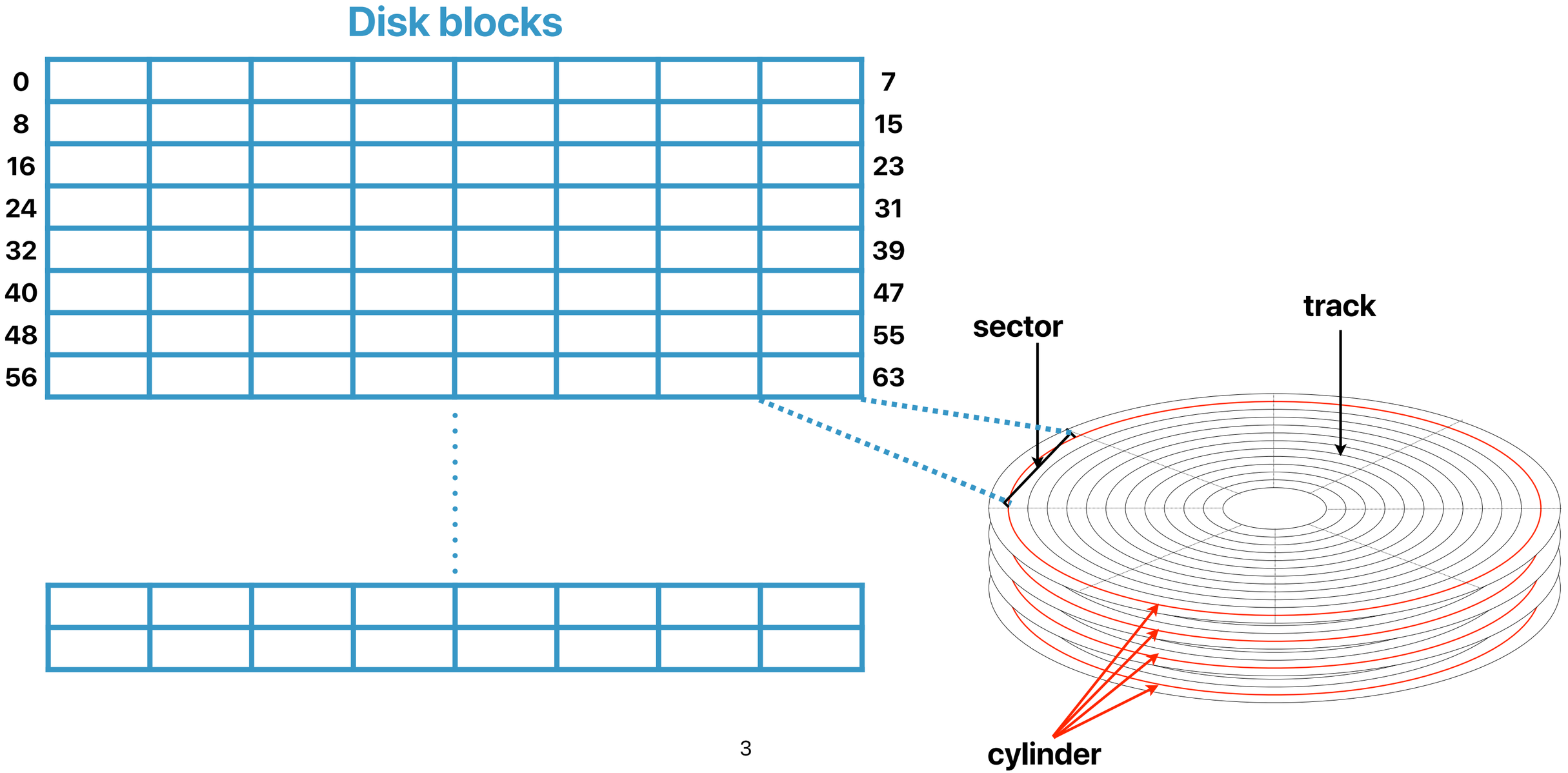
Streamline Data Exchanges & In-Storage Processing (1)

Hung-Wei Tseng

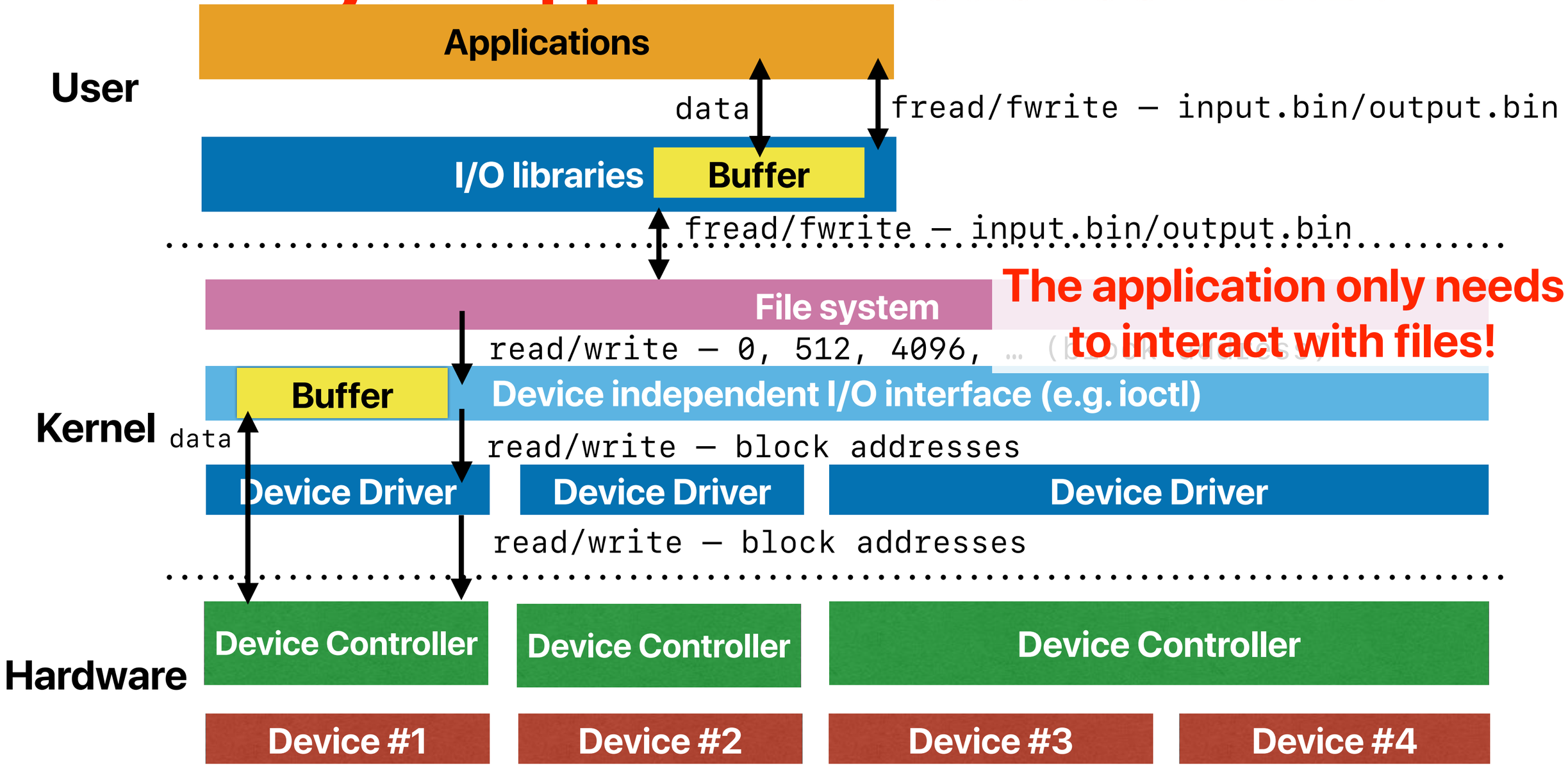
Polling/Interrupt/DMA/PIO?

	Polling	Interrupt
DMA	CPU OPs more than IRQ Shorter latency High throughput	Lowest CPU Operations Longer latency High throughput
PIO	Most CPU operations Shorter latency Low throughput	CPU OPs more than IRQ Longer latency Low throughput

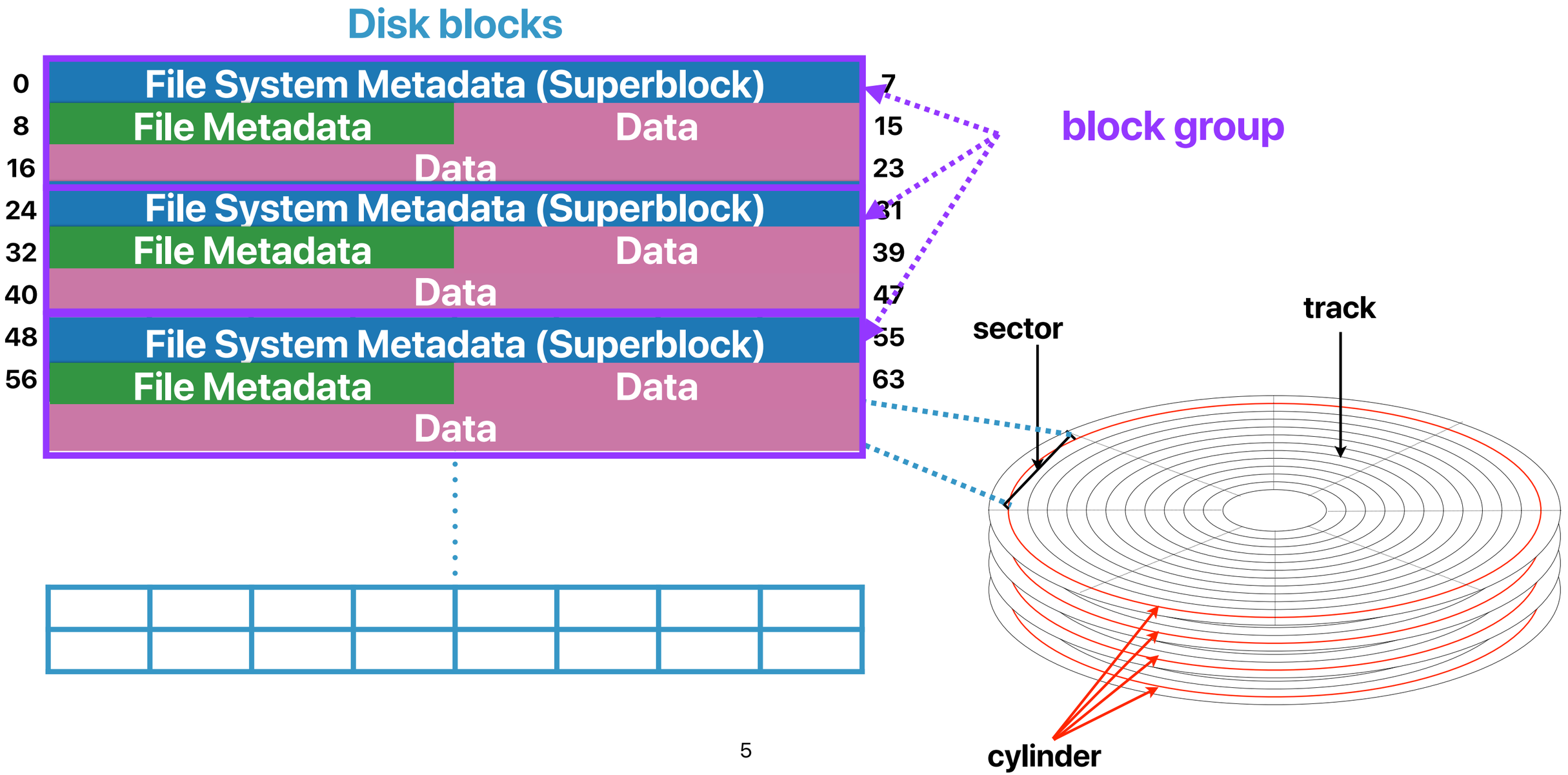
Numbering the disk space with block addresses



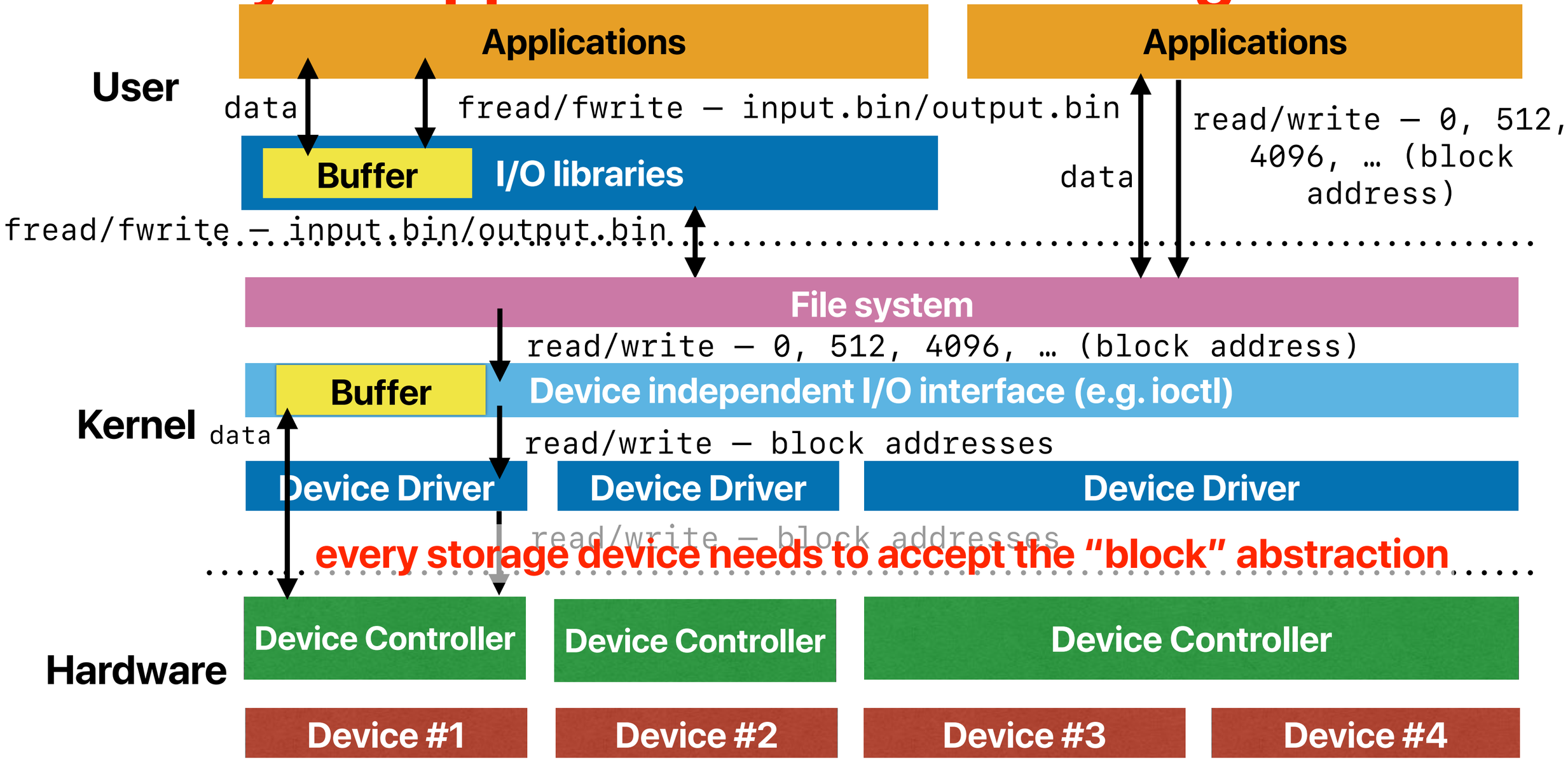
How your application reaches H.D.D.



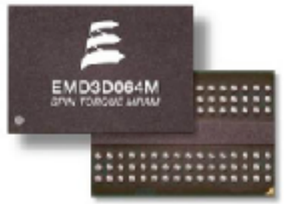
How ExtFS use disk blocks



How your application reaches storage devices



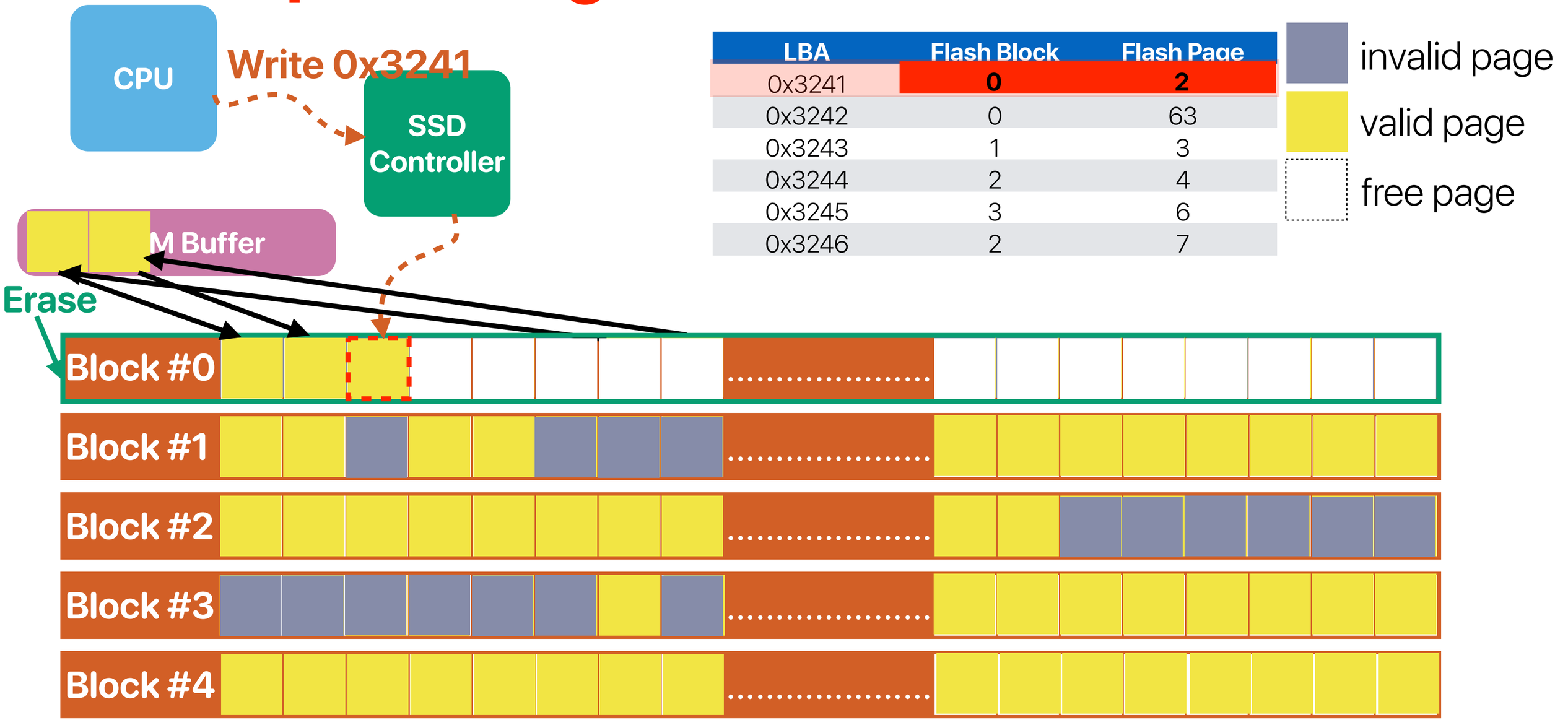
Non-volatile memory technologies



	H.D.D	Flash	Optane	STT-MRAM
Latency	~ 10-15 ms	~ 100 us (read) ~ 1 ms (write)	7 us (read) 18 us (write)	35 ns
Bandwidth	~200 MB/Sec	3.5 GB/sec (read) 2.1 GB/sec (write)	1.35 GB/sec (read) 290 MB/sec (write)	
Dollar/GB	0.0295	0.583	2.18	

Flash is still the most convincing technology for now

Recap: Garbage Collection in Flash SSDs



LBA	Flash Block	Flash Page
0x3241	**0**	**2**
0x3242	0	63
0x3243	1	3
0x3244	2	4
0x3245	3	6
0x3246	2	7

File systems for flash-based SSDs

- Still an open research question
- Software designer should be aware of the characteristics of underlying hardware components
- Revising the layered design to expose more SSD information to the file system or the other way around

Apple M1 Macs appear to be chewing through their SSDs

By Alan Dexter 4 hours ago

The same SSDs that are soldered-in and almost impossible to rep

COMMENTS

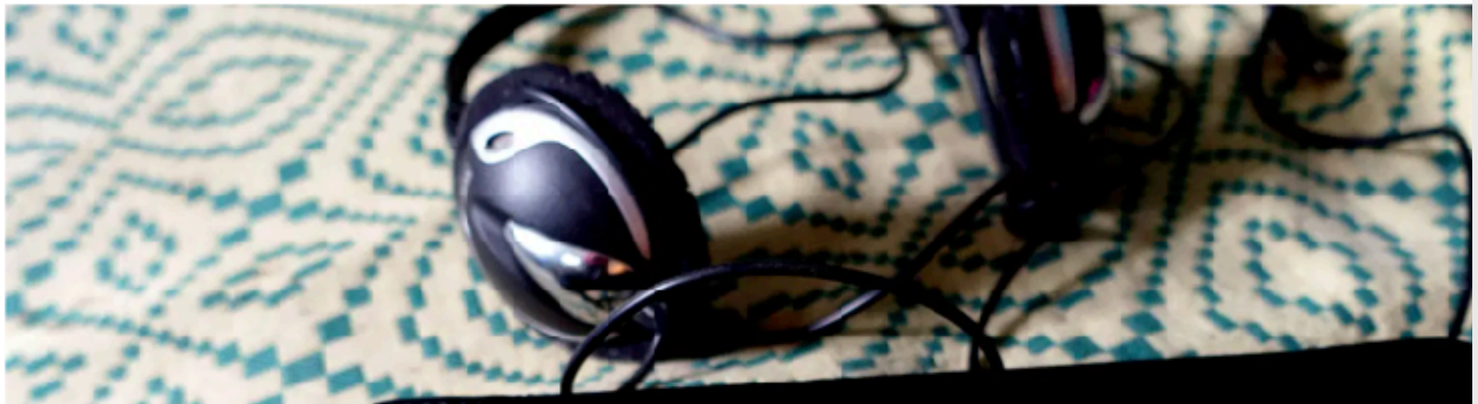
(Image credit: Apple)

Spotify is

BGR TECH ENTERTAINMENT DEALS BUSINESS

TECH

Spotify has been quietly killing your SSD's life



The tasks of an SSD controller/firmware

Parse NVMe/SATA/PCIe commands

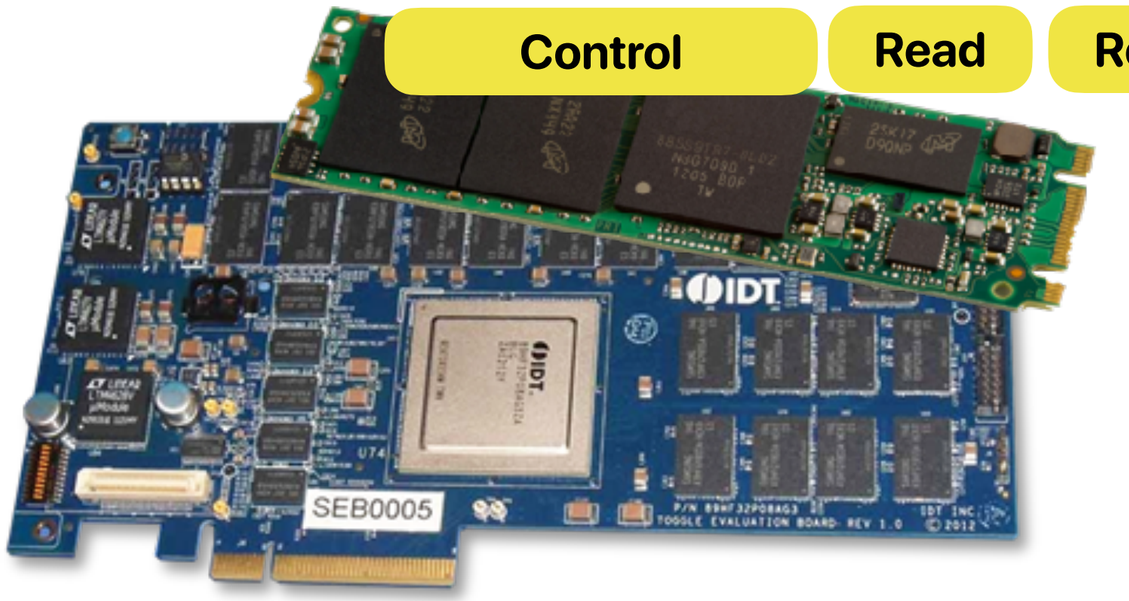
Address mapping/lookup

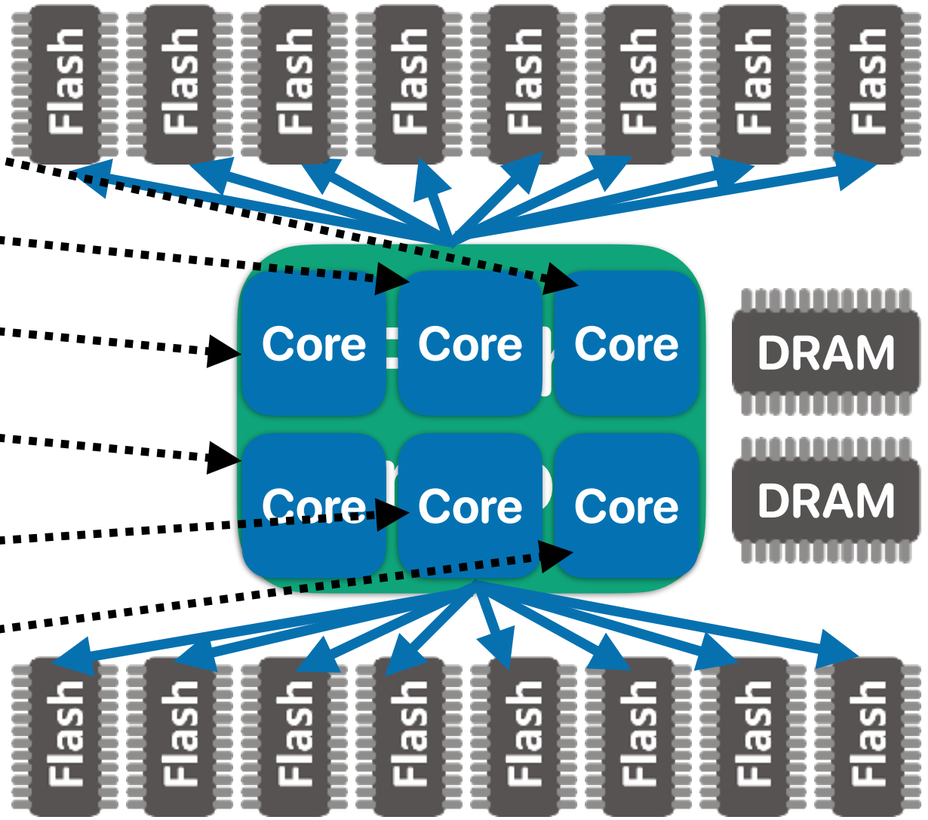
Wear-leveling/Garbage Collection

Program | Read | Read

Program | Read | Read

Control | Read | Read

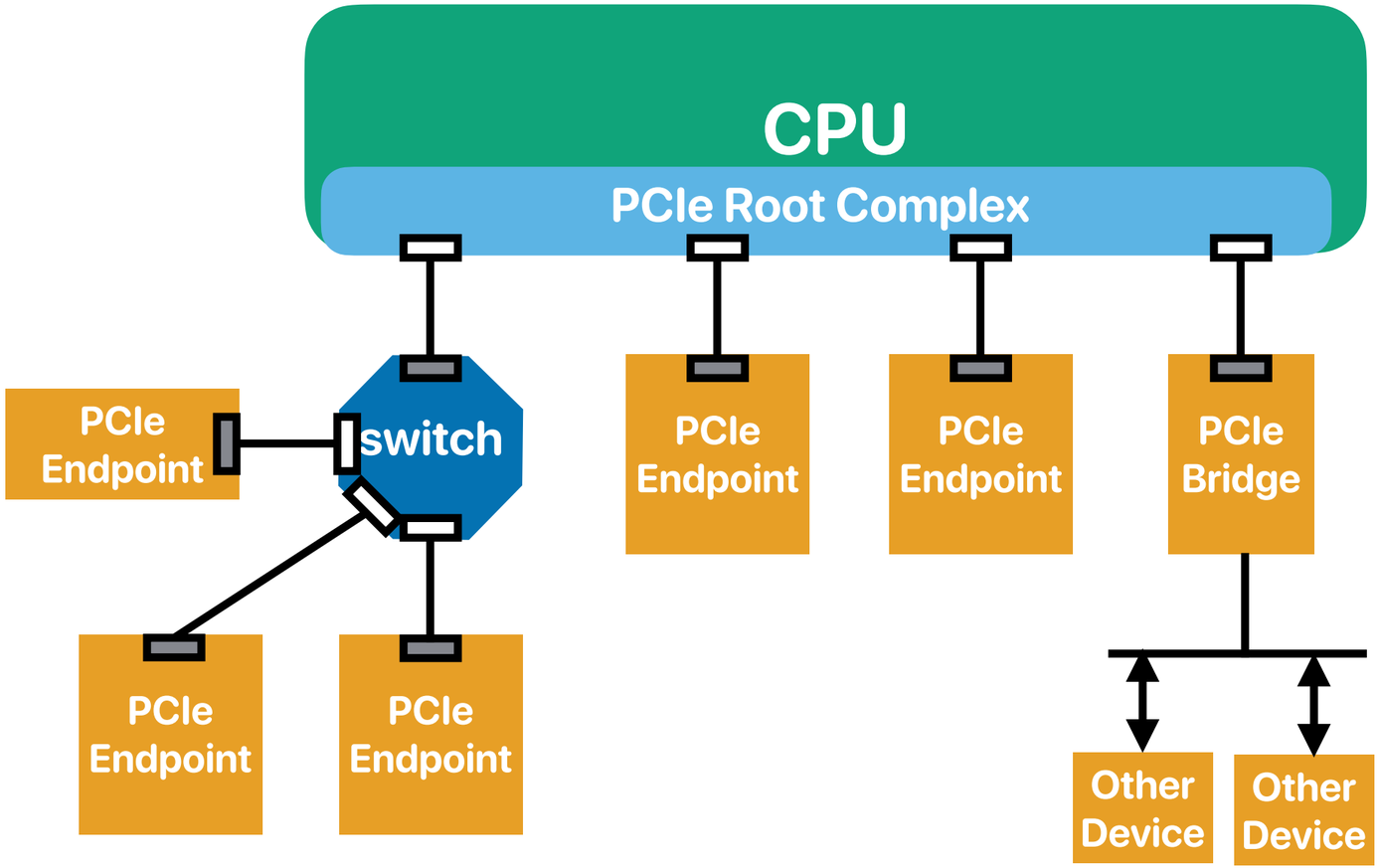


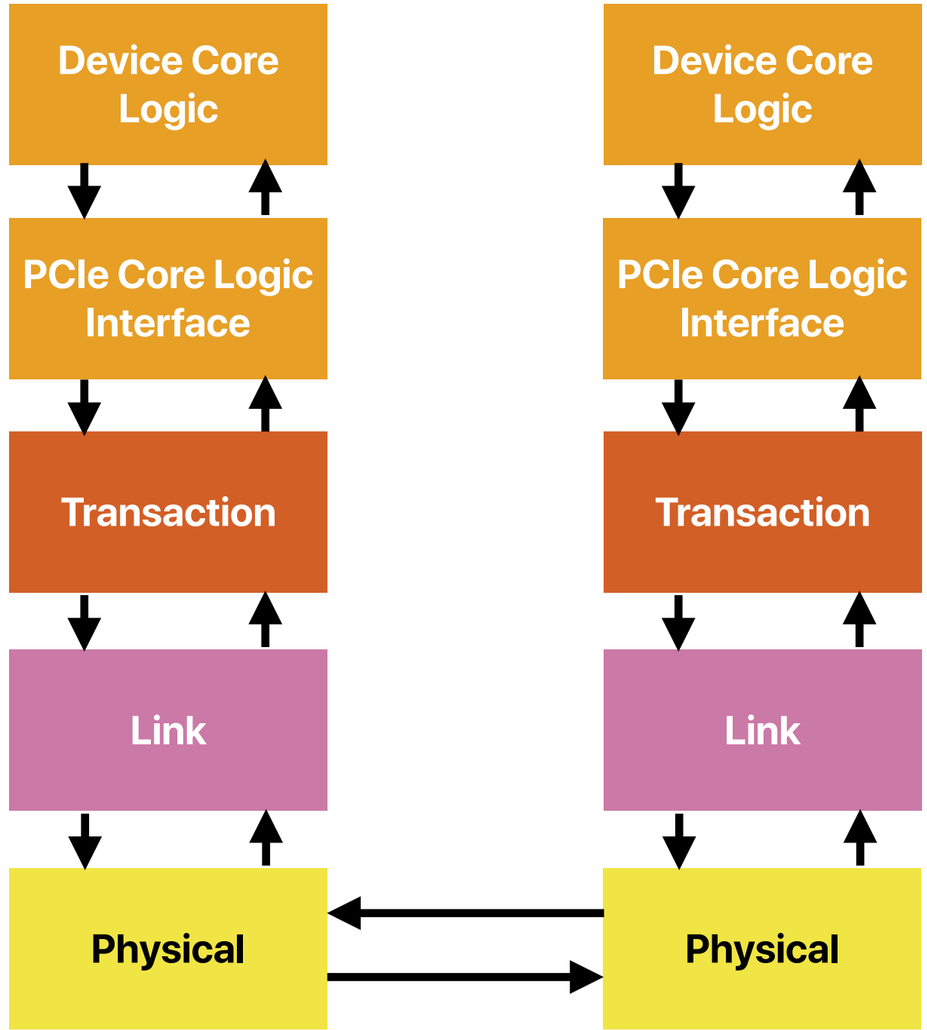


Outline

- NVMe (Non-volatile memory express)
- Efficient communication on PCIe
- Near-storage processing
  - SmartSSDs
  - ActiveDisks

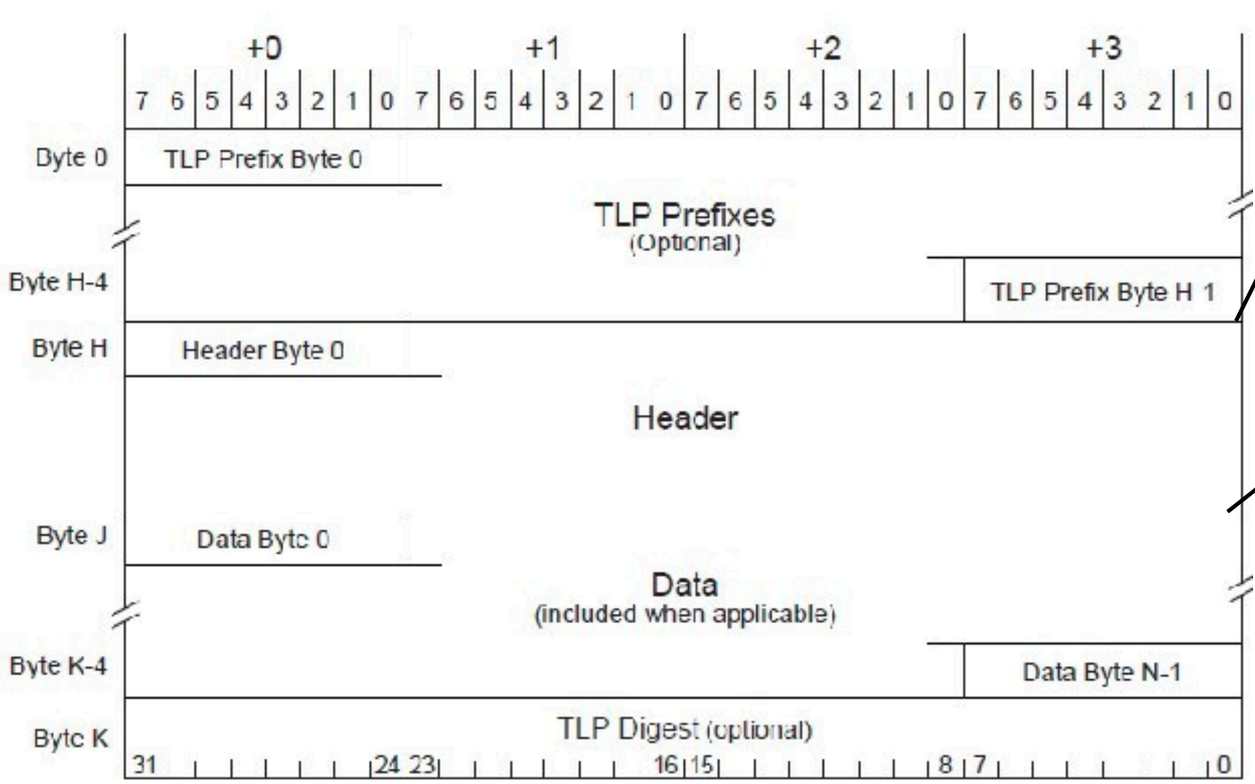
PCIe "Interconnect"

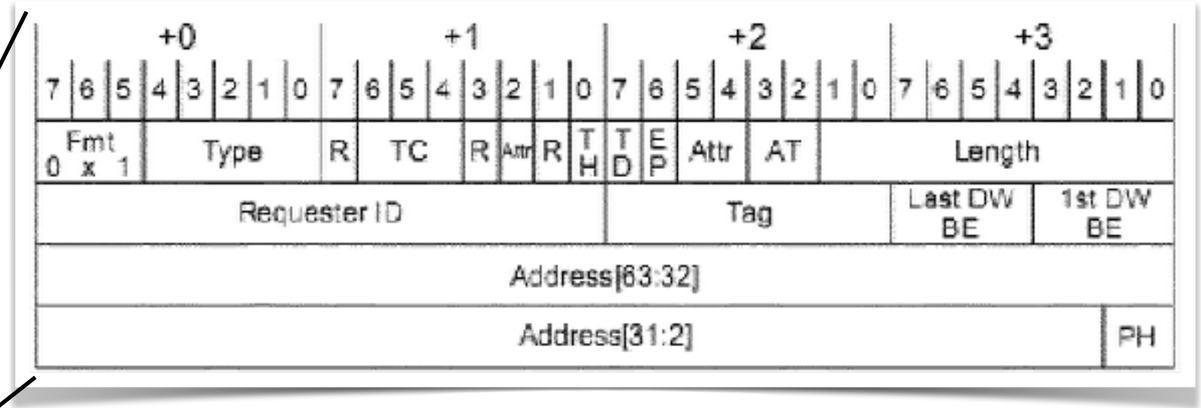




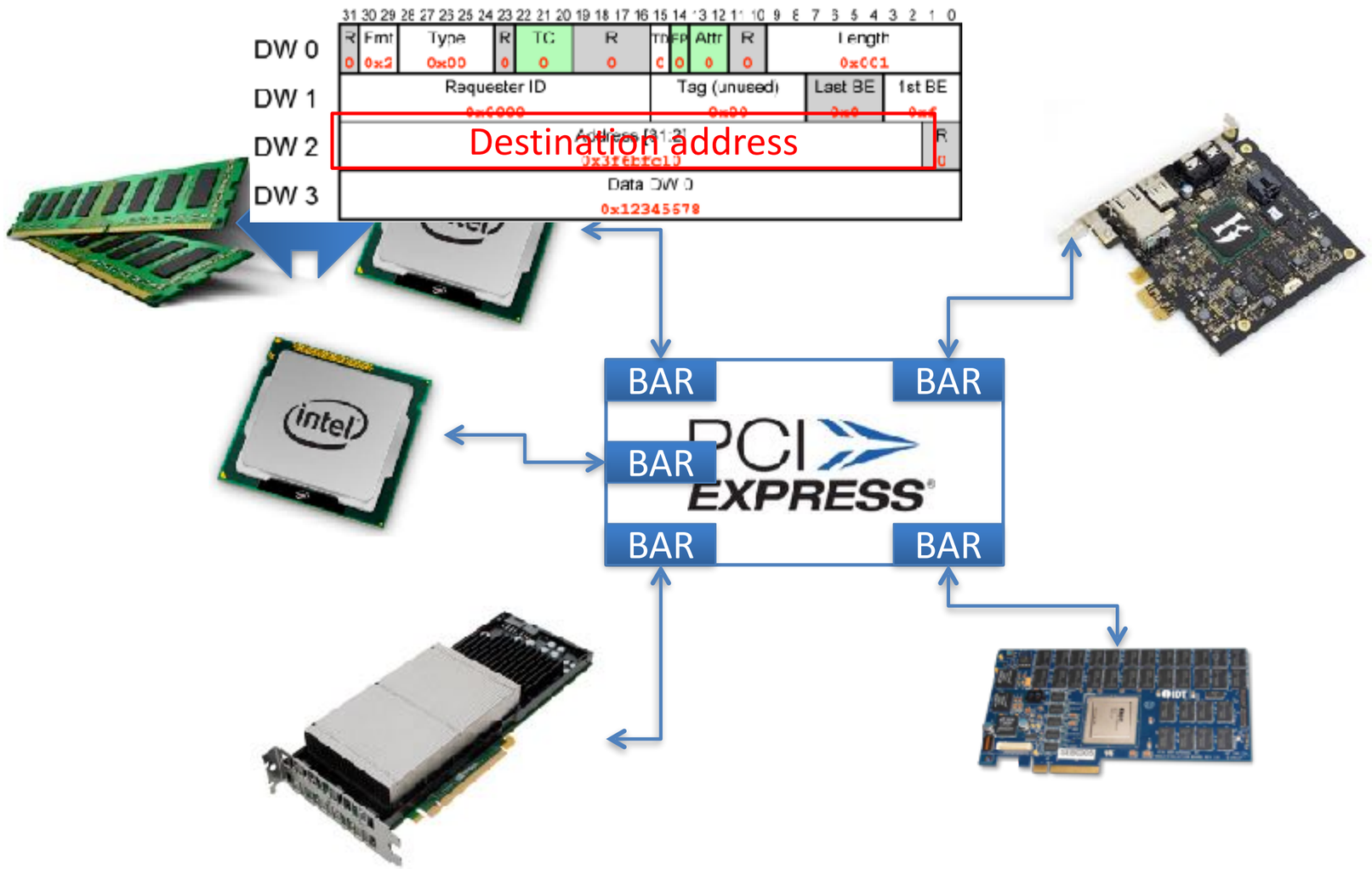
PCIe interconnect

- Very similar to computer networks
- Use “memory addresses” as the identifier for routing





PCIe


DW 0
DW 1
DW 2
DW 3
Fmt
Type
TC
Length
Requester ID
Tag (unused)
Last BE
1st BE
Destination address
Data DW 0
0x12345678
BAR
BAR
BAR
BAR
PCI EXPRESS

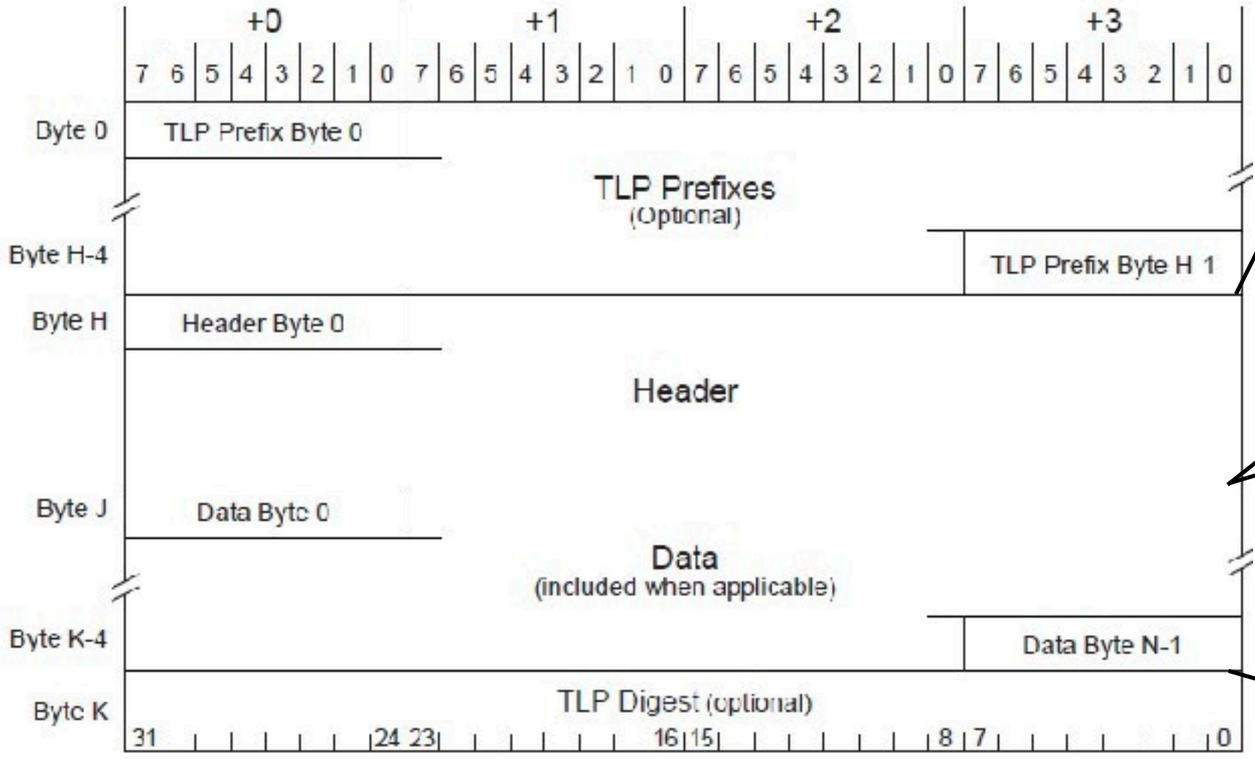
NVMe

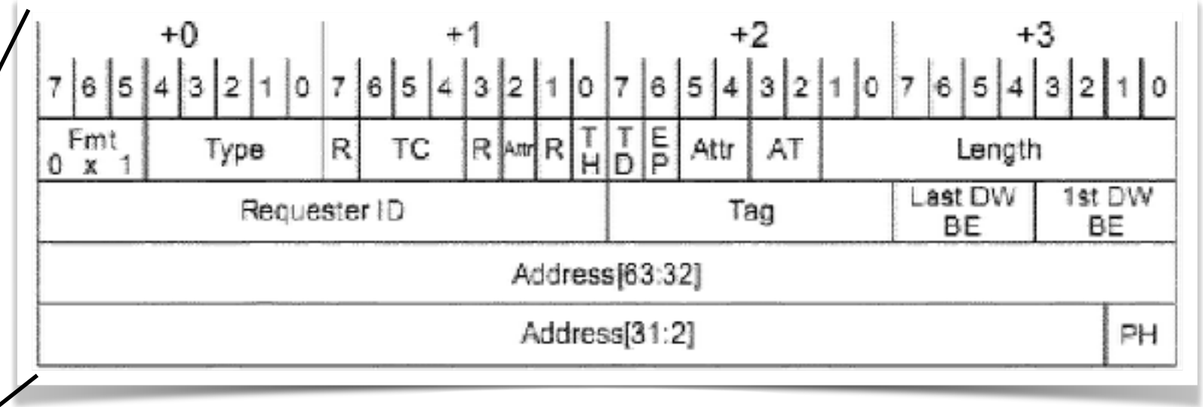
- The standard of PCIe SSD devices now
  - Provides multiple command queues to better support multithreading hardware
  - Allows more parallelism inside the SSD
- The “payload” of a PCIe packet

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**	

NVMe is the payload/data of PCIe

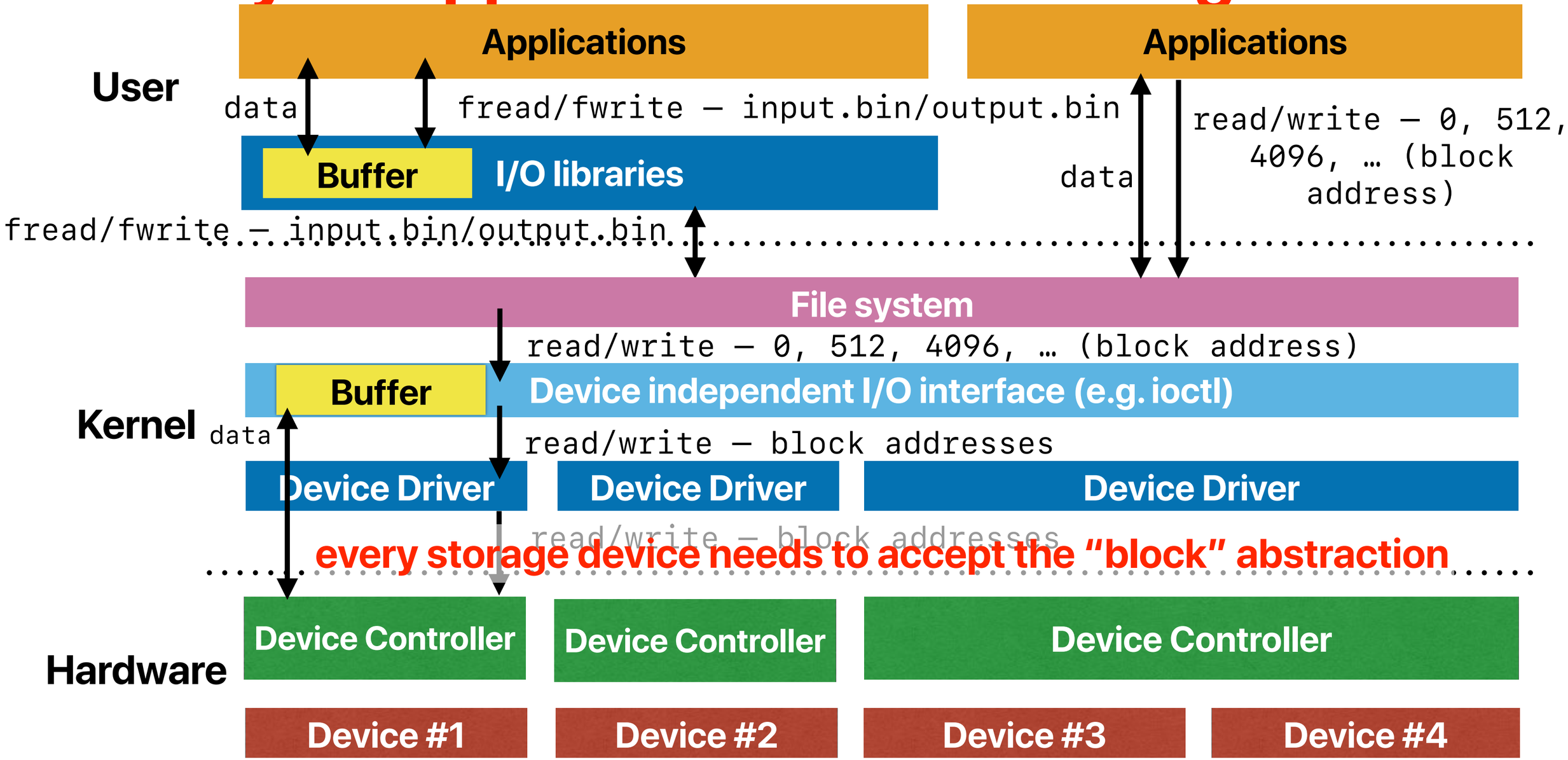
- Very similar to computer networks
- Use “memory addresses” as the identifier for routing





OPCODE FLAGS	command id	Namespace ID	
reserved			
metadata			
PRP1			
PRP2			
Start LBA			
length	control	Dataset management	
Reference tag		App tag	App mask

How your application reaches storage devices



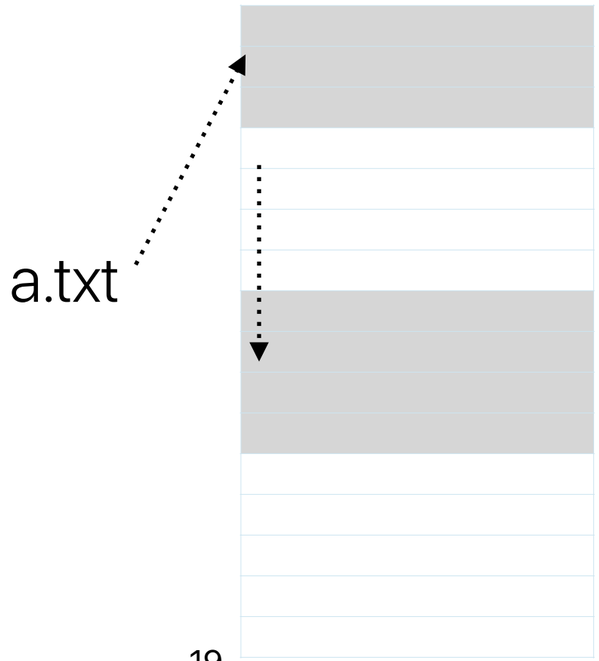
How to get the starting address?

- The I/O library fills the starting LBA for you
- The file system knows the address of each file

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	

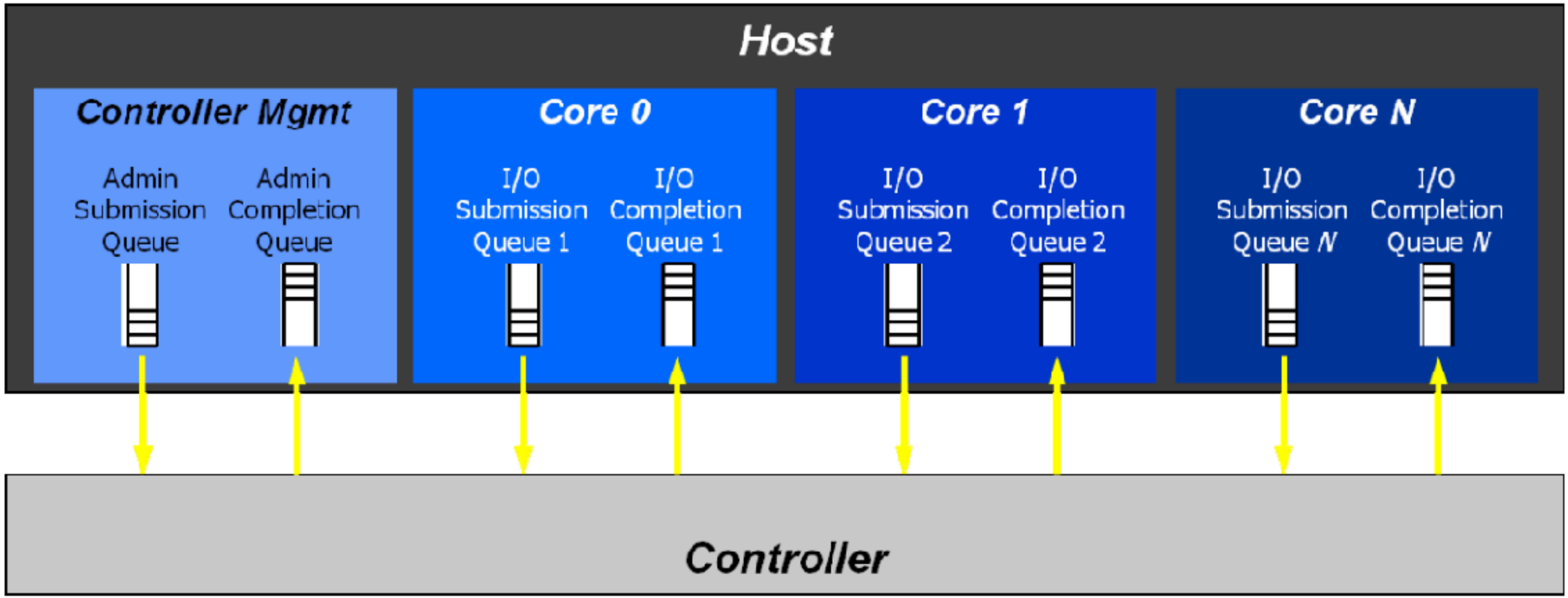
Extent-based file systems

- Contiguous blocks only need a pair <start, size> to represent
- Improve random seek performance
- Save inode sizes
- Encourage the file system to use contiguous space allocation



NVMe Model

Figure 3: Queue Pair Example, 1:1 Mapping



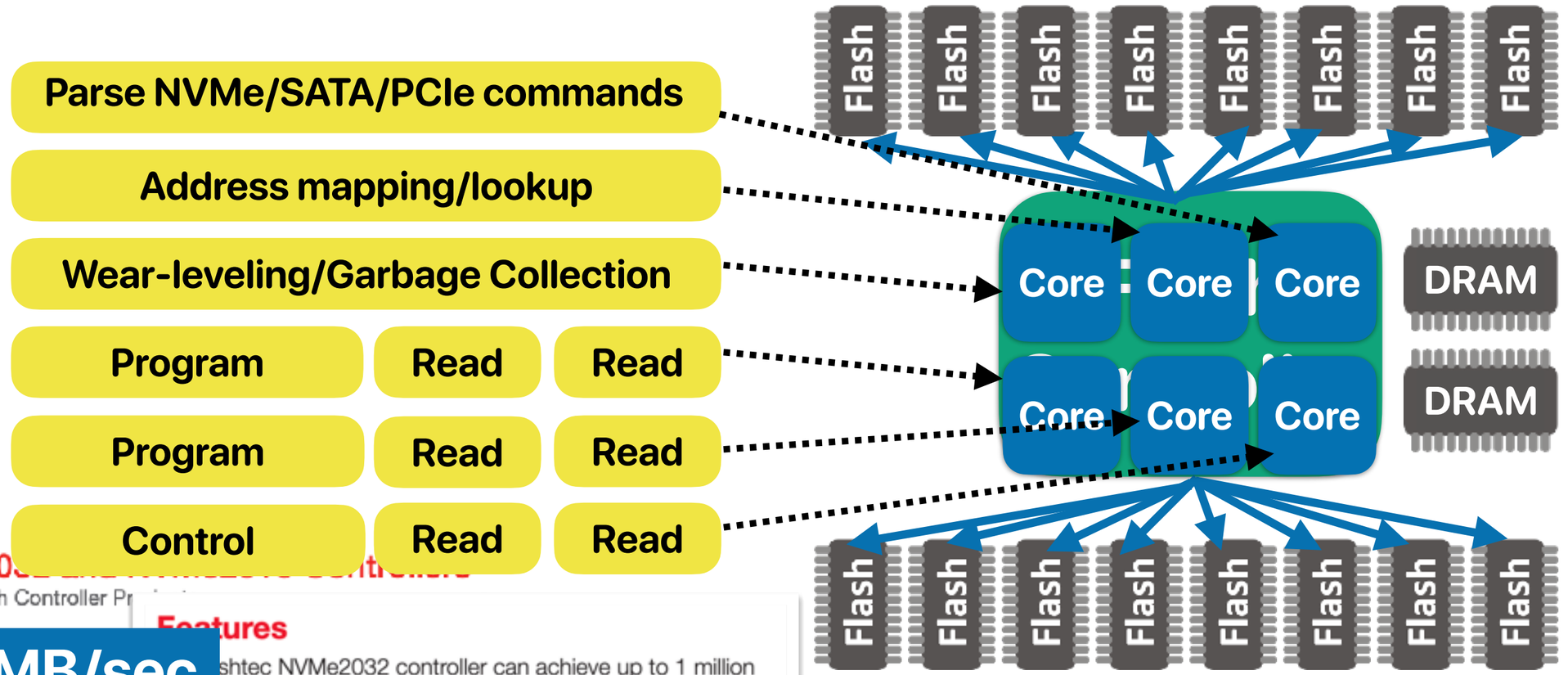
https://nvmexpress.org/wp-content/uploads/2013/04/NVM_whitepaper.pdf

3:00

Why do you think the NVMe take the multi queue design?

Why does MQ make sense?

Multi-channel to optimize bandwidth



Each ~500MB/sec
8 channels ==
4GB/sec

Flashtec™ NVMe20

32- and 16-Channel PCIe Flash Controller Pr

Features

shtec NVMe2032 controller can achieve up to 1 million
dom read IOPS on 4 KB operations
to 20 TB Flash capacity using 256 GB Flash
C, MLC, Enterprise MLC, and TLC Flash with to
FI interface
le Gen 3 x8 or dual independent PCIe Gen 3 x4 (active,
tive/standby) host interface
- 16 and 32 independent Flash channels, each suppo
up to 8 CE

Micron

- Organization
  - Page size x8: 18,592 bytes (16,384 + 2208 bytes)
  - Block size: 2304 pages, (36,864K + 4968K bytes)
  - Plane size: 4 planes x 504 blocks
  - Device size: 512Gb: 2016 blocks; 1Tb: 4032 blocks; 2Tb: 8064 blocks; 4Tb: 16,128 blocks; 8Tb: 32,256 blocks

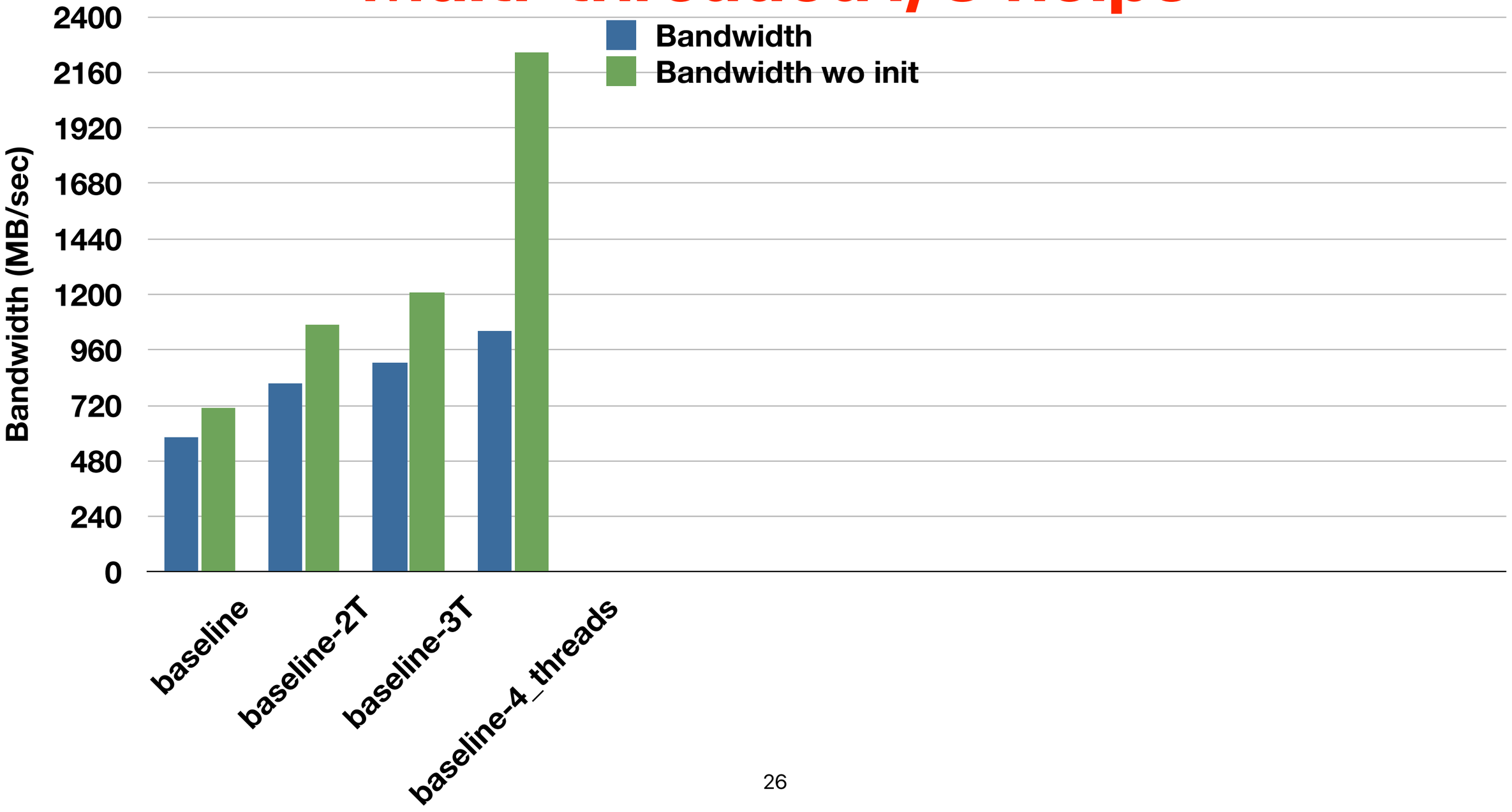
TLC 512Gb-8Tb NAND
– Array Characteristics

is suspended or ongoing					
READ PAGE operation time without/with V_{PP}	^tR	57/41	60	µs	2, 12
SNAP READ operation time	^tRSNAP	27	37	µs	
the read busy time	^tRCBSY	11	60	µs	2, 8, 12

Why does MQ makes sense?

- Flash SSDs have multiple channels inside!
  - You need multiple requests to fill the requests
- The same design cannot work on HDDs — HDDs have only one head

Multi-threaded I/O helps



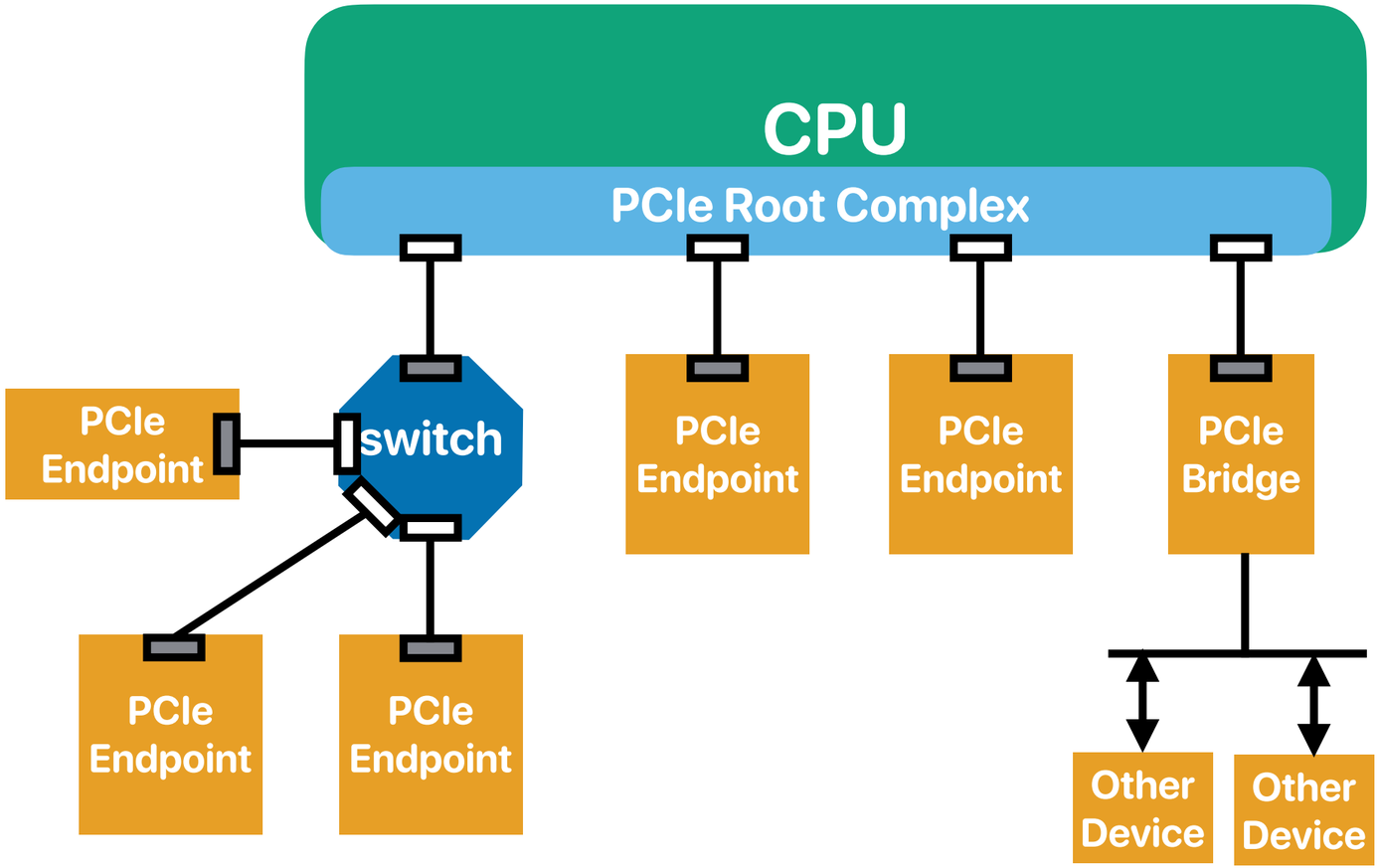
Let’s walk through an NVMe driver

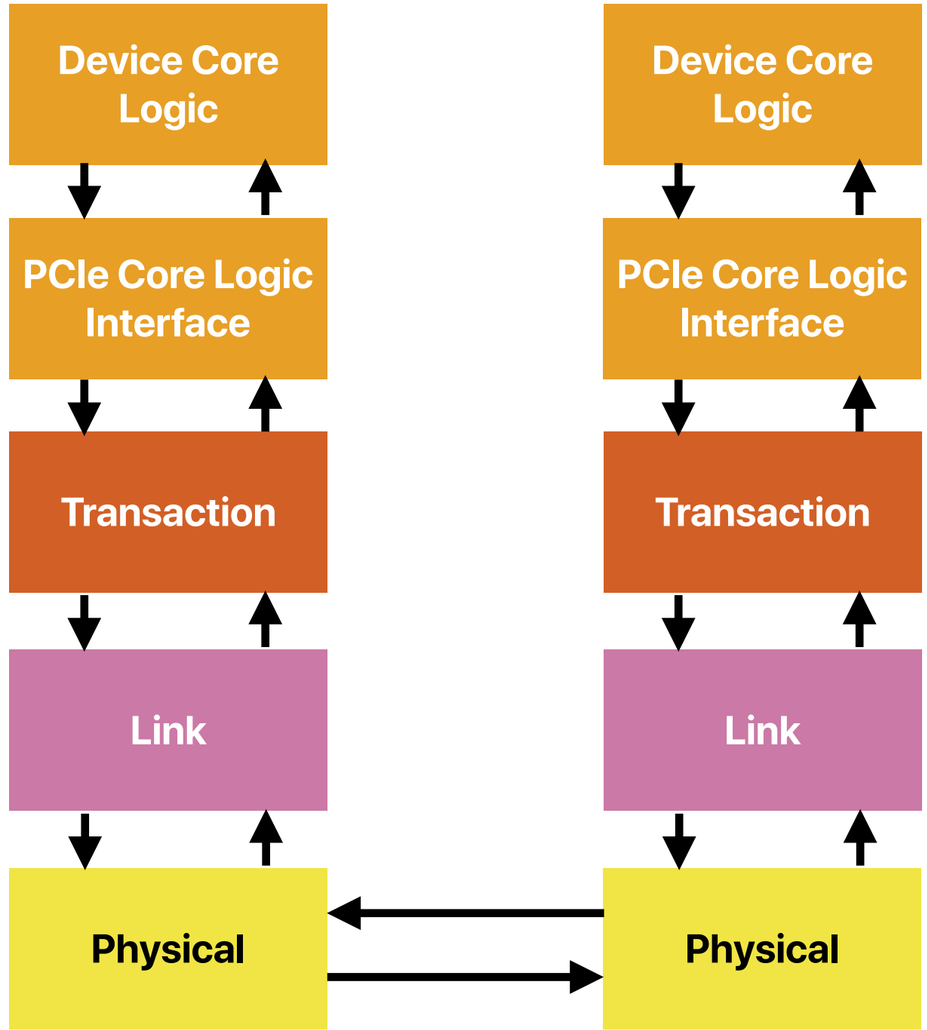
3:00

How can you let SSDs to communicate with GPUs directly & why does SSD-GPU P2P make sense?

How to implement GPU-SSD P2P?

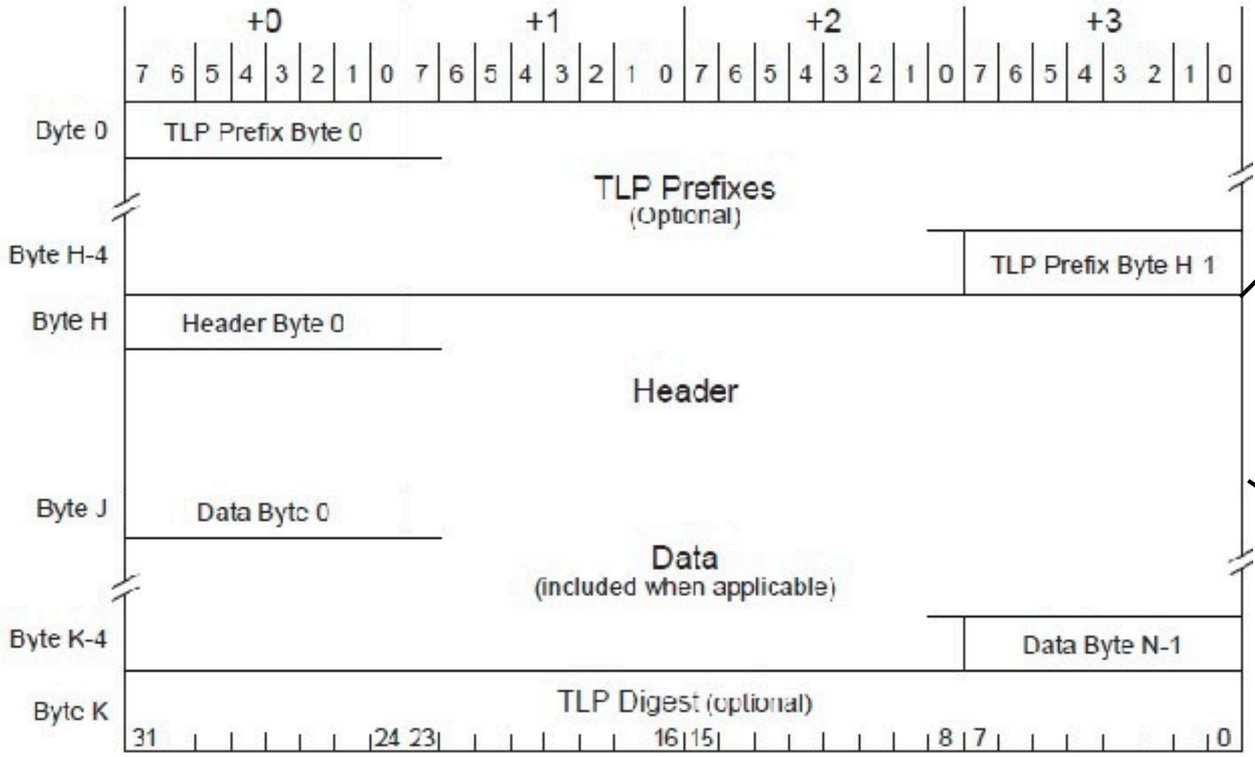
PCIe "Interconnect"

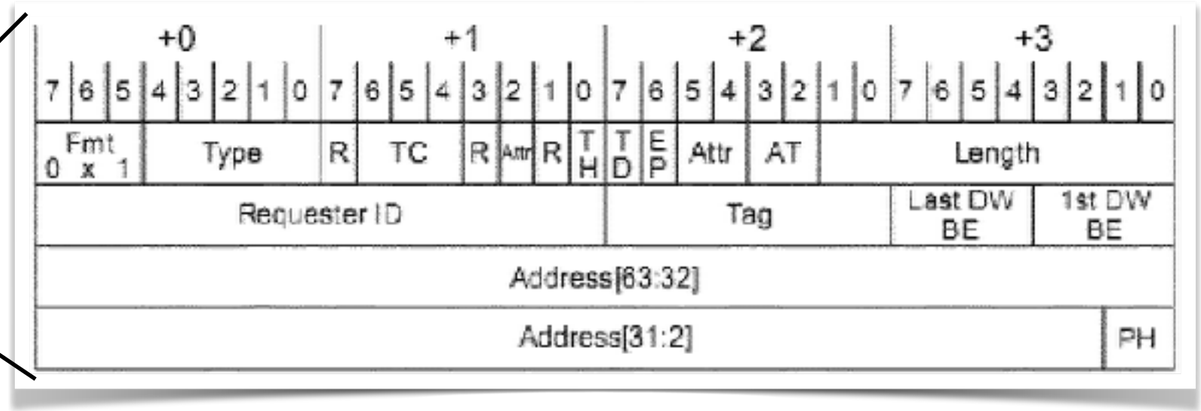




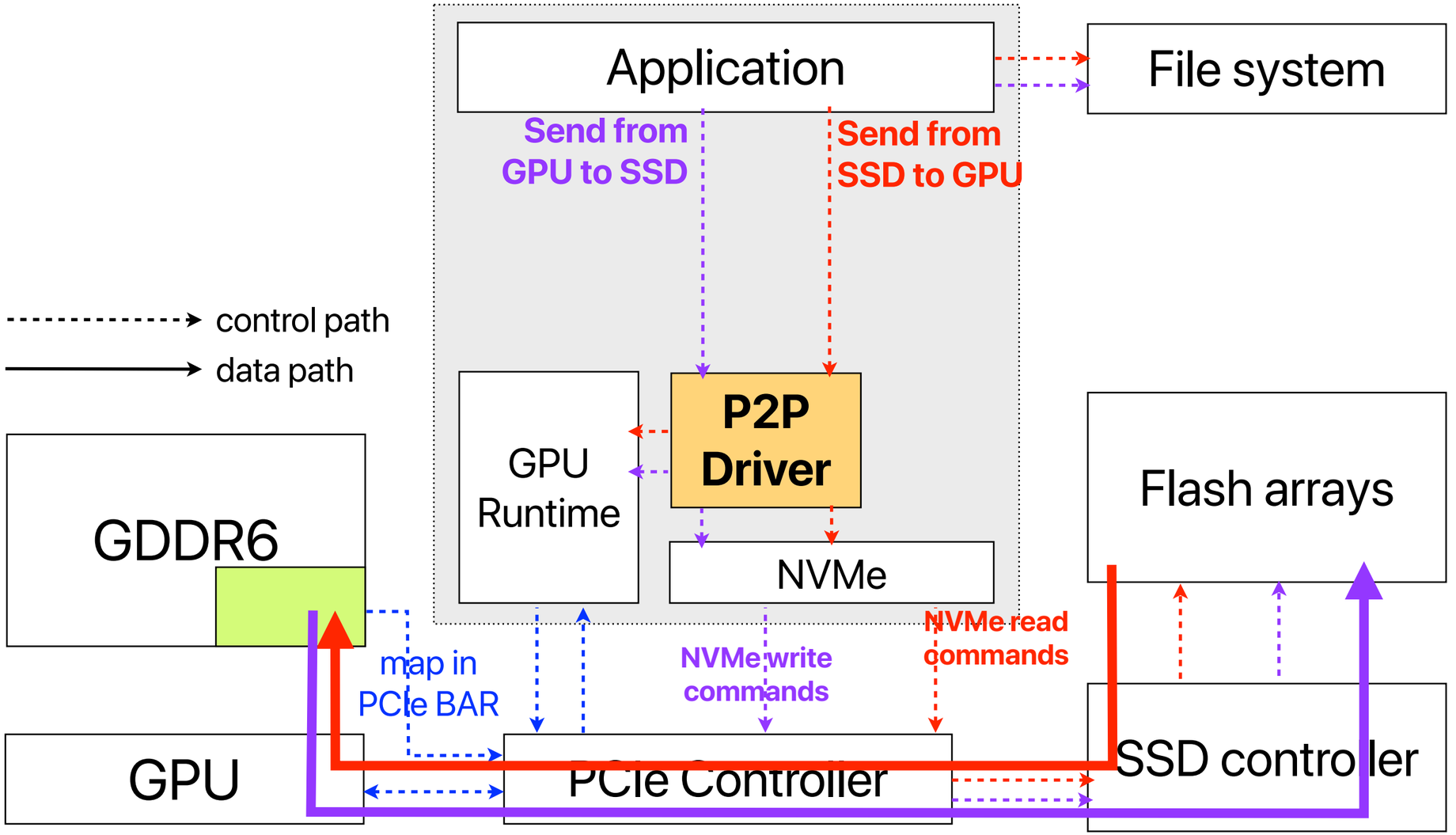
PCIe interconnect

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

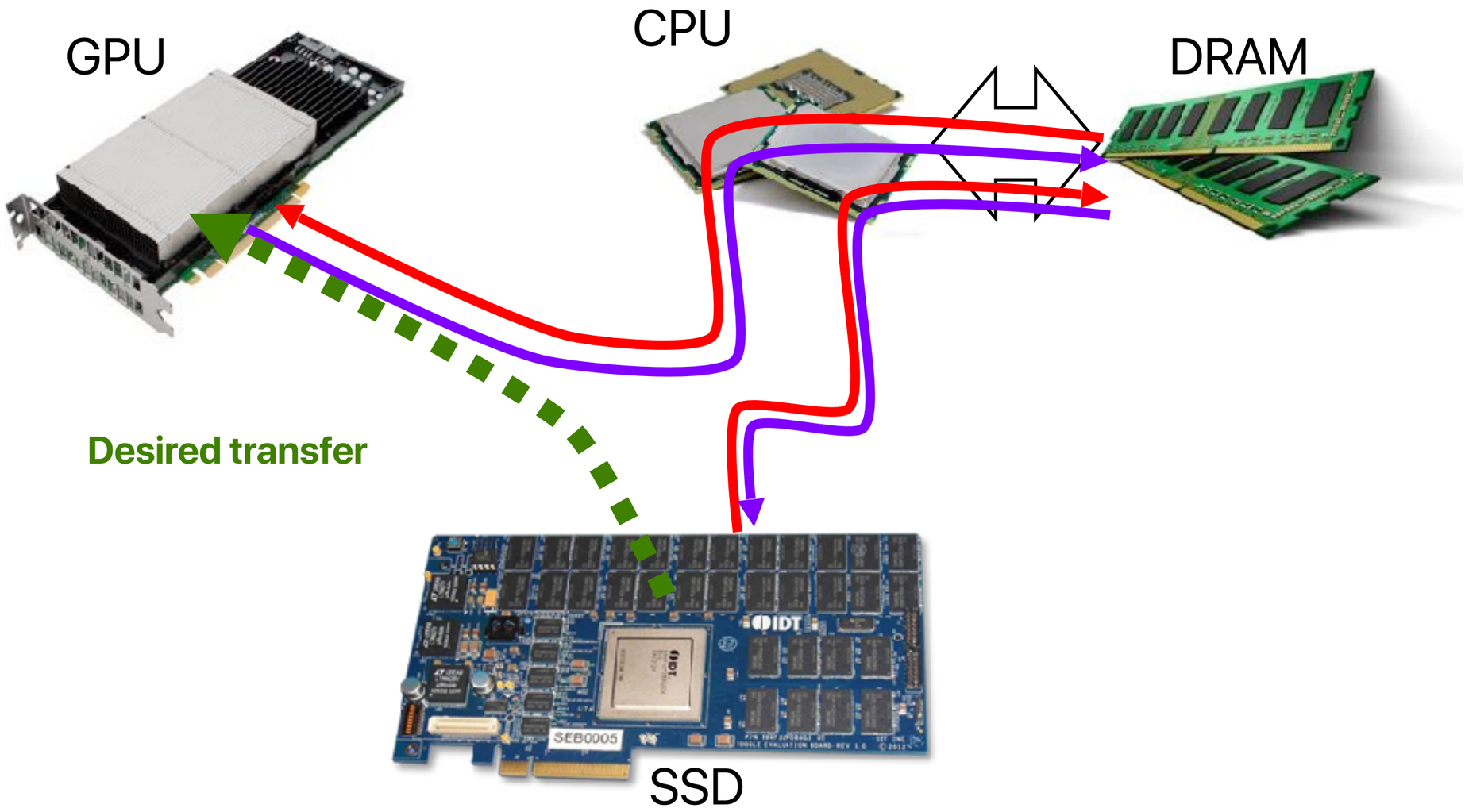




P2P between GPU and SSD



GPU-accelerated architecture



CPU-centric programming model

Moving data

```
1:  m = (float*) malloc(sizeof(float)*matrix_dim*matrix_dim);  // allocate m
2:  fp = open(filename, O_RDONLY|O_DIRECT);                   // open file
3:  read(fp,m,sizeof(float)*matrix_dim*matrix_dim);           // read file
4:  close(fp);
    // allocate GPU memory
5:  cudaMalloc((void**)&d_m, matrix_dim*matrix_dim*sizeof(float));
    //copy from main memory to GPU memory
6:  cudaMemcpy(d_m, m, matrix_dim*matrix_dim*sizeof(float),
                cudaMemcpyHostToDevice);
```

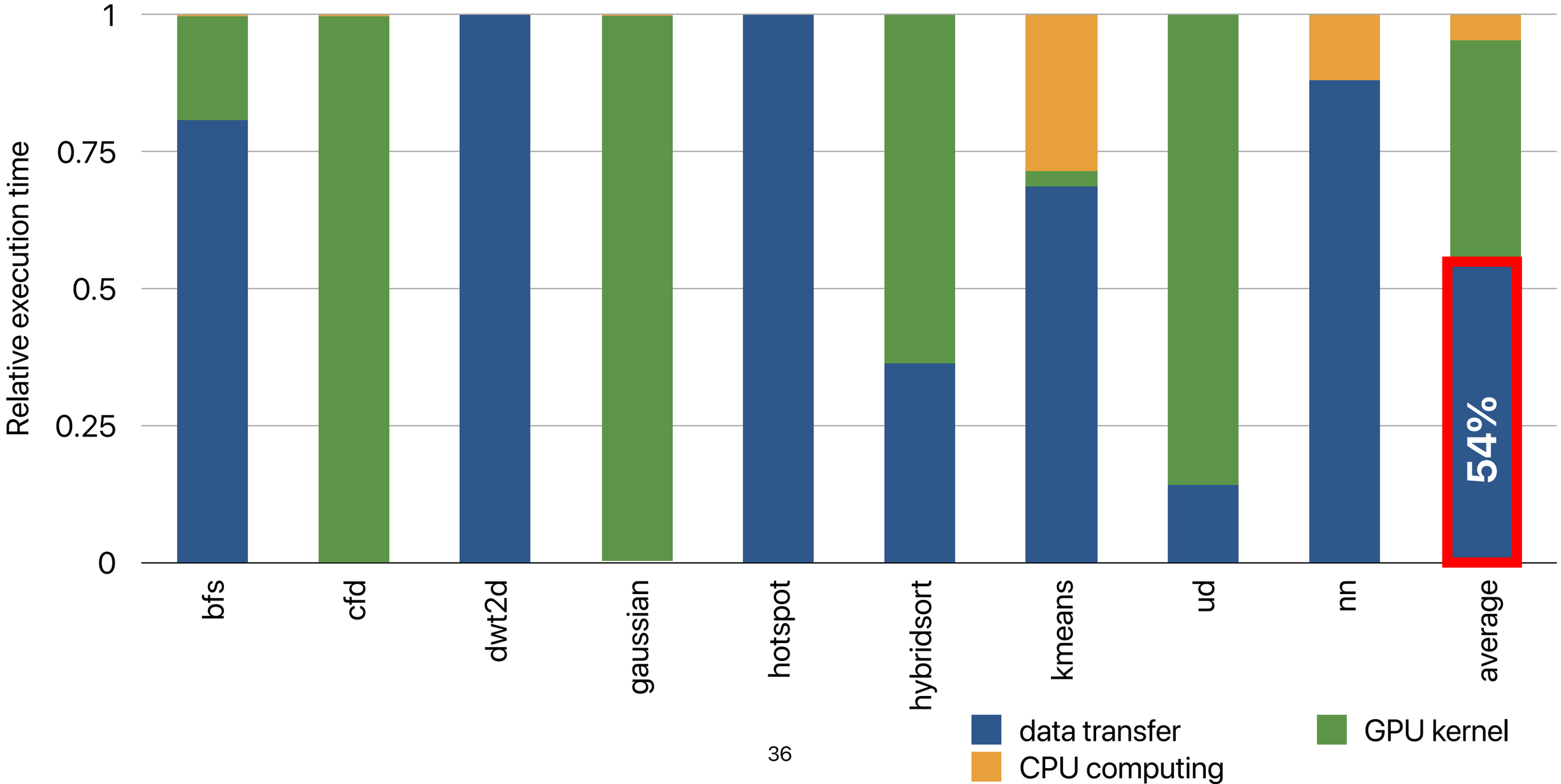
```
7: lud_diagonal<<<1,BLOCK_SIZE>>>(d_m, matrix_dim, i);
```

GPU computing

```
//copy from GPU memory to main memory
8: cudaMemcpy(m, d_m, matrix_dim*matrix_dim*sizeof(float),
                cudaMemcpyDeviceToHost);
9: output_matrix(m, matrix_dim);
```

Moving data

Cost of moving data



Code revisited

Moving data

storing input in m

```
1:  m = (float*) malloc(sizeof(float)*matrix_dim*matrix_dim); // allocate m
2:  fp = open(filename, O_RDONLY|O_DIRECT);                   // open file
3:  read(fp, m, sizeof(float)*matrix_dim*matrix_dim);         // read file
4:  close(fp);
    // allocate GPU memory
5:  cudaMalloc((void**)&d_m, matrix_dim*matrix_dim*sizeof(float));
    //copy from main memory to GPU memory
6:  cudaMemcpy(d_m, m, matrix_dim*matrix_dim*sizeof(float),
                cudaMemcpyHostToDevice);
```

copying m to GPU memory

```
7: lud_diagonal<<<1,BLOCK_SIZE>>>(d_m, matrix_dim, i);
```

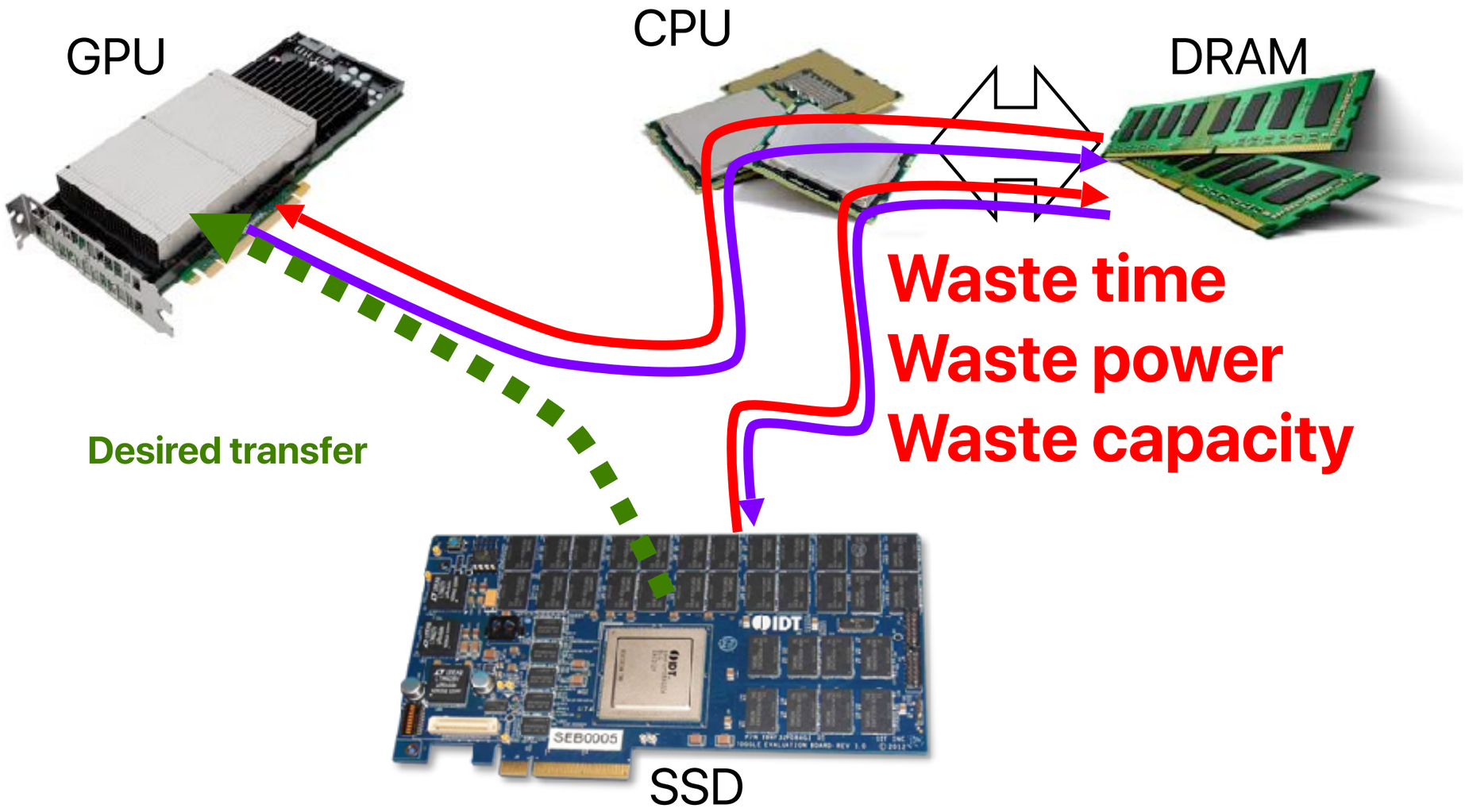
GPU computing

```
//copy from GPU memory to main memory
8: cudaMemcpy(m, d_m, matrix_dim*matrix_dim*sizeof(float),
                cudaMemcpyDeviceToHost);
9: output_matrix(m, matrix_dim);
```

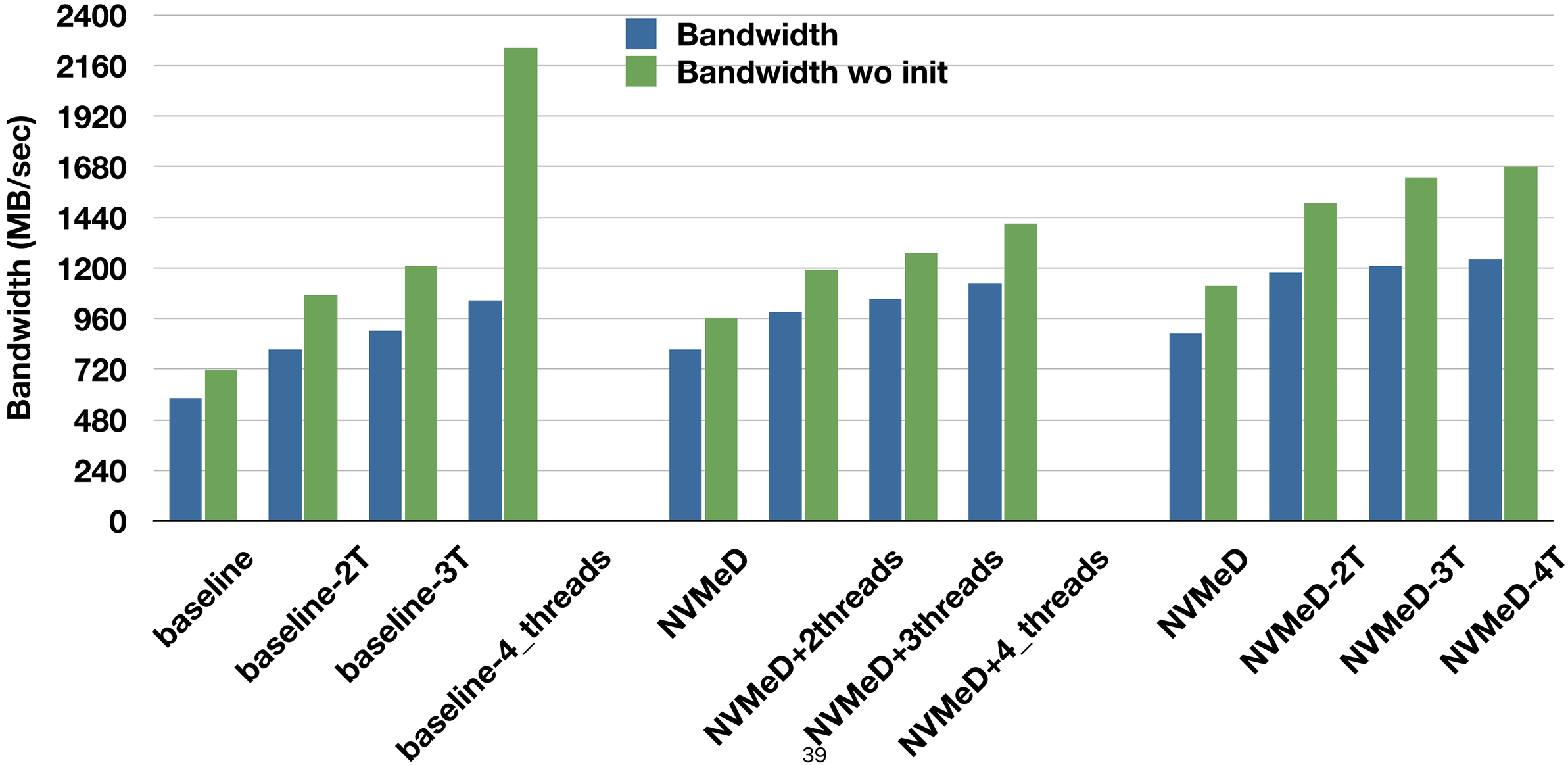
Moving data

m is overwritten, previous m is only used as a“buffer”

GPU-accelerated architecture



Multi-threaded I/O helps

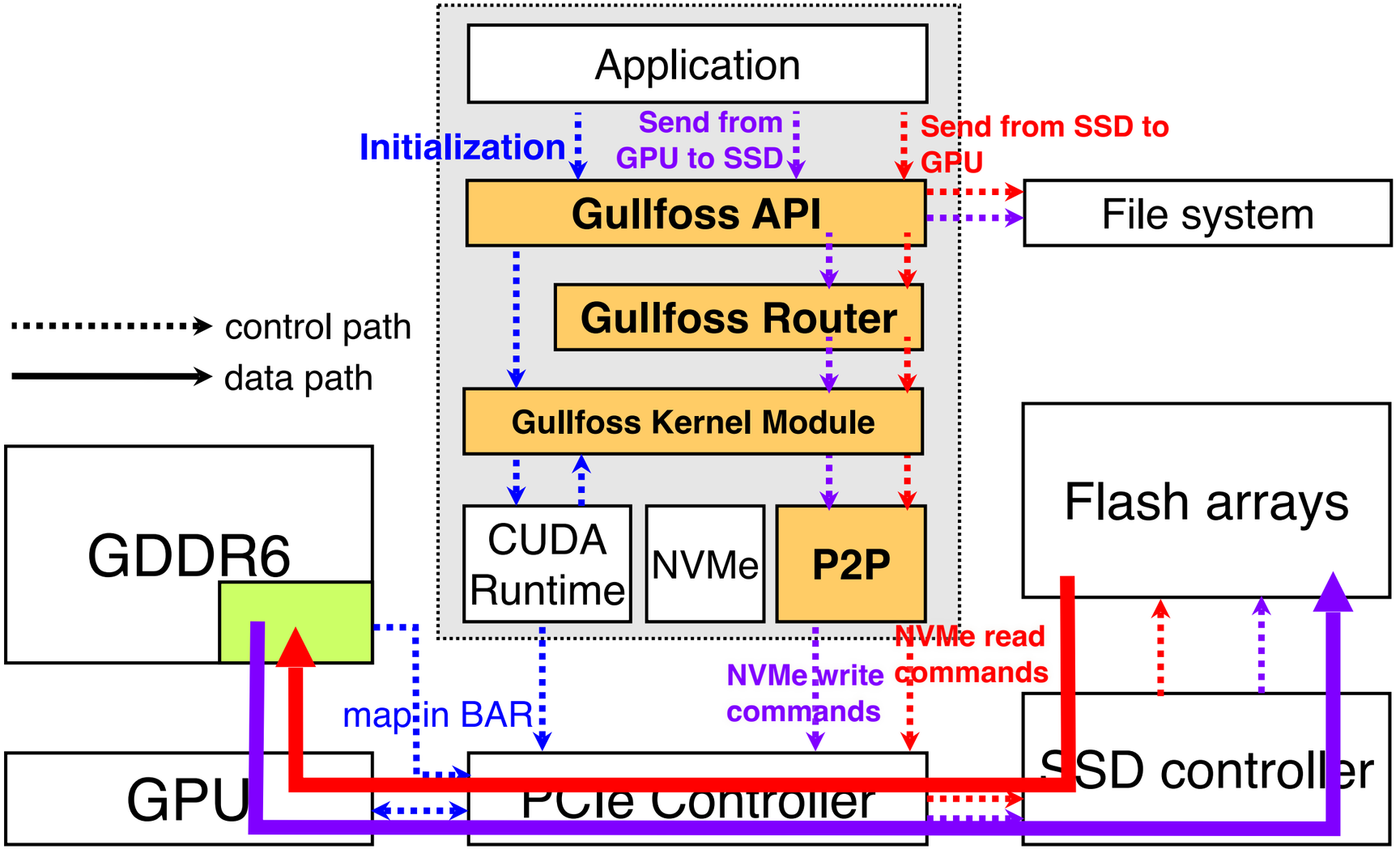


3:00

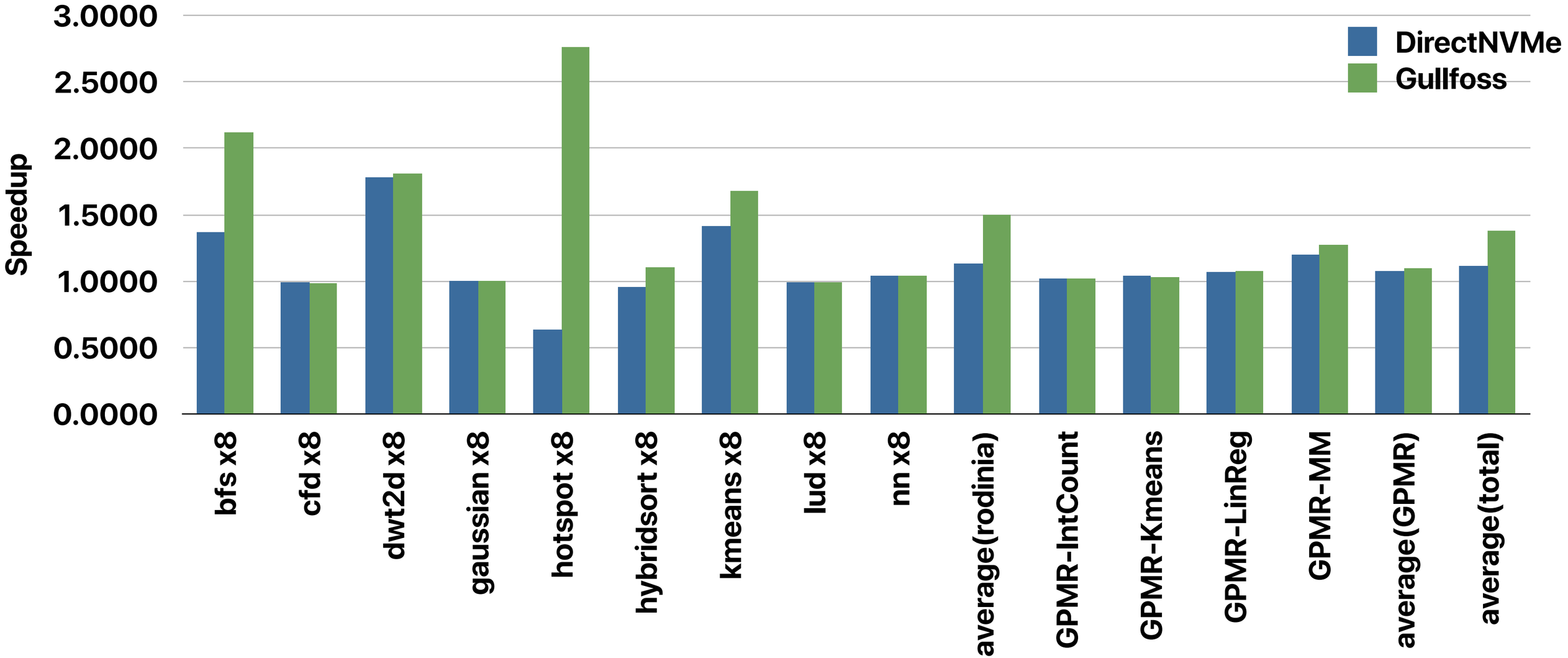
When can GPU-SSD P2P be a bad idea?

When doesn’t GPU-SSD P2P make sense?

Gullfoss: decouples control/data planes



GPU-SSD P2P + traditional route and main memory buffers



UC RIVERSIDE | Computer Engineering

Please consider attending

CEN TECH TALK

Join Us to Learn How to Shape Tech Careers:
Insights and Guidance from Industry Experts

Ameen Akel

Micron Technology

Friday, February 23rd, 2024
12:30 pm - 1:30 pm

Winston Chung Hall
205/206

Food will be provided!

Guest speaker biography: As a Distinguished Member of Technical Staff at Micron Technology, Ameen Akel leads a systems research and pathfinding organization that researches,

Electrical
Computer Science
Engineering

277

つづく