EE/CS277: Data-Centric Computer Architecture

Hung-Wei Tseng

Brief introduce yourself & one thing fun you’ve done during the holiday season

Instructor — Hung-Wei Tseng

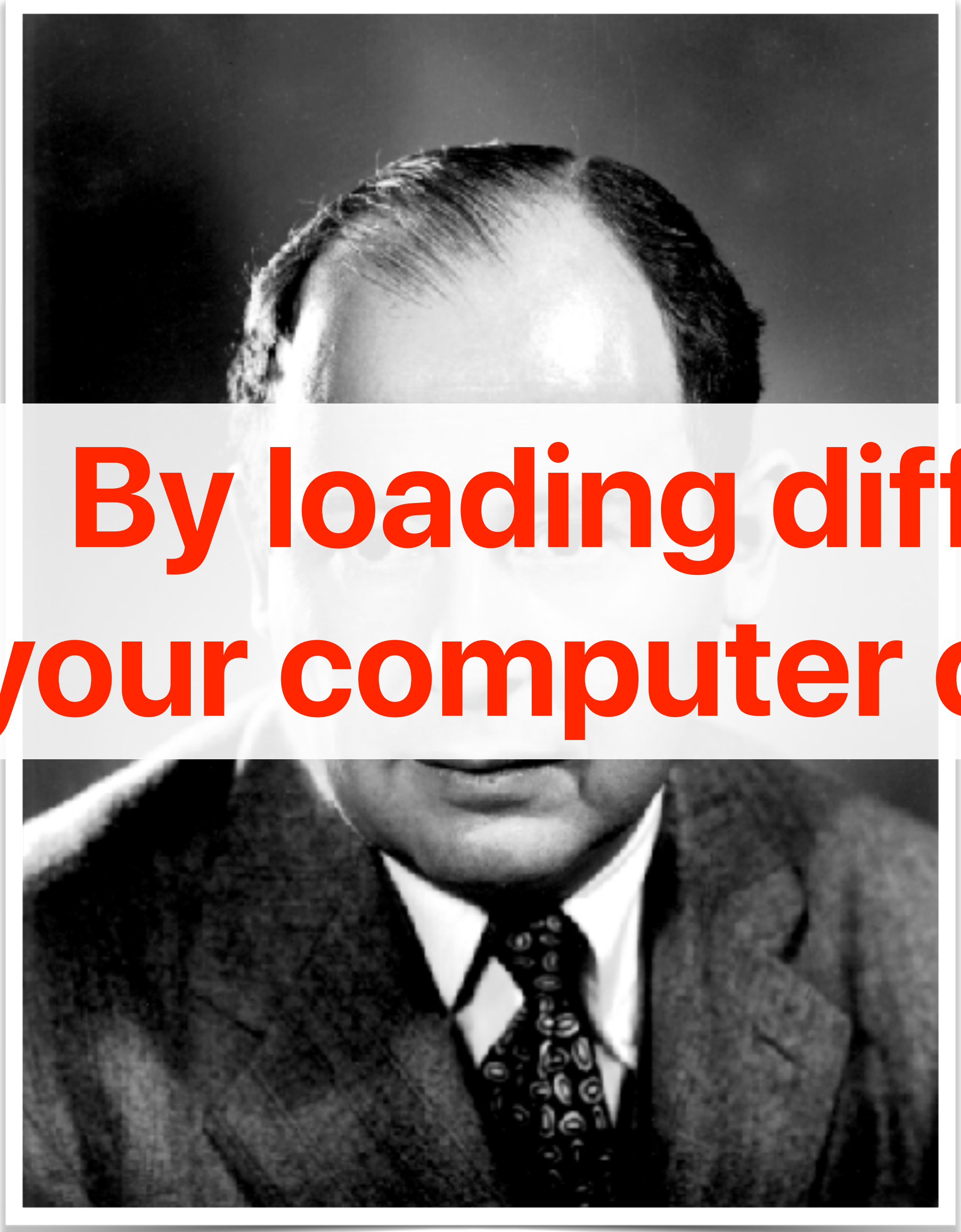
- Associate Professor @ UC Riverside, 05/2019—
- Website: https://intra.engr.ucr.edu/~htseng/
- E-mail:
- Visiting Researcher @ Google, 01/2023—03/2023
    - Working for TensorFlow Lite
- PhD in **Computer Science**, University of California, San Diego, 2014
- Research Interests
    - General-purpose computing on AI/ML/NN accelerators
    - Intelligent storage devices & near-data processing
    - Or anything else fun — we have an OpenUVR project recently
- Fun fact: Hung-Wei was once considering a career path as a singer but went back to academia due to the unsuccessful trial

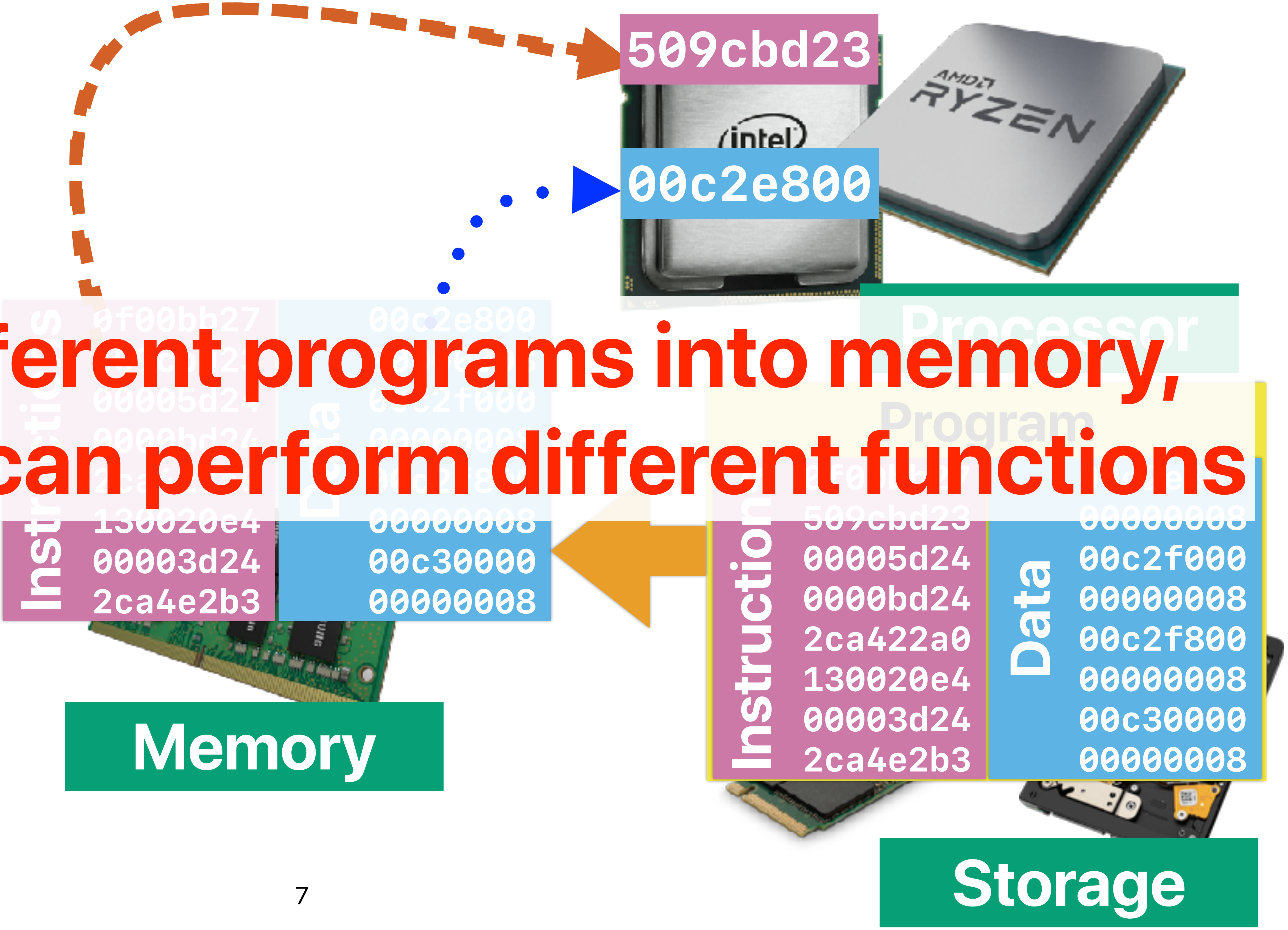

3:00

How does a computer process & store data? How does that reflect in programming?

Ideas?

von Neumman Architecture

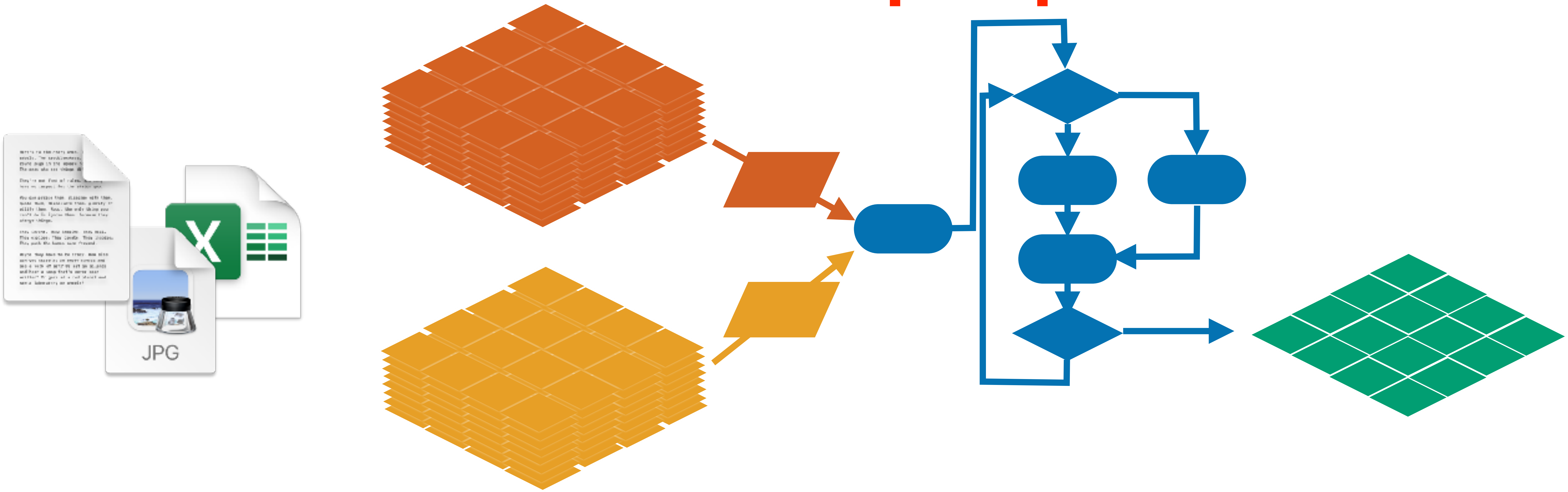




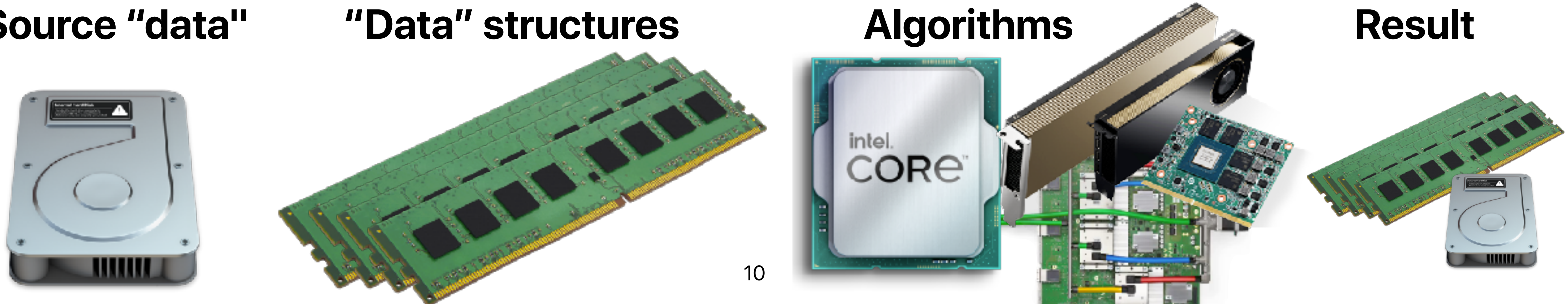
By loading different programs into memory, your computer can perform different functions

Let’s take a look of a program

From the software perspective

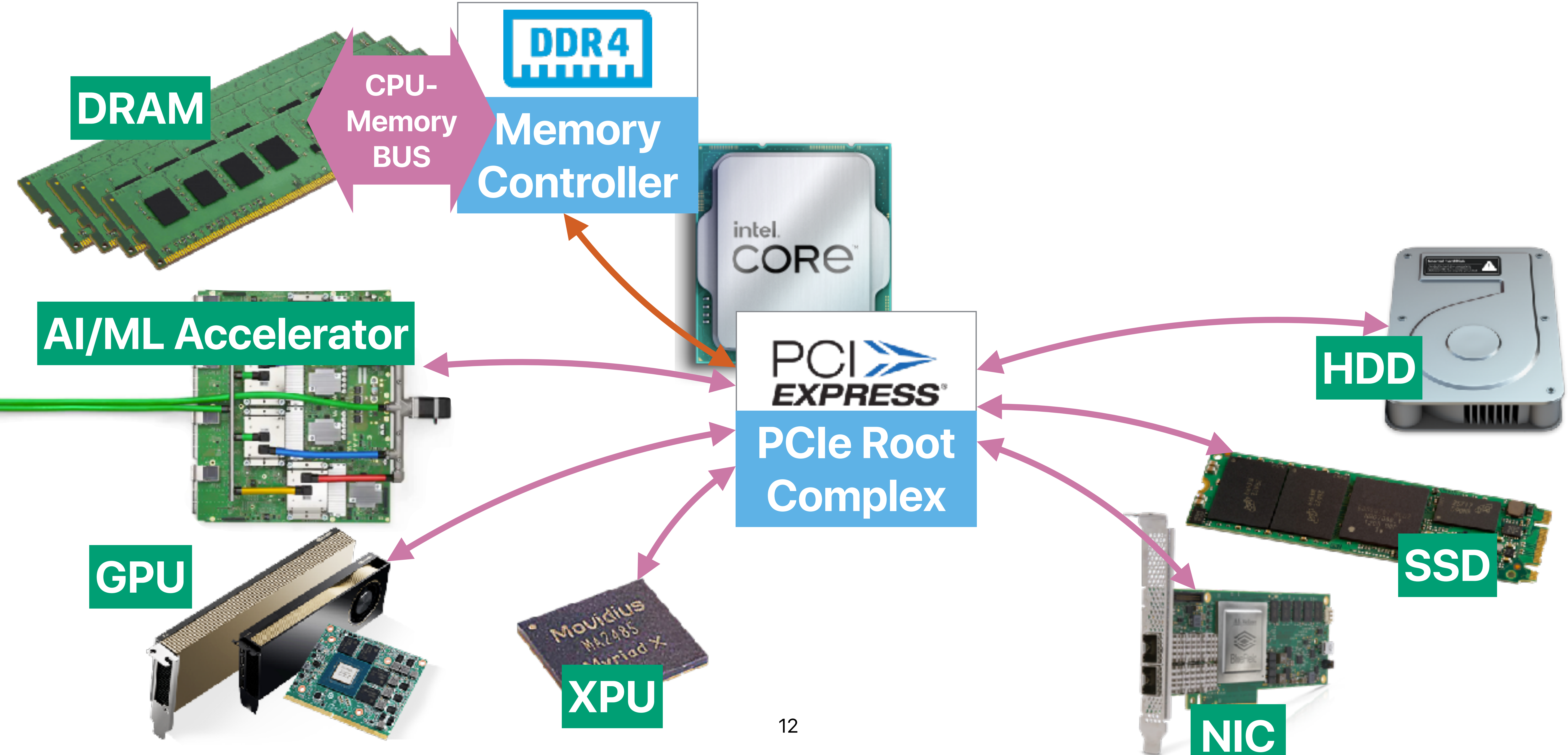


Source "data" | "Data" structures | Algorithms | Result



What’s the “center” of current computer architectures?

The "real" datapath

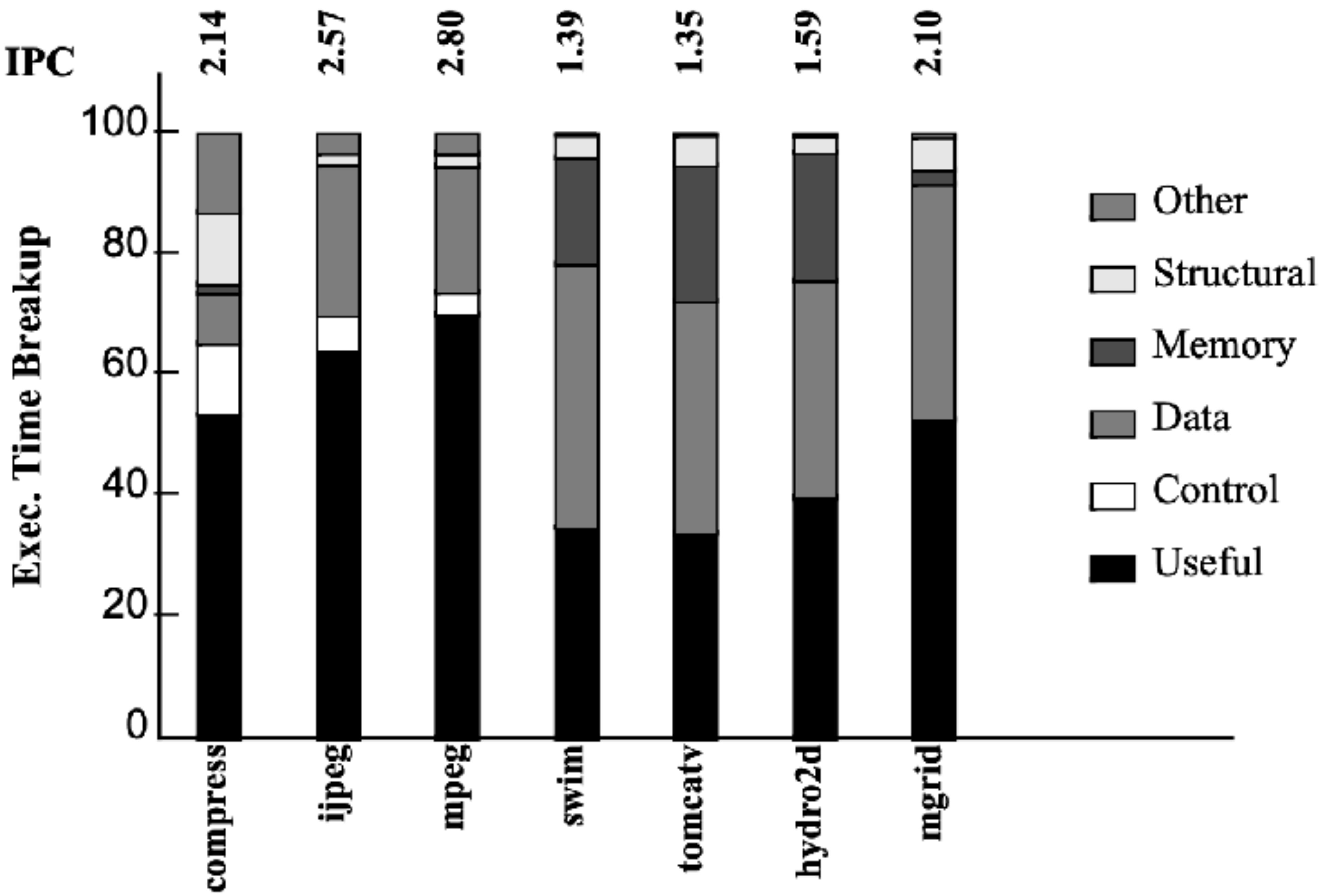


The current architecture is “CPU-centric”

3:00

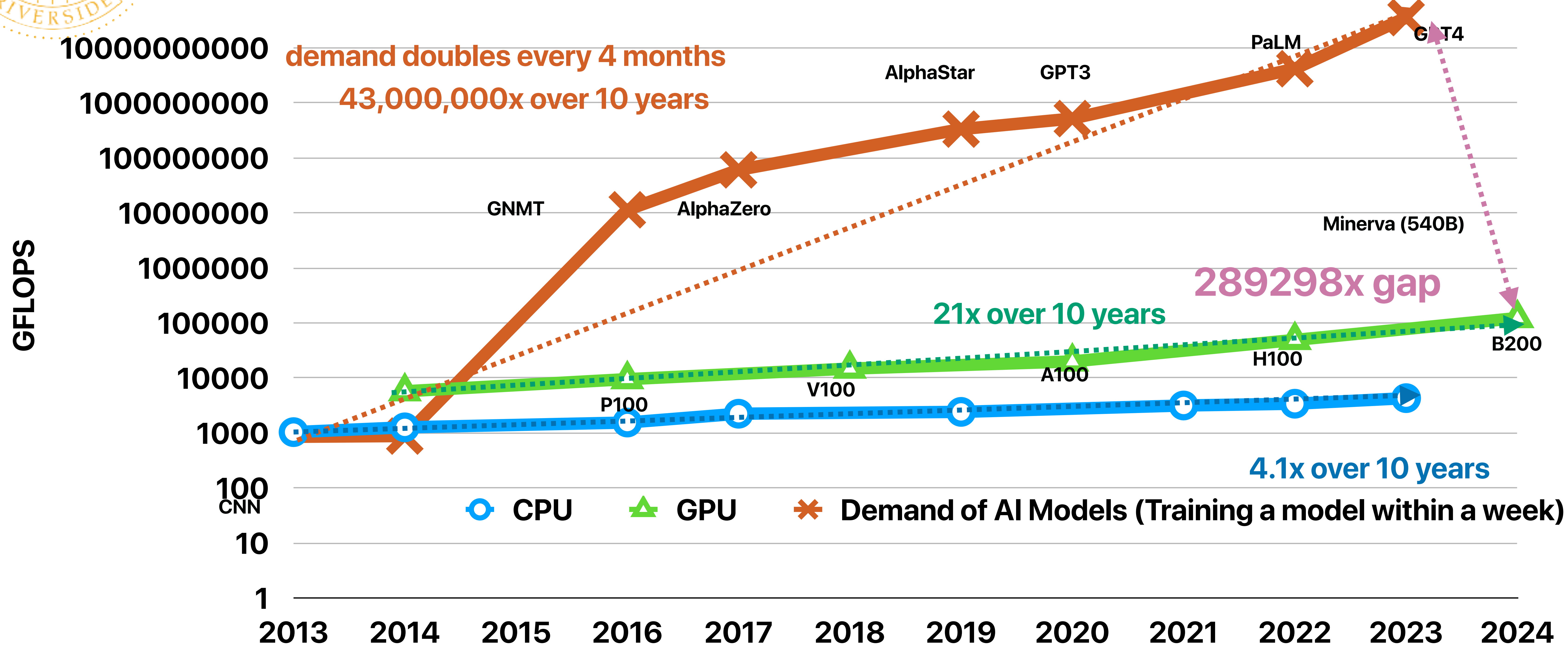
Is it an “efficient” computing model?

Why is traditional architecture "CPU-centric"



V. Krishnan and J. Torrellas, "A direct-execution framework for fast and accurate simulation of superscalar processors," Proceedings. 1998 International Conference on Parallel Architectures and Compilation Techniques

Mis-matching AI/ML demand and general-purpose processing



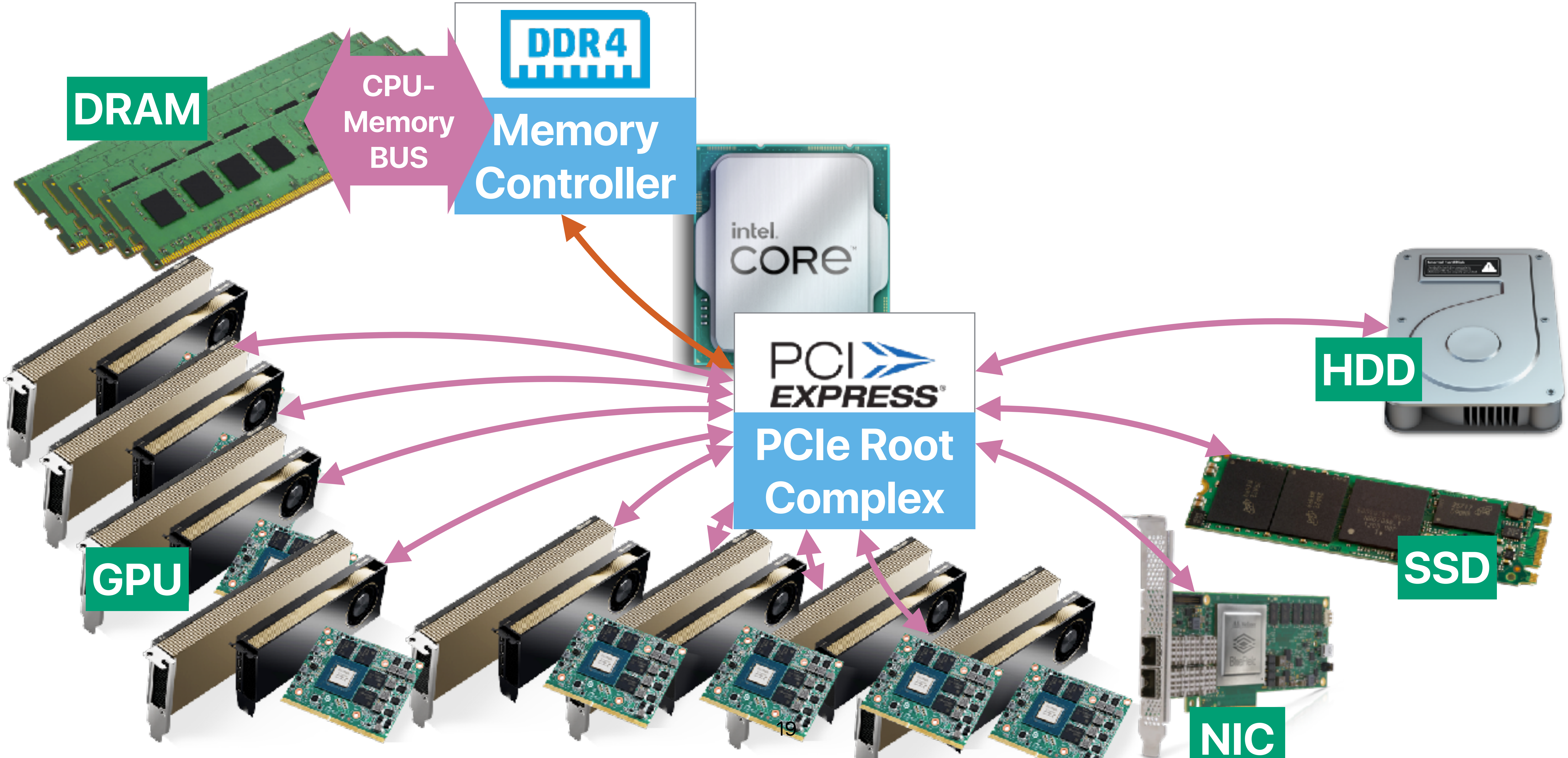
https://ourworldindata.org/grapher/artificial-intelligence-training-computation

Can we make the computer more "compute-centric"

3:00

How about a multi-GPU server? Is it efficient?

Can we support parallelism efficiently?



The capability of CPU's PCIe root complex

Essentials — Export spe

Product Collection	5th Generation Intel® Xeon® Scalable Processors
Code Name	Products formerly SAPPHIRE RAPIDS
Vertical Segment	Server
Processor Number	4510
Lithography	Intel 7
Use Conditions	Server/Enterprise

Expansion Options

Scalability	2S
PCI Express Revision	5
Max # of PCI Express Lanes	80

Essentials — Download Specific

Product Collection	13th Generation Intel® Core™ i9 Processors
Code Name	Products formerly Raptor Lake
Vertical Segment	Desktop
Processor Number	i9-13900K
Lithography	Intel 7
Use Conditions	PC/Client/Tablet, Workstation

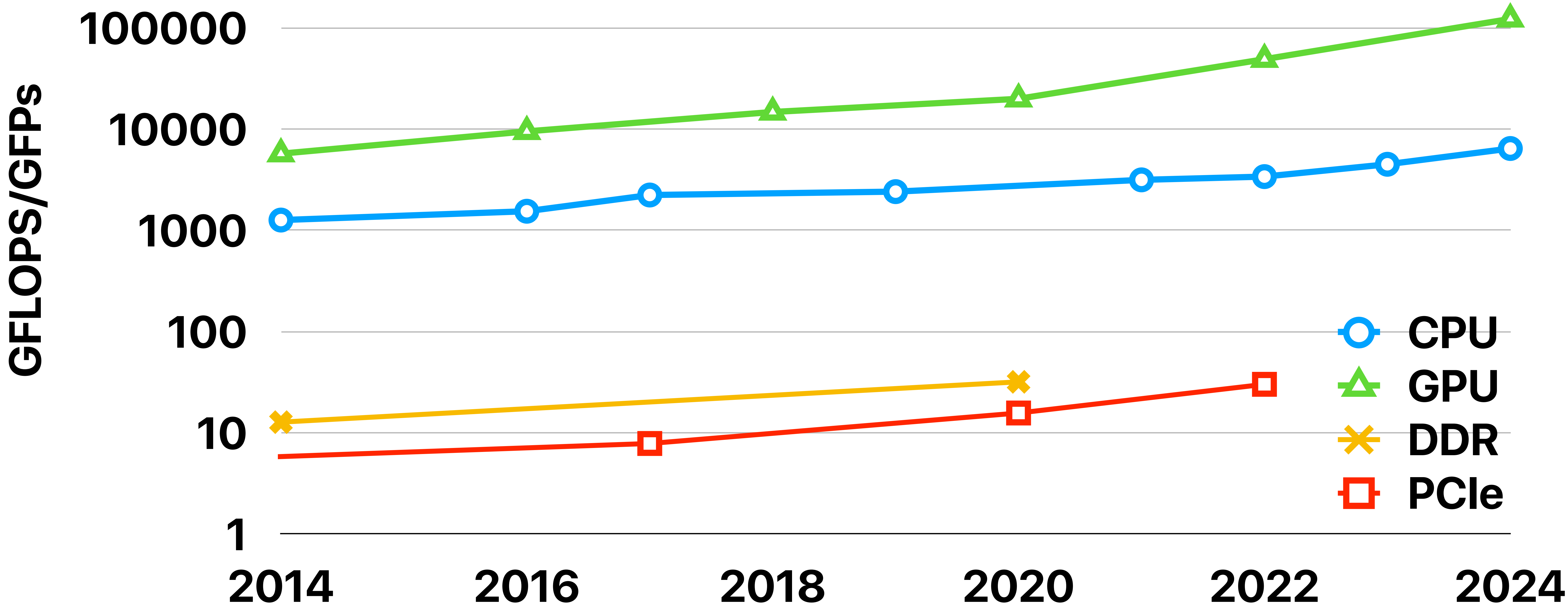
Expansion Options

Direct Media Interface (DMI) Revision	4.0
Max # of DMI Lanes	8
Scalability	1S Only
PCI Express Revision	5.0 and 4.0
PCI Express Configurations [‡]	Up to 1x16+4, 2x8+4
Max # of PCI Express Lanes	20

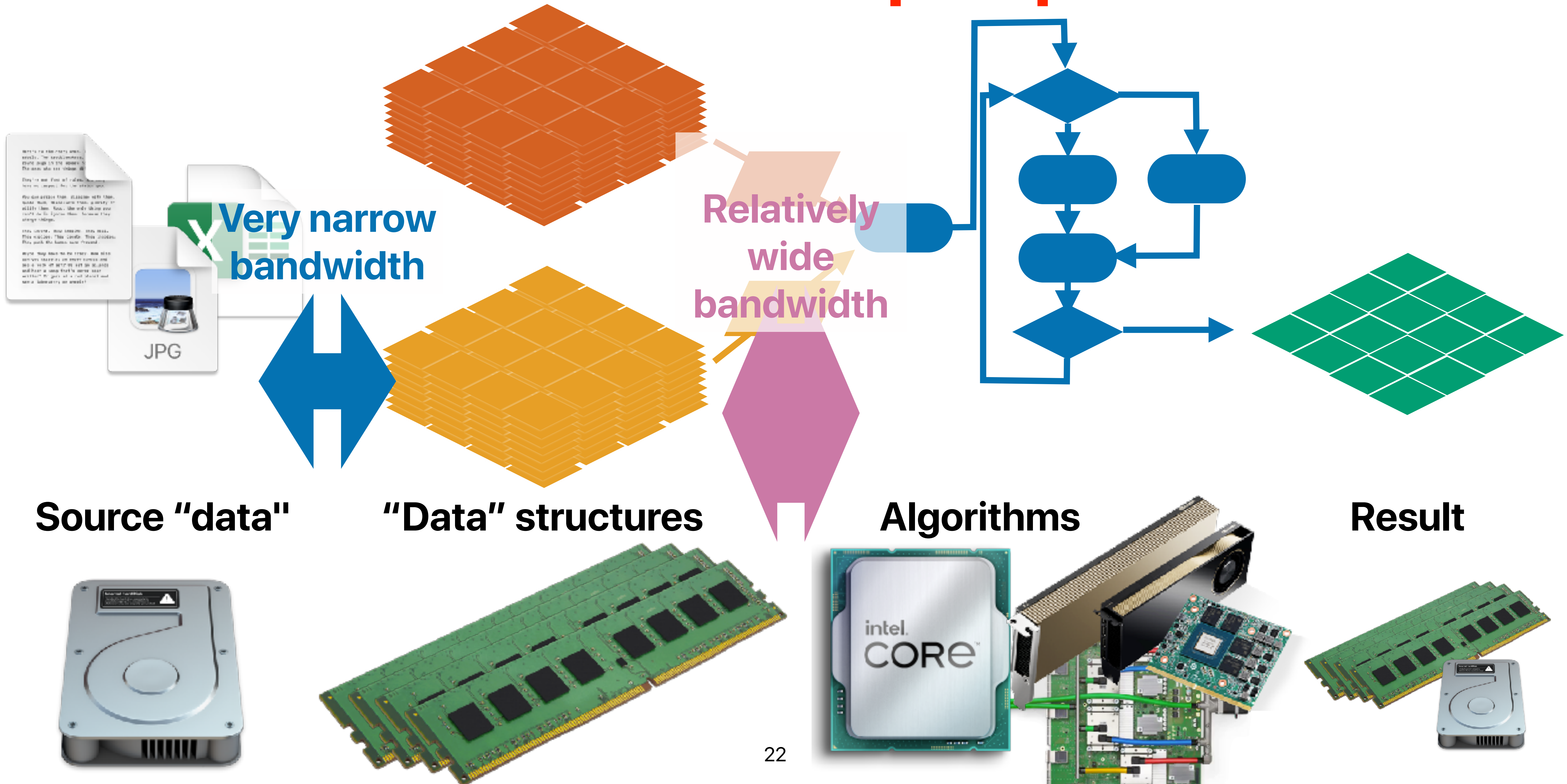
AMD Ryzen™ 7 7700X

Native SATA Ports:	0
PCI Express® Version:	PCIe® 5.0
Native PCIe® Lanes (Total/Usable):	28 / 24

The “speed” of interconnects compared to computing



From the software perspective



3:00

What will you do differently if the application spends more time interacting with data/storage/memory?

Ideas?


大改造!!
劇的
ビフォーアフター
SEASON II
That'S a DRAMATIC CHANGE!
毎週日曜ごご4:00~5:00放送
BEFORE

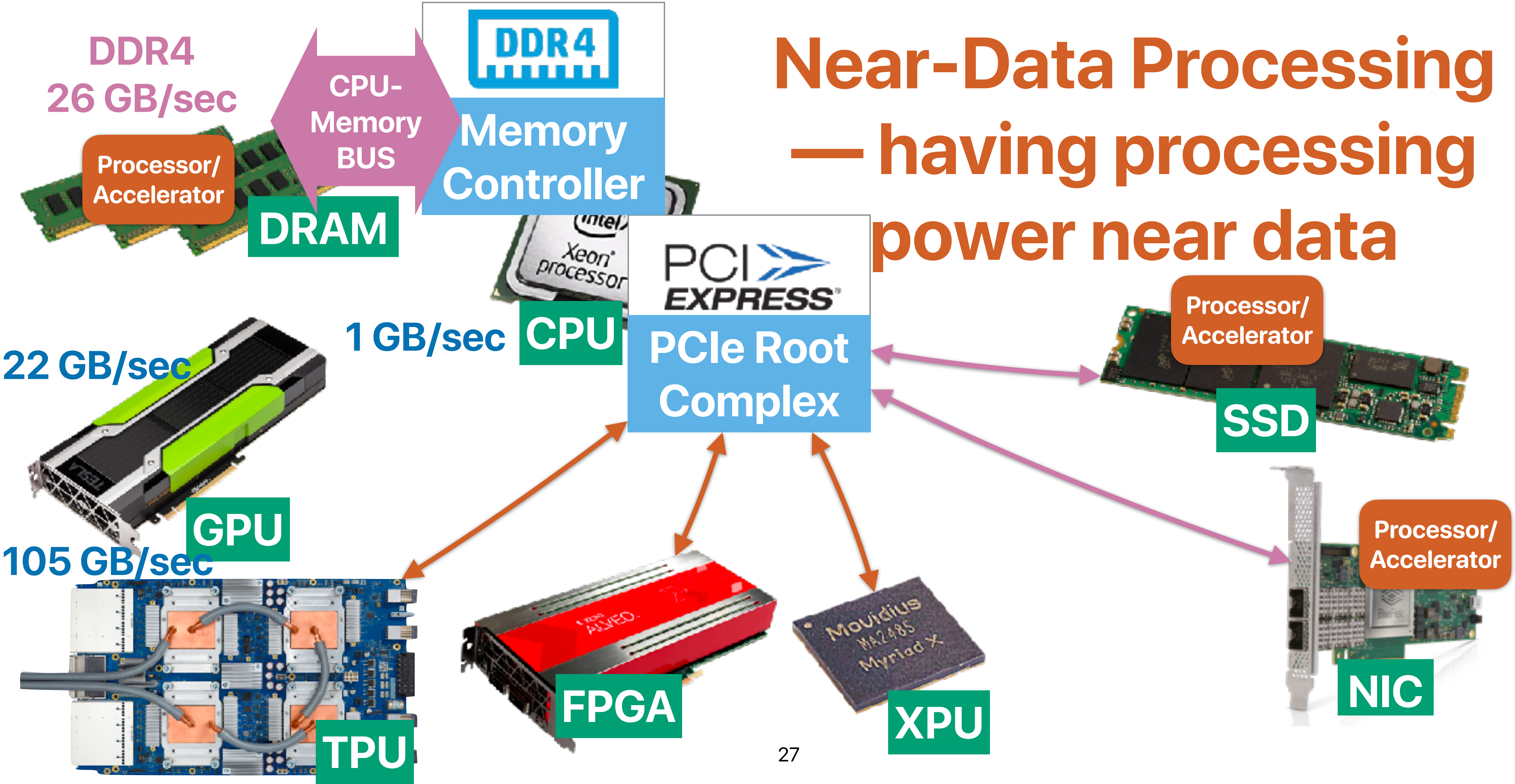

FORE
BEFORE
AFTER


Living room centric


Kitchen-centric

What if ...

Near-Data Processing — having processing power near data



Do I really fully utilize the hardware? How to make more efficient use of them?

Should every computer be data-centric?

It depends

But how do I know what should do when it's "it depends"?

That’s the “why” of EE/CS277!

Technical perspectives

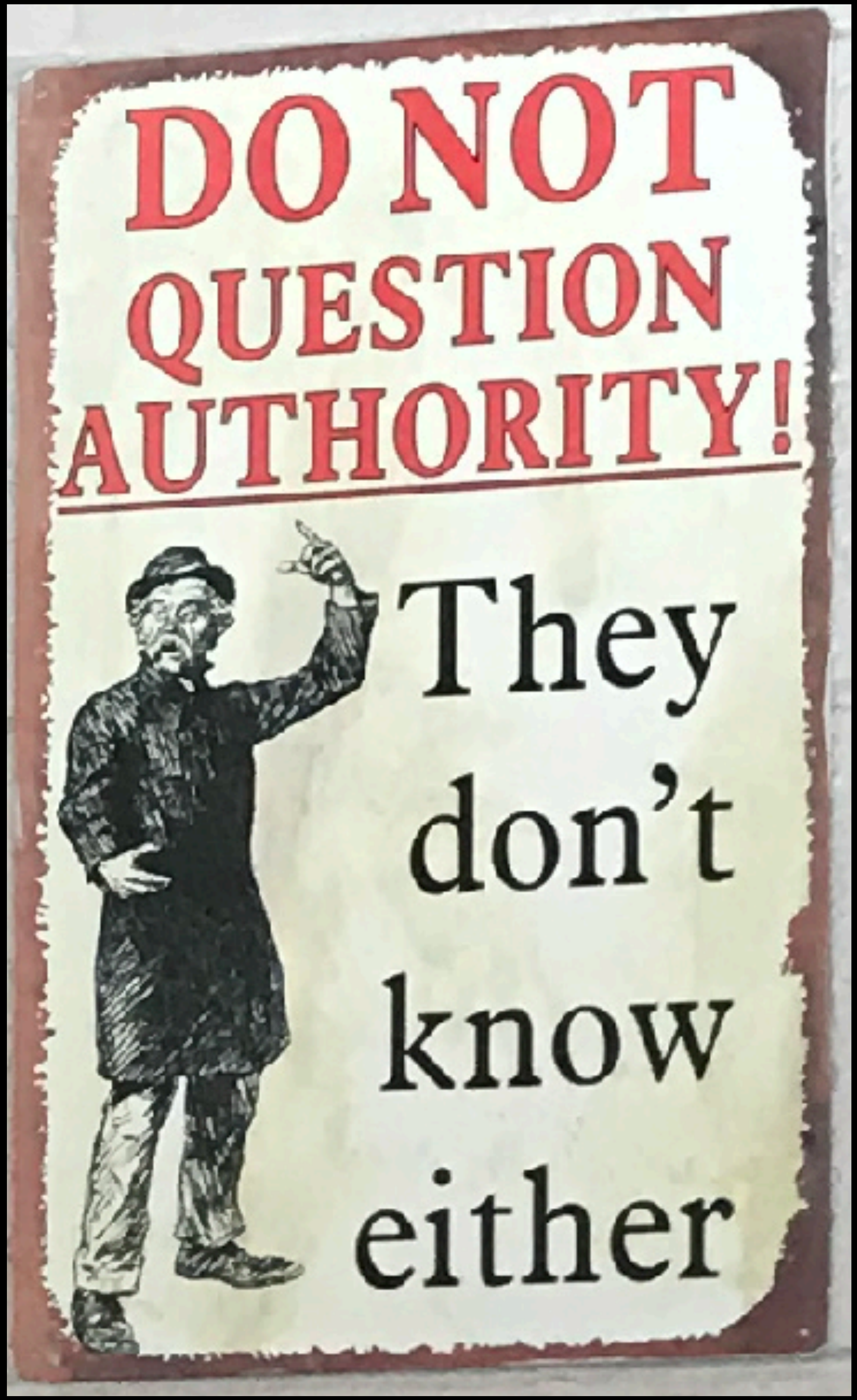
- Performance analysis
  - Evaluate the maximum potential of applications/algorithms using the roofline model
  - Design the "right"/most efficient architecture for your applications
  - Maximize the utilization of your system resources
- Emerging computer architectures
  - Heterogeneous computer architectures
  - Hardware accelerators
  - Near-data processing
- Performance programming
  - New programming models (e.g., TF Lite!)
  - New programming platforms (e.g., smartSSDs)
- Efficient data storage
  - Data representations
  - Storage system architectures

Soft skills

- Identifying “big” problems in our community
- Reading papers and pick up new ideas efficiently
- Coming up with new research ideas
- Learning the way of drafting a solution to an idea
- Working in a group

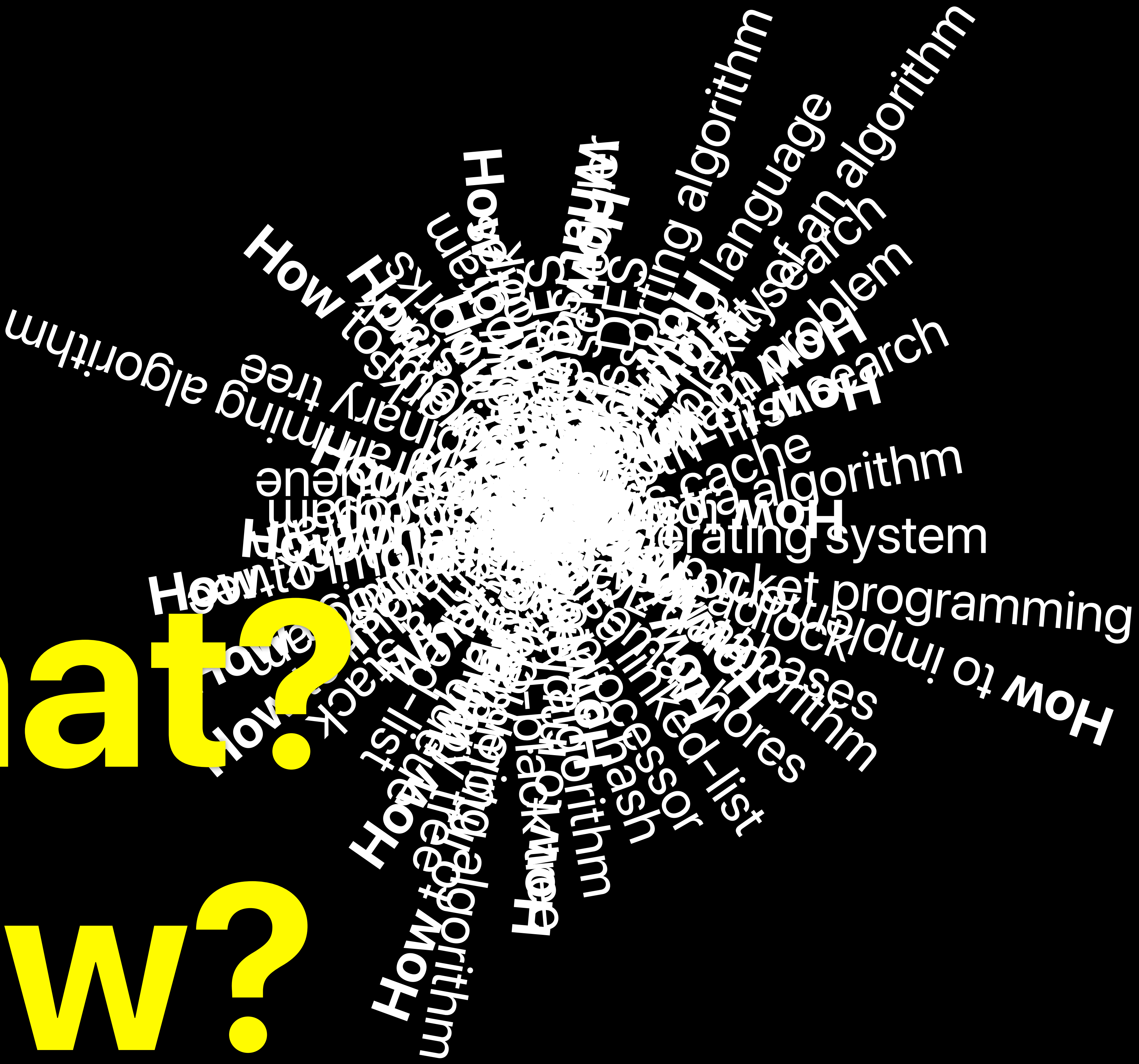
Learning eXperience

Why?
What?
How?



What?

How?



sorting algorithm
language
of an algorithm
search
problem
search
cache
algorithm
operating system
socket programming
How to implement
deadlock
databases
algorithm
semaphores
linked-list
processor
hash
algorithm
binary tree
algorithm

Lecture

Why?

What?

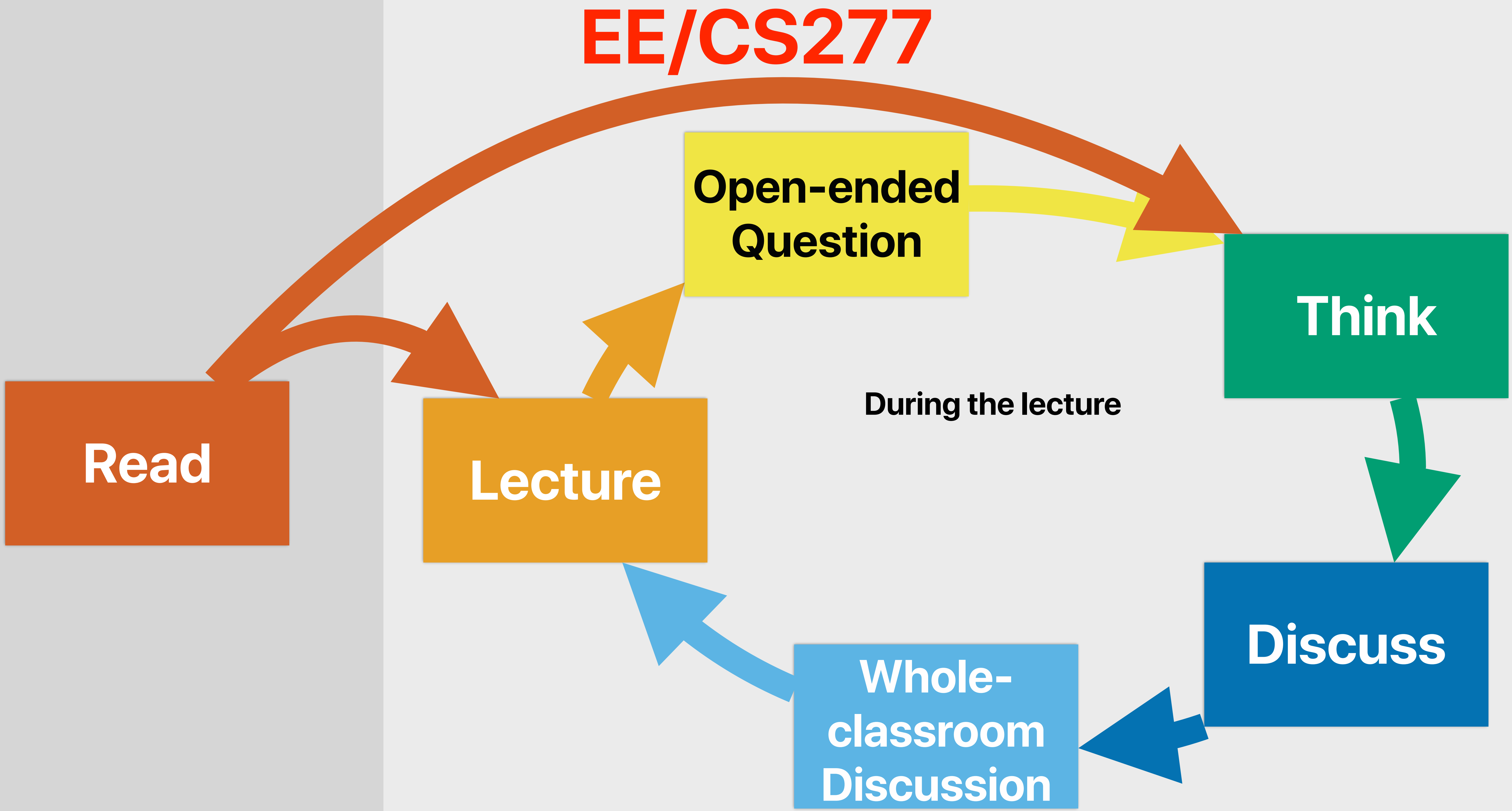
How?

Project

Traditional lectures

Peer Instructions

What kind of show is ours?


EE/CS277
Open-ended Question
Read
Lecture
Think
During the lecture
Discuss
Whole-classroom Discussion

Your tasks — 50%

- Read papers!
  - We will read papers reflecting the most trending data-centric computing
  - I will try my best to teach you how to read papers quickly
  - I will try my best to lead you come up with research ideas naturally
  - Submit paper reading summary before lectures
- Discuss
  - Participate in the discussion questions during the lectures

Why papers?

Papers give you insights!

- Papers contain **design principles** that are missing in your textbook or online documents
  - You can apply these design principles and the skills of analyzing these principles to anywhere
  - They provide case-studies to those "it depends" answers
- You can learn those **whys** for those proposed work
  - Submit your paper summary through Gradescope!

Industry cares

[redacted]@intel.com> 2011/2/15
寄給 h1tseng

Hi Hung-Wei,

I am very interested in your topic you presented yesterday. If possible, may I get a copy of

Best Regards,
[redacted]

[redacted]freescale.com 透過 cs.ucsd.edu 2012/1/10
寄給 h1tseng

Hung-Wei

I just finished reading your paper "Understanding the Impact of Power Loss on Flash Memory", very interesting information, do you have a PowerPoint presentation that goes along this paper?

[redacted]fb.com> 2019年7月9日 週二 下午2:12
寄給 Hung-Wei [redacted]

Hung-wei

[redacted]

Given we are also working on in-memory and near-memory computing at my Boston team, I would like to see how do we work more closely to churn out even more useful results and applications for Facebook's ML models/workloads in both datacenters and edge devices and instigate new research directions.

[redacted]

[redacted]sap.com 透過 cs.ucsd.edu 2012/11/12
寄給 h1tseng

Hi Tseng,

I have read your paper titled "Understanding the Impact of Power Loss on Flash Memory". It
work. I would like to understand what specific tools did you use to observe the page-read and
the FTL level. Did you use some sort of Flash simulator to get all the statistics about the num
and the energy consumption? My second question would be regarding FTL algorithms. Did y
real SSD or you used some kind of simulator and simulated the FTL algorithm?

Thanks.

[redacted]
SAP Research

[redacted]@huawei.com> 2016/6/24
寄給 h1tseng

Hi, Hung-Wei,

[redacted] from Huawei, and I am impressed by your ISCA 2016 presentation in Seoul. Near-data processing in ssds may be a promising solution for future data centers. Would you mind sending me your slides presented in the conference? I really appreciate your kindness. Thank you!

Best regards,
[redacted]

How to read research papers

How to read research papers

- For each paper, you should identify the followings:
  - Why? **The most important thing when you're reading/writing a paper**
    - Why should we care about this paper?
    - What's the problem that this paper is trying to address?
  - What? **The second most important thing when you're reading/writing a paper**
    - What has been proposed? and again, why the proposed idea makes sense
    - What's new about the paper: contributions of the paper and challenges addressed?
  - How? **They are important only if you want to implement the proposed idea**
    - How does the paper accomplish the proposed idea?
    - How does this paper verify and evaluate the effectiveness of the idea?
    - How does the result perform?

Let's read this paper together!

DOI:10.1145/3286588

Programmable software-defined solid-state drives can move computing functions closer to storage.

BY JAEYOUNG DO, SUDIPTA SENGUPTA, AND STEVEN SWANSON

Programmable Solid-State Storage in Future Cloud Datacenters

THERE IS A major disconnect today in cloud datacenters concerning the speed of innovation between application/operating system (OS) and storage infrastructures. Application/OS software is patched with new/improved functionality every few weeks at months and application-demanded requirements from the storage system grow quickly over time. This notable lag in the adaptability and velocity of movement of the storage infrastructure may ultimately affect the ability to innovate throughout the cloud world.

In this article, we advocate creating a software-defined storage substrate of solid-state drives (SSDs) that are as programmable, agile, and flexible as the applications/OS accessing from servers in cloud datacenters. A fully programmable storage substrate promises opportunities to better bridge the gap between application/OS needs and storage capabilities/limitations, while allowing application developers to innovate in-house at cloud speed.

The move toward software-defined control for IO devices and co-processors has played out before in the datacenter. Both GPUs and network interface cards (NICs) started as black-box devices that provide acceleration for CPU-intensive operations (such as graphics and packet processing). Internally, they implemented acceleration features with a combination of specialized hardware and proprietary firmware. As customers demanded greater flexibility, vendors slowly exposed programmability to the rest of the system, unleashing the vast processing power available from GPUs and a new level of agility in how sys-

https://dl.acm.org/doi/pdf/10.1145/3286588

Why?

- Why should we care about this paper?
- What's the problem that this paper is trying to address?

THERE IS A major disconnect today in cloud datacenters concerning the speed of innovation between application/operating system (OS) and storage infrastructures. Application/OS software is patched with new/improved functionality every few weeks at "cloud speed," while storage devices are off-limits for such sustained innovation during their hardware life cycle of three to five years in datacenters. Since

life cycle of three to five years in datacenters. Since
the software inside the storage device is written by storage vendors as proprietary firmware not open for general application developers to modify, the developers are stuck with a device whose functionality and capabilities are frozen in time, even as many of them are modifiable in software. A period of five years is

What?

- What has been proposed? and again, why the proposed idea makes sense
- What's new about the paper: contributions of the paper and challenges addressed

In this article, we advocate creating a software-defined storage substrate of solid-state drives (SSDs) that are as programmable, agile, and flexible as the applications/OS accessing from servers in cloud datacenters. A fully programmable storage substrate promises opportunities to better bridge the gap between application/OS needs and storage capabilities/limitations, while allowing application developers to innovate in-house at cloud speed.

Interestingly, SSDs generally have a far larger aggregate internal bandwidth than the bandwidth supported by host I/O interfaces (such as SAS and PCIe). Figure 2 outlines an example

In-Storage Programming

Modern SSDs combine processing—embedded processor—and storage components—SRAM, DRAM, and flash memory—to carry out routine functions required for managing the SSD. These computing resources present interesting opportunities to run general user-defined programs. In 2013, Do et

First, the computing capabilities available inside the SSD are limited by design. The low-performance embed-

Moreover, the embedded software-development process is complex and makes programming and debugging very challenging. To maximize perfor-

How?

- How does the paper accomplish the proposed idea?
- How does this paper verify and evaluate the effectiveness of the idea?
- How does the result perform?

Figure 6. Preliminary results using a programmable SSD yield approximately 5x speedups for full scans of ZLIB-compressed ORC files within the device, compared to native ORC readers running on x86 architecture.

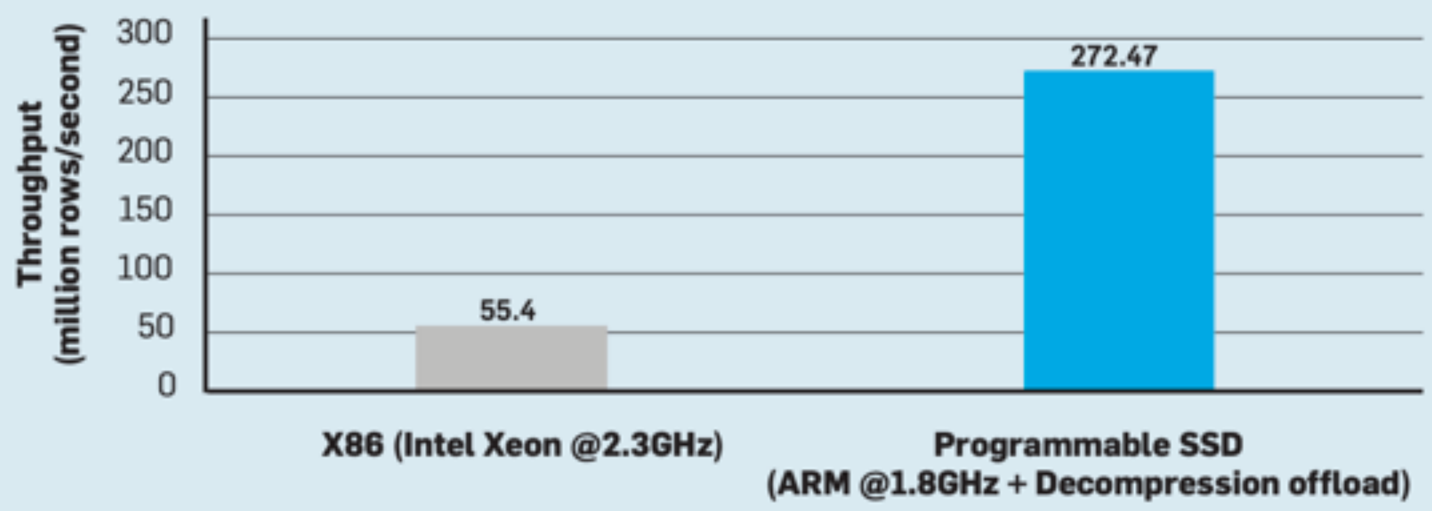


Figure 5. A prototype programmable SSD developed for research purposes.

(a)
Device with a storage board with an embedded storage controller and DIMM slots for flash or other forms of NVM

(b)
Device with a storage board where M.2 SSDs can be plugged into.

Your tasks (cont.) — 50%

- Project ideas
  - Accelerating applications through AI/ML accelerators (e.g., EdgeTPUs or tensor cores)
  - Accelerating applications through innovative parallel programming models that hardware accelerators enable and evaluate the result using FPGAs
  - Static tools to perform roofline analysis on AI/ML workloads
  - Accelerating applications through intelligent storage devices (smartSSD)
  - Literature review (more than 20 papers) on a certain topic related to this class
  - Anything related to what we discussed in this class!
- Decide & discuss your topics of interests with me before the 3rd week
- Working on escalab.org/datahub
  - Containerized environment with TF, TF Lite, CUDA
  - GPUs, Edge TPUs, smartSSD available
- Presentations
  - One on the 5th week to present your project ideas
  - The other on the 9th/10th week to present your current progress/outcome

Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: https://intra.engr.ucr.edu/~htseng/
- E-mail:
- Visiting Researcher @ Google, 01/2023—03/2023
  - Working for TensorFlow Lite
- PhD in **Computer Science**, University of California, San Diego, 2014
- Research Interests
  - General-purpose computing on AI/ML/NN accelerators
  - Intelligent storage devices & near-data processing
  - Or anything else fun — we have an OpenUVR project recently
- Fun fact: Hung-Wei was once considering a career path as a singer but went back to academia due to the unsuccessful trial

UC RIVERSIDE

Google

TensorFlow

UC San Diego

Course materials

- Course webpage
  - https://www.escalab.org/classes/eecs277-2026wi
- Discussion
  - Google Space — you should be invited
- Recording
  - Prof Usagi's Youtube Channel — https://www.youtube.com/profusagi

Let’s act today

- Form your group!
  - At most 3
  - Discuss your research interests
  - Having our first milestone meeting between you and I next week
- Check out our Google Spaces
- Read two papers listed on the website and submit your paper summary through Gradescope!

Electrical
Computer Science
Engineering

277

つづく