computer systems performance
lecture 6 – hardware acceleration

Pınar Tözün

March 11, 2025

agenda

lecture – part 1

- general-purpose vs specialized hardware
  - pros/cons
  - CPUs, GPUs, FPGAs, ASICs
- switch to more hardware specialization
- today’s landscape for specialized hardware

lecture – part 2 & exercises

- GPUs & CUDA

hardware acceleration

wording courtesy of Wikipedia

"… is the use of computer hardware specially made to perform some functions more efficiently than is possible in software running on a general-purpose CPU."

systems stack overview

application

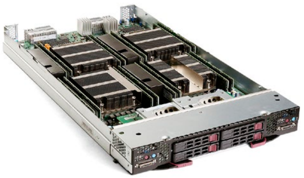
e.g., online shopping page, database system, code to read/write a file, etc.

operating system

e.g., linux, windows, etc.

hardware

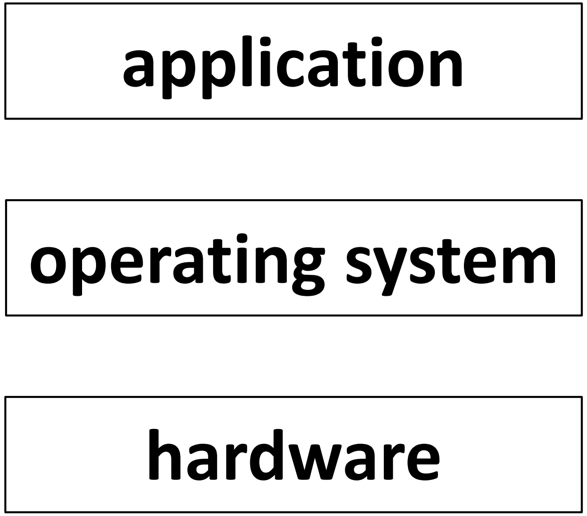
e.g., intel server, disks, etc.



systems stack overview

hardware acceleration …

- need to write code for diverse hardware
- need for efficient data movement across hardware devices



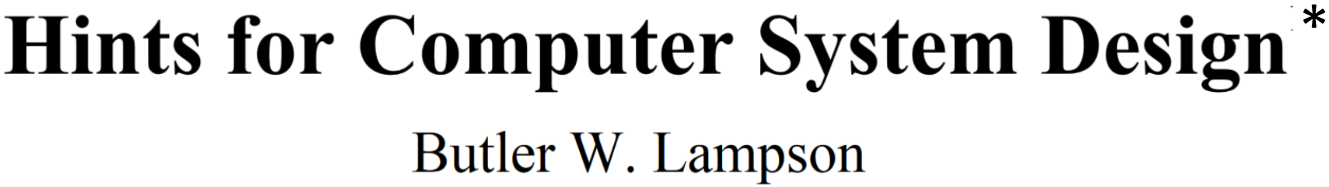
application

operating system

hardware

- more common to side-step OS when dealing with non-general-purpose hardware and directly manage it
- though, an active research topic

- is part of hardware



Hints for Computer System Design*

Butler W. Lampson

“Do one thing at a time, and do it well. An interface should capture the minimum essentials of an abstraction. ***Don’t generalize; generalizations are generally wrong.”***

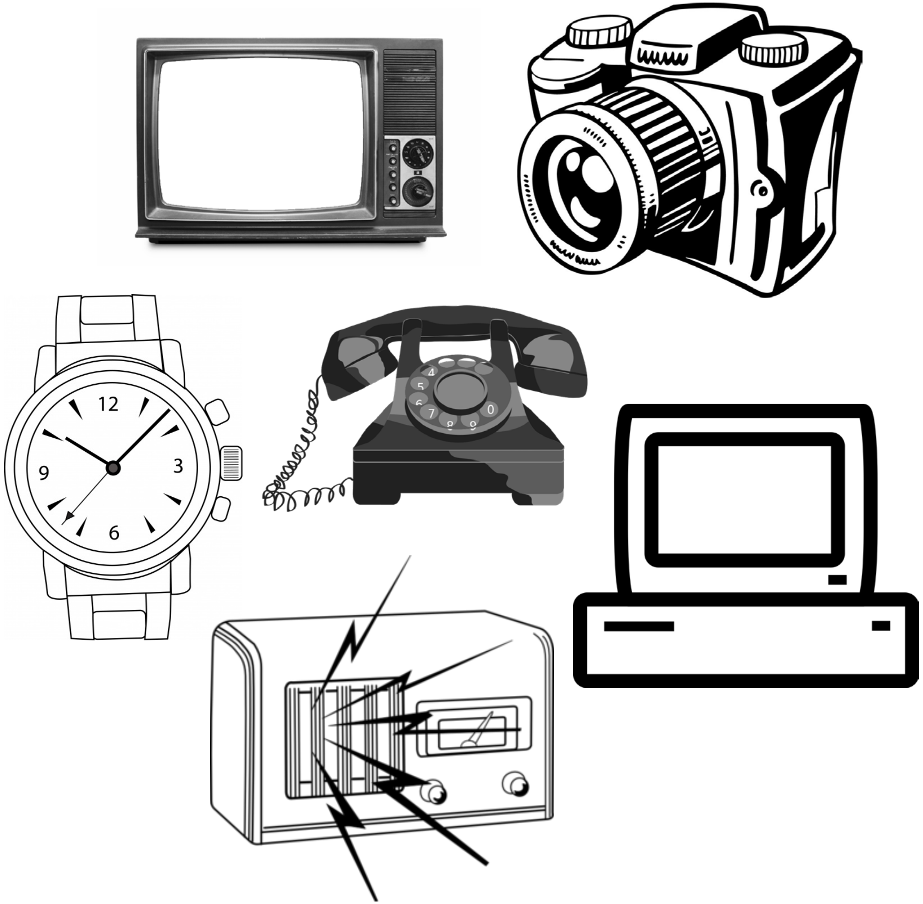
case for specialized / no one-size

*https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/acrobat-17.pdf

case for general-purpose / one-size fits all

analogy with cell phones

one system to rule them all



specialized systems for all

agenda

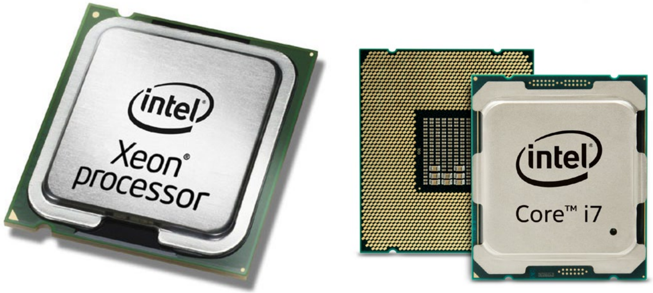
lecture – part 1

- general-purpose vs specialized hardware
    - pros/cons
    - CPUs, GPUs, FPGAs, ASICs
- switch to more hardware specialization
- today’s landscape for specialized hardware

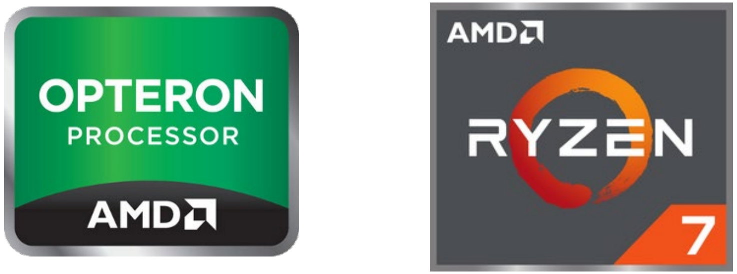
general-purpose – CPU

central processing unit

CISC (complex instruction set computing) & x86 family

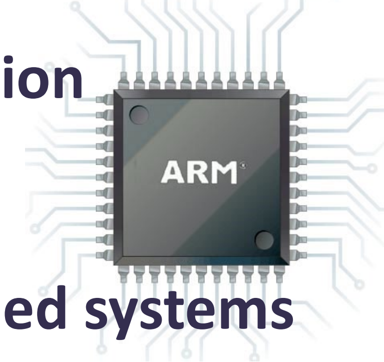


dominates desktops, laptops, servers

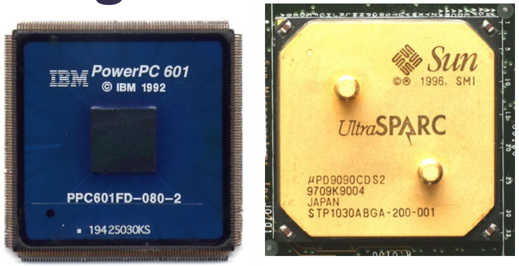


other big competitor in x86

RISC (reduced instruction set computing)



- lighter core designs
- dominates embedded systems
- today, also competes in the server market
  - AWS gravitons



SPARC & POWER targets server market mainly, which is dominated by Intel Xeons

specialized – GPU

graphics processing unit

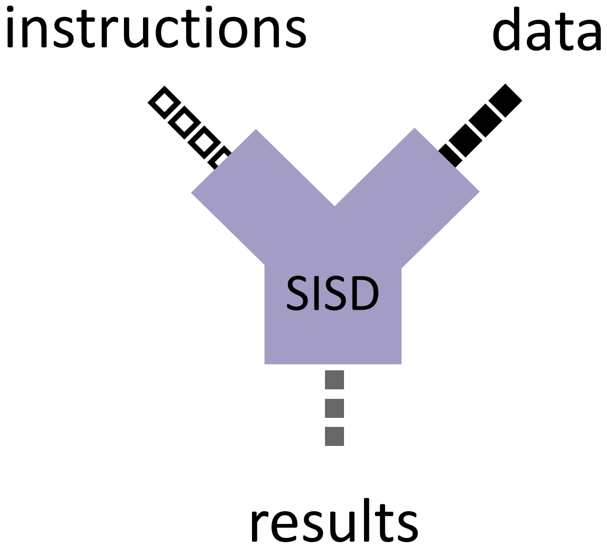
- designed to accelerate operations for computer graphics (e.g., rendering images)
  - used across embedded systems, mobile phones, personal computers, game consoles
  - based on SIMT (single instruction multiple thread)

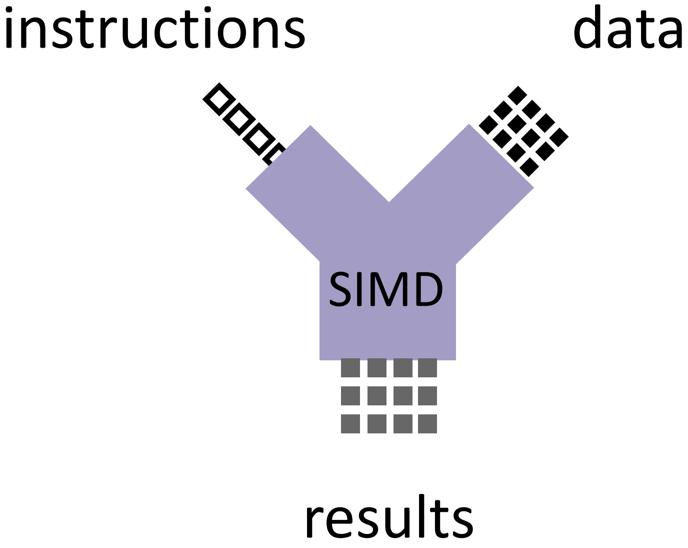
- general-purpose GPUs (GPGPUs)
  - GPUs that are used to perform operations traditionally performed by CPUs (e.g., sorting data)

- NVIDIA dominates server market followed by AMD
- your personal devices have a form of integrated GPUs (e.g., to accelerate graphics or AI in personal computers)

single instruction multiple data (SIMD)

from lecture 3

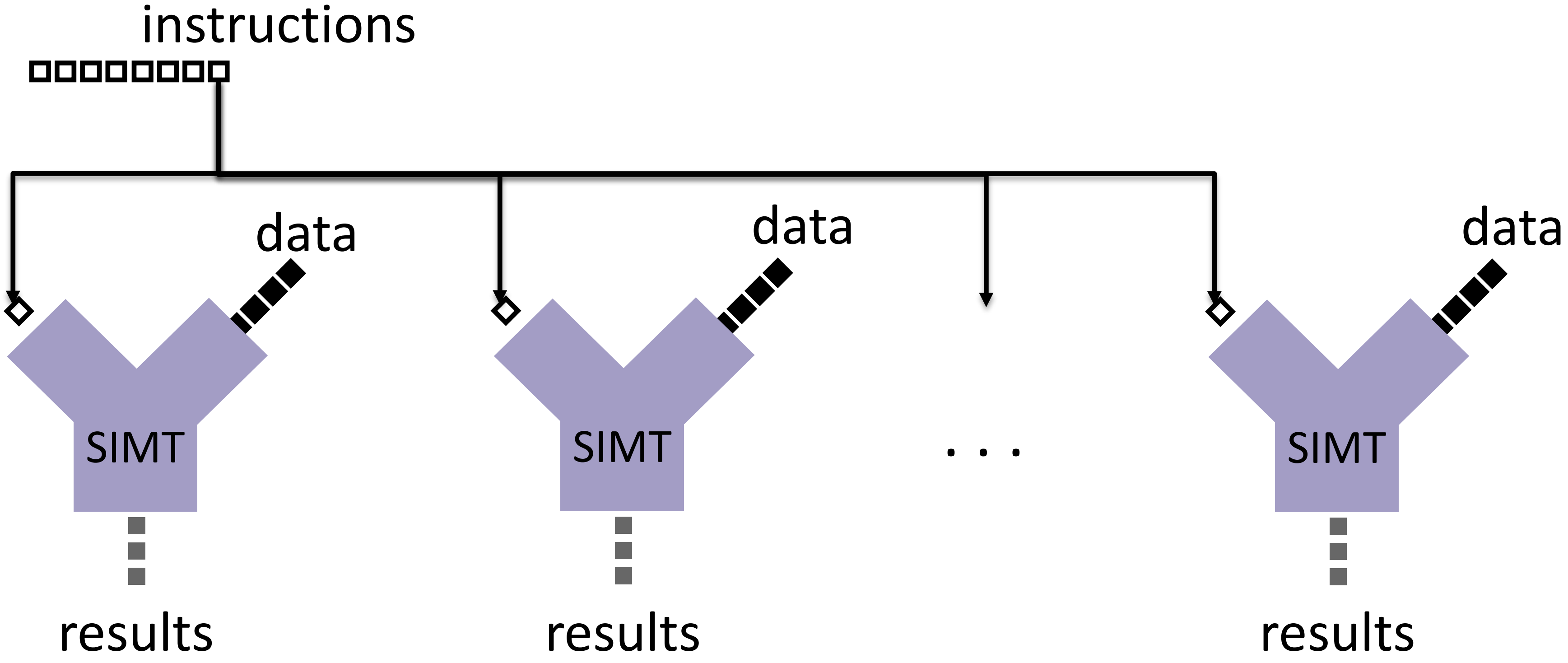




GPUs are like SIMD machines

they support extreme parallelism

single instruction multiple thread (SIMT)



GPUs are based on SIMT

GPU execution model

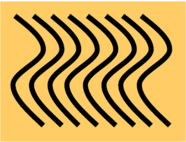
content from Jeff Larkin's slides

Software **Hardware**

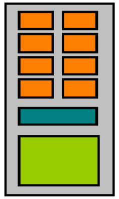
Thread

Scalar Processor

Threads are executed by scalar processors



Thread Block

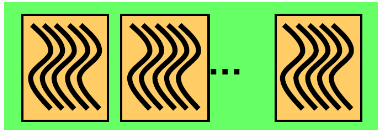


Multiprocessor

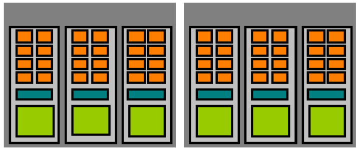
Thread blocks are executed on multiprocessors

Thread blocks do not migrate

Several concurrent thread blocks can reside on one multiprocessor - limited by multiprocessor resources (shared memory and register file)



Grid



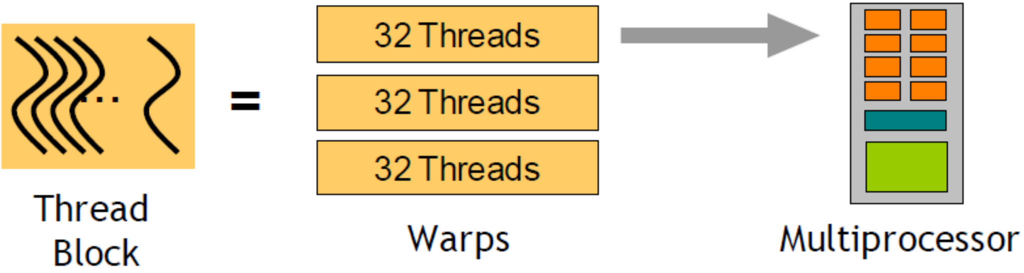
Device

A kernel is launched as a grid of thread blocks

function to execute

GPU execution model

content from Jeff Larkin's slides



A thread block consists of 32-thread warps

A warp is executed physically in parallel (SIMT) on a multiprocessor

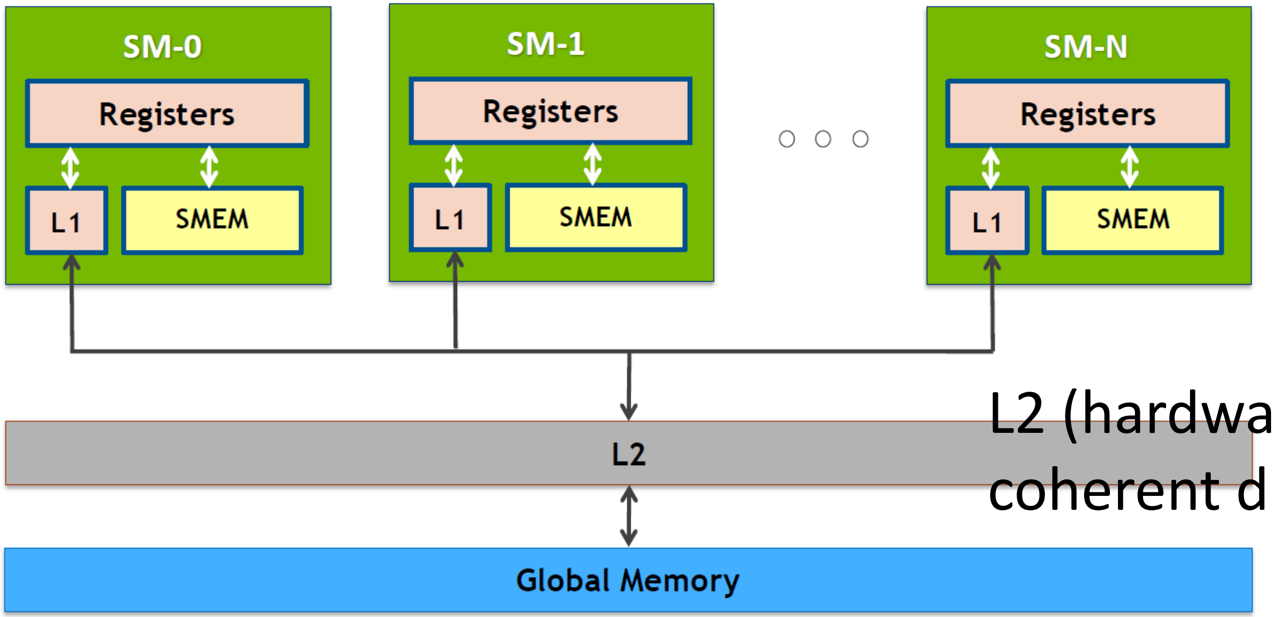
- **memory access latency is overlapped by execution of different warps**
- **SIMT doesn't require data to be in contiguous memory like SIMD**

GPU memory hierarchy

content from Jeff Larkin’s slides

L1 (hardware managed) is used for things like register spilling

SMEM (user-managed) scratch-pad memory

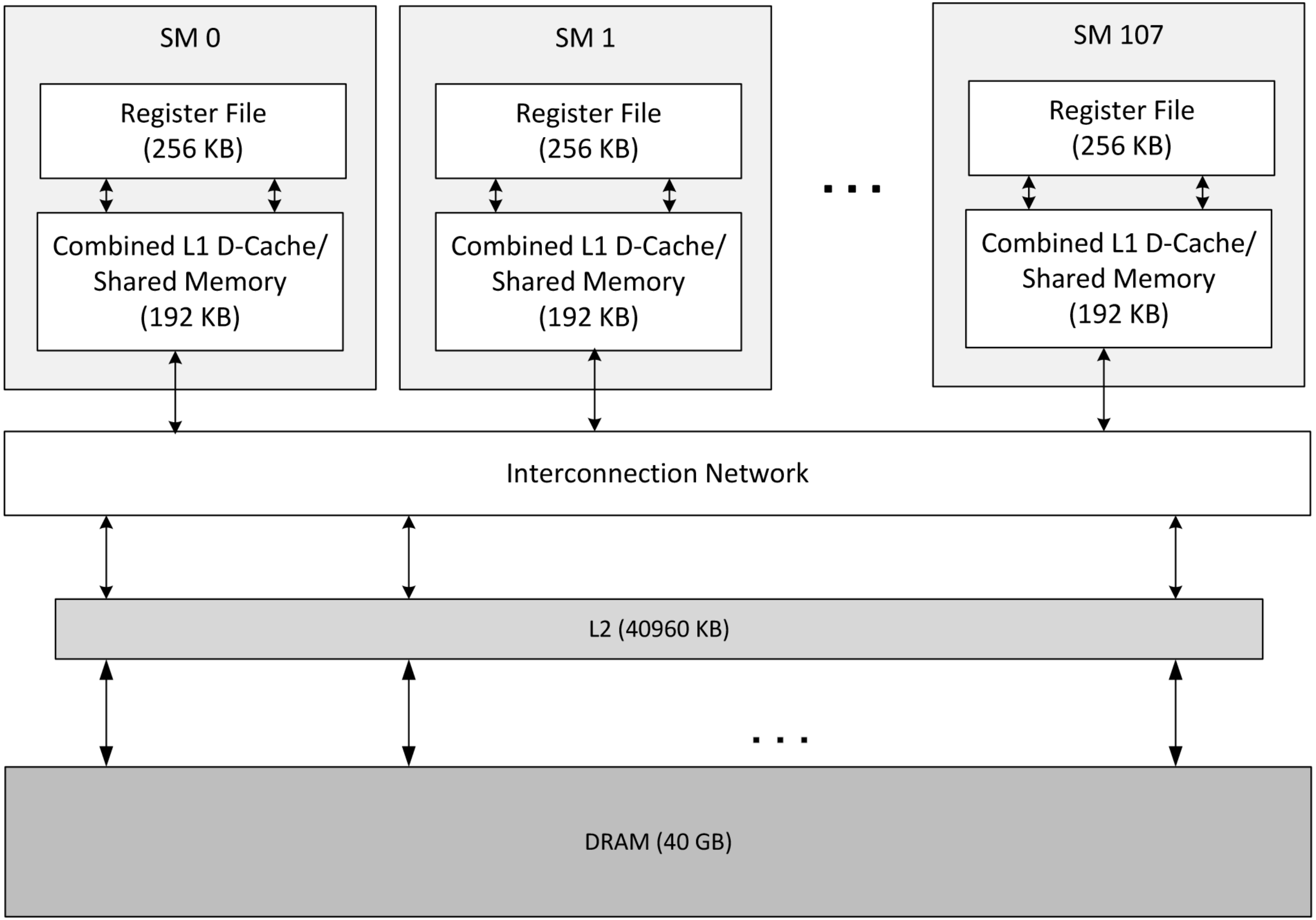


SM = streaming multiprocessor

L2 (hardware managed) coherent data sharing across

global memory handles communication with hosts (e.g., CPU in a CPU-GPU co-processor)

example: A100



max threads per SM = 2048

CPUs vs GPUs – what are they good for?

CPU

- latency-oriented tasks
  - even though within CPU domain, we have throughput- vs latency-oriented designs
- if you need single-core performance
- general-purpose computing

GPU

- throughput-oriented & embarrassingly parallel tasks
  - graphics
  - matrix multiplications (deep learning)
  - large sort operations

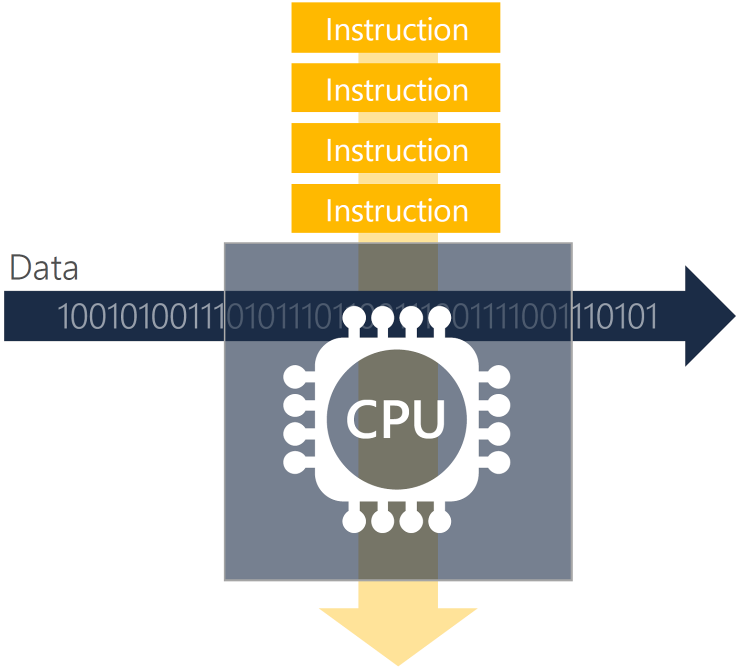
specialized – FPGA

field-programmable gate array

- not specialized by default,
  but (re-)programmable
- telecommunication and networking were primary application domains in the beginning
- today part of many data centers
  (from networking layer to processing layer)
- improved a lot over time in terms of efficiency (speed and energy) compared to ASICs
- Altera & Xilinx (now AMD’s) are market leaders

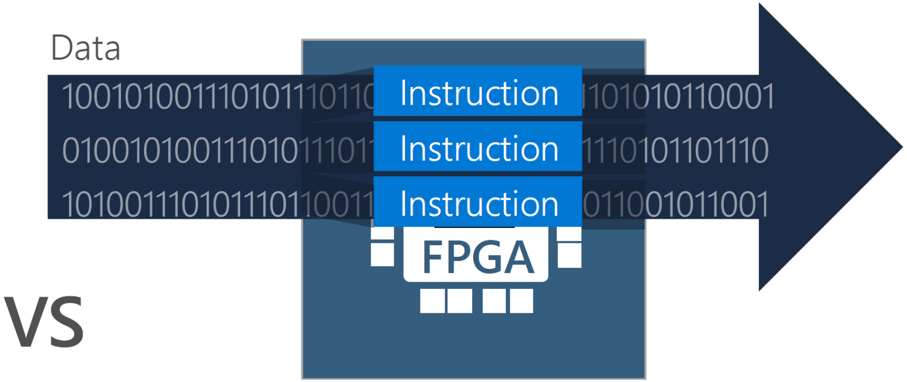
specialized – FPGA

figure from Doug Burger's HTPS 2020 talk



Temporal Computation

VS



Spatial Computation

specialized – ASIC

application-specific integrated circuit

- chip customized for a specific use
(e.g., crypto-mining, voice recording)

- programmed by a hardware description language (HDL)
such as Verilog / VHDL

- very fast & energy-efficient

- requires a very large volume to be cost-effective
otherwise, use an FPGA instead

- FPGAs can also be used as a platform to test hardware
like ASICs before production

existential questions

what is specialized?
what is general-purpose?

- GPUs are specialized hardware for computer graphics, but can also be used for other tasks (e.g., machine learning), so more flexible than an ASIC

- SUN SPARC, IBM Power processors are general-purpose, but designed with database applications in mind

- specialized evolution of the general-purpose CPU [CIDR15]
  - floating-point arithmetic
  - SIMD (single instruction multiple data)
  - hardware transactional memory
  - Intel Software Guard Extensions (SGX)

certain specialized features may find their place in general-purpose hardware eventually!

agenda

lecture – part 1

- general-purpose vs specialized hardware
  - pros/cons
  - CPUs, GPUs, FPGAs, ASICs
- switch to more hardware specialization
- today’s landscape for specialized hardware

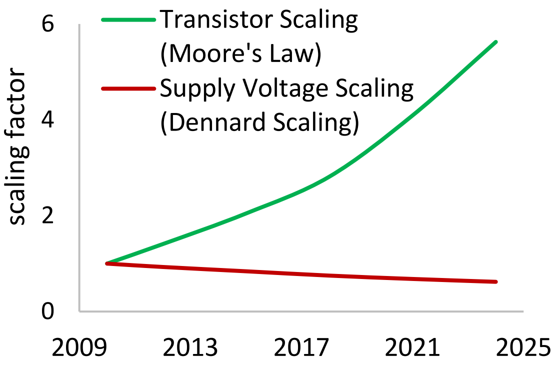
data management community back in 2014

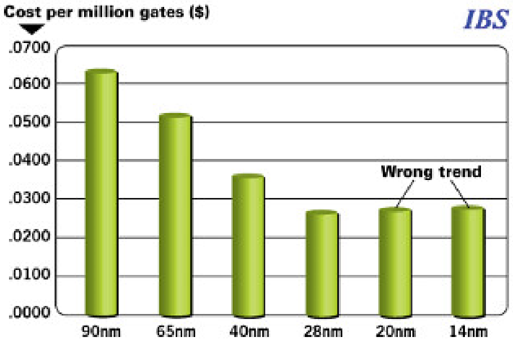
Michael Stonebraker

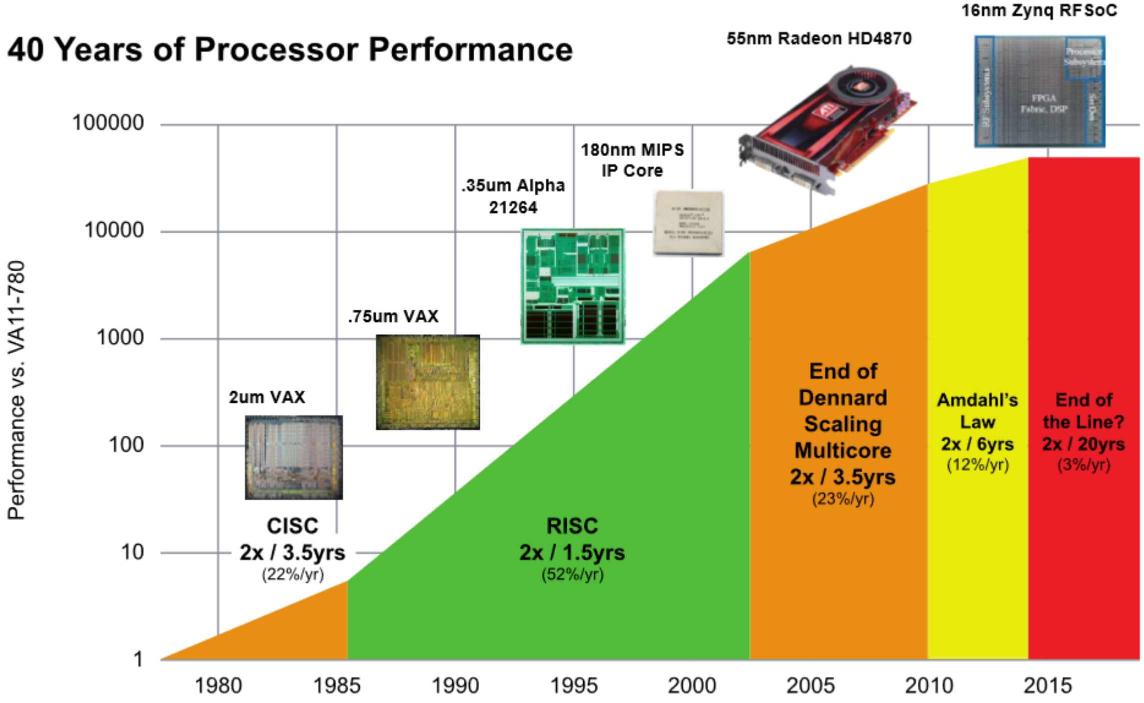
if you think, Intel will ever produce hardware for you, you are smoking something

what changed today? – for hardware

- general-purpose multicores doesn’t scale anymore







Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e 2018

what changed today? – for hardware

- general-purpose multicores doesn’t scale anymore
  - heat dissipation concerns – Dennard scaling doesn’t hold
  - too small transistors – expensive to get right

  if we want forward progress in hardware,
  need to change the way hardware specialization is viewed

- FPGAs got way better in terms of efficiency,
  manufacturing ASICs aren’t as necessary for specialization

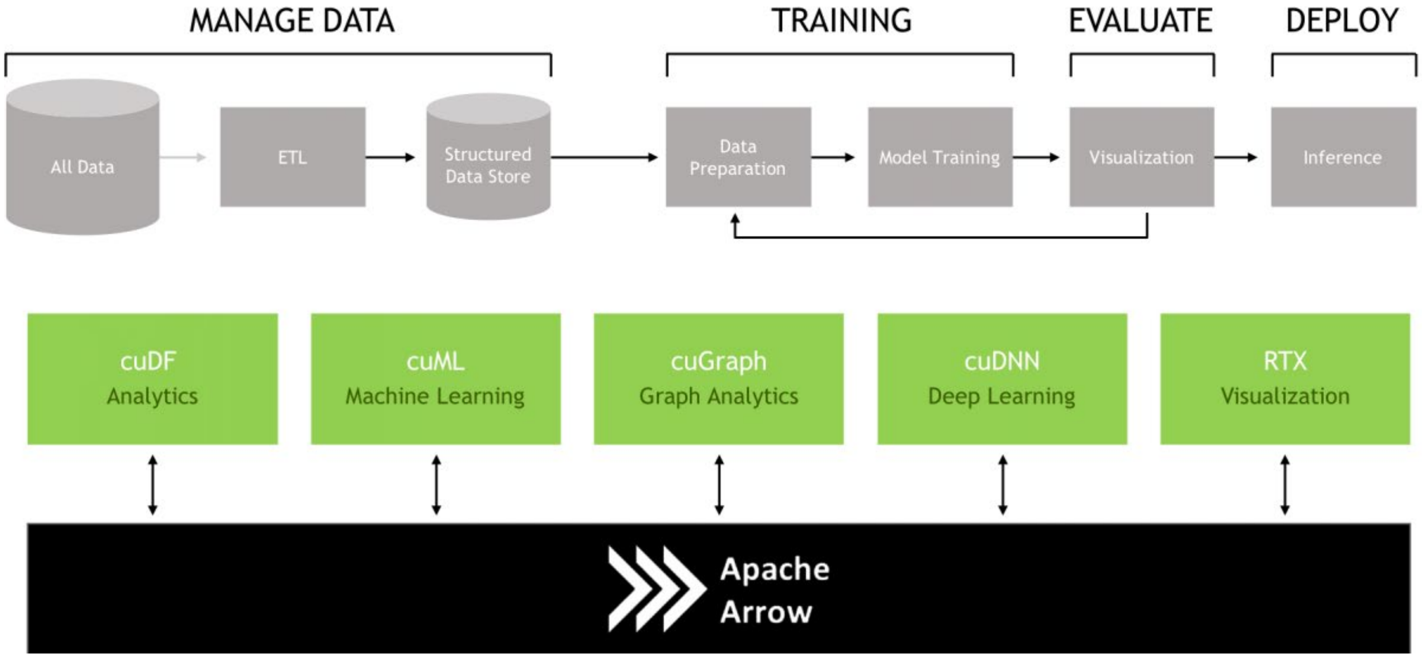
what changed today? – for software

- huge demand for AI (machine learning, deep learning, …)
  - which benefit from hardware specialization (GPUs, TPUs)

- scale of data-intensive applications in the cloud / in IoT makes hardware specialization more economically viable

- rise of python
  or to be more generic: high-level tools that make running code on specialized hardware like GPUs/FPGAs easier

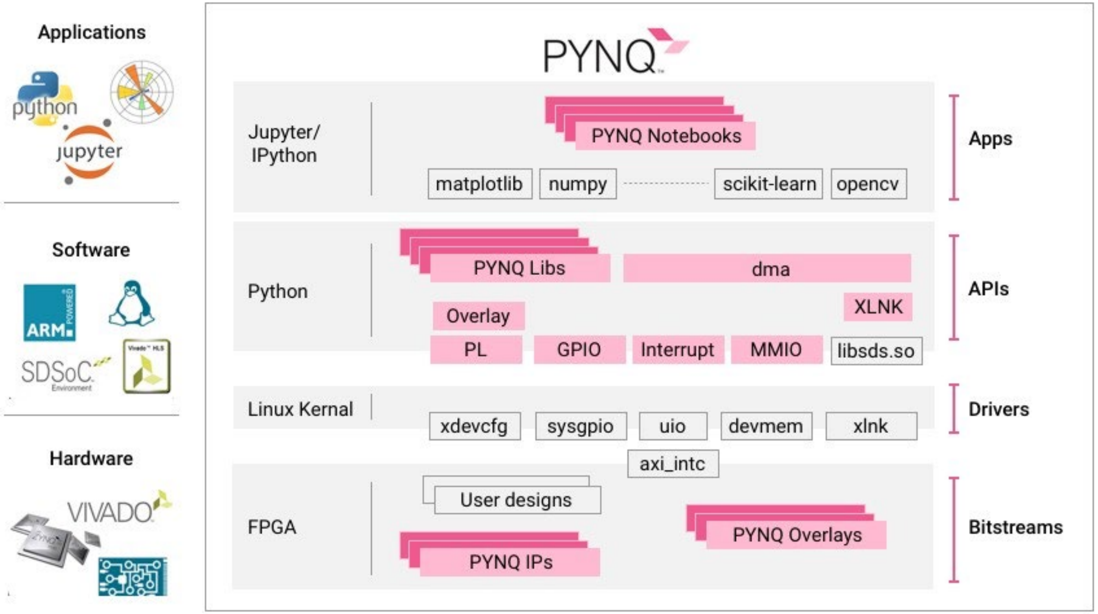
what changed today? – for software

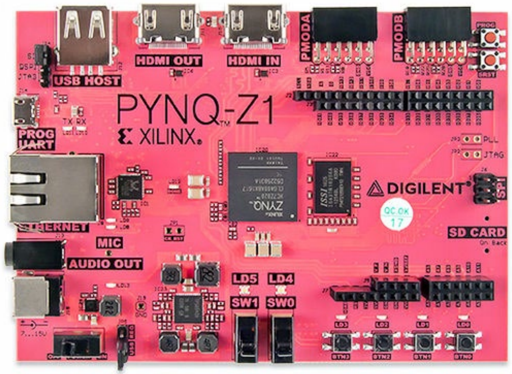


rapids.ai from NVIDIA open-source libraries to run data science pipelines on GPUs

- rise of python
  or to be more generic: high-level tools that make running code on specialized hardware like GPUs/FPGAs easier

what changed today? – for software





pynq.io from Xilinx

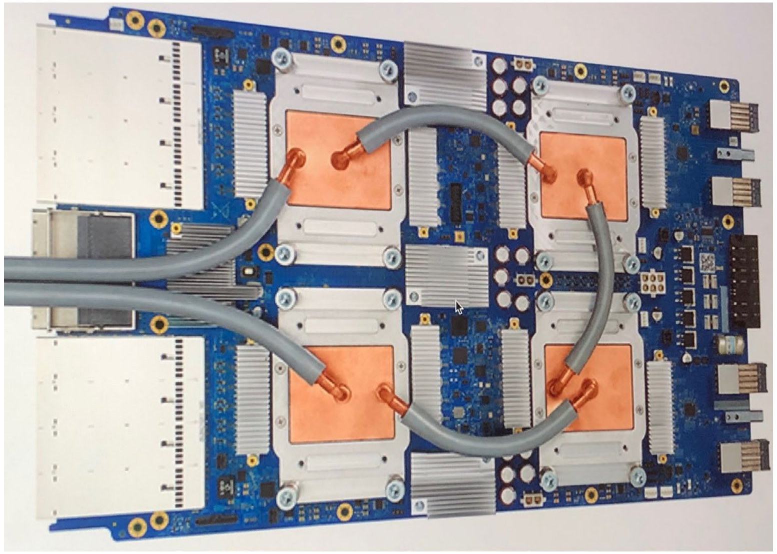
- rise of python
  or to be more generic: high-level tools that make running code on specialized hardware like GPUs/FPGAs easier

agenda

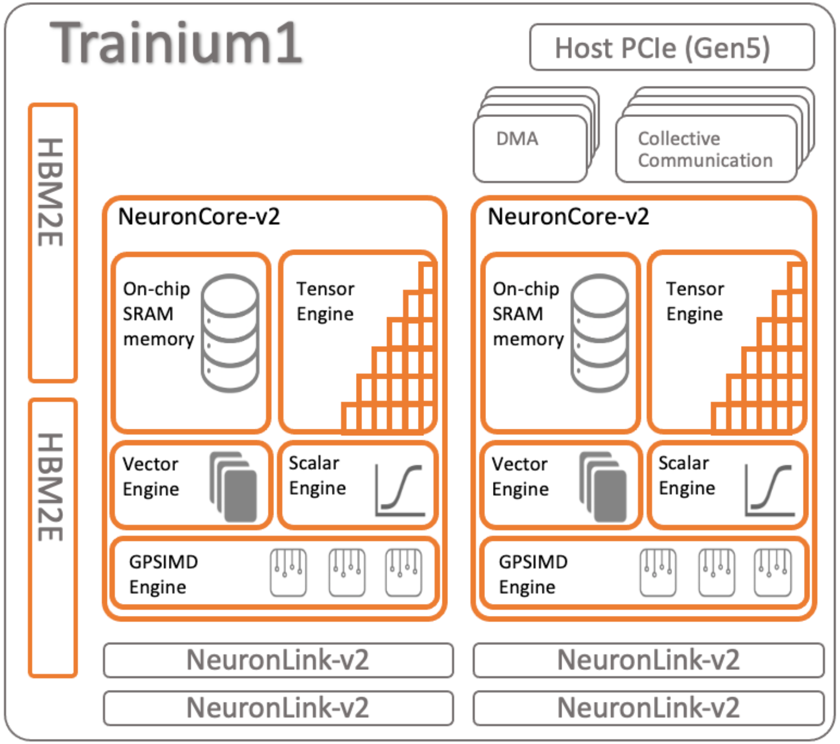
lecture – part 1

- general-purpose vs specialized hardware
  - pros/cons
  - CPUs, GPUs, FPGAs, ASICs
- switch to more hardware specialization
- today’s landscape for specialized hardware

deep learning accelerators in the cloud

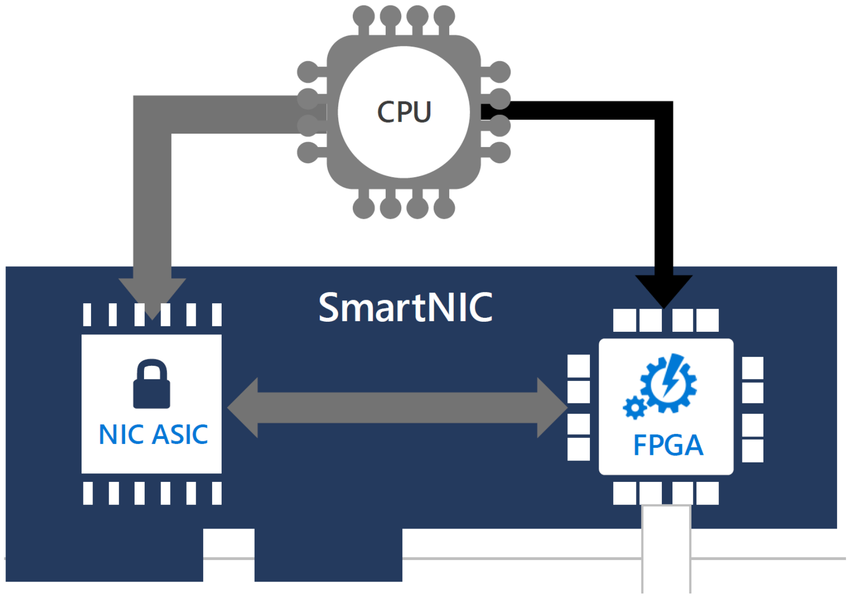


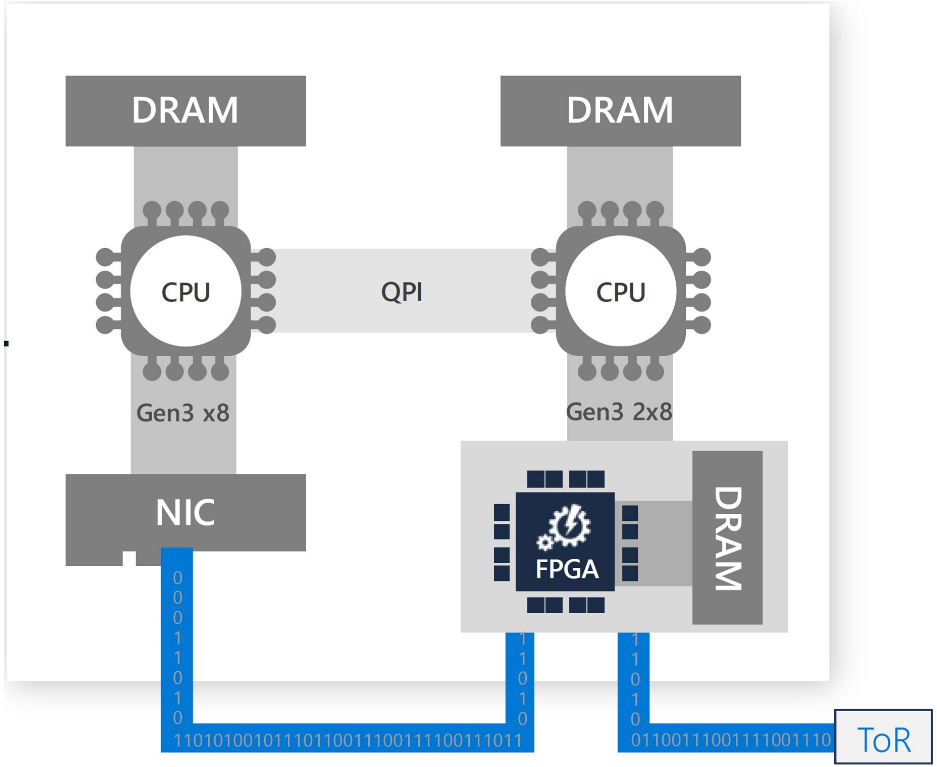
Google TPU
(Tensor Processing Unit)



AWS Trainium & Inferentia

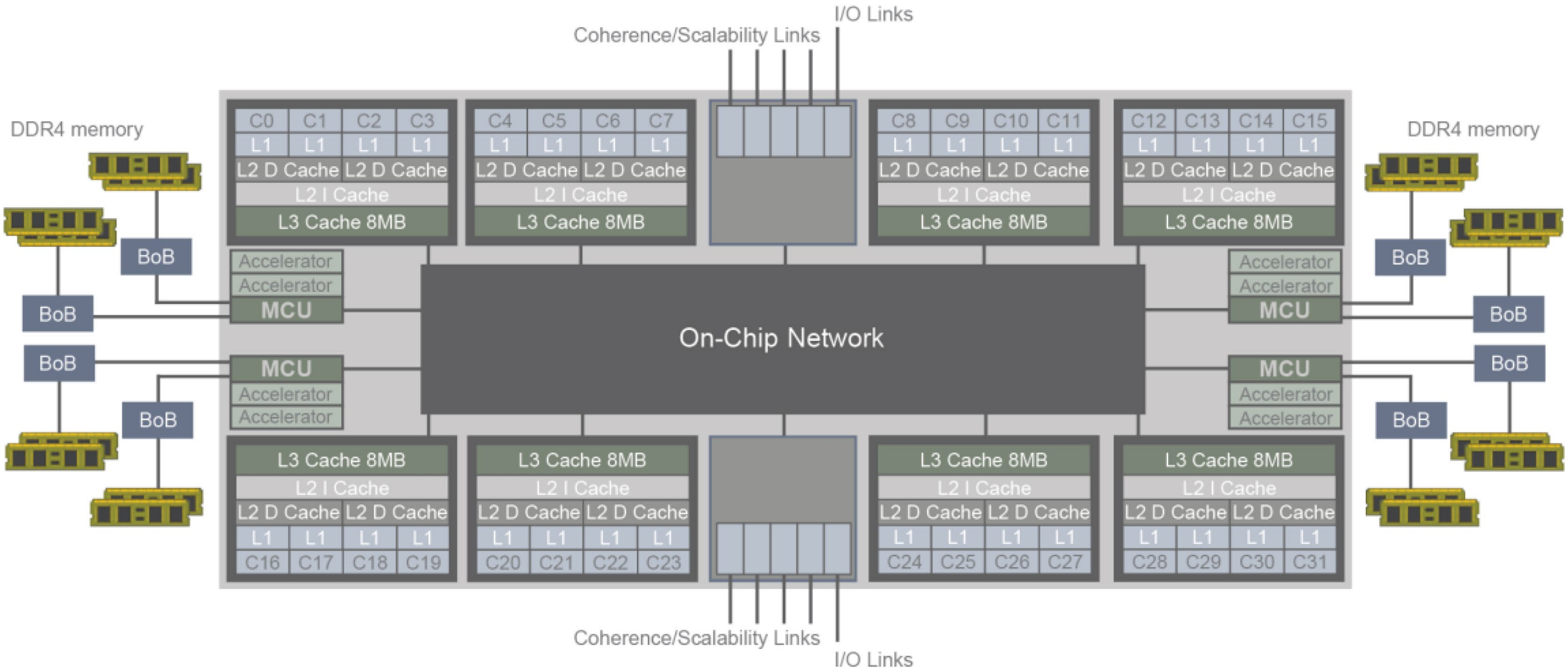
Microsoft Catapult/Brainwave @ Azure





- CPU-FPGA co-processors or in-network processing to accelerate data processing operations
  - e.g., crypto, filtering data, bing search, AI
- rolling out hardware updates just like software ones

Oracle DAX (Data Analytics Accelerator)



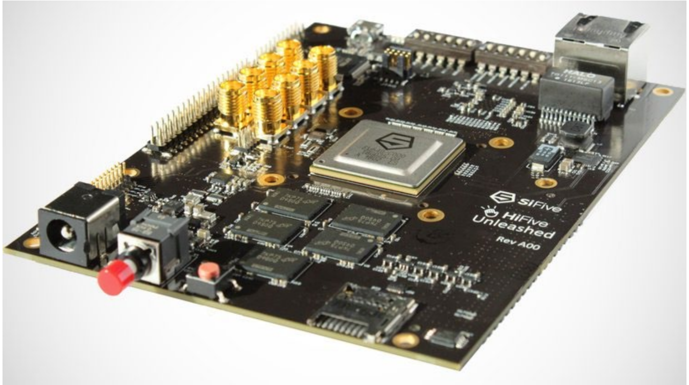
- part of SPARC M7 processors, can be found in Oracle cloud
- in-memory data processing (e.g., compression, filtering)
- technology developed part of the cancelled RAPID project

RISC-V & Agile Hardware Development

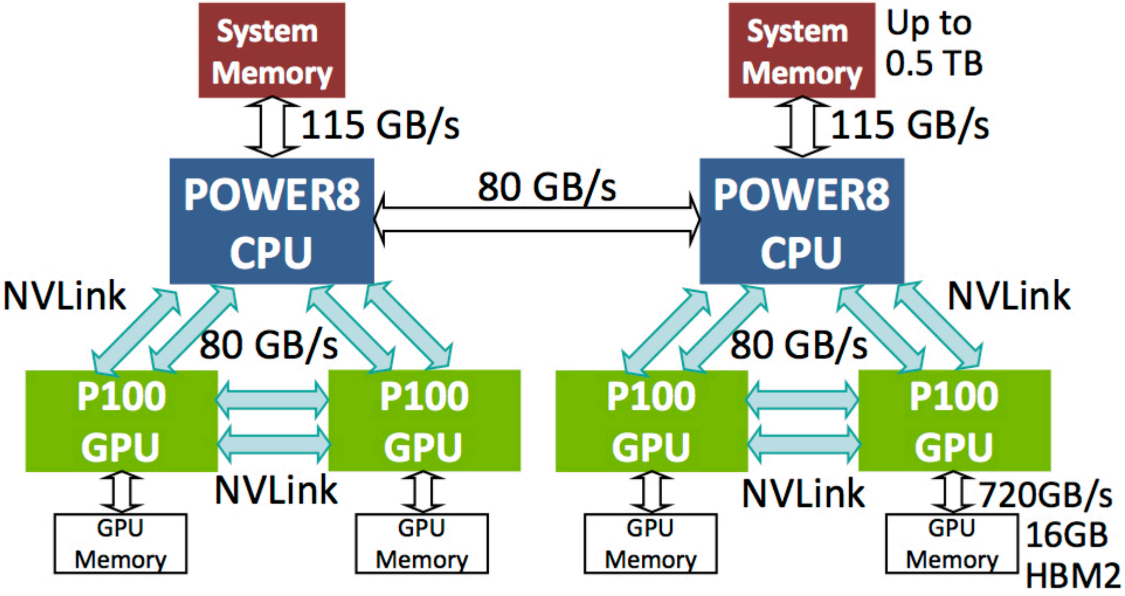
motivation:

- hardware acceleration became inevitable
- hardware may need software-like update cycles
- why not have open-source hardware instruction sets
- why not have "linux for processors"

- main development in Verilog
  but ecosystem is improving



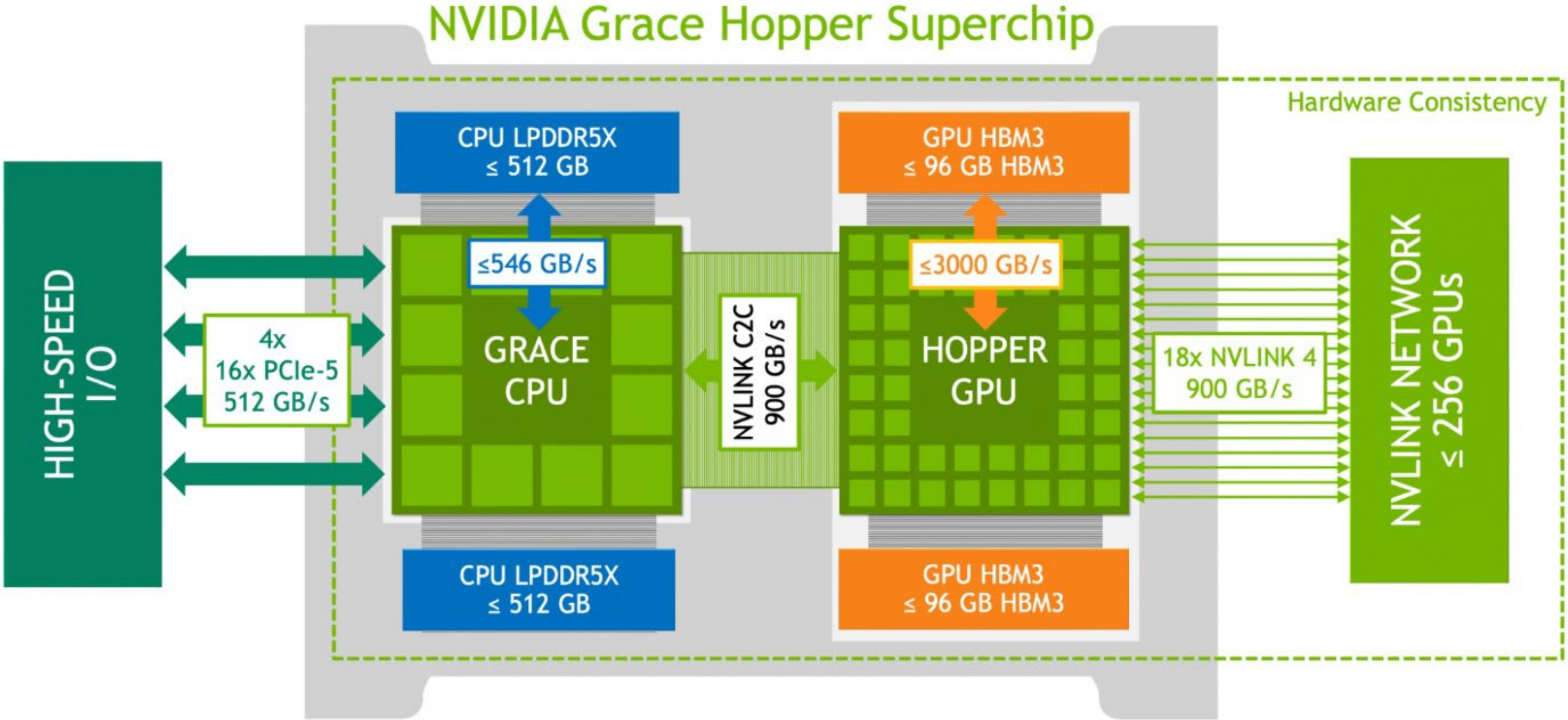
IBM Power 8 & 9 – back in ~2017



the link between co-processors is a big concern! communication may outweigh the benefits of acceleration.

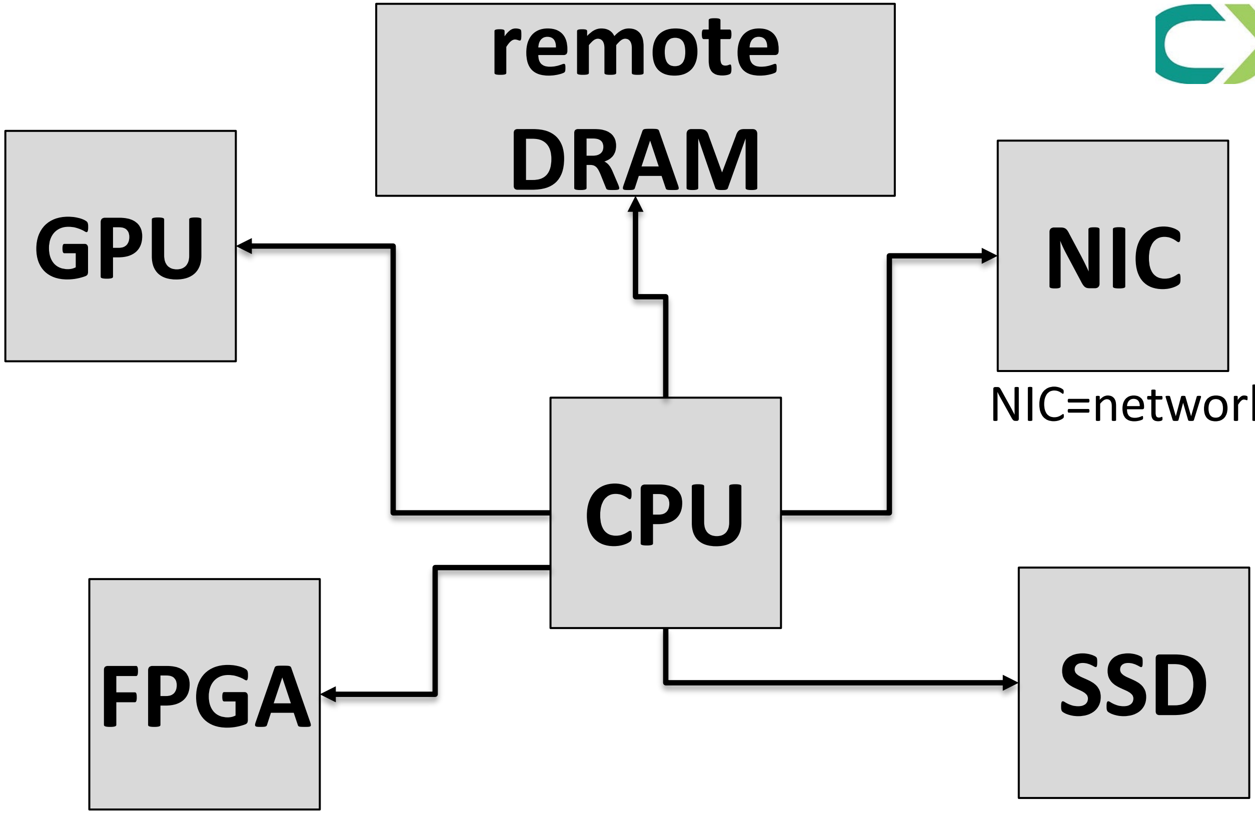
- NVIDIA developed NVLink,
  which is faster alternative to PCIe
- new-generation Power servers have them
- expensive though (compared to PCIe)

NVIDIA Grace Hopper – today



coherent interconnect across the Grace CPU (based on ARM) and Hopper GPU through NVLink

CXL: Compute Express Link – today



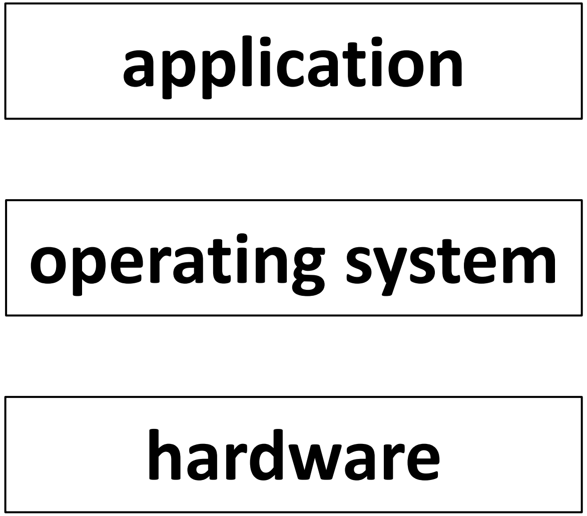
NIC=network interface card

- allows using remote memory on other devices more efficiently than regular network protocols

summary

hardware acceleration …

- need to write code for diverse hardware
- need for efficient data movement across hardware devices



application

operating system

hardware

- more common to side-step OS when dealing with non-general-purpose hardware and directly manage it
- though, an active research topic

- is part of hardware

… is widely available today!