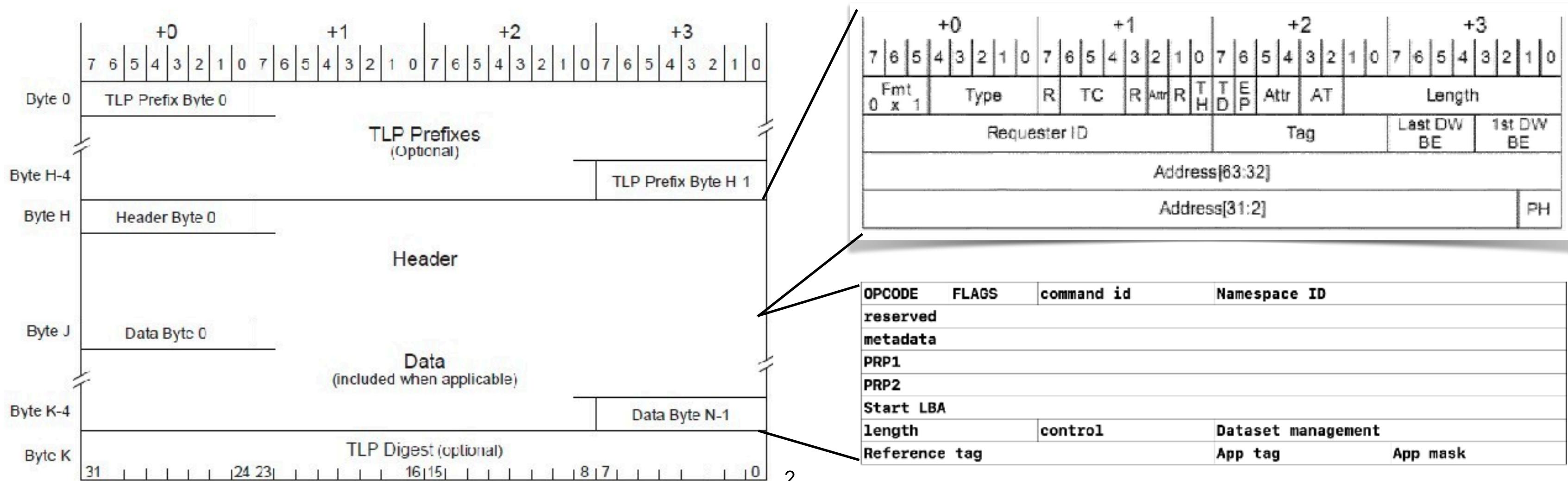


In-Storage Processing (1)

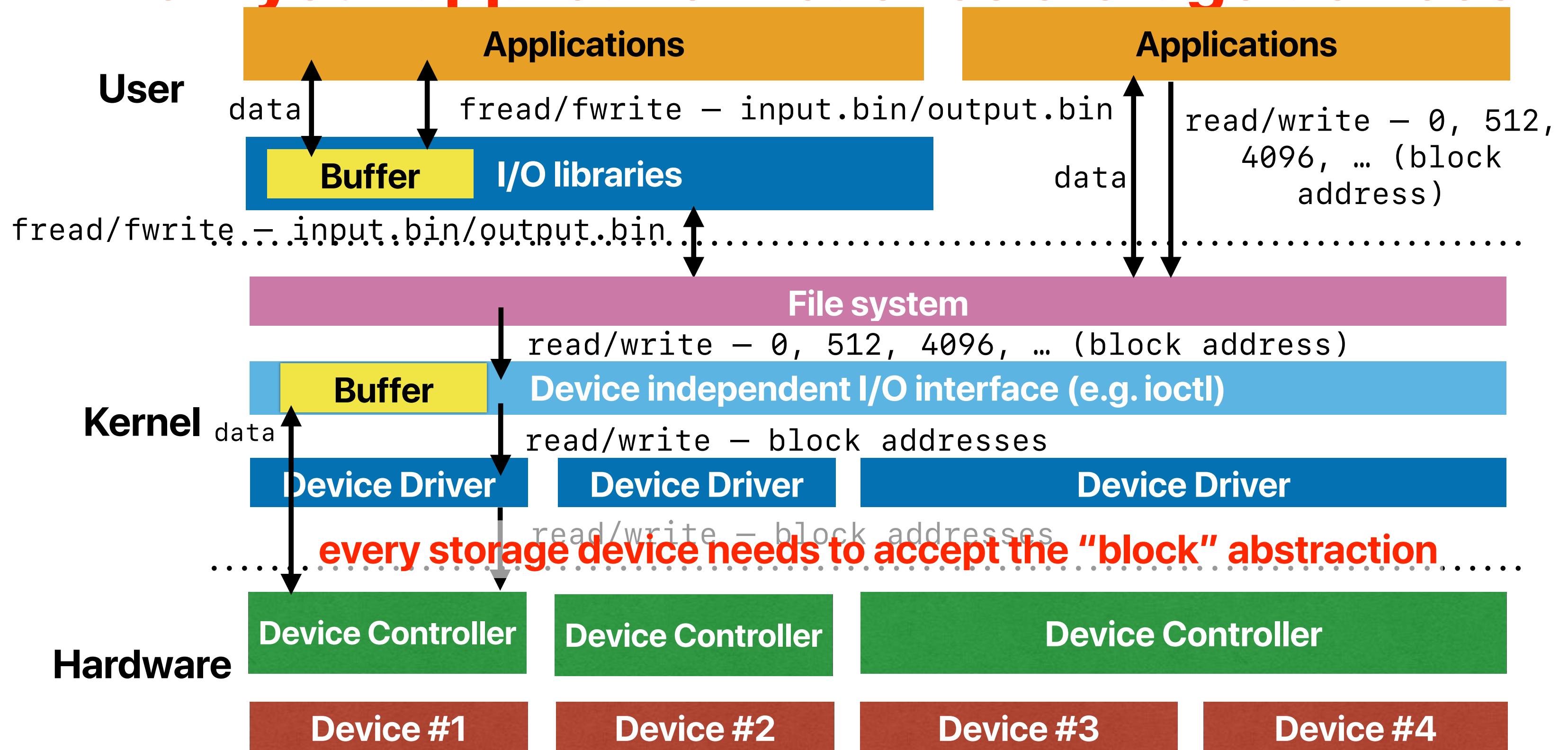
Hung-Wei Tseng

Recap: NVMe is the payload/data of PCIe

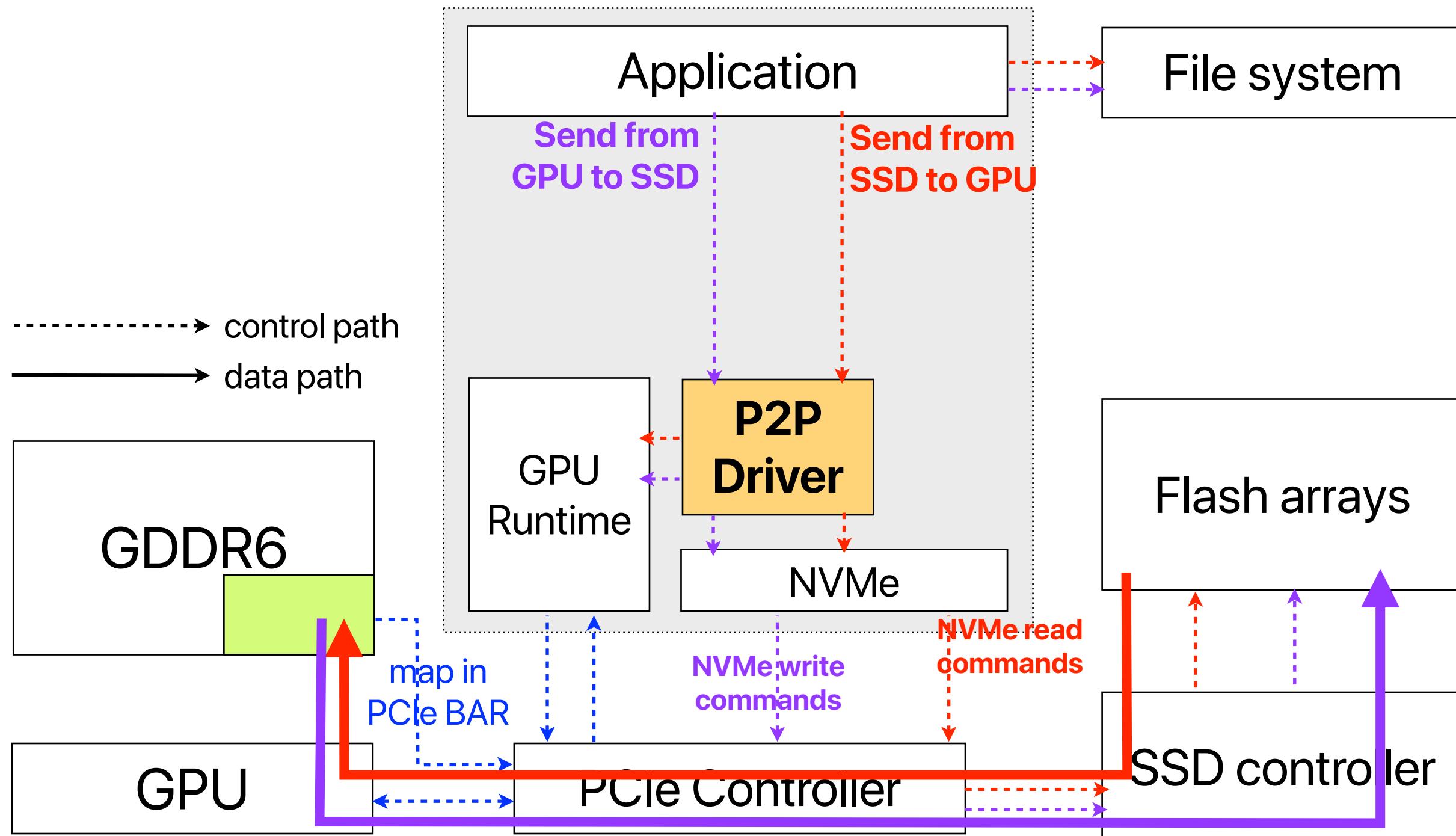
- Very similar to computer networks
- Use “memory addresses” as the identifier for routing



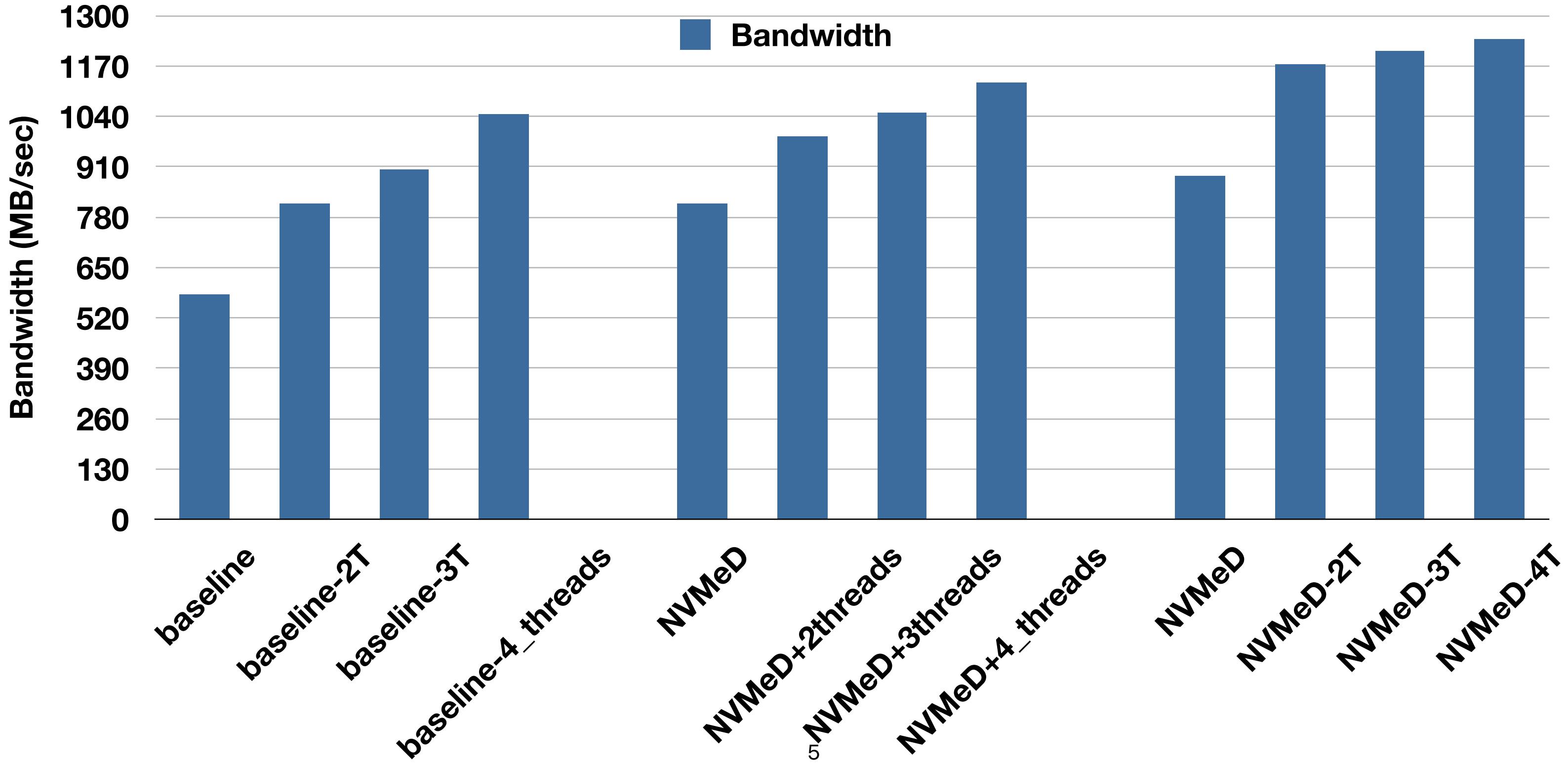
How your application reaches storage devices



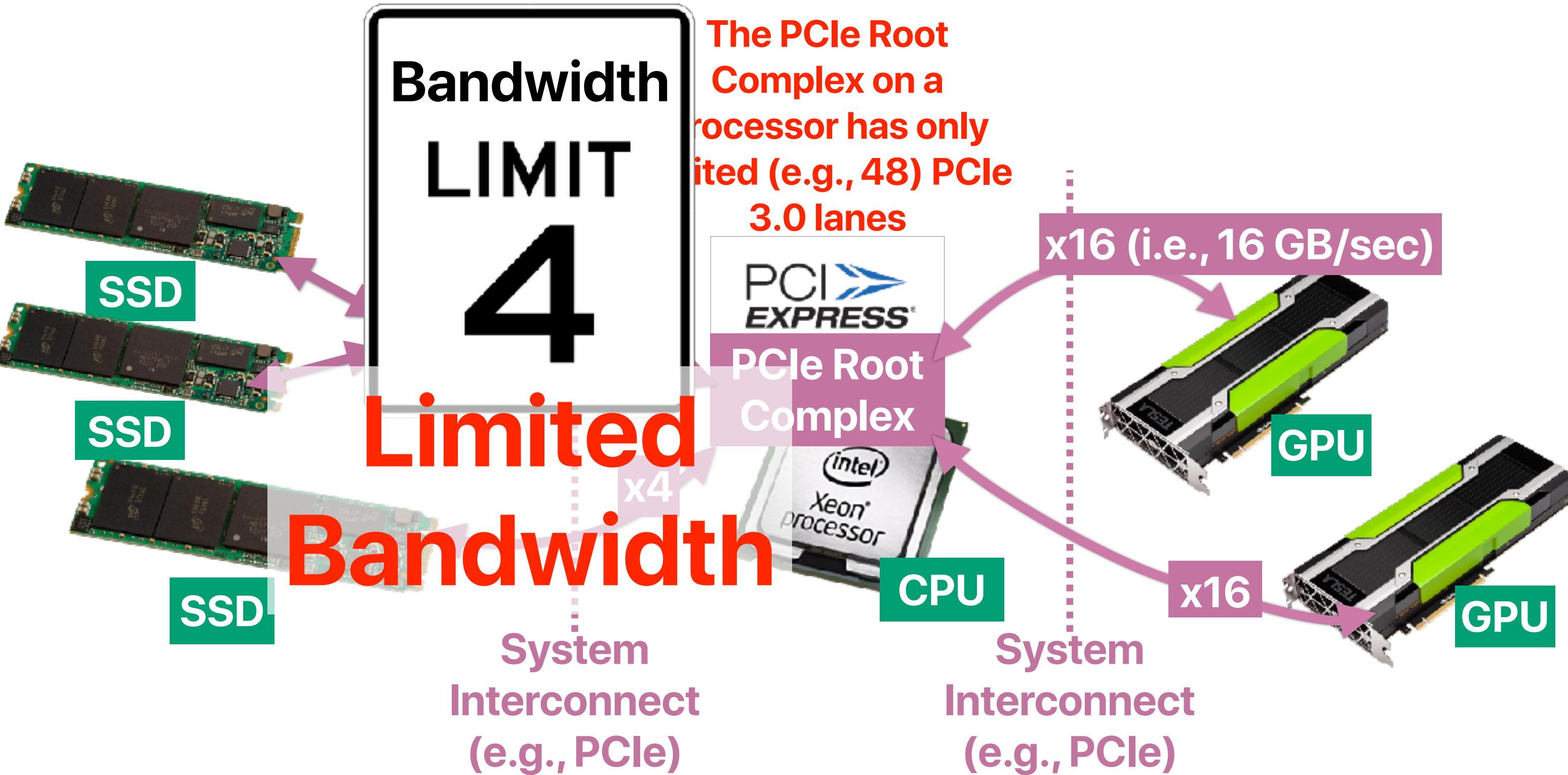
P2P between GPU and SSD



Recap: Multi-threaded I/O helps

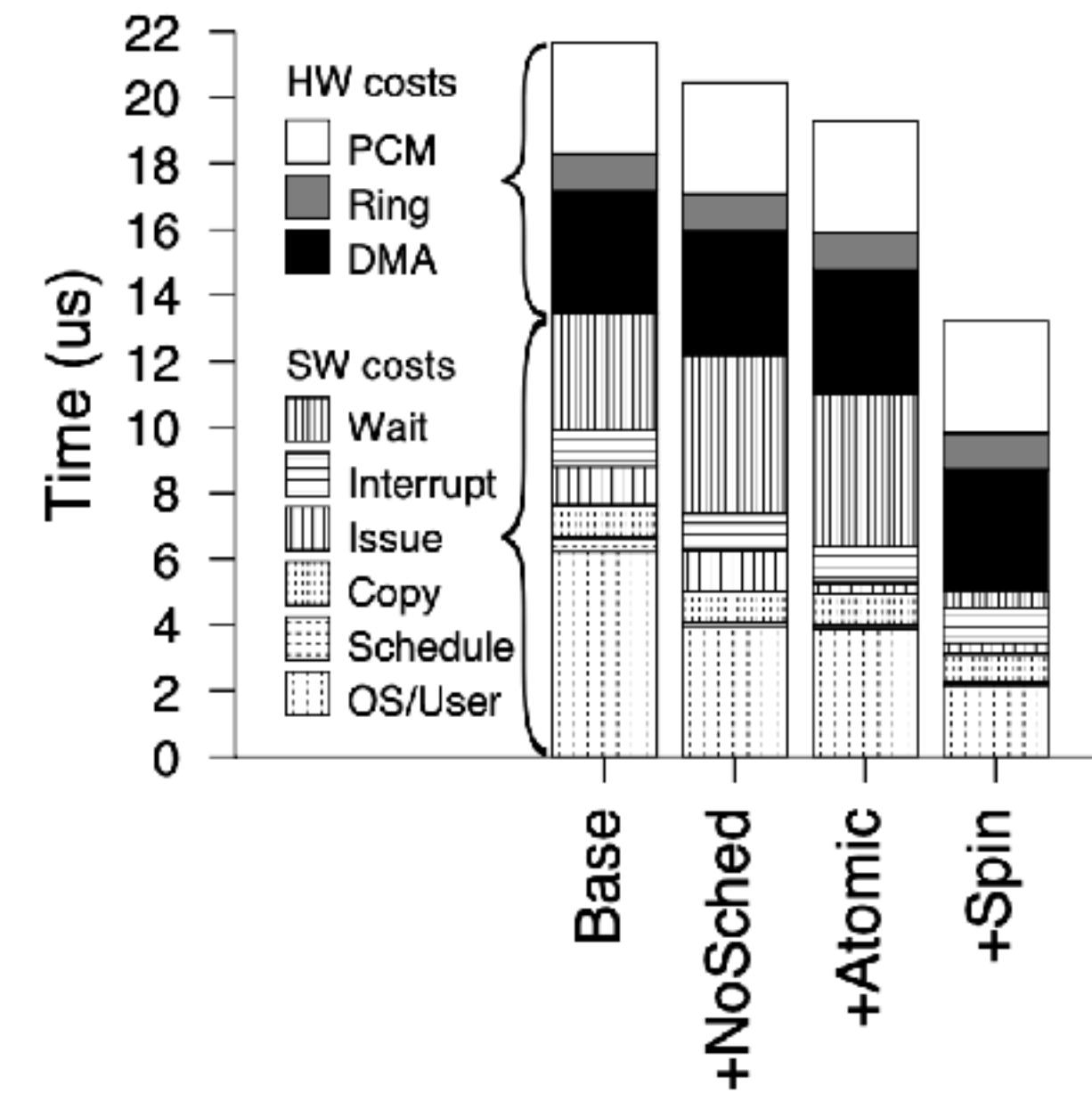


Limited Interconnect Bandwidth



Software overhead

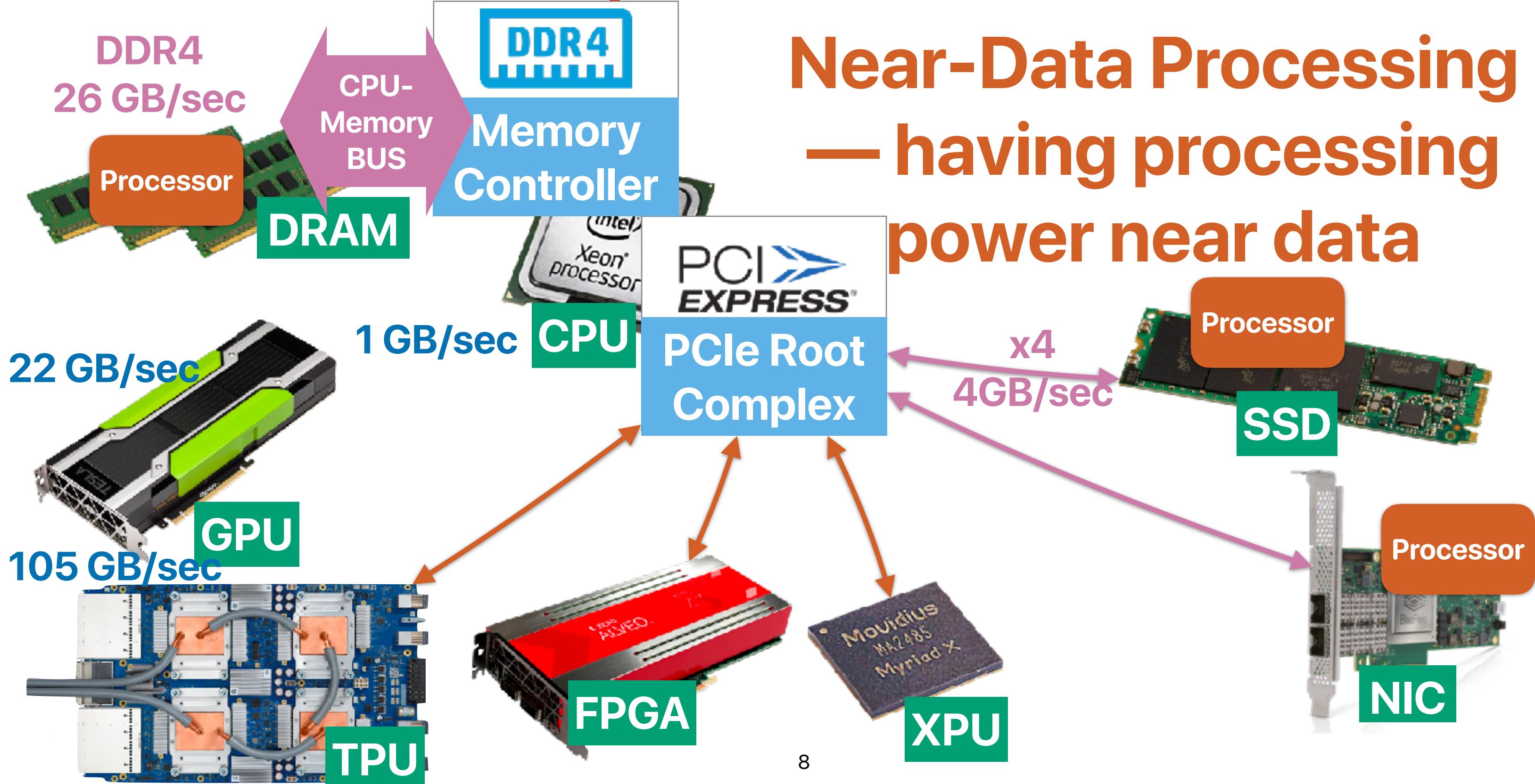
Label	Description	Baseline latency (μ s)	
		Write	Read
OS/User	OS and userspace overhead	1.98	1.95
OS/User	Linux block queue and no-op scheduler	2.51	3.74
Schedule	Get request from queue and assign tag	0.44	0.51
Copy	Data copy into DMA buffer	0.24/KB	-
Issue	PIO command writes to Moneta	1.18	1.15
DMA	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.

Recap: Think different

Near-Data Processing
— having processing power near data



Where do you see the opportunity of in-storage processing in address the bandwidth limitation and software overhead?

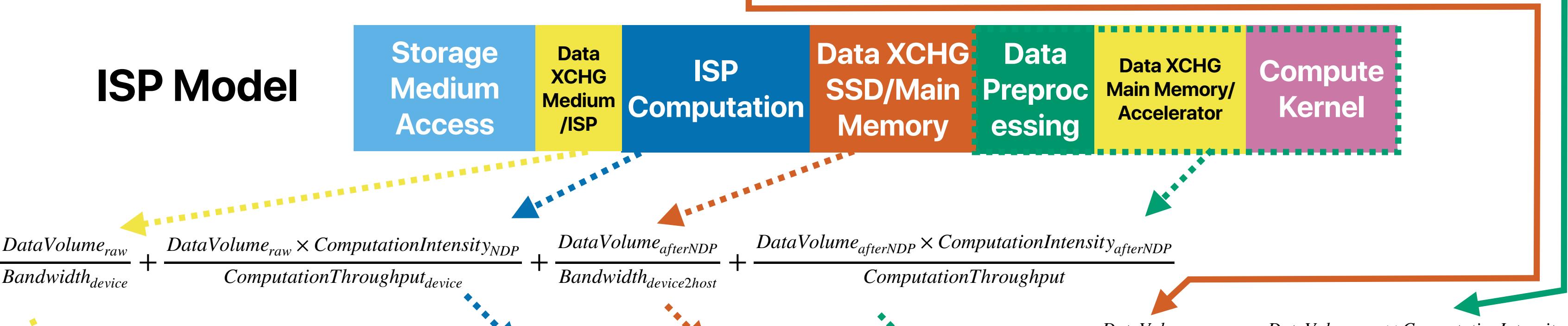
Why does ISP/SmartSSD make sense?

The “Winning Formula” of In-Storage Processing

Conventional Model



ISP Model



The ISP processor must have efficient internal access to the device

The ISP computation should not take too long

The ISP should reduce data volume

The ISP should offload the processor

Opportunities of SmartSSDs

Multi-channel to optimize bandwidth

Flashtec™ NVMe2032 and NVMe2016 Controllers

32- and 16-Channel PCIe Flash Controller Processor

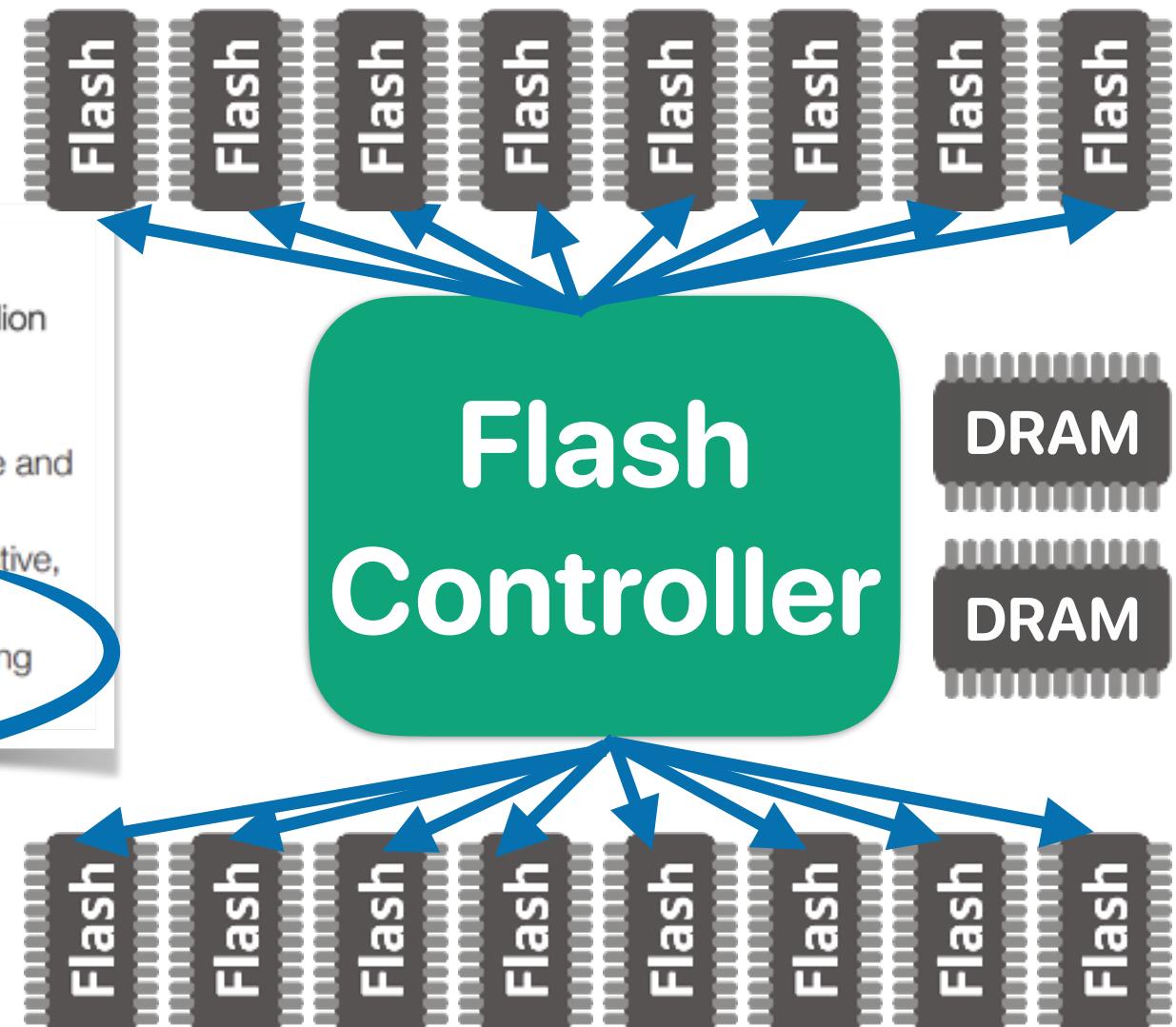
Summary

The Flashtec™ 2nd generation NVMe Controller Family enables the world's first 32 and 16 channel controllers to realize the highest performance SSDs utilizing next-generation memory technologies. Combining world-class capacity and flexibility, the Flashtec NVMe2032 and NVMe2016 controllers are the most reliable choice. The Flashtec NVMe2032 and NVMe2016 controllers support the PCIe Gen 3 x8 or dual independent PCIe Gen 3 x4 (active, active/standby) host interface and are optimized for high-performance operations, performing all Raed management operations on-chip and processing and memory resources.

Features

- Flashtec NVMe2032 controller can achieve up to 1 million random read IOPS on 4 KB operations
- Up to 20 TB Flash capacity using 256 GB Flash
- SLC, MLC, Enterprise MLC, and TLC Flash with toggle and ONFI interface
- PCIe Gen 3 x8 or dual independent PCIe Gen 3 x4 (active, active/standby) host interface
- 16 and 32 independent Flash channels, each supporting up to 8 CE

Each ~500MB/sec
8 channels == 4GB/sec
16 channels == 16 GB/sec

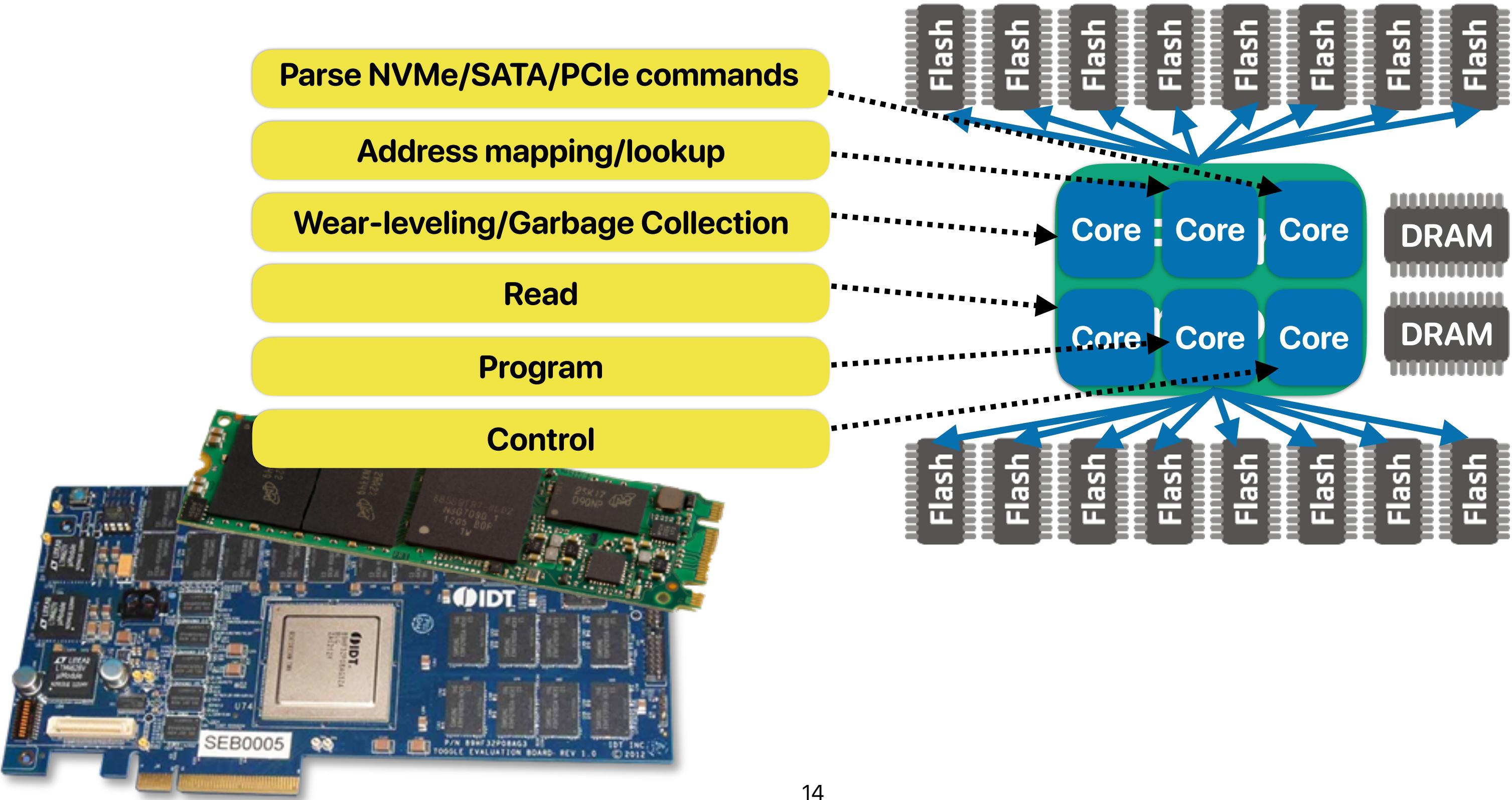


- Organization
 - Page size x8: 18,592 bytes (16,384 + 2208 bytes)
 - Block size: 2304 pages, (36,864K + 4968K bytes)
 - Plane size: 4 planes x 504 blocks
 - Device size: 512Gb: 2016 blocks; 1Tb: 4032 blocks; 2Tb: 8064 blocks; 4Tb: 16,128 blocks; 8Tb: 32,256 blocks

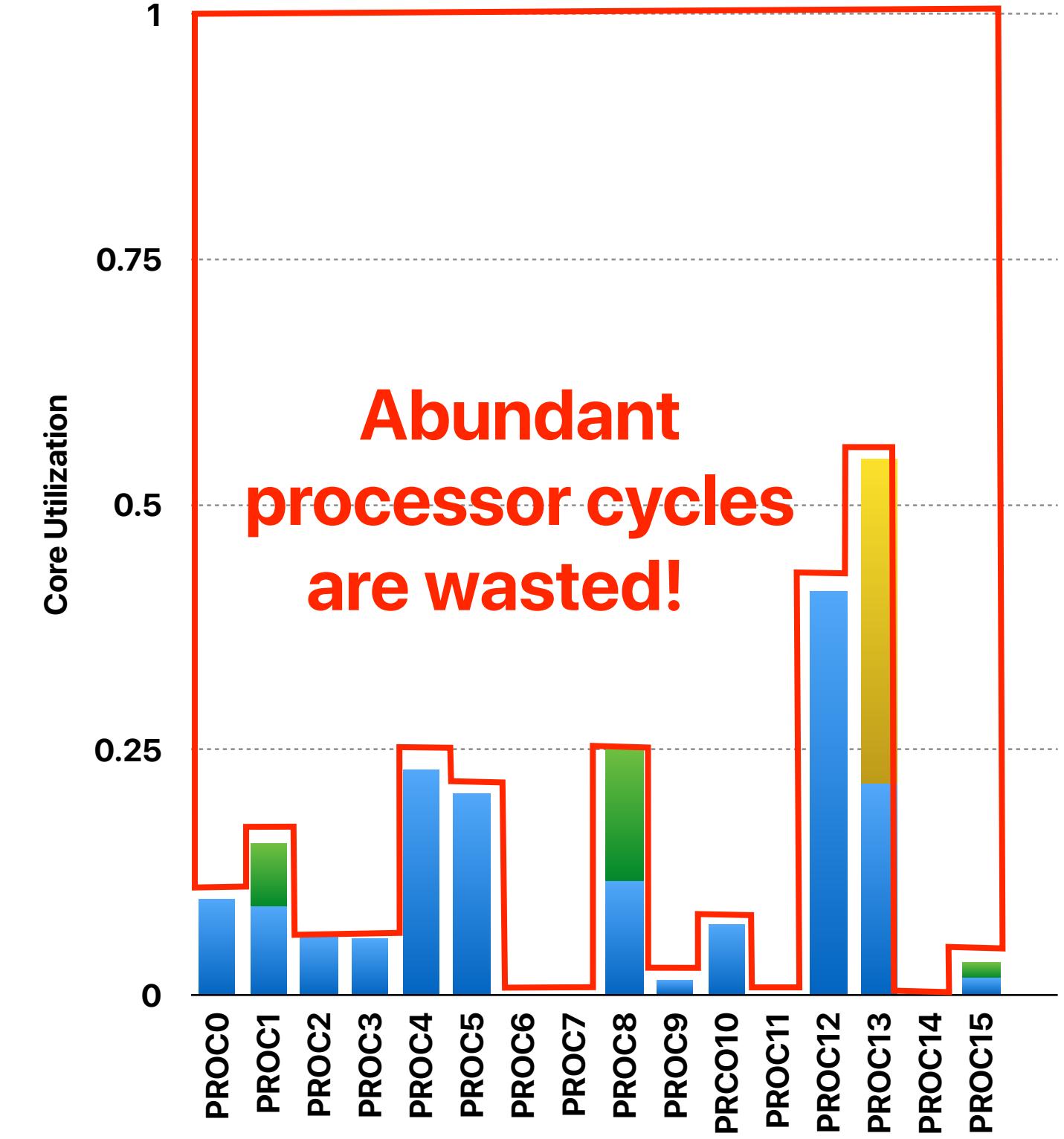
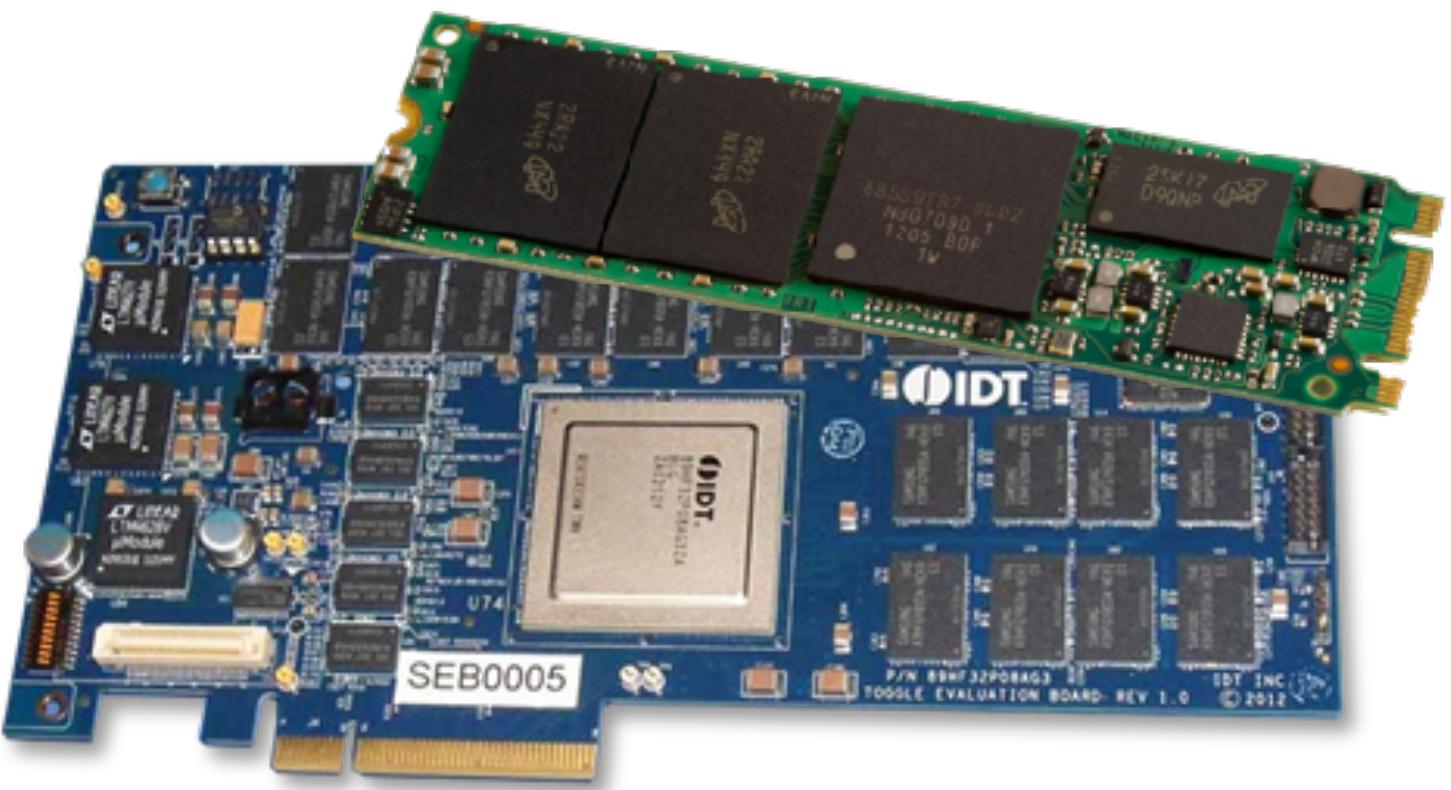
TLC 512Gb-8Tb NAND - Array Characteristics

is suspended or ongoing	t_R	57/41	60	μs	2, 12
READ PAGE operation time without/with V_{PP}	t_R	57/41	60	μs	2, 12
SNAP READ operation time	t_{RSNAP}	27	37	μs	
Cache read busy time	t_{RCBSY}	11	60	μs	2, 8, 12

SSDs already have processors inside



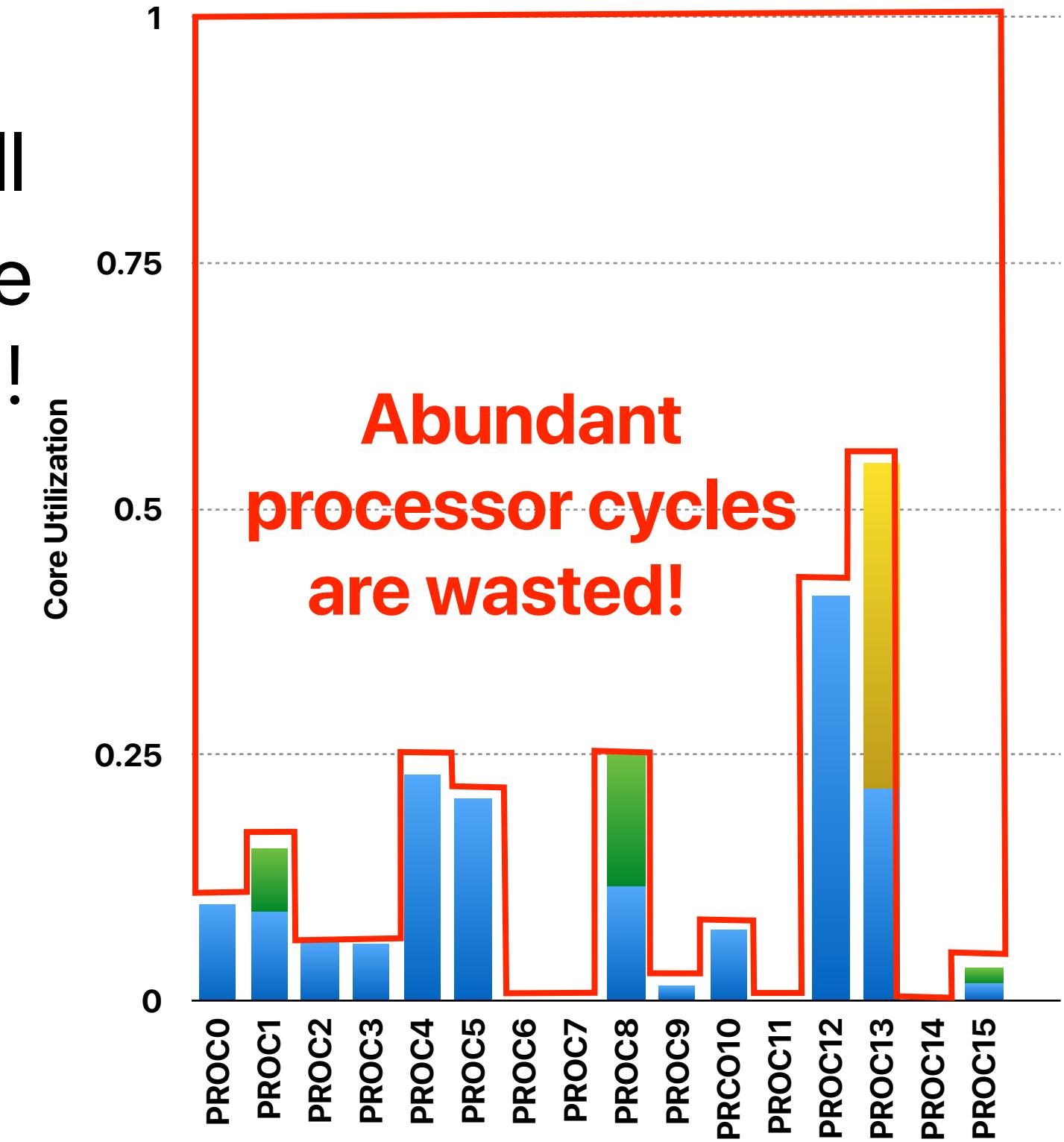
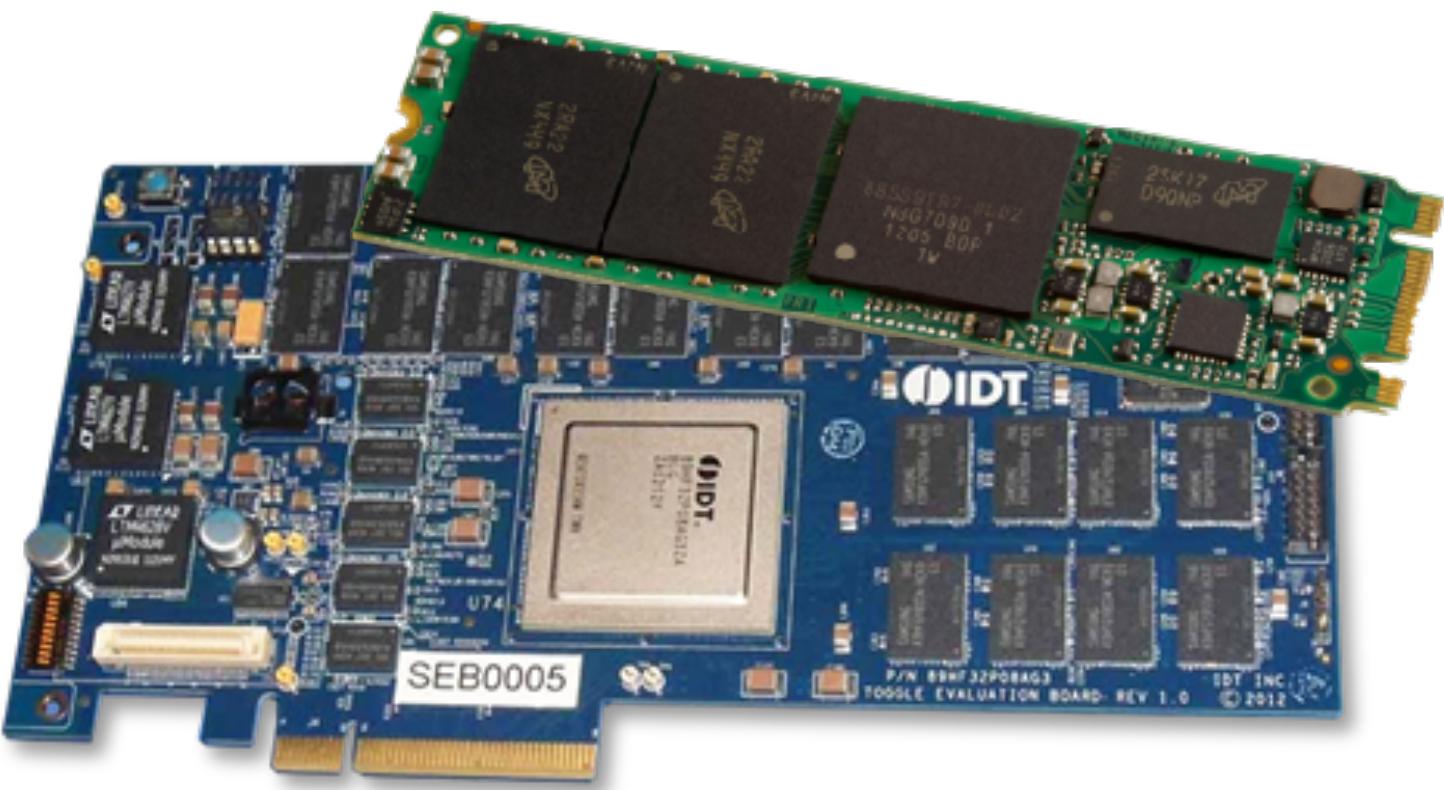
Are we using them “efficiently”?



Why are we under utilizing these processors?

SSDs already have processors inside

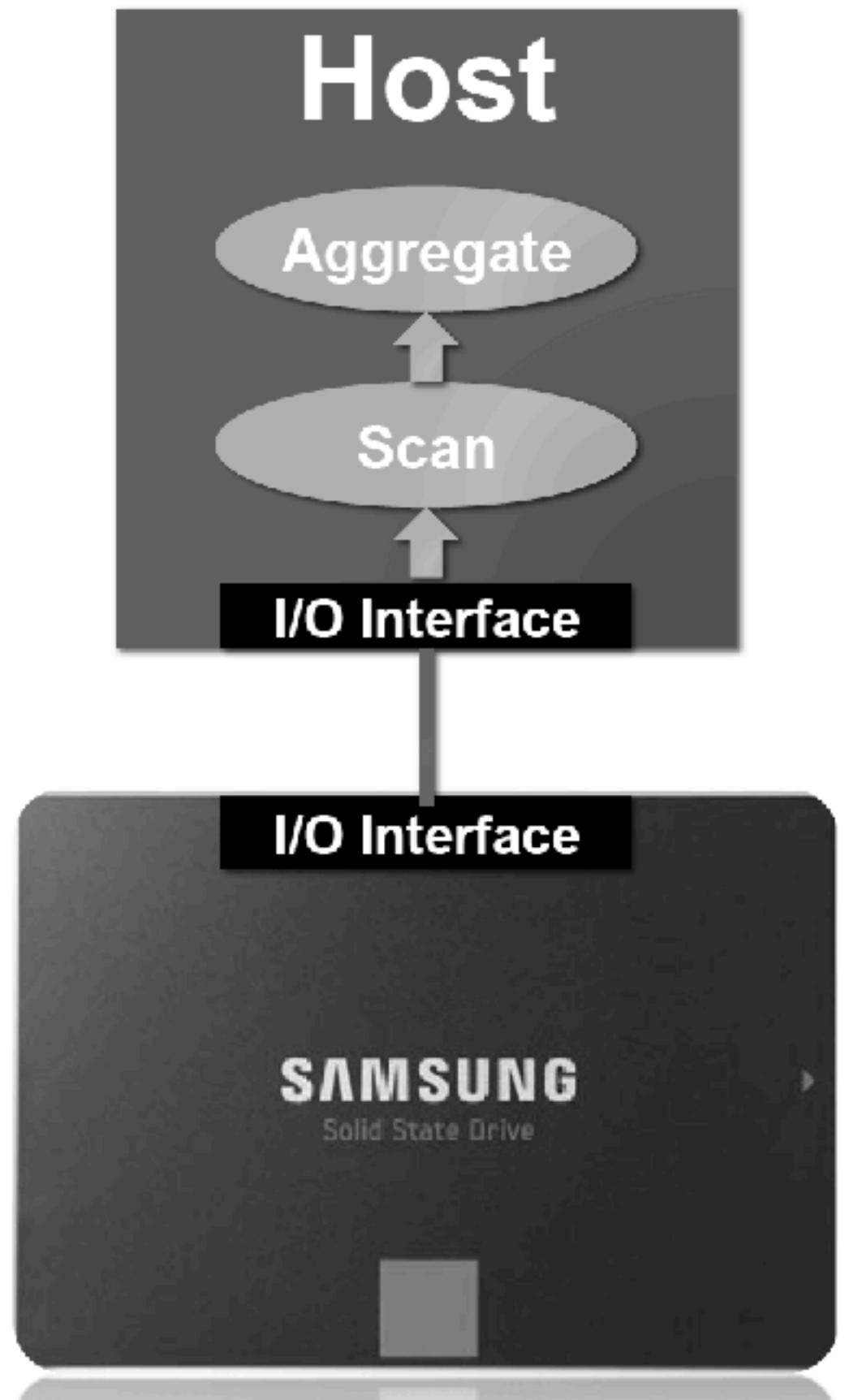
- Firmware does not rely on OS
- Utilization is not optimized as well
- We can “tweak” and “extend” the firmware to perform computation!



Query processing on smart SSDs: opportunities and challenges.

**Jaeyoung Do, Yang-Suk Kee, Jignesh M. Patel, Chanik Park, Kwanghyun Park,
and David J. DeWitt**

**In Proceedings of the 2013 ACM SIGMOD International Conference on
Management of Data (SIGMOD '13)**



Change in the System

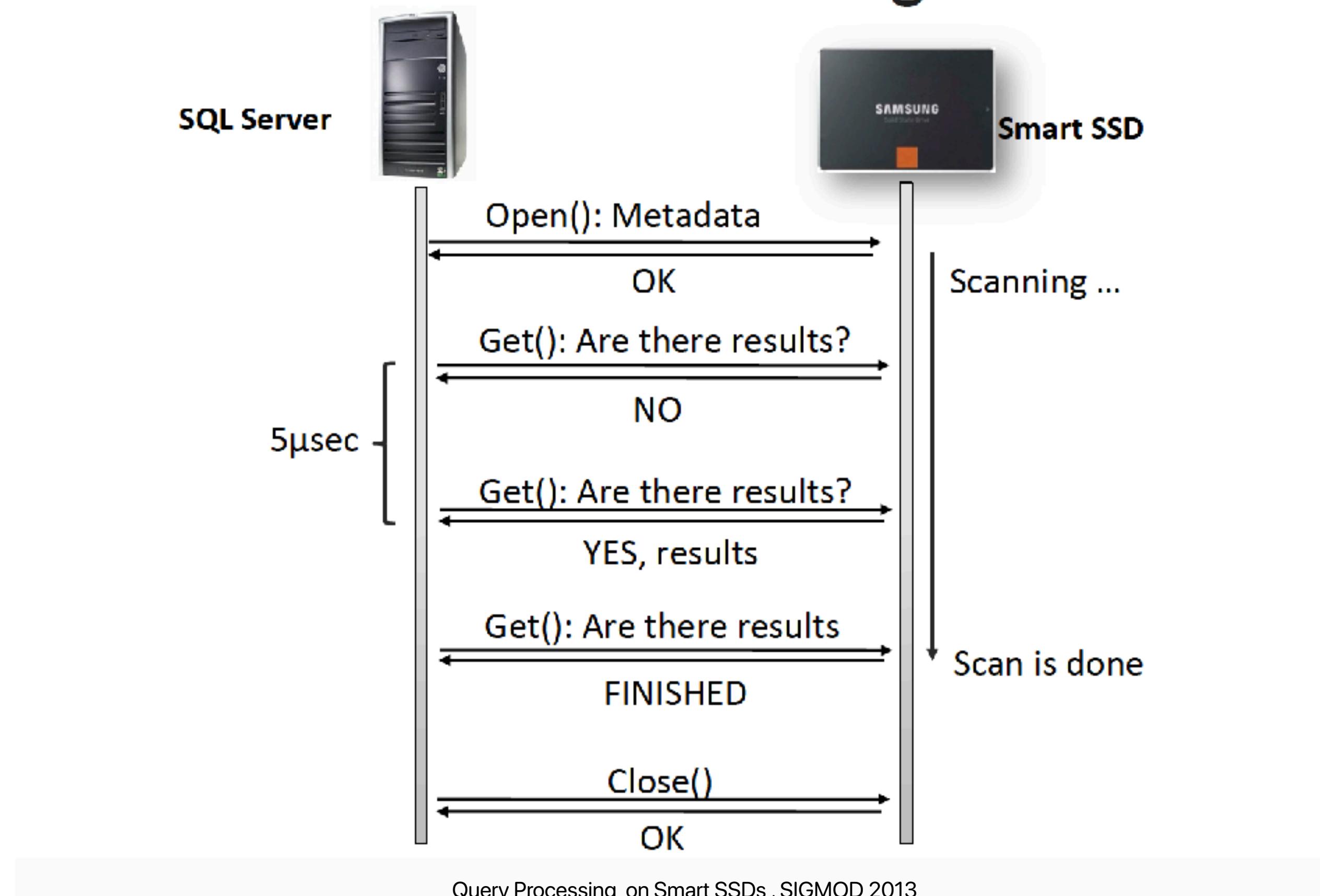
Samsung
Smart
SSD

- **New Smart SSD APIs** to run database operations inside the SSD
- Open(), Get(), Close(), SendToHost()

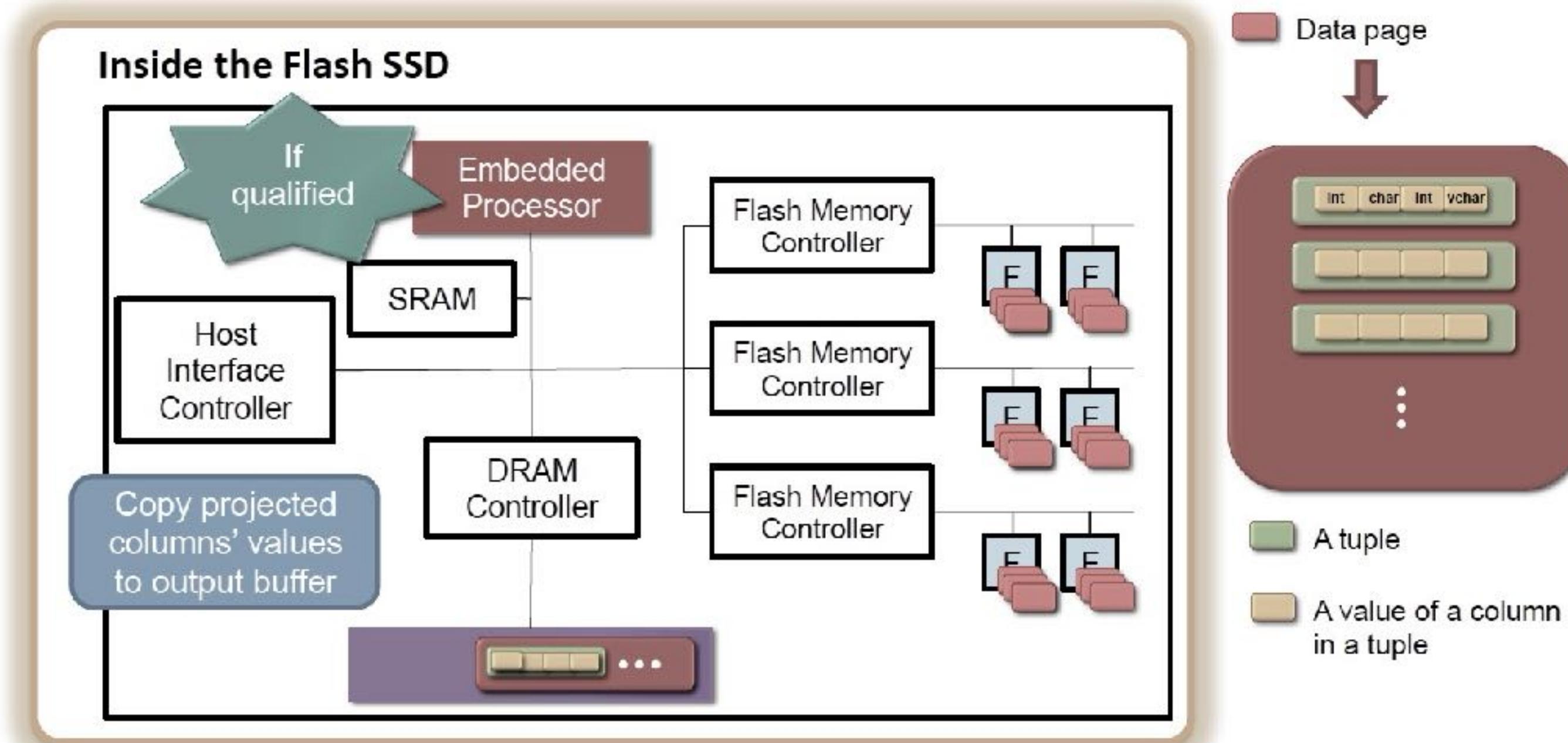
SQL
Server

- Modified the query processor to hard-code selected Smart SSD plans
- Implement Smart SSD executors

Communication Flow – e.g. Scan



Data Format in SSD



Data is read in tuples from the table

Data Format in SSD

- NSM layout



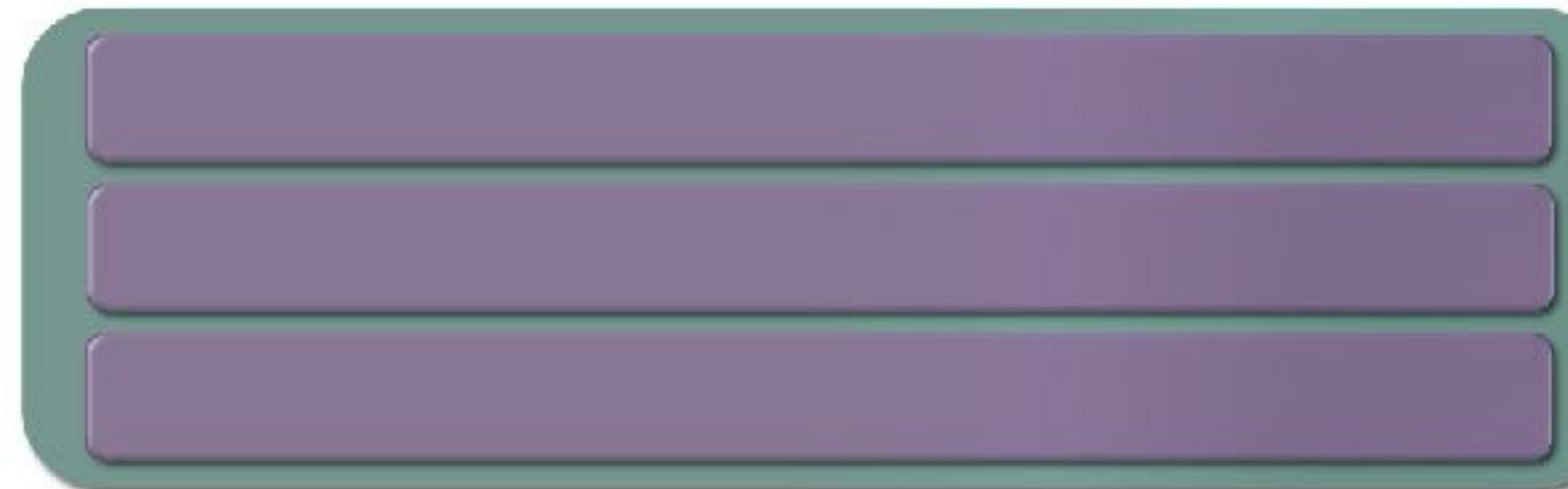
Data Page

Tuple

Column

Tuple based organization

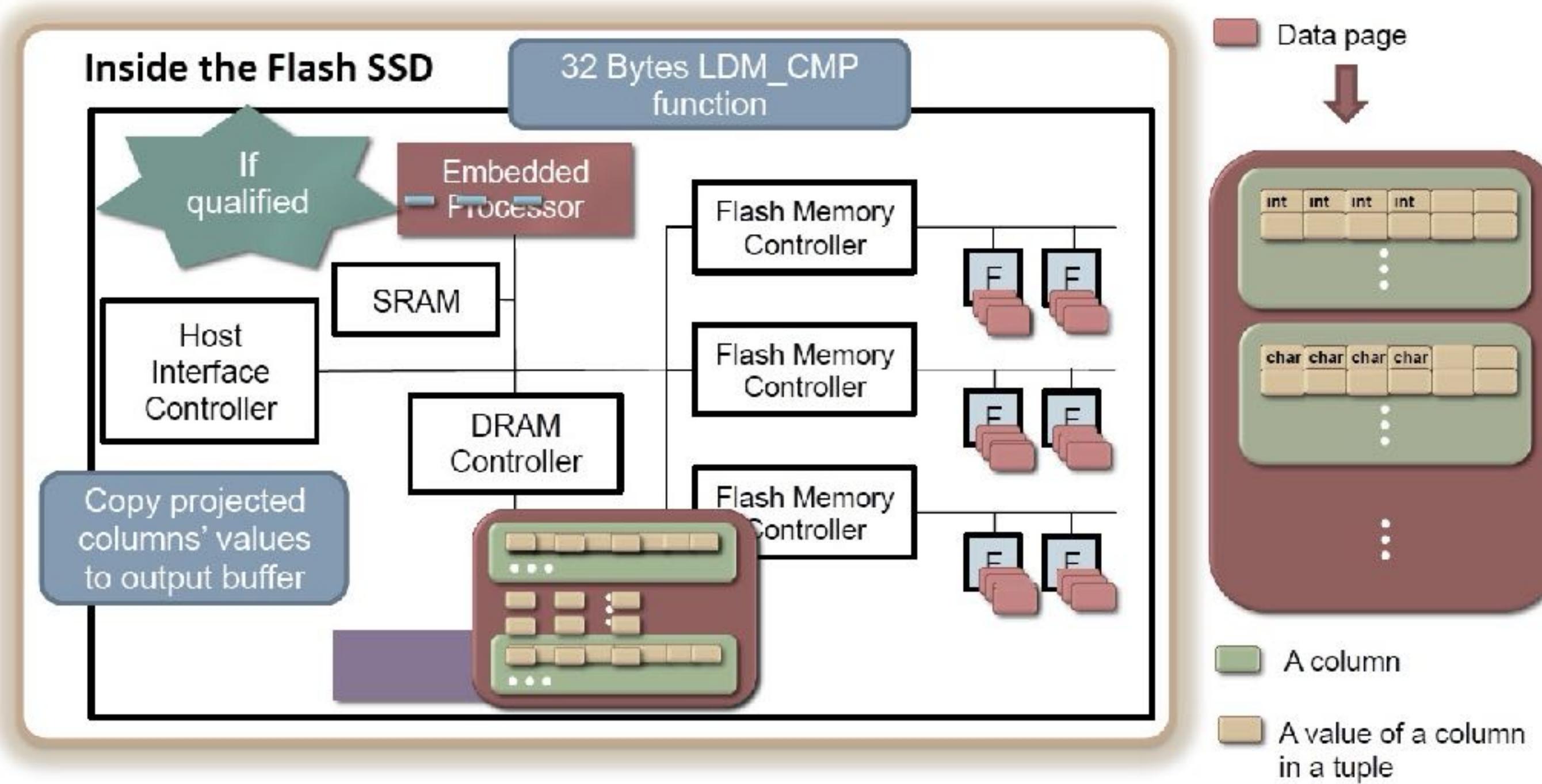
- PAX layout



Column based organization

Operations are performed on values in the same column

Column Format in SSD



Operations are performed on values in the same column

System Configuration



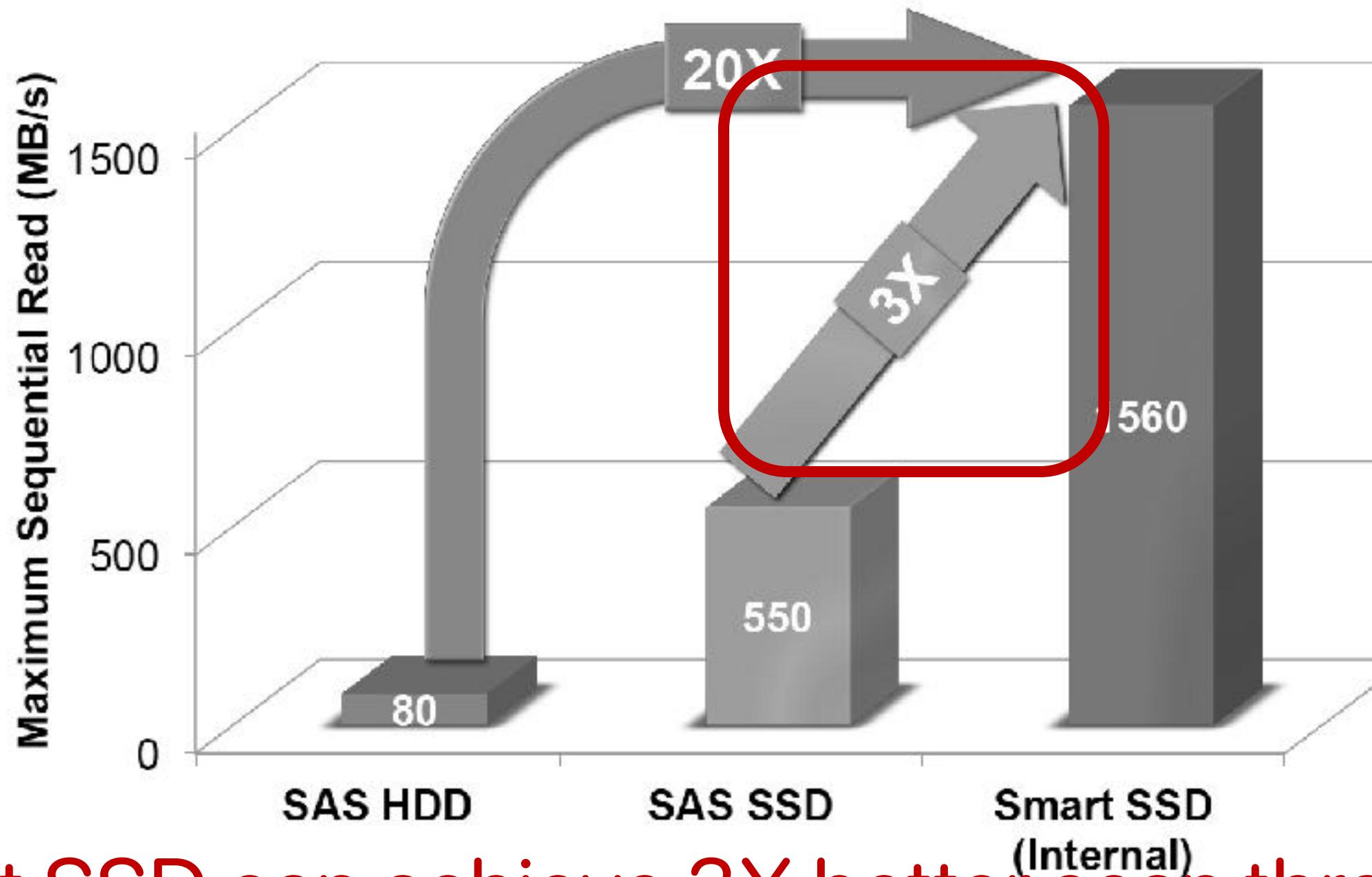
SQL server (Host machine)
64-bit Windows 7
32GB DRAM (24GB dedicated to the DBMS)
2.66 GHz Dual Quad Core Intel Xeon5430
32KB L1 cache, 6MB L2 cache
Two 7.5 RPM SATA HDDs one for the OS and one for the Log
LSI four-port SATA/SAS 6Gbps HBA (host bus adapter)

Flash SSD
Samsung 400GB SAS SSD
Two ARM Cores
16KB SRAM for each core
256KB DRAM for each core
1MB DRAM output buffer

SSD cores are lightweight, DRAM capacity is small

Upper Bound on Expected Performance Gains

Speed of a Simple Sequential Scan Program



Smart SSD can achieve 3X better scan throughput
than normal SSD

Evaluated Data Sets

Each table has a number of integer-typed columns

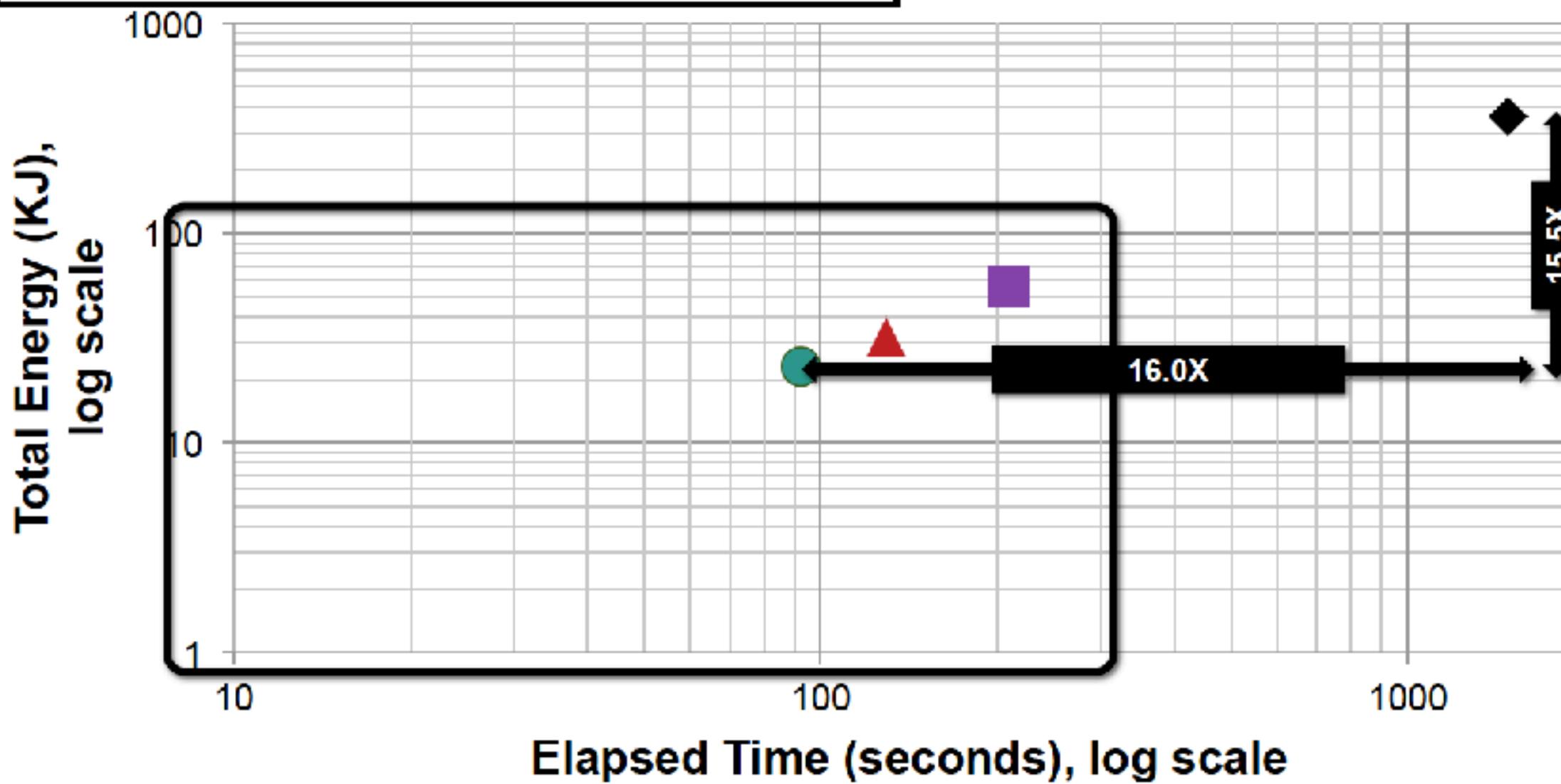
Table	# Columns	Table Size (GB)	# tuples (millions)	# tuples/page
Synthetic 4	4	10	400	323
Synthetic16	16	30	400	109
Synthetic64	64	110	400	29

Synthetic64 has the largest data size

10% Selection Query

```
SELECT COLUMN_2  
FROM Synthetic64  
WHERE COLUMN_1 < [VALUE]
```

- SAS HDD
- SAS SSD
- Smart SSD (NSM)
- Smart SSD (PAX)

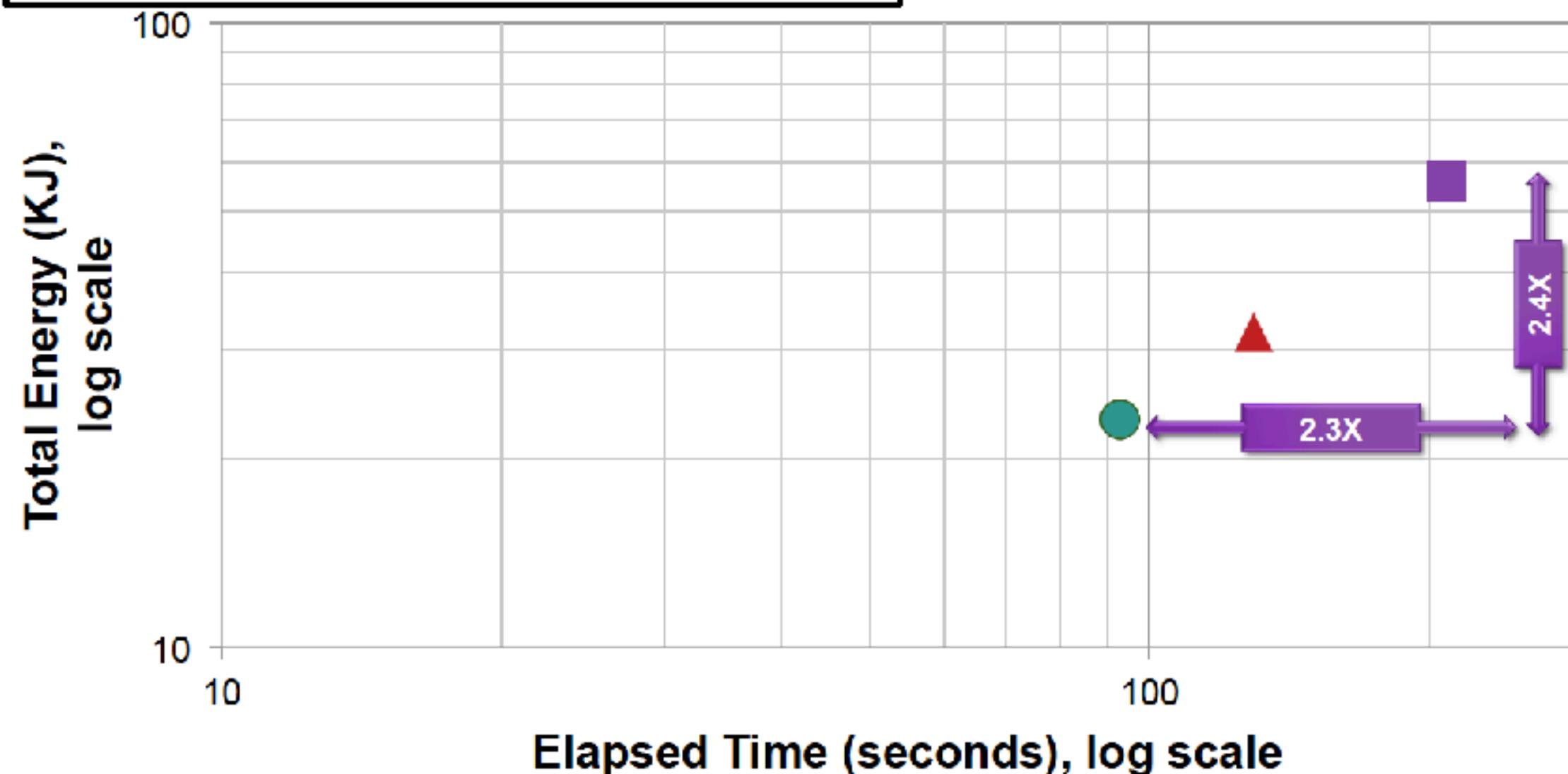


16X better energy and performance than disk

10% Selection Query

```
SELECT COLUMN_2  
FROM Synthetic64  
WHERE COLUMN_1 < [VALUE]
```

- SAS HDD
- SAS SSD
- Smart SSD (NSM)
- Smart SSD (PAX)



2.5X better energy and performance than SSD

What do you expect the performance of SmartSSDs on Synthetic 4 of the selection query?

`SELECT SecondColumn FROM SyntheticTable WHERE FirstColumn < [VALUE]`

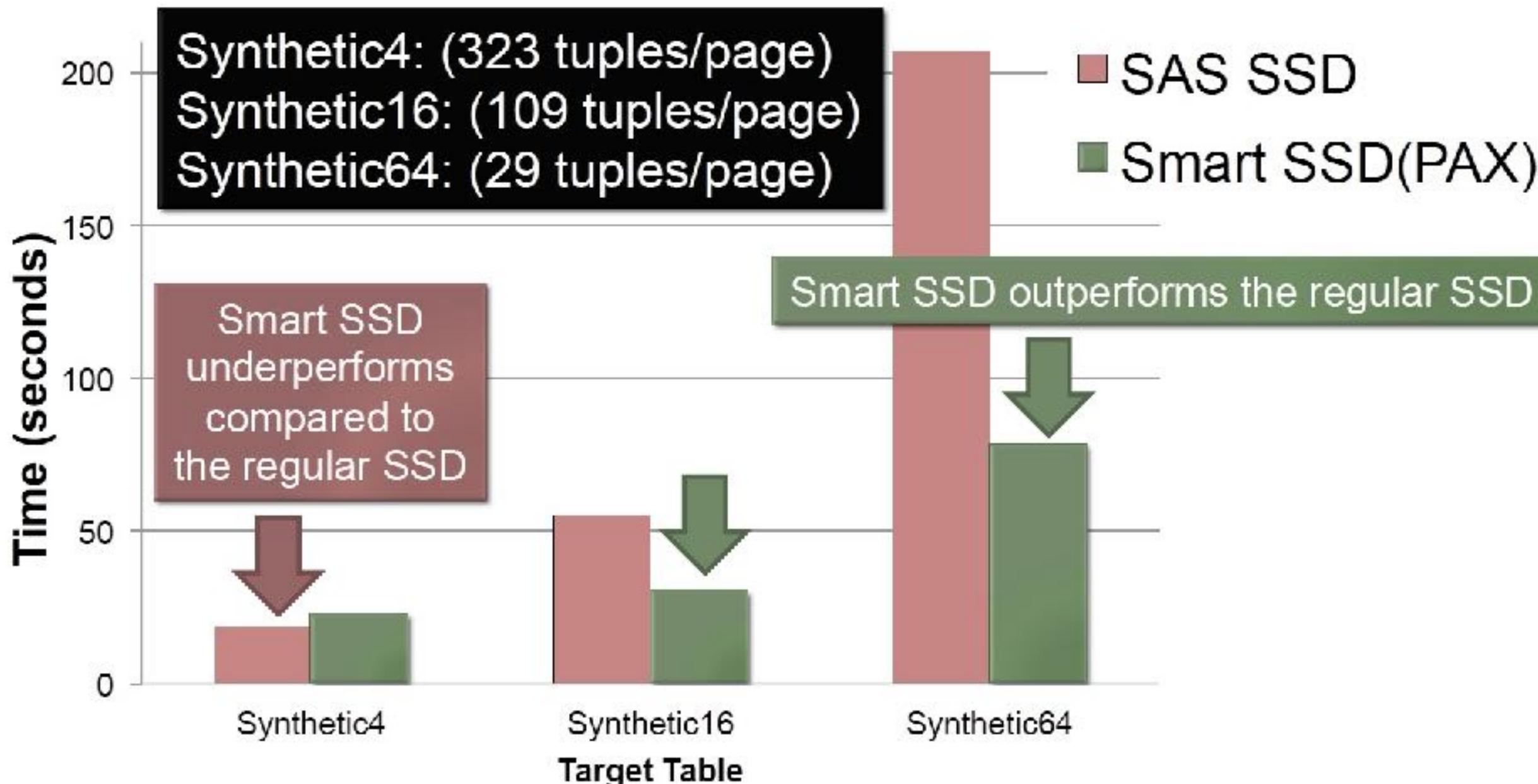
Each table has a number of integer-typed columns

Table	# Columns	Table Size (GB)	# tuples (millions)	# tuples/page
Synthetic 4	4	10	400	323
Synthetic16	16	30	400	109
Synthetic64	64	110	400	29

Is SmartSSD always a good idea?

Effect of the # of Tuples on a Page

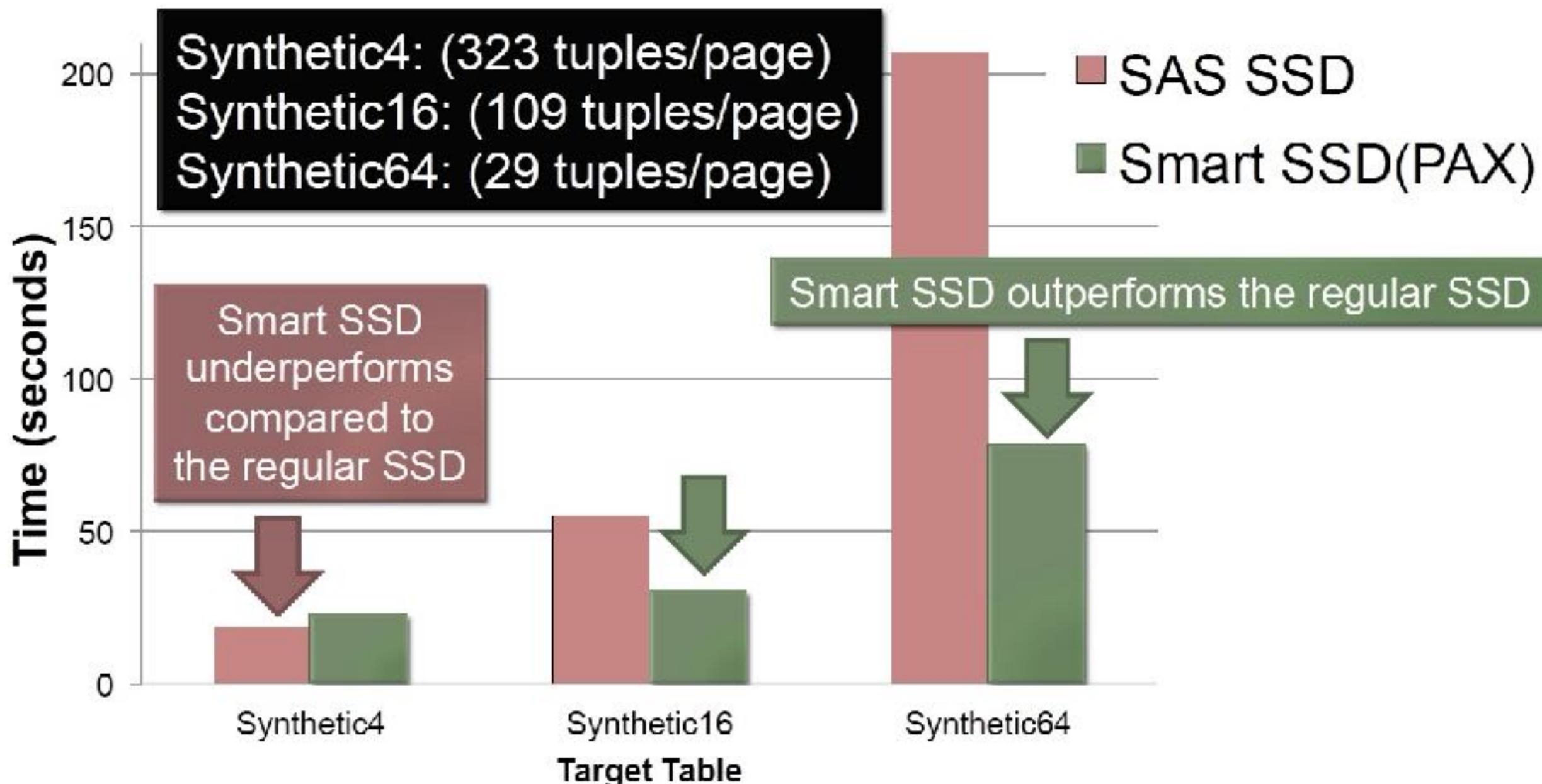
```
SELECT COLUMN_2, FROM [Synthetic_table] WHERE COLUMN_1 < [VALUE]
```



Benefit depends on the dataset size

Effect of the # of Tuples on a Page

```
SELECT COLUMN_2, FROM [Synthetic_table] WHERE COLUMN_1 < [VALUE]
```



Benefit depends on the dataset size

The “Winning Formula” of Near-Data Processing

Conventional Model



The compute intensity is higher in synthetic 4 as there are more tuples per unit of data

The diagram illustrates the ISP Model as a parallel process:

- ISP Computation (blue)
- Data XCHG SSD/Main Memory (orange)
- Data Preprocessing (green)
- Data XCHG Main Memory/Accelerator (yellow)
- Compute Kernel (purple)

Arrows show data being processed in parallel by the ISP Computation and other stages. A large bracket indicates the total computation time.

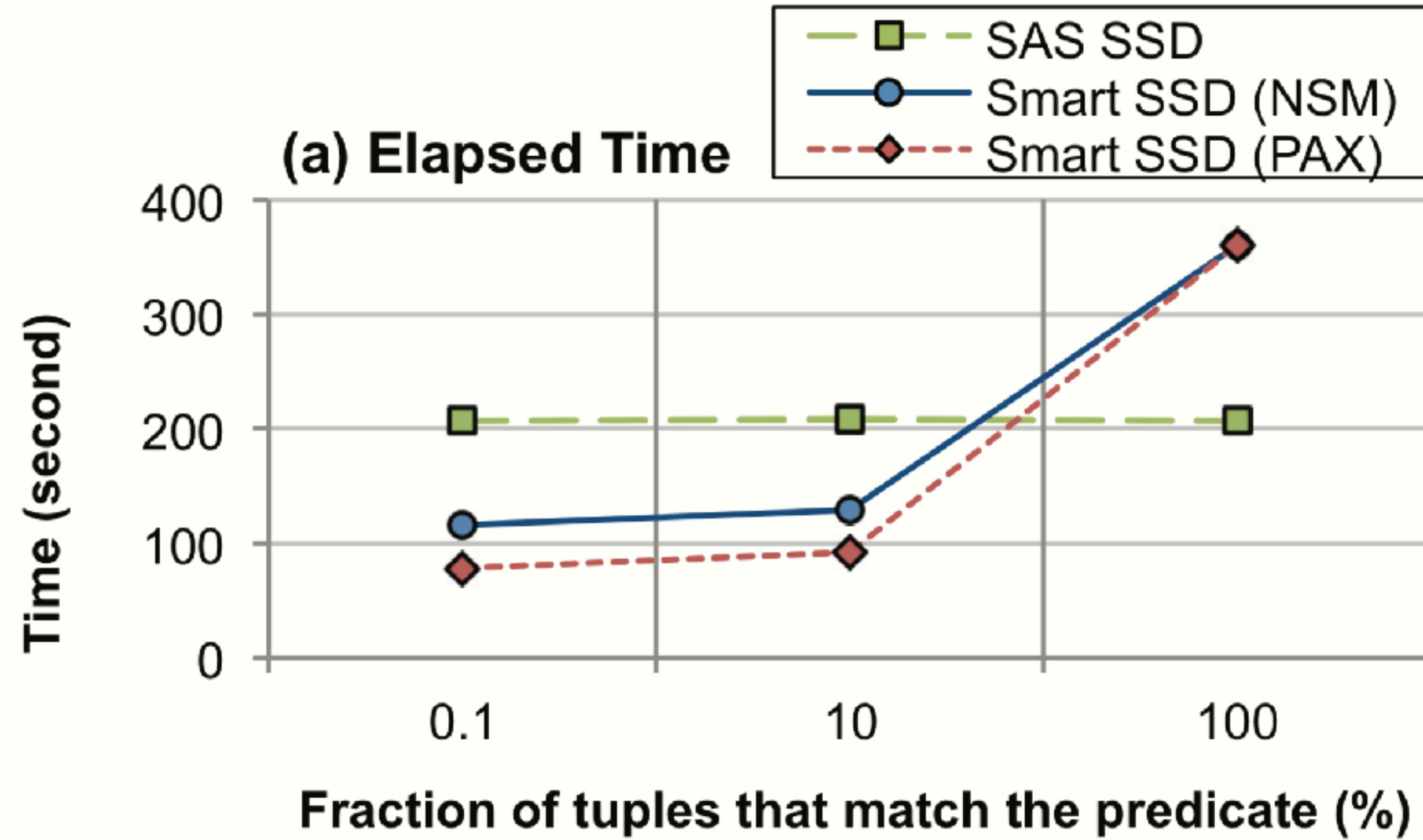
$$\frac{DataVolume_{raw}}{Bandwidth_{device}} + \frac{DataVolume_{raw} \times ComputationIntensity_{NDP}}{ComputationThroughput_{device}} + \frac{DataVolume_{afterNDP}}{Bandwidth_{device2host}} + \frac{DataVolume_{afterNDP} \times ComputationIntensity_{afterNDP}}{ComputationThroughput}$$
$$<= \frac{DataVolume_{raw}}{Bandwidth_{device2host}} + \frac{DataVolume_{raw} \times ComputationIntensity}{ComputationThroughput}$$

The ISP processor must have efficient internal access to the device

The ISP computation should not take too long

The ISP should offload the processor

SSD becomes a bottleneck



Memory and compute in SSD is a bottleneck

Why the performance of using
SmartSSD is worse if the selectivity is
high?

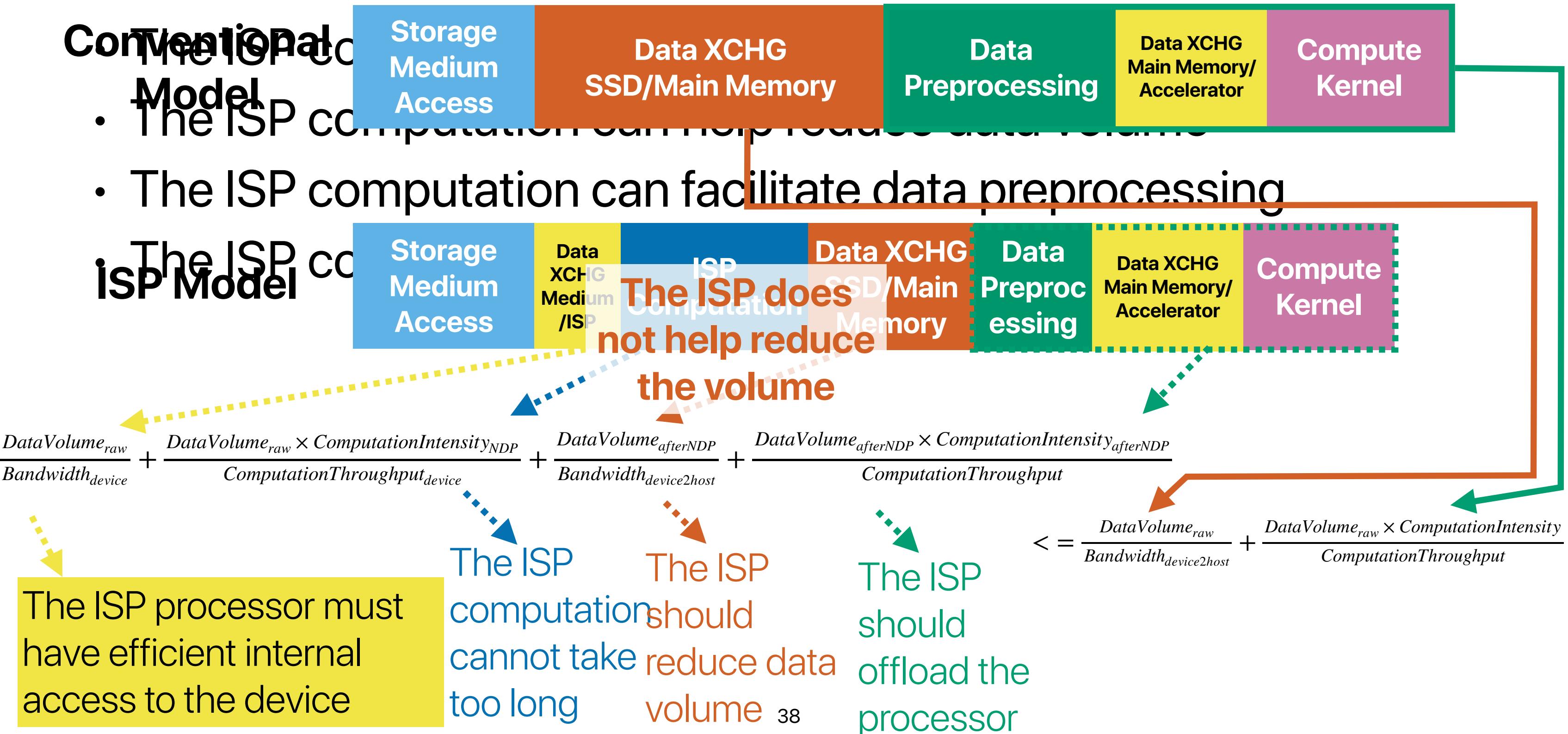
When can SmartSSDs fail?

Recap: The “Winning Formula” of In-Storage Processing

Conventional Model

- The ISP computation can help reduce data volume
- The ISP computation can facilitate data preprocessing

The ISP Model

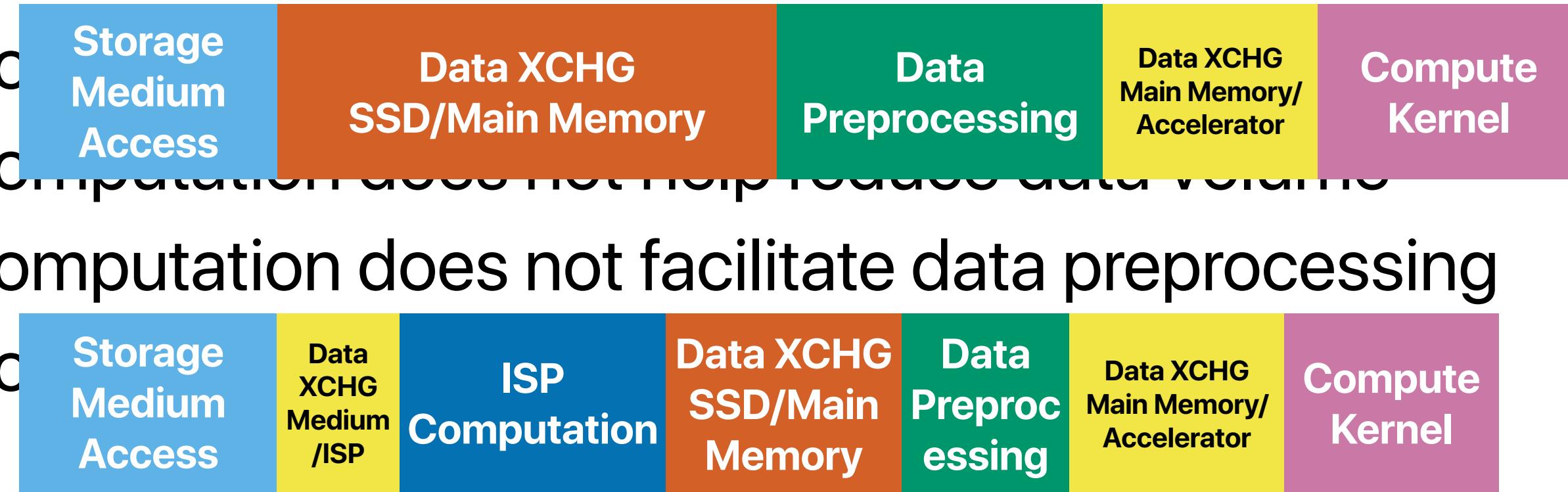


When can ISP fail?

Conventional Model

- The ISP computation does not help reduce data volume
- The ISP computation does not facilitate data preprocessing

ISP Model



SmartSSD® COMPUTATIONAL STORAGE DRIVE

OVERVIEW

The Samsung SmartSSD computational storage drive (CSD) is the industry's first adaptable computational storage platform. It empowers a new breed of software developers to easily build innovative hardware-accelerated solutions in familiar high-level languages. The SmartSSD dramatically accelerates data intensive applications by 10X or more by pushing compute to where the data lives.

At the heart of the SmartSSD is the Xilinx Adaptive Platform, a powerful toolkit which enables customers to create customized, scalable applications to solve a broad range of data center problems.

CHALLENGE

Turning Big Data Into Fast Data

The global datasphere will grow from 45 zettabytes in 2019 to 175 by 2025, and nearly 30% of the world's data will need real-time processing. This data explosion provides tremendous opportunity for business insights, but the sheer volume

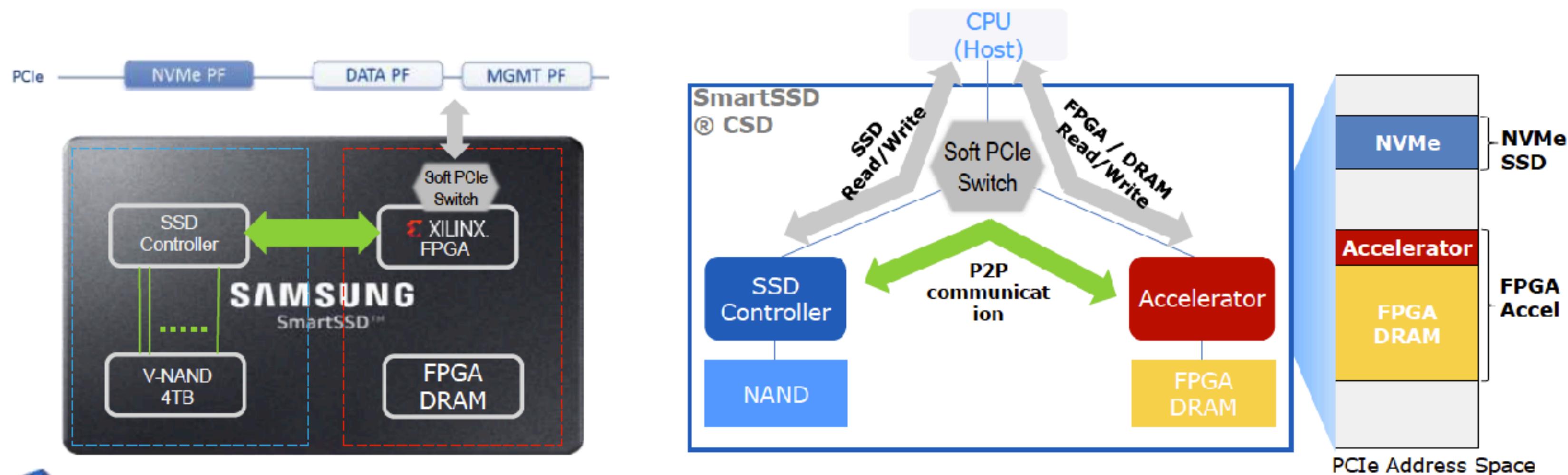


TARGET APPLICATIONS

- AI/ML Inference
- Big Data Analytics

SmartSSD® CSD HW Architecture

- Peer-to-peer (P2P) communication enables unlimited concurrency
 - SSD:Accelerator data transfers use internal data path
 - Save precious L2:DRAM Bandwidth (Compute Nodes)
 - Avoid the unnecessary funneling and data movement of standalone accelerators
 - FPGA DRAM is exposed to Host PCIe address space
 - NVMe commands can securely stream data from SSD to FPGA peer-to-peer



Developing on SmartSSD® CSD

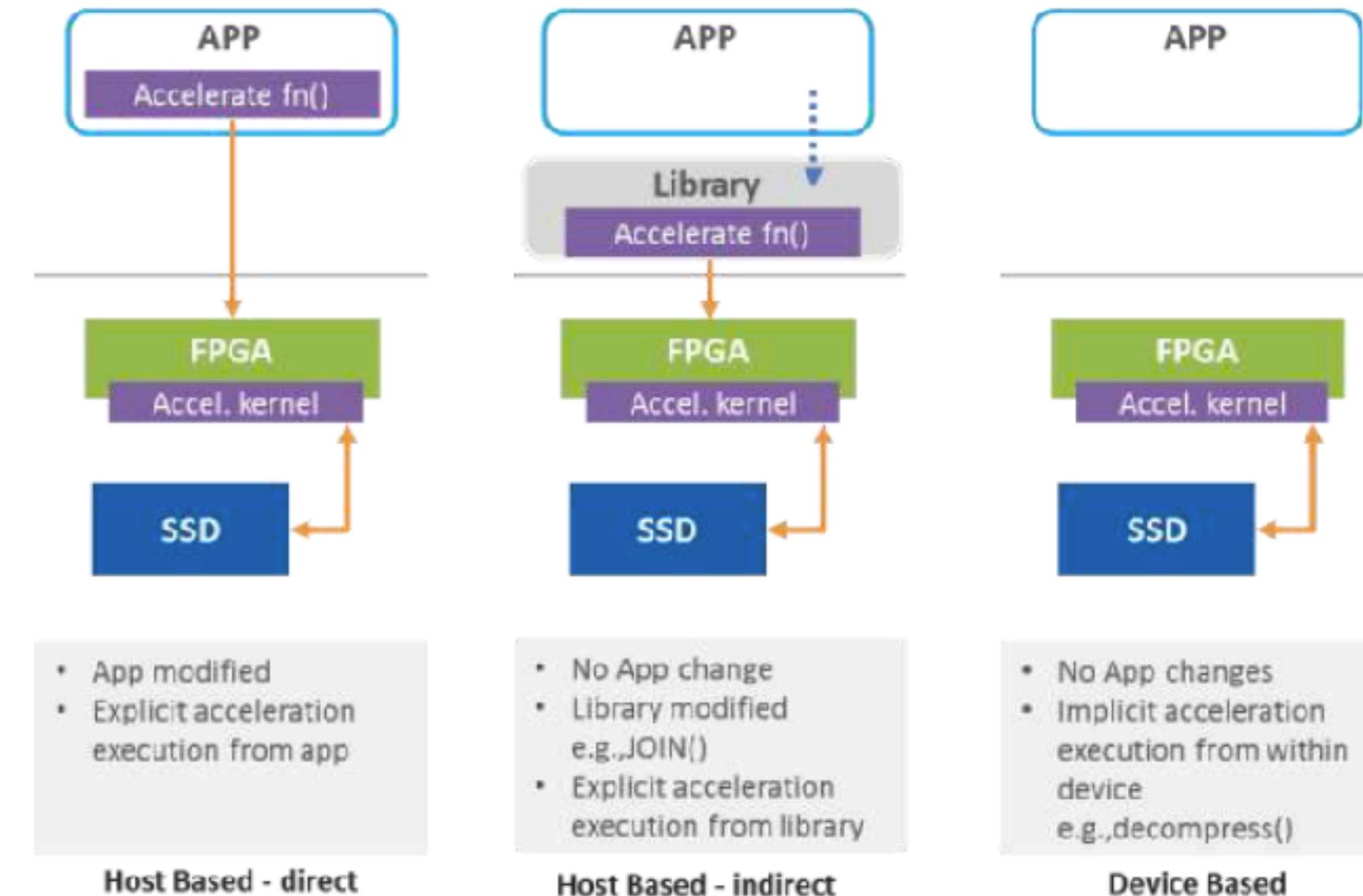
- Frameworks supported by partners

- Spark, Kafka available
- FFmpeg coming
- Many more in development

- Supported OSes

- Linux
- FreeBSD
- Windows Server

- Ease of porting for SDAccel OpenCL developers



- Vivado-friendly for RTL developers

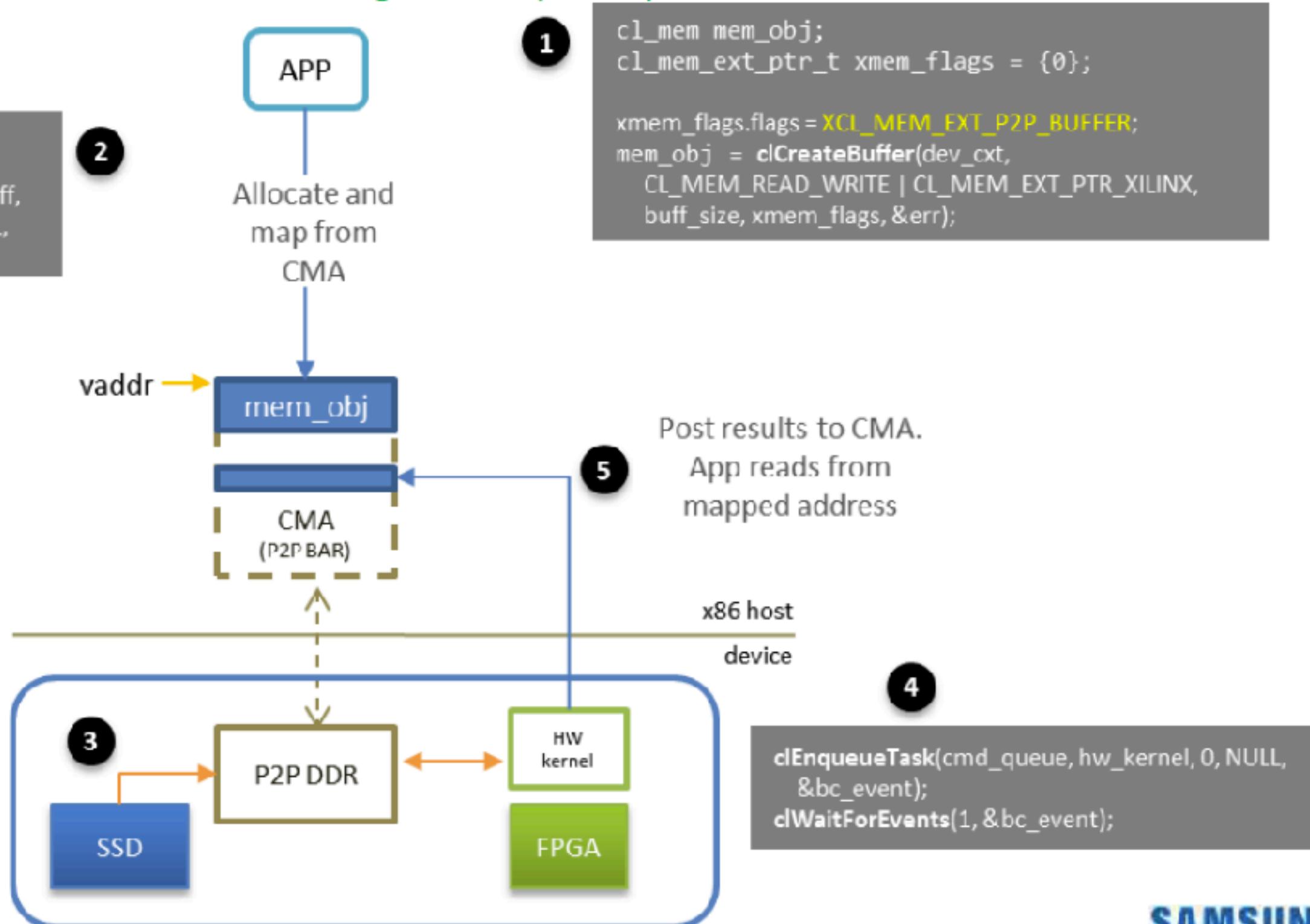
SmartSSD® OpenCL Programming in 5 Steps

- Secure, P2P data movement
 - Data moves in/out of SSD only under control of storage stack (NVMe)

```
void *vaddr;  
  
vaddr = clEnqueueMapBuffer(cmd_queue, cl_buff,  
    CL_TRUE, host_access, 0, buff_size, 0, NULL, NULL,  
    &err);
```

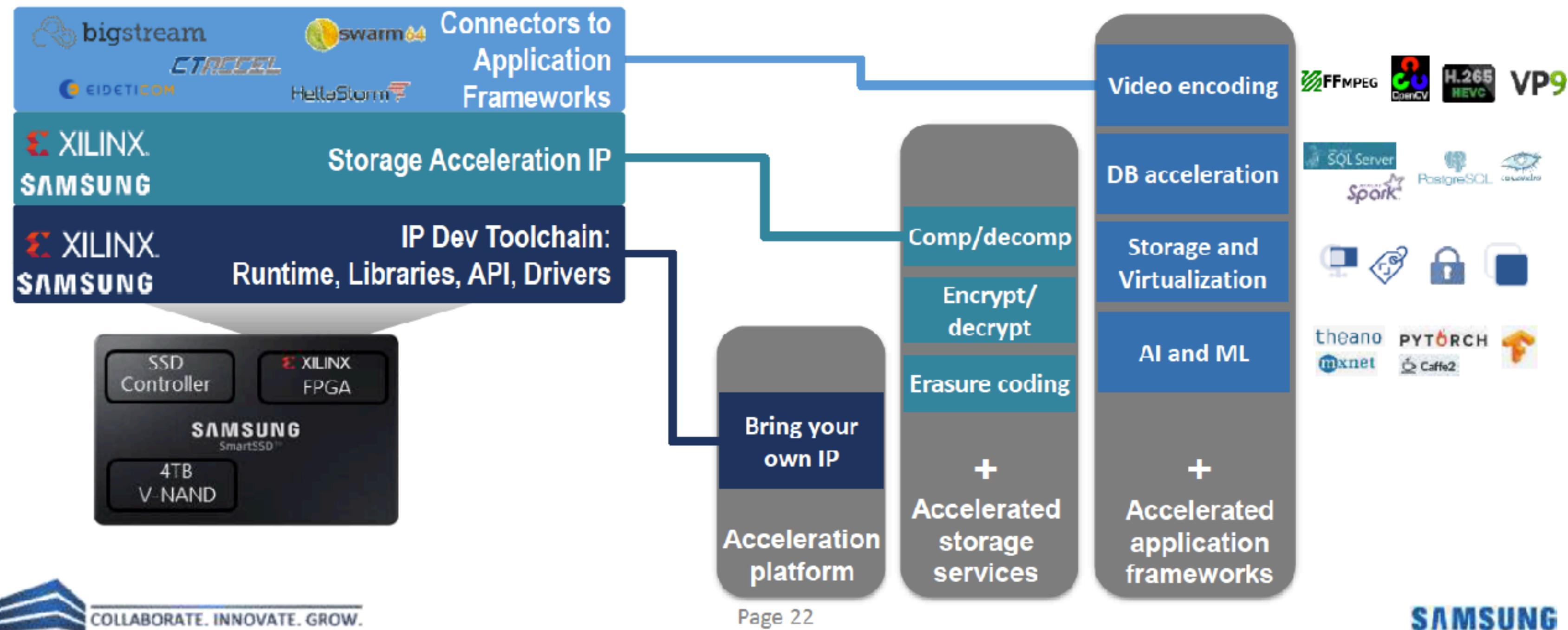
```
bytes_read = read(fd, vaddr, buff_size);
```

Initiate peering
transfers from SSD



SmartSSD® CSD Use Cases and Ecosystem

- **Storage Services:** Comp/Decomp, Encryp/Decrypt, Metadata management, Erasure Coding,
- **Real-time Analytics & Biz Intelligence:** DB Query (Spark, PostgreSQL, ..), Log analytics, genomics, physics
- **Rich Media and ML:** transcoding, live streaming, object detection



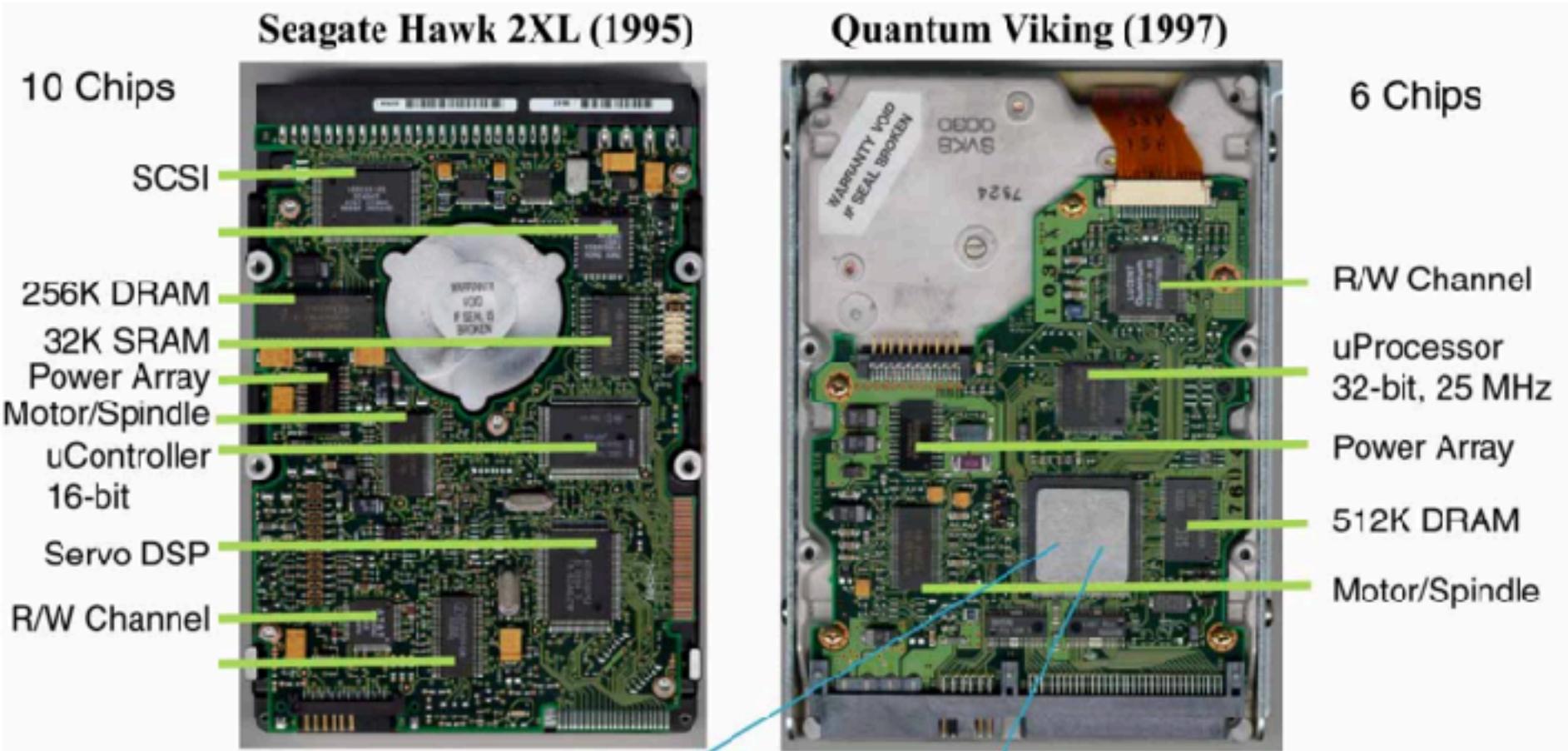
**In-storage processing isn't a new
idea...**

Active disks: programming model, algorithms and evaluation

Anurag Acharya, Mustafa Uysal, and Joel Saltz.

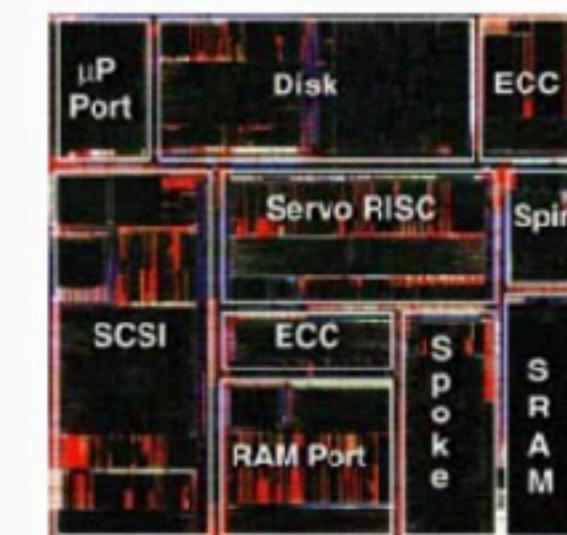
**In Proceedings of the eighth international conference on Architectural supports
for programming languages and operating systems (ASPLOS VIII), 1998**

Evolution of Disk Drive Electronics

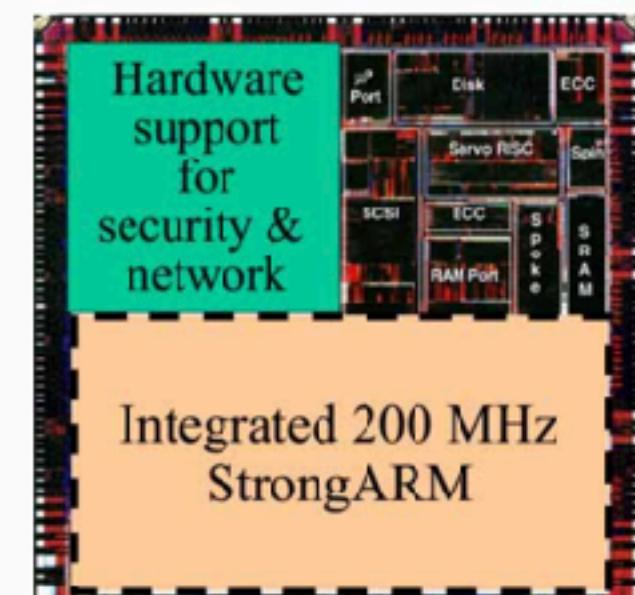


Integration

- **reduces chip count**
- **improves reliability**
- **reduces cost**
- **future integration to processor on-chip**
- **but there must be at least one chip**



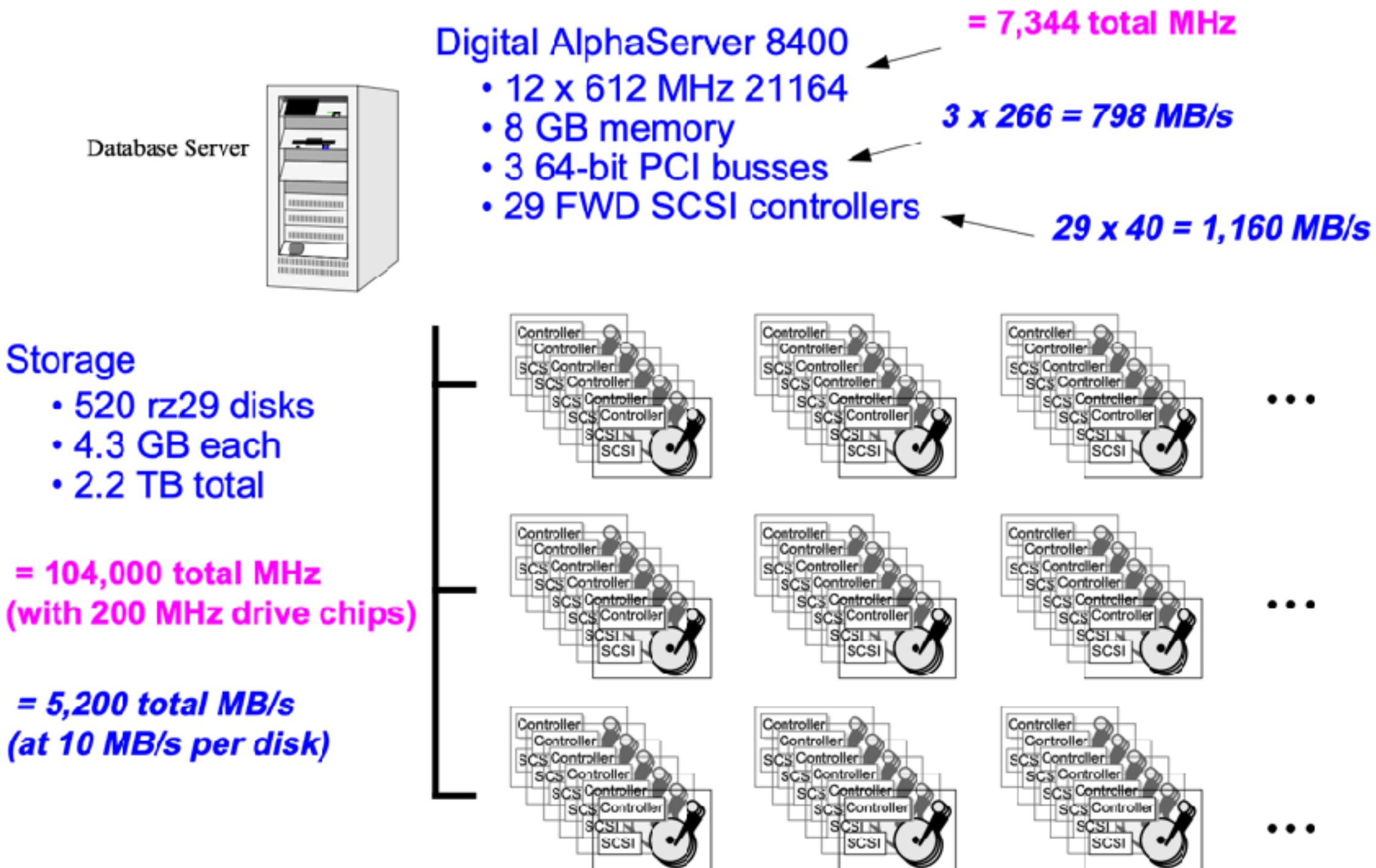
Trident ASIC



Future Generation ASIC

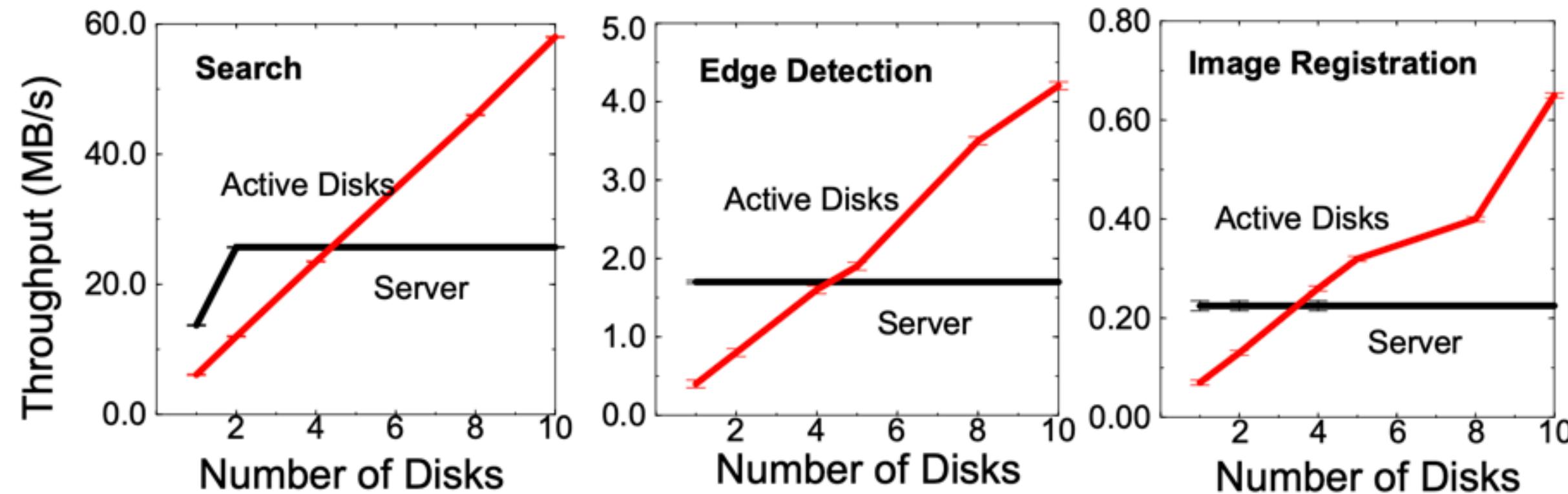
Opportunity

TPC-D 300 GB Benchmark, Decision Support System



Performance with Active Disks

application	input	computation throughput (inst/byte)	throughput (MB/s)	memory (KB)	selectivity (factor)	bandwidth (KB/s)
Search	$k=10$	7	28.6	72	80,500	0.4
Frequent Sets	$s=0.25\%$	16	12.5	620	15,000	0.8
Edge Detection	$t=75$	303	0.67	1776	110	6.1
Image Registration	-	4740	0.04	672	180	0.2



Why “active disks” did not work
out in real practice?

Why isn't ActiveDisk working?

Why Active Disks did not work out?

- Computation still dominates at that era
- Magnetic disks do not really have too much to do with “internal bandwidth”
- Independent advances like distributed storage is similar to active disk
- New features at added cost may not be used by many and convincing system vendors to implement support is hard
- Business model — hard disk drives focus on low cost per unit of data

Announcement

- Final project presentation schedule?
 - 3/7 and 3/14 in class or
 - 3/21 11:30a-2:30p (the scheduled final exam time)
- CEN Tech Talk this Friday @ 12:30p /WCH 205/206 by Zoe Wang from Google

Electrical Computer Science Engineering

277

つづく

