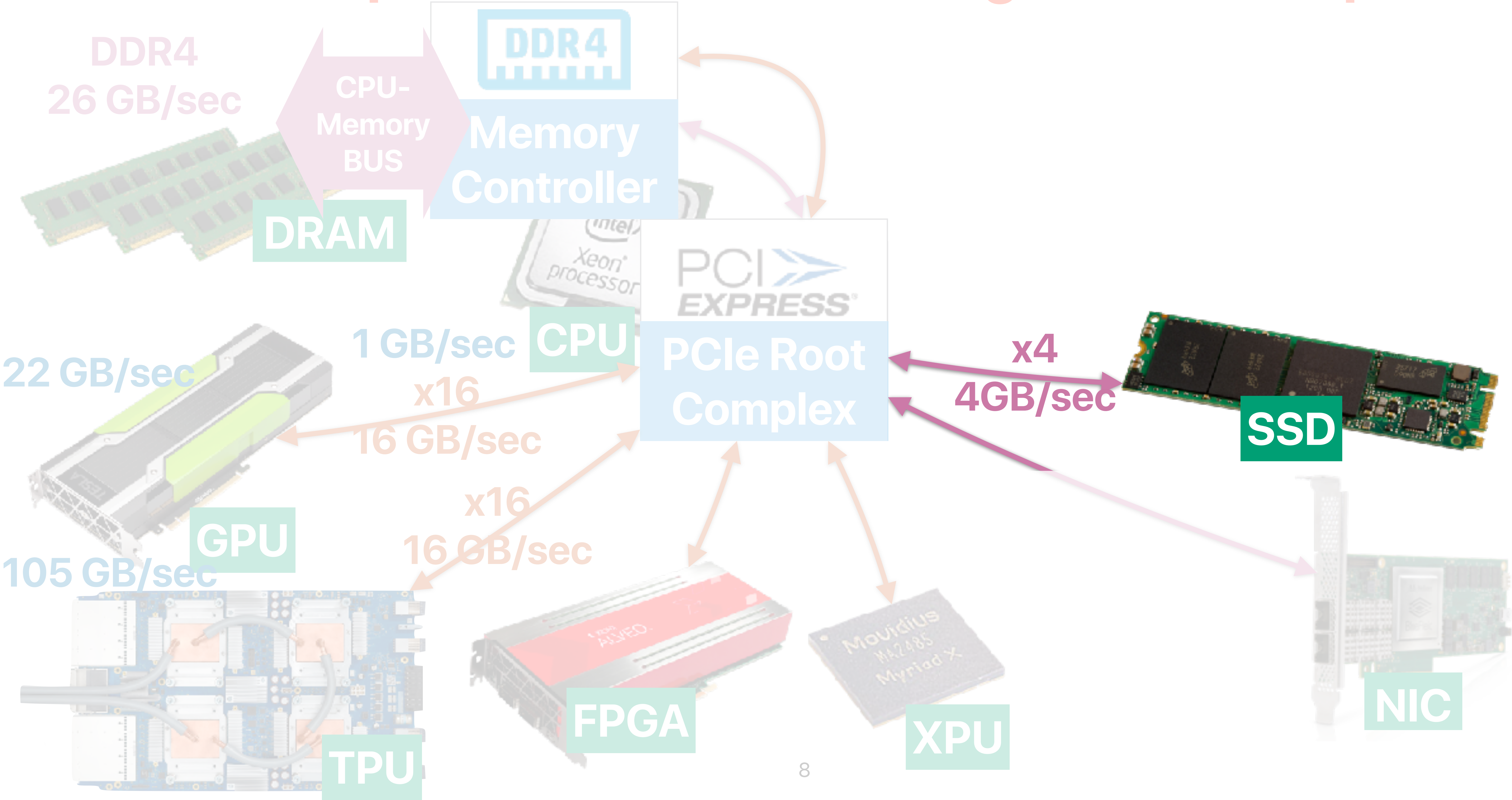
Implications

- The cost of loading a word is a lot
  - More than just a load — you need to calculate the effective address
  - That's why we want AVX to load 256-bit (32B or 4 64-bit words) to load in one instruction
- The cost of a page fault is significant
  - That's why we see the first write/read is a lot longer
  - Huge page can be helpful
- Performance optimization in software is hard!!!

How can we lower data volume

- Compression
  - Too much computation overhead if no accelerator is presented
- Near/In-memory processing
  - Embed “logic”/“intelligence” near memory locations
  - Will talk about this later!

The "data path" in modern heterogeneous computers



3:00

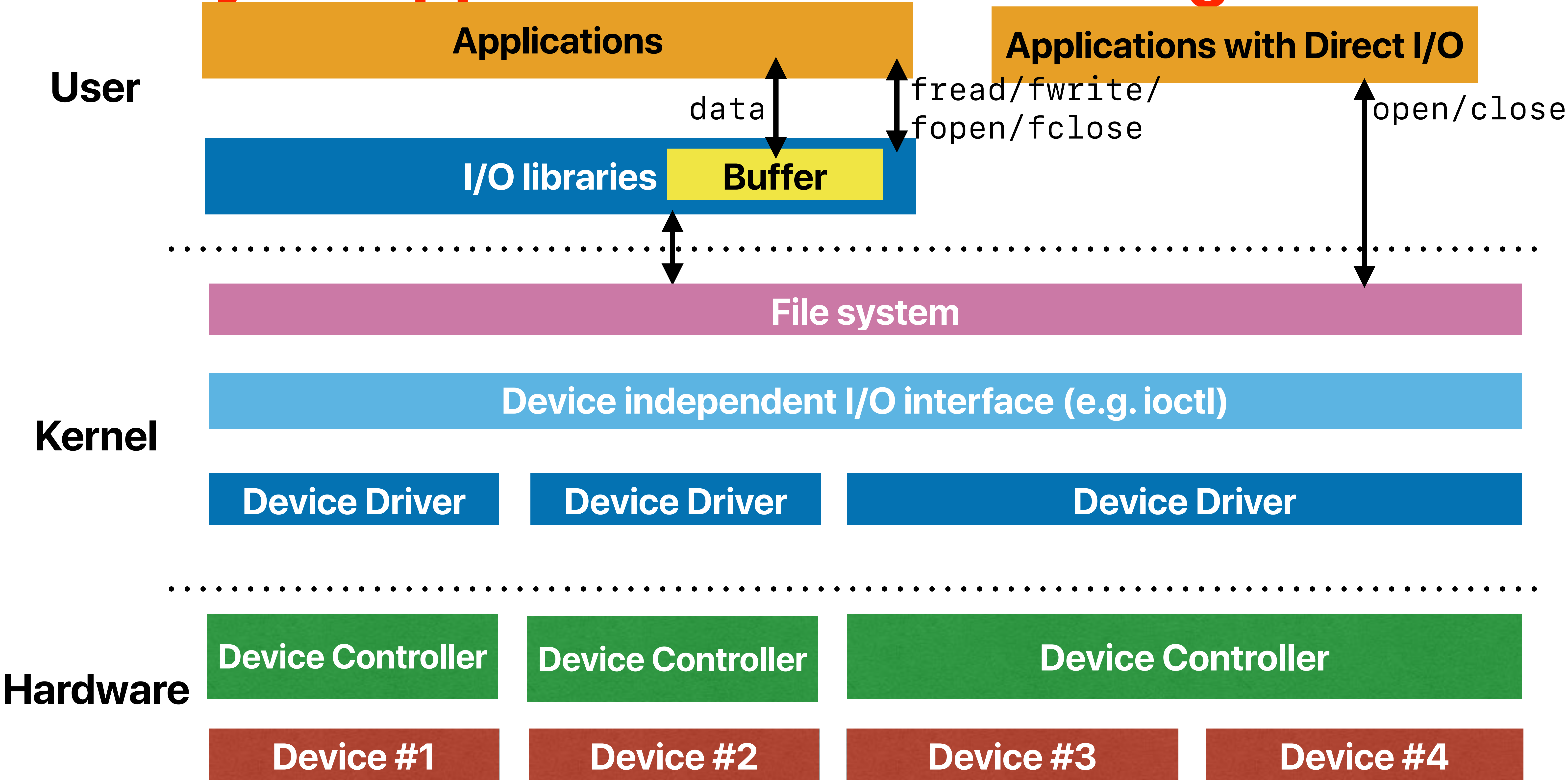
Regarding an SSD, or in general, storage devices, what do you have in mind? (everything you can think of)

Data Storage — particularly, modern SSDs

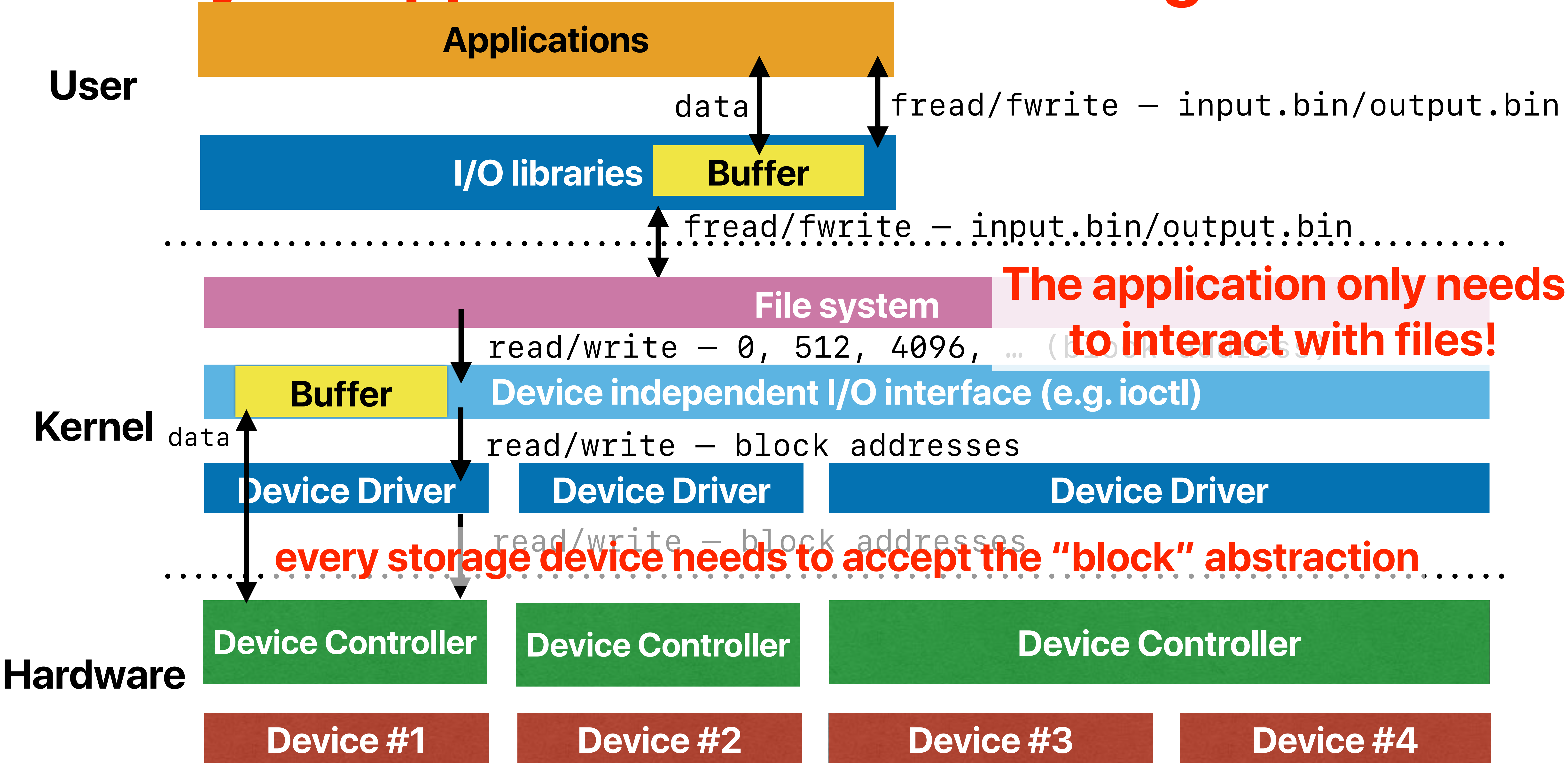
3:00

How does your system interact with a storage device?

How your application reaches storage devices



How your application reaches storage devices



User

Applications

data

fread/fwrite — input.bin/output.bin

I/O libraries

Buffer

fread/fwrite — input.bin/output.bin

File system

read/write — 0, 512, 4096, … (block address)

Kernel

Buffer

Device independent I/O interface (e.g. ioctl)

data

read/write — block addresses

Device Driver

Device Driver

Device Driver

read/write — block addresses

Hardware

Device Controller

Device Controller

Device Controller

Device #1

Device #2

Device #3

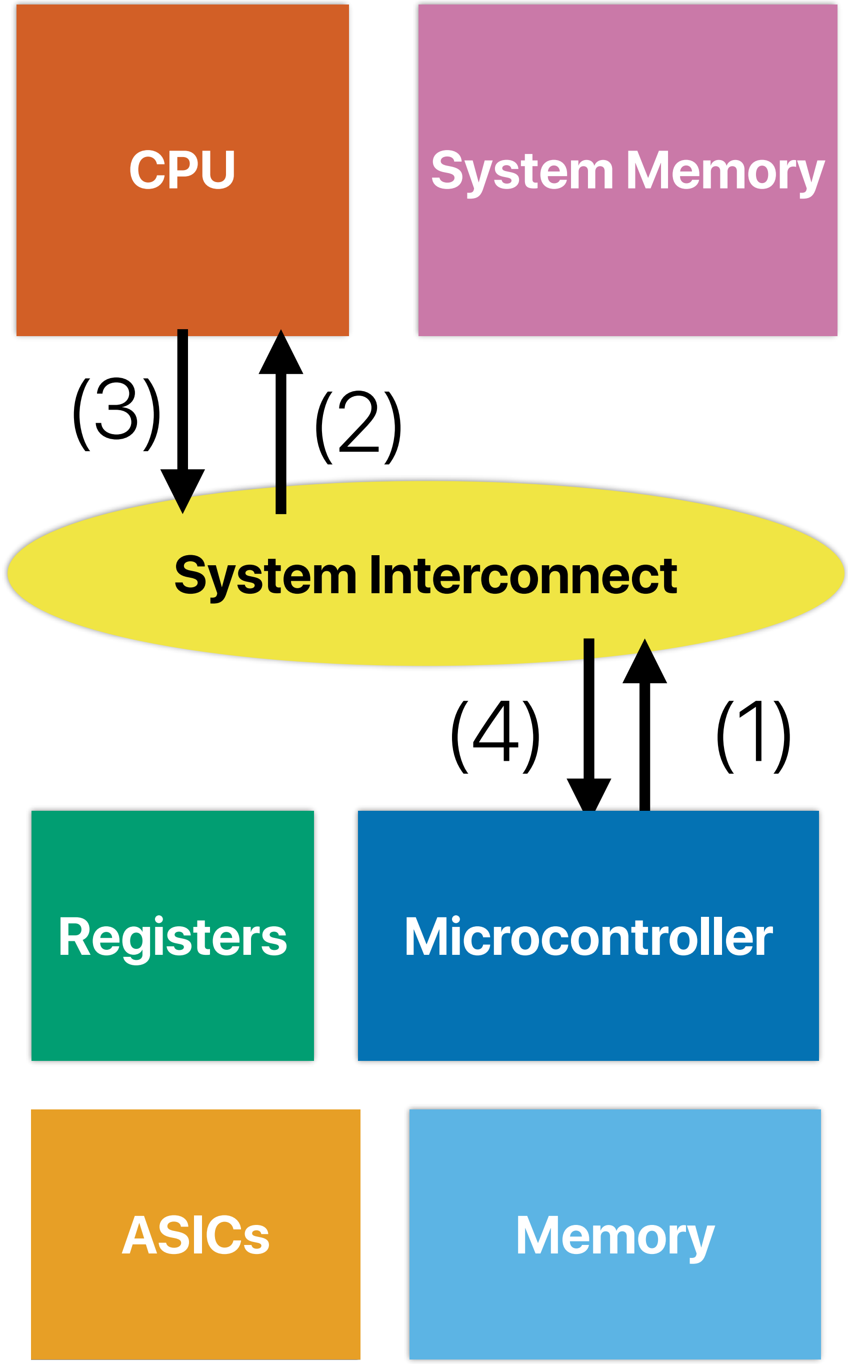
Device #4

The application only needs to interact with files!

every storage device needs to accept the “block” abstraction

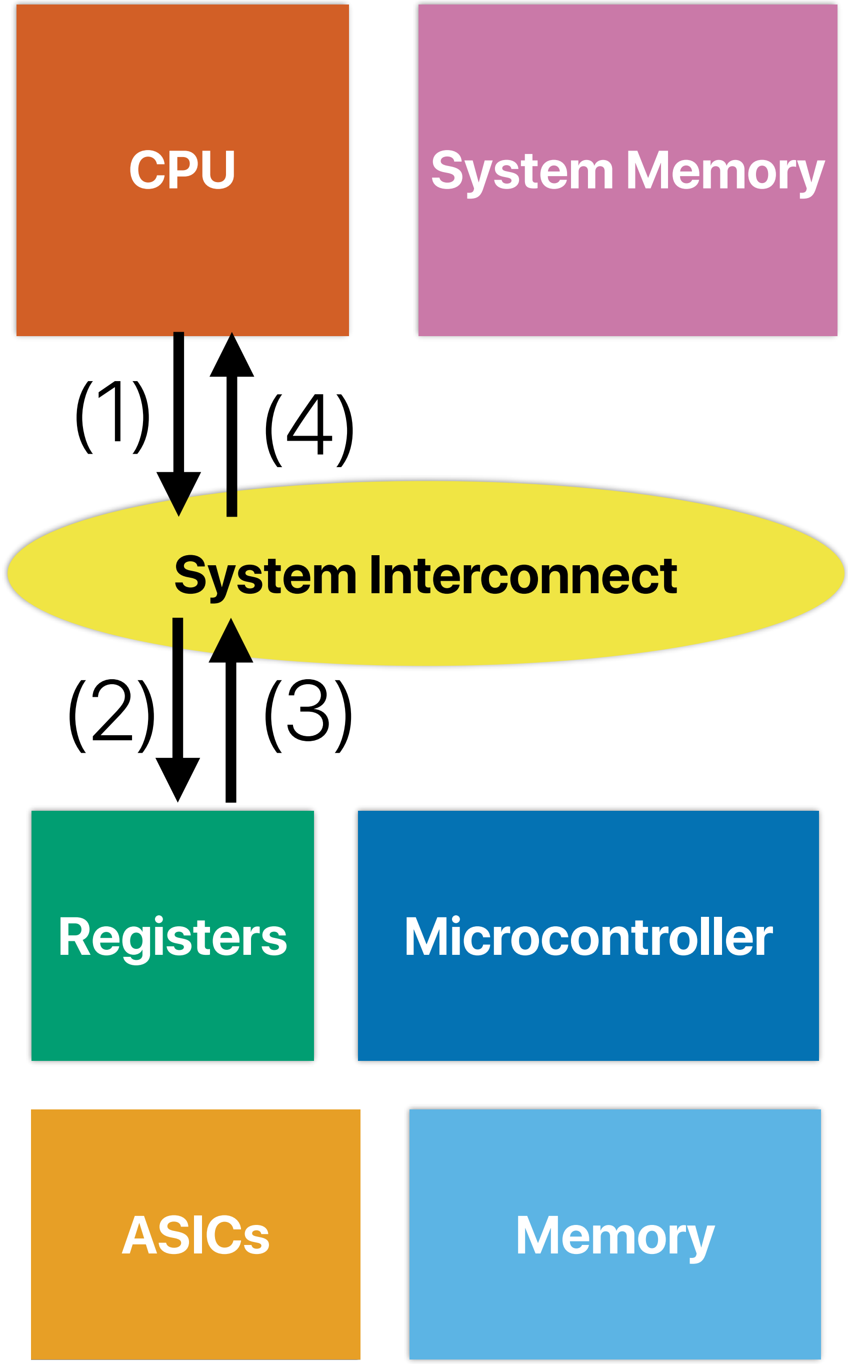
Interrupt

- The device signals the processor only when the device requires the processor/OS handle some tasks/data
- The processor only signals the device when necessary



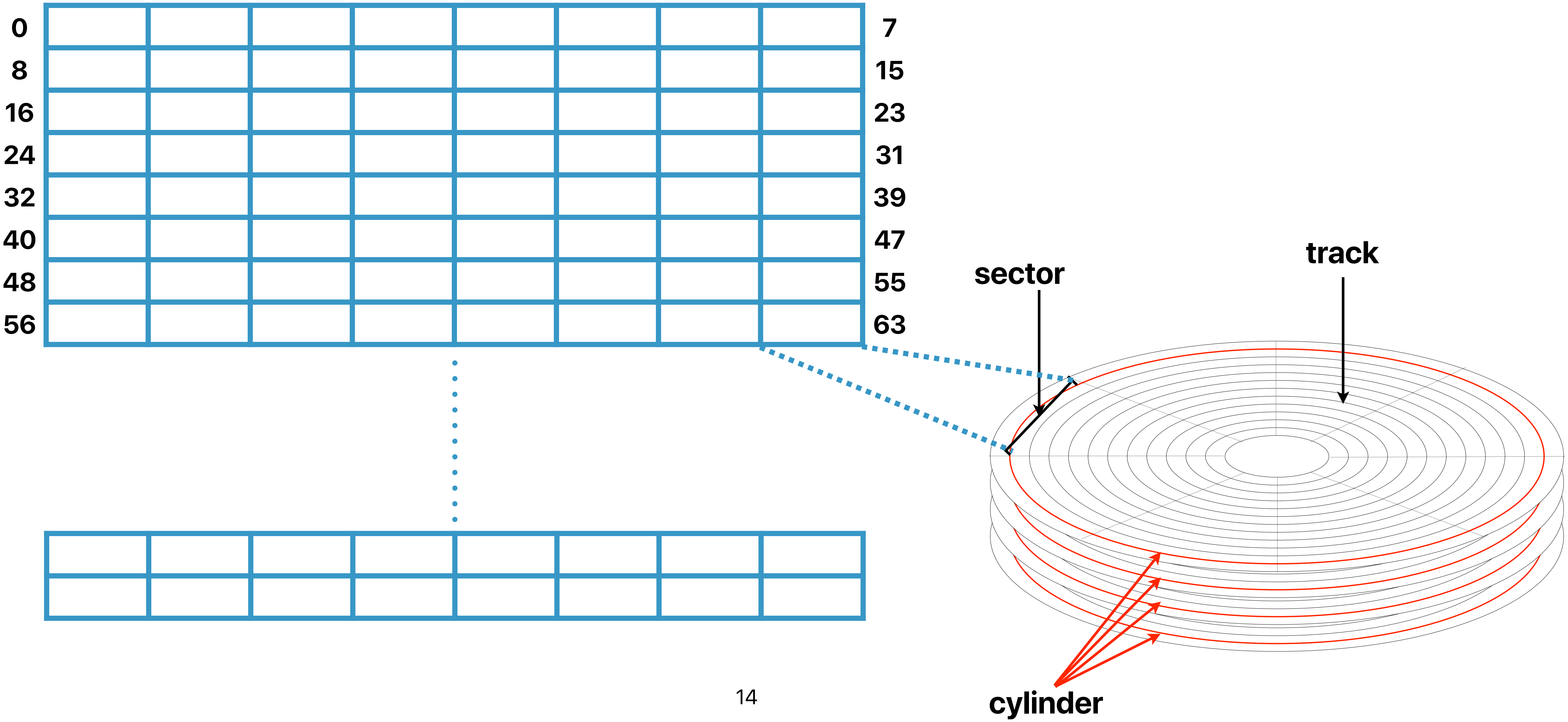
Polling

- The processor/OS constantly asks if the device (e.g. examine the status register of the device) is ready to or requires the processor/OS handle some tasks/data
- The OS/processor executes corresponding handler if the device can handle demand tasks/data or has tasks/data ready

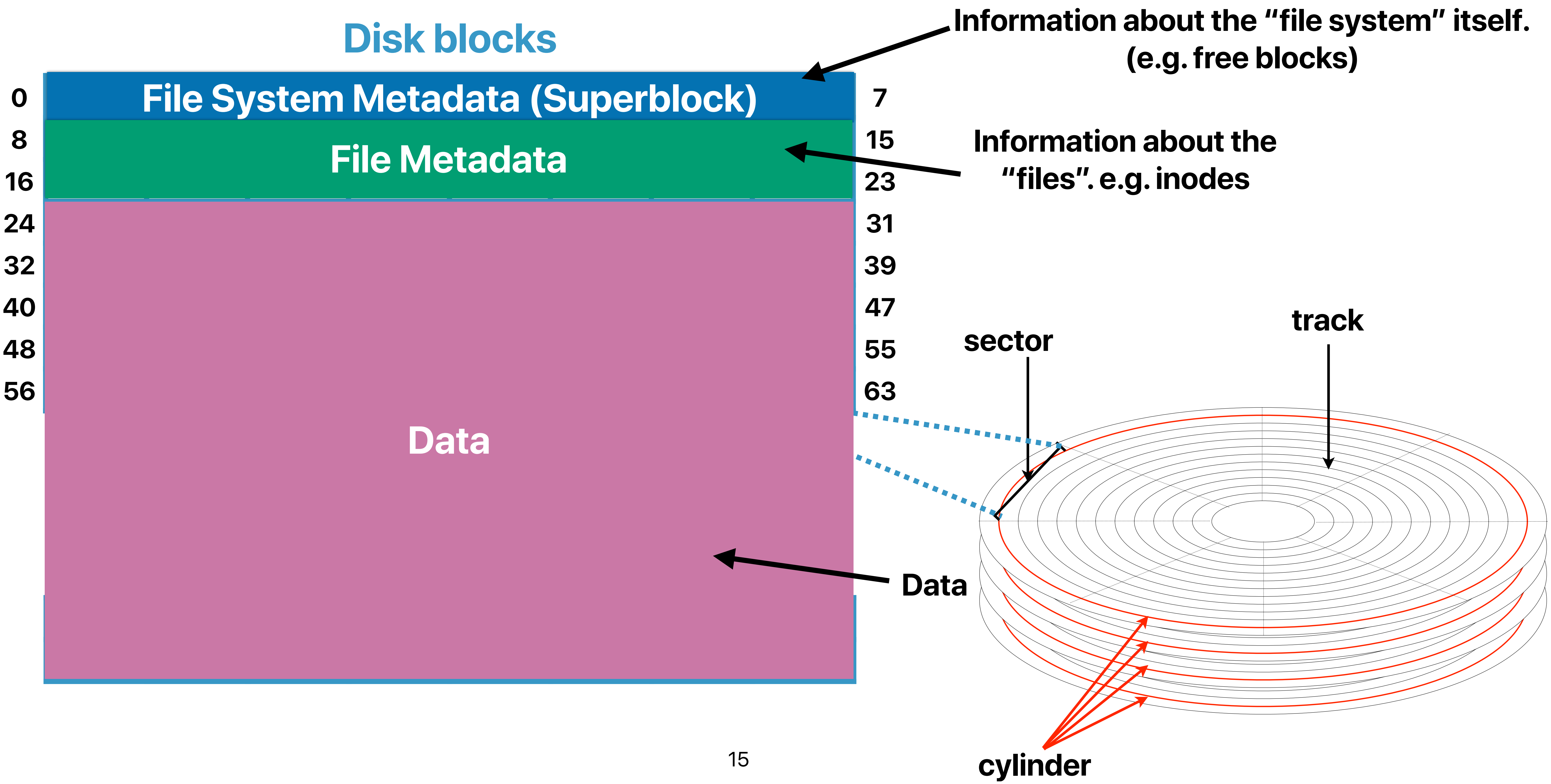


Numbering the disk space with block addresses

Disk blocks



How the original UNIX file system use disk blocks

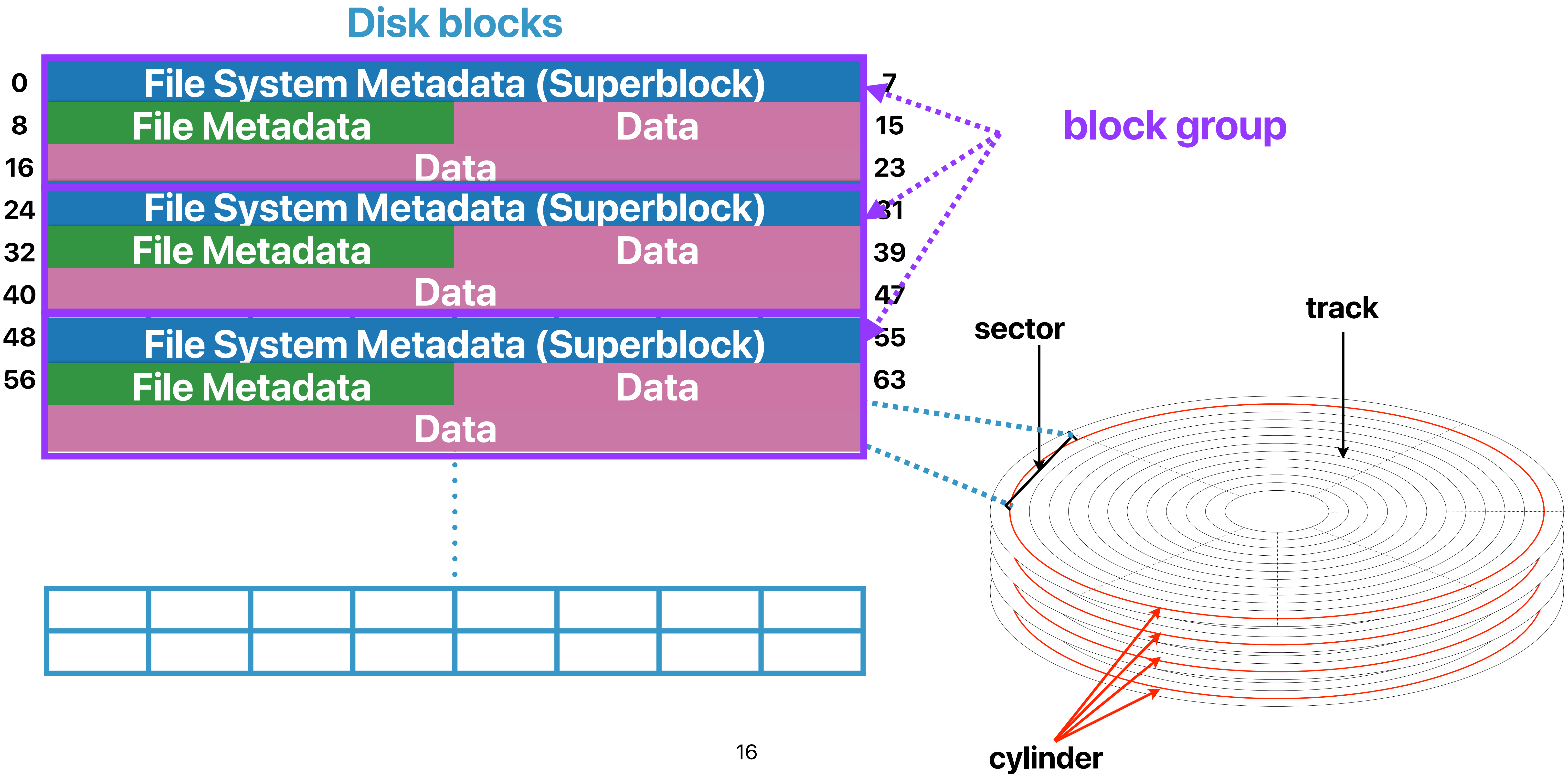


Information about the "file system" itself.
(e.g. free blocks)

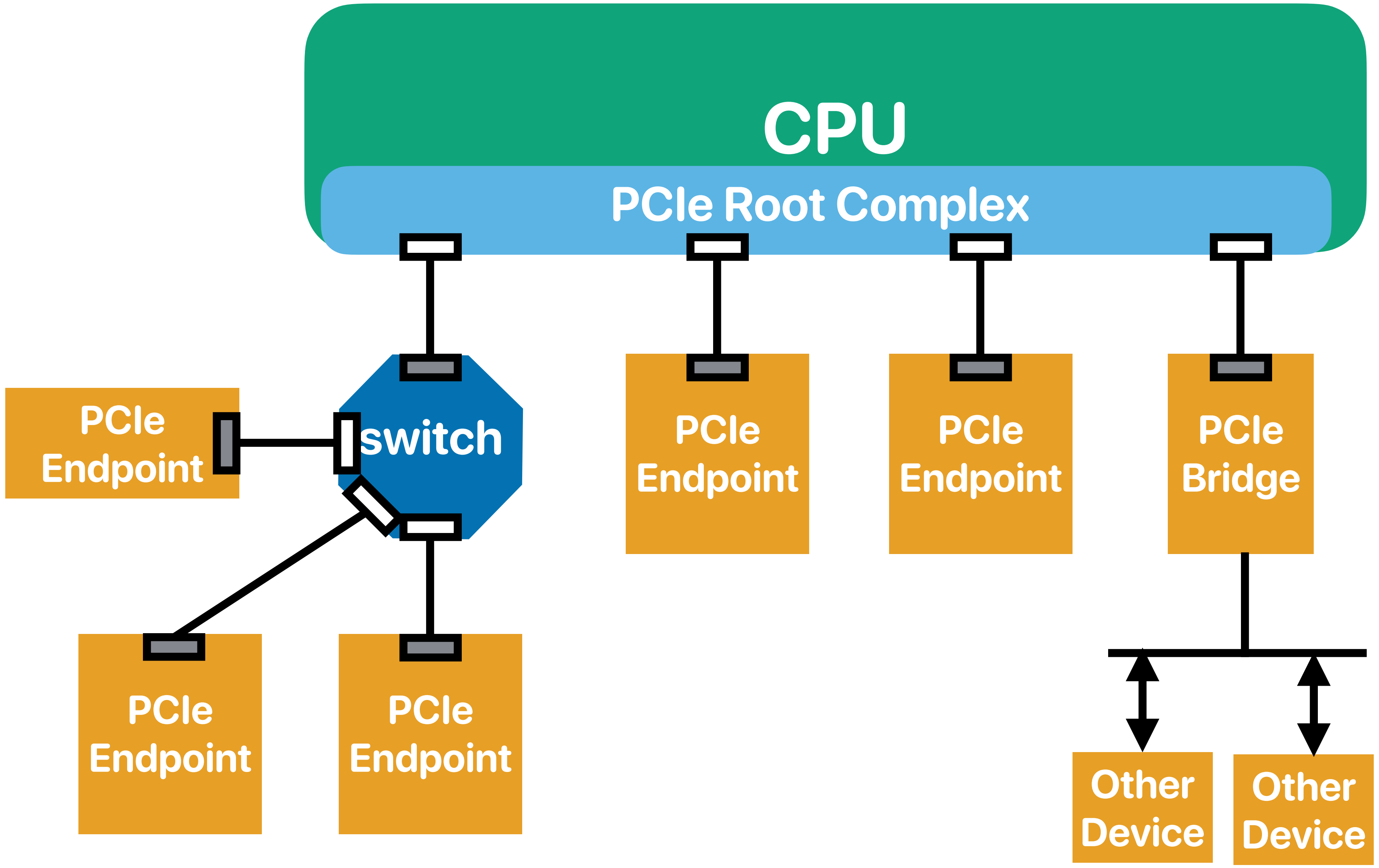
Information about the
"files". e.g. inodes

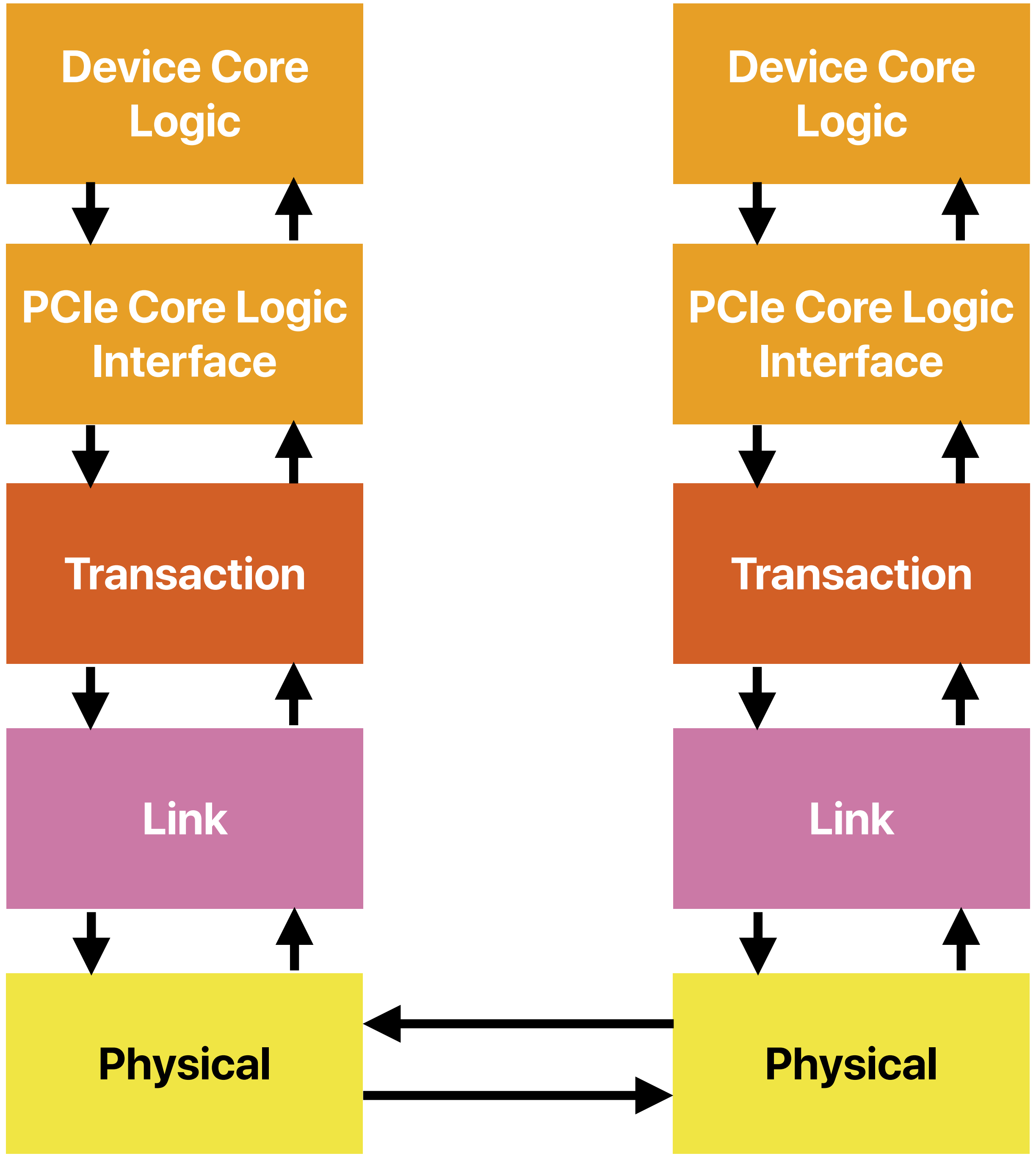
Data

How ExtFS use disk blocks



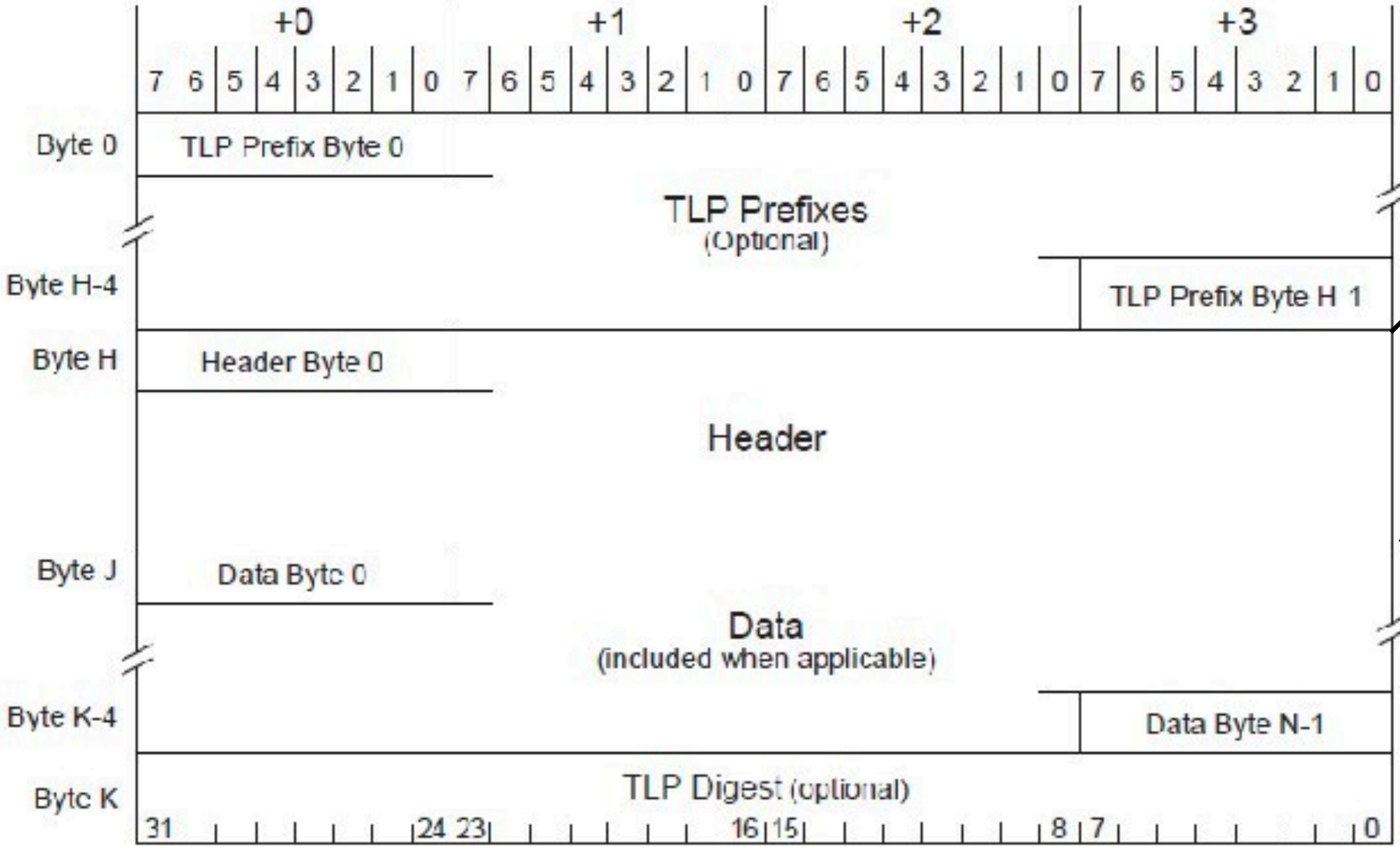
PCIe "Interconnect"

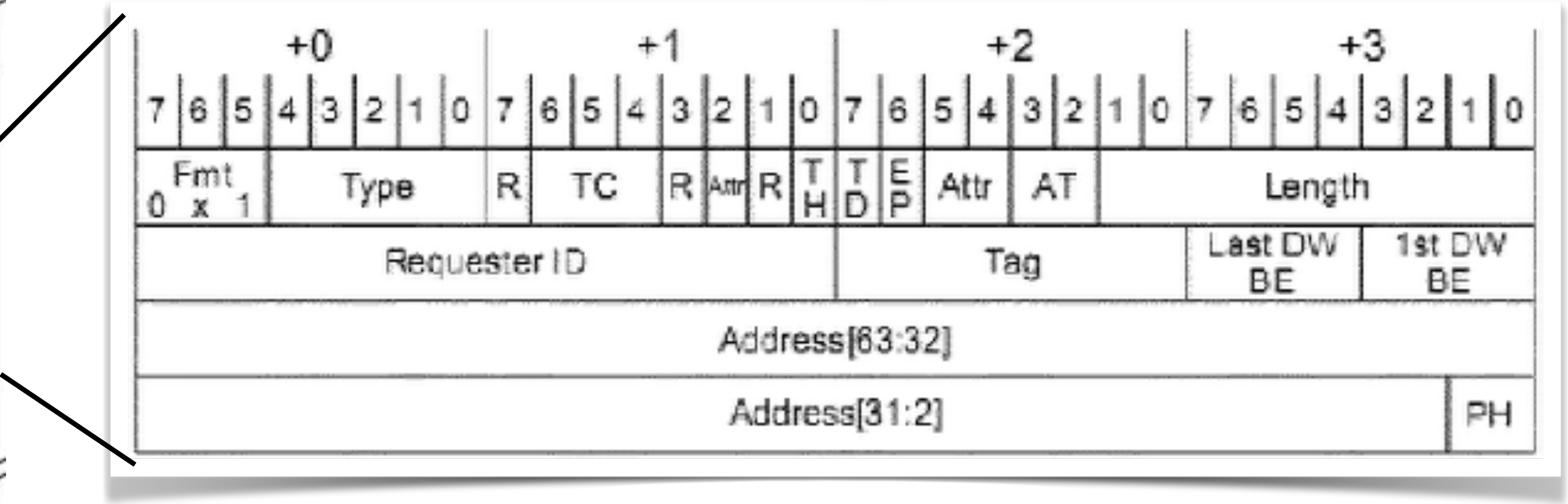




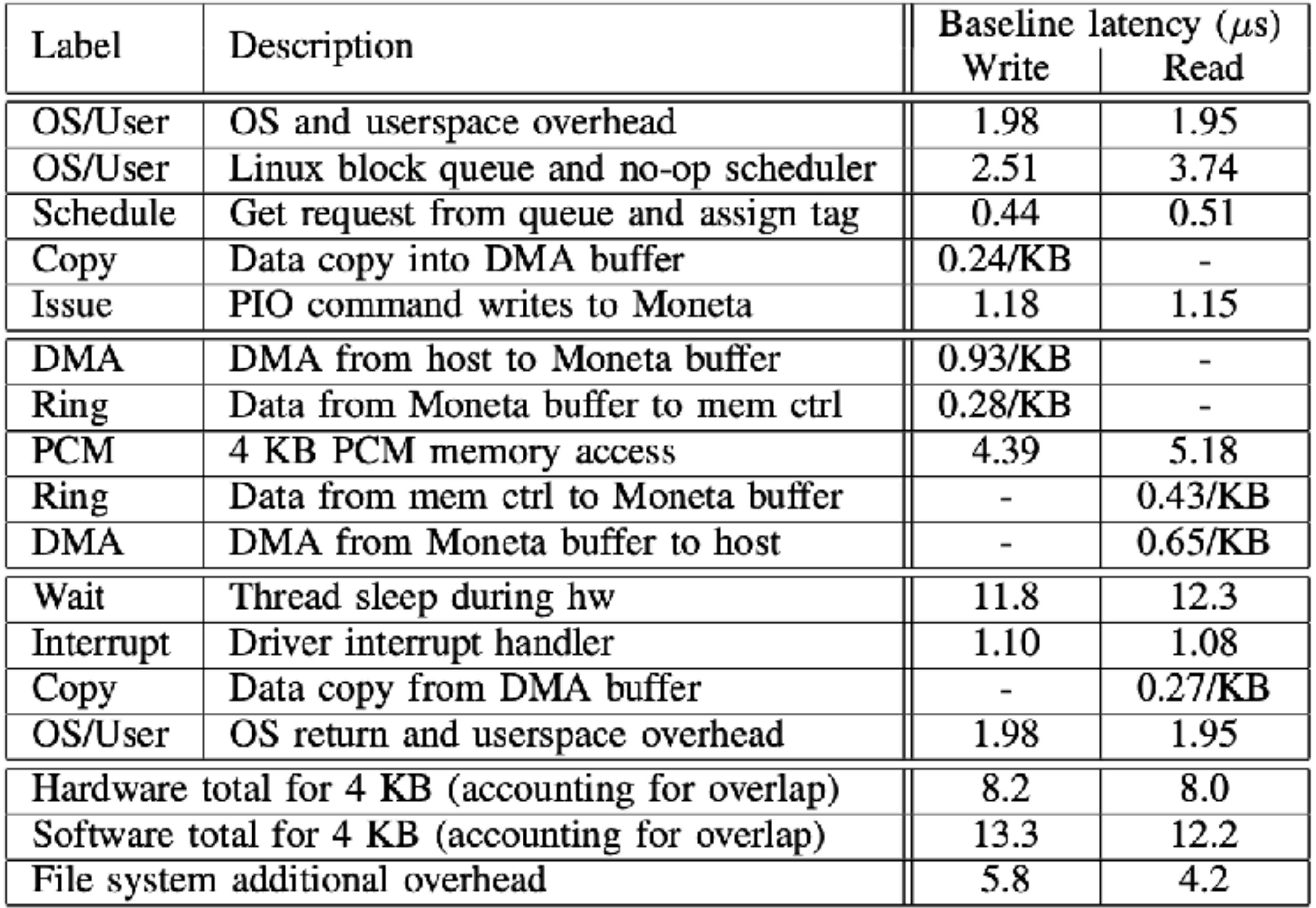
PCIe interconnect

- Very similar to computer networks
- Use "memory addresses" as the identifier for routing
- Peer-to-peer communication is possible

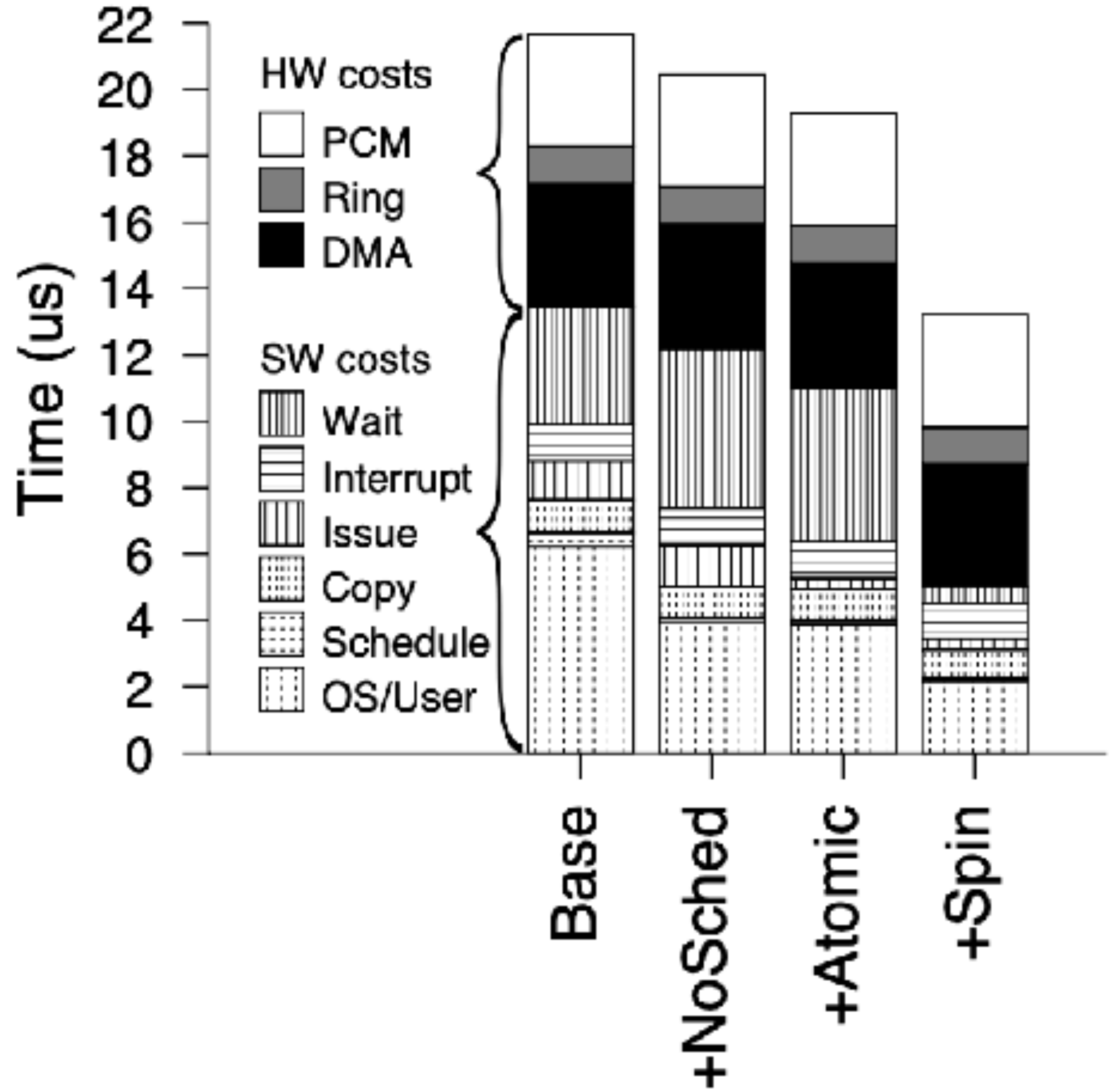




Software overhead



Label	Description	Baseline latency (μs)	
		Write	Read
OS/User	OS and userspace overhead	1.98	1.95
OS/User	Linux block queue and no-op scheduler	2.51	3.74
Schedule	Get request from queue and assign tag	0.44	0.51
Copy	Data copy into DMA buffer	0.24/KB	-
Issue	PIO command writes to Moneta	1.18	1.15
DMA	DMA from host to Moneta buffer	0.93/KB	-
Ring	Data from Moneta buffer to mem ctrl	0.28/KB	-
PCM	4 KB PCM memory access	4.39	5.18
Ring	Data from mem ctrl to Moneta buffer	-	0.43/KB
DMA	DMA from Moneta buffer to host	-	0.65/KB
Wait	Thread sleep during hw	11.8	12.3
Interrupt	Driver interrupt handler	1.10	1.08
Copy	Data copy from DMA buffer	-	0.27/KB
OS/User	OS return and userspace overhead	1.98	1.95
Hardware total for 4 KB (accounting for overlap)		8.2	8.0
Software total for 4 KB (accounting for overlap)		13.3	12.2
File system additional overhead		5.8	4.2



A. M. Caulfield, A. De, J. Coburn, T. I. Mollow, R. K. Gupta and S. Swanson, "Moneta: A High-Performance Storage Array Architecture for Next-Generation, Non-volatile Memories," 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture, 2010, pp. 385-395, doi: 10.1109/MICRO.2010.33.

NVMe

- The standard of PCIe SSD devices now
  - Provides multiple command queues to better support multithreading hardware
  - Allows more parallelism inside the SSD
- The “payload” of a PCIe packet

0	8	16	32	48	63
OPCODE	FLAGS	command id	Namespace ID		
reserved					
metadata					
PRP1					
PRP2					
Start LBA					
length		control	Dataset management		
Reference tag			App tag	App mask	

Let’s walk through an NVMe driver

Electrical
Computer Science
Engineering

277

つづく