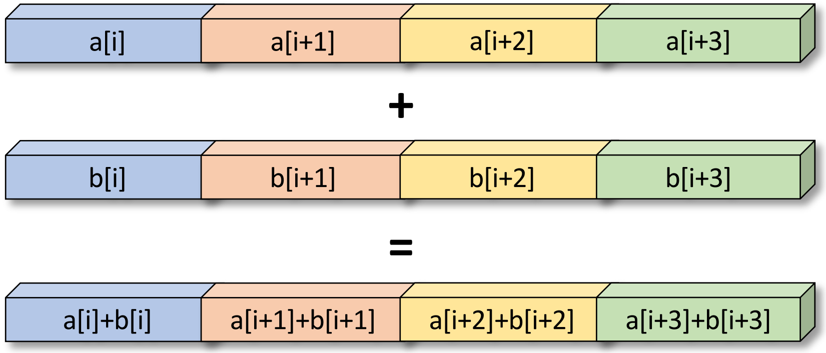
What Every Computational Physicist Should Know About Computer Architecture*



a[i] a[i+1] a[i+2] a[i+3]

+

b[i] b[i+1] b[i+2] b[i+3]

=

a[i]+b[i] a[i+1]+b[i+1] a[i+2]+b[i+2] a[i+3]+b[i+3]

*Plus some other useful technology

Ian Cosden, Princeton University

Steve Lantz, Cornell University

Cornell University
Center for Advanced Computing

CoDaS-HEP Summer School, August 1, 2022

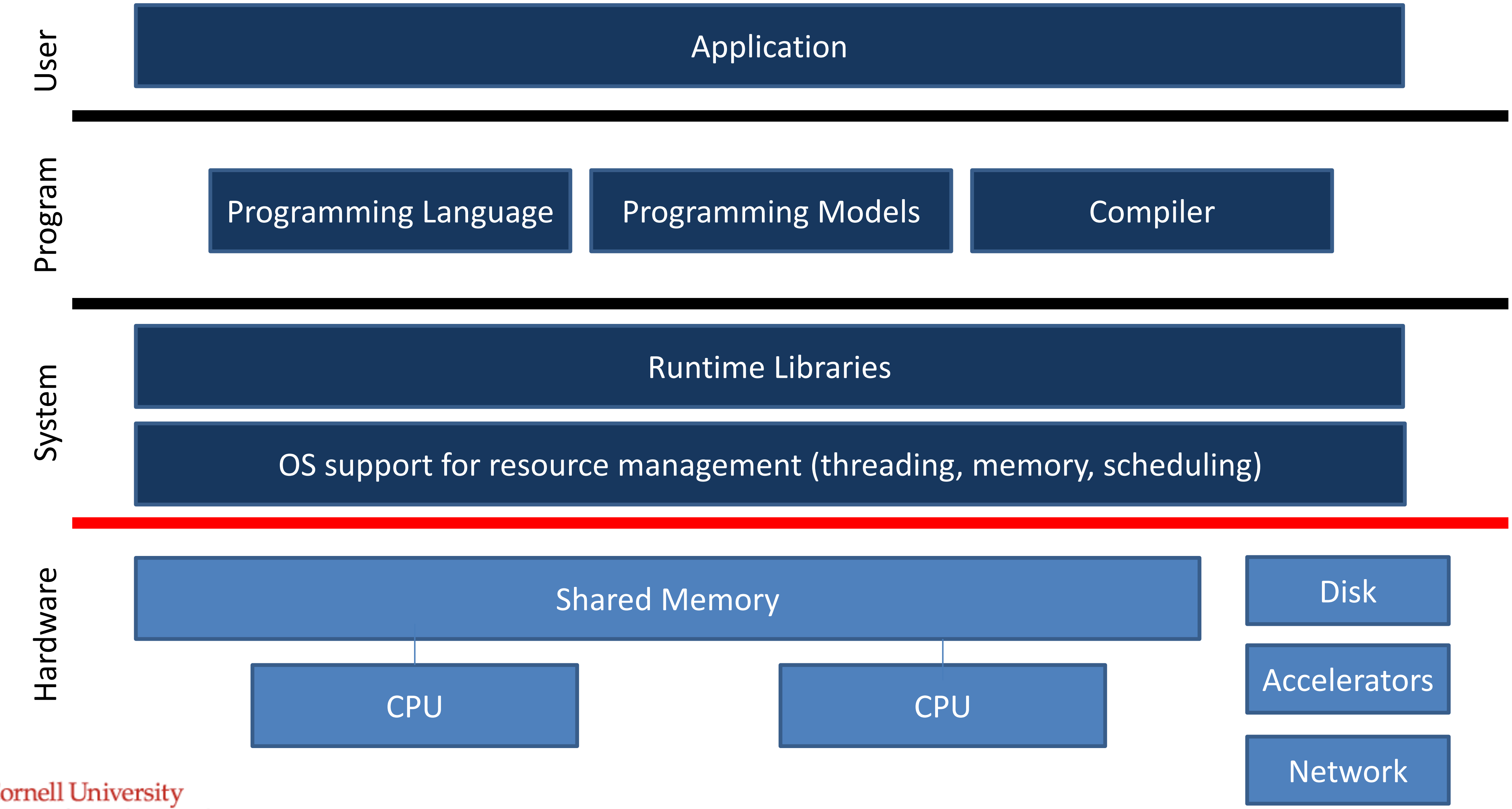
Outline

- Motivation
- HPC Cluster
- Compute
- Memory
- Disk
- Other Architectures

Why Care About the Hardware?

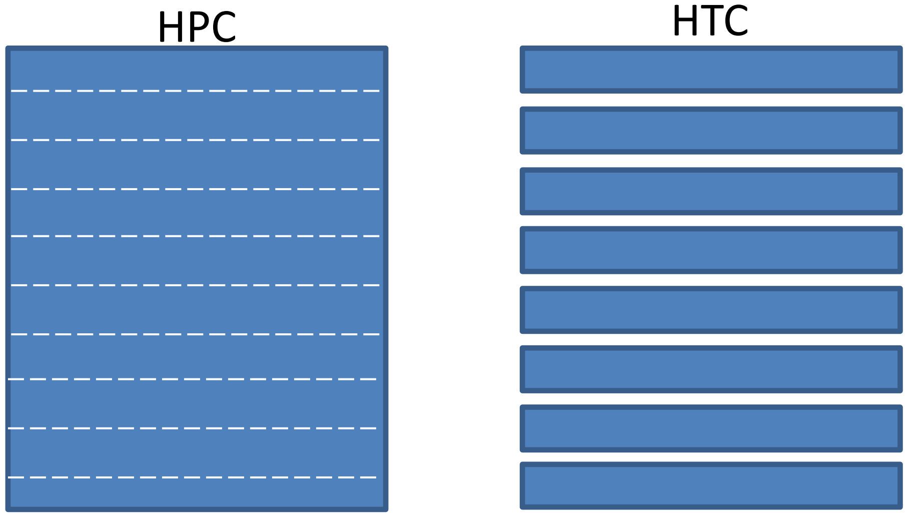
- Get the most out of this week
- Be able to ask better questions and understand the answers
- Architecture changes can force new design and implementation strategies
  - Prevent programming mistakes
  - Make educated decisions
- Exploit the hardware to your advantage (performance)

Hardware is the Foundation



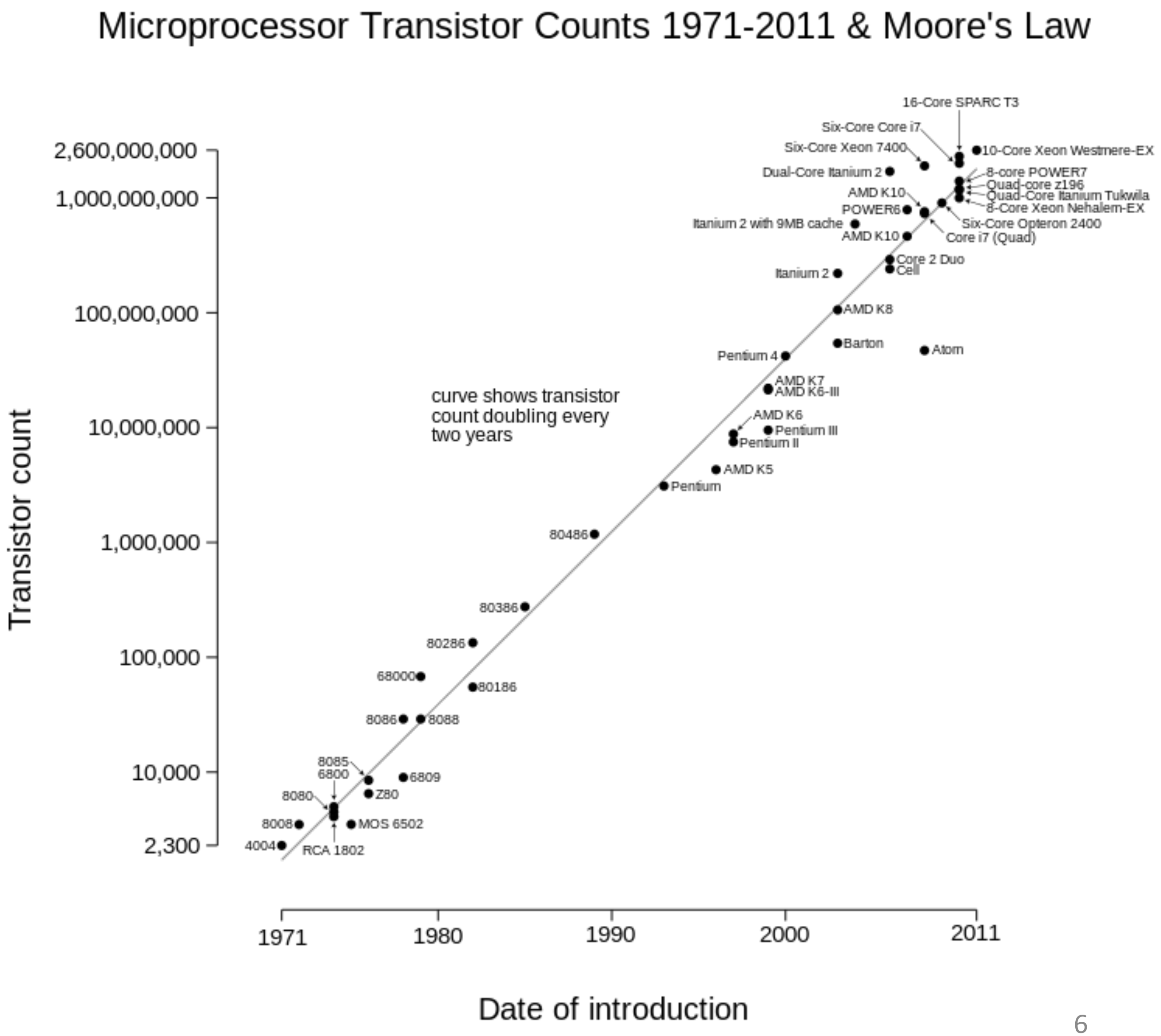
Why Care About HPC Hardware?

- High-Performance Computing (HPC)
  - Aggregate computing power in order to deliver far more capacity than a single computer
  - Supercomputers
- Some problems demand it from the onset; <u>many</u> problems evolve to need it
  - When you outgrow your laptop, desktop, departmental server
  - Ask: do you need greater *capability*, or just *capacity*?
  - Either way, HPC clusters run lots of High-Throughput Computing (HTC) jobs



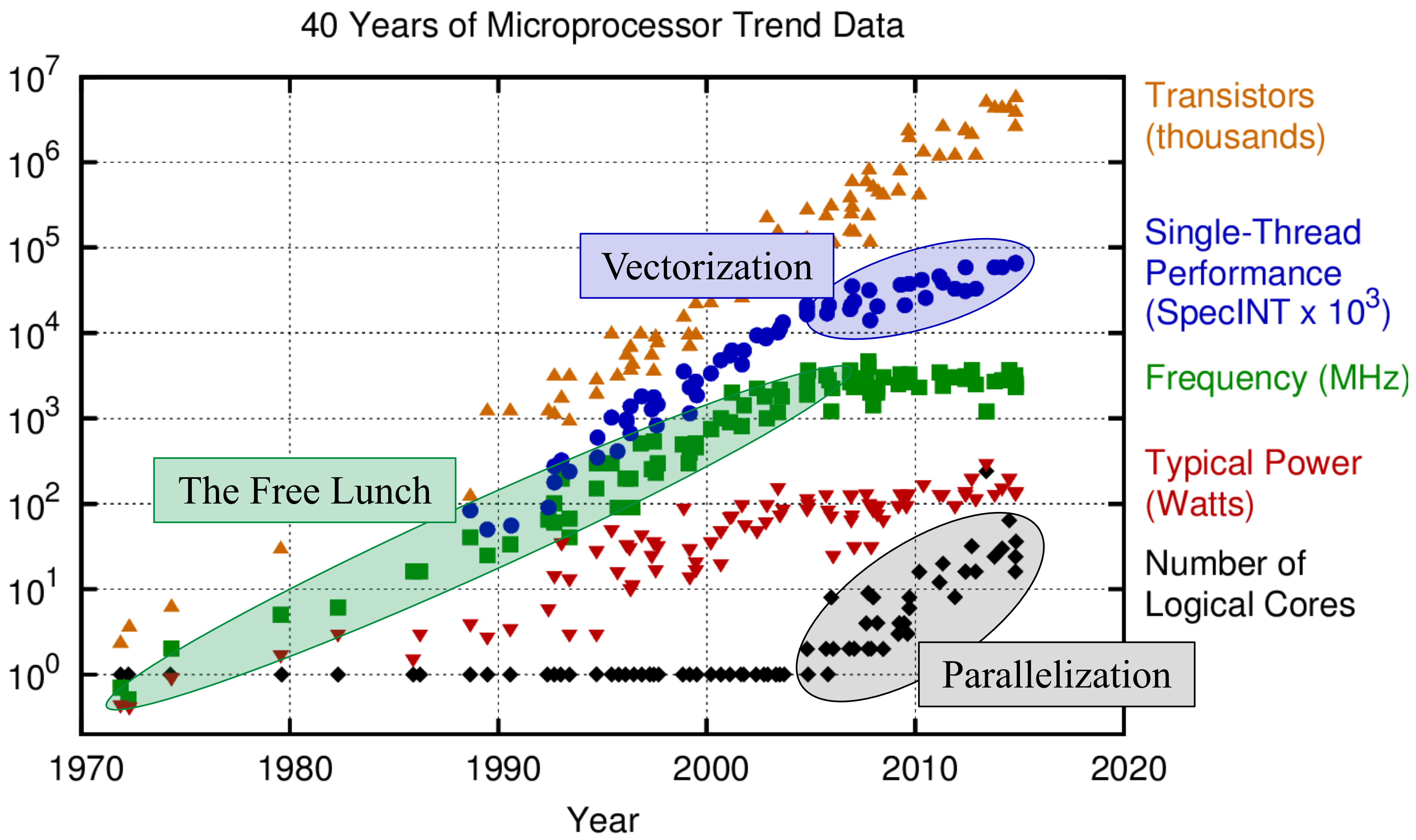
Moore’s Law

- Number of transistors doubles ever 18-24 months
- More transistors = better, right?



Microprocessor Transistor Counts 1971-2011 & Moore's Law

The Free Lunch is Over

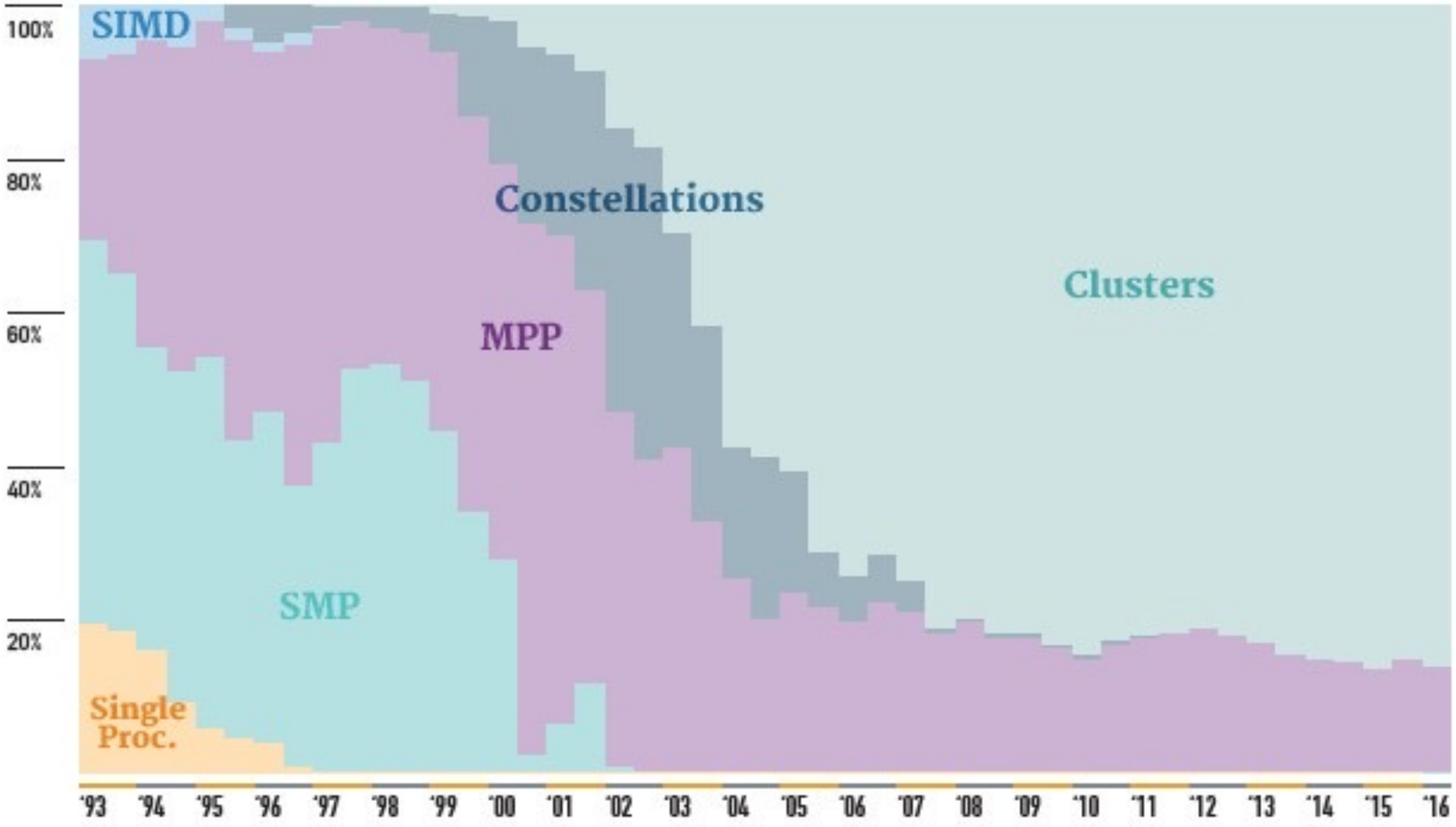


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

https://www.karlrupp.net/2015/06/40-years-of-microprocessor-trend-data/

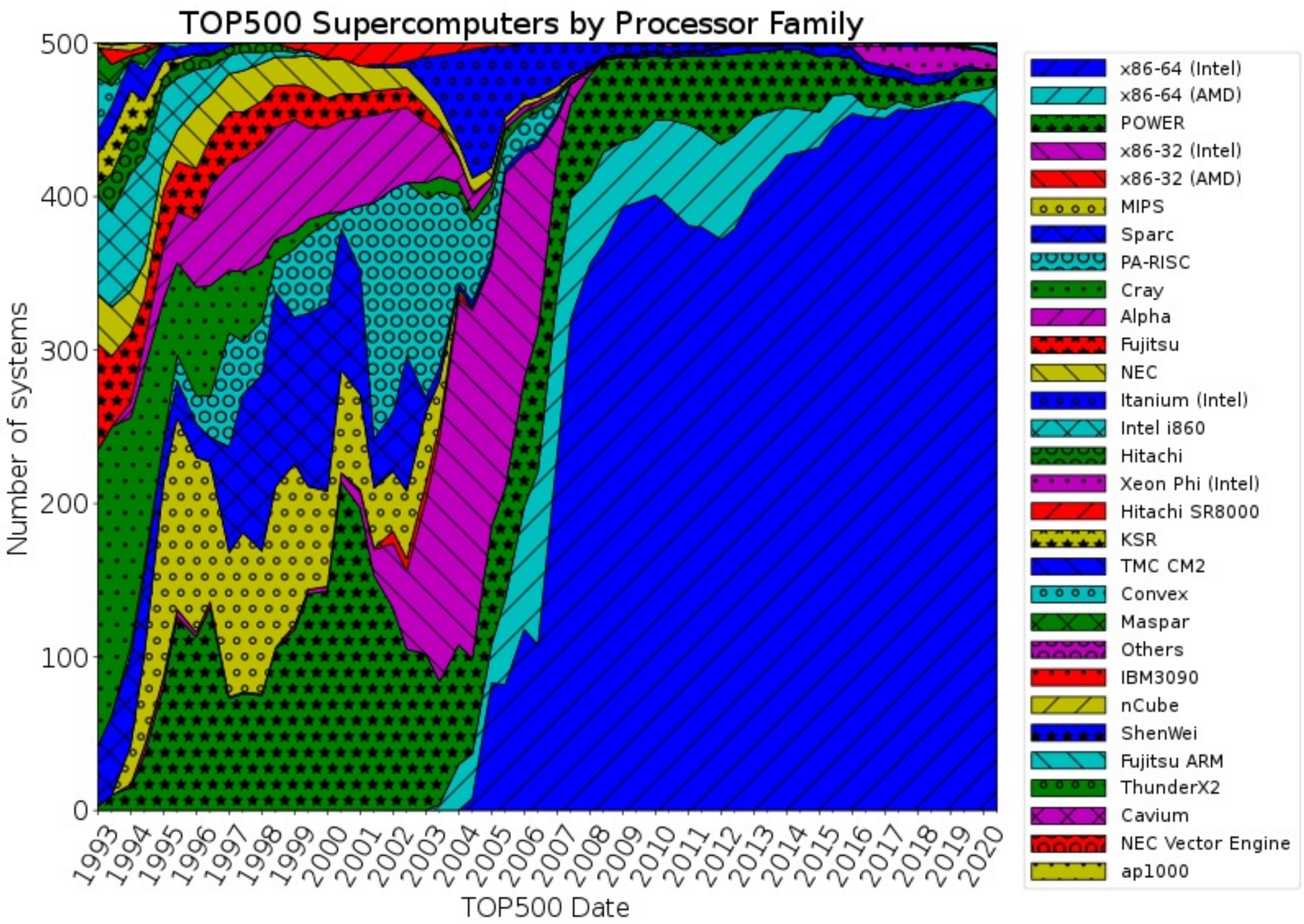
Top 500 Supercomputers

https://www.top500.org



https://www.nextplatform.com/2016/06/20/china-topples-united-states-top-supercomputer-user/

Why Talk About Intel Architecture?



https://en.wikipedia.org/wiki/TOP500

Outline

- Motivation
- HPC Cluster
- Compute
- Memory
- Disk
- Other Architectures

Zoom In On A Cluster

- A cluster is just *connected* collection of computers called nodes
- The high-speed network that connects the cluster is called the interconnect
- A single, unconnected node is usually just called a server
- Entire clusters, as well as stand-alone servers, are sometimes referred to as machines

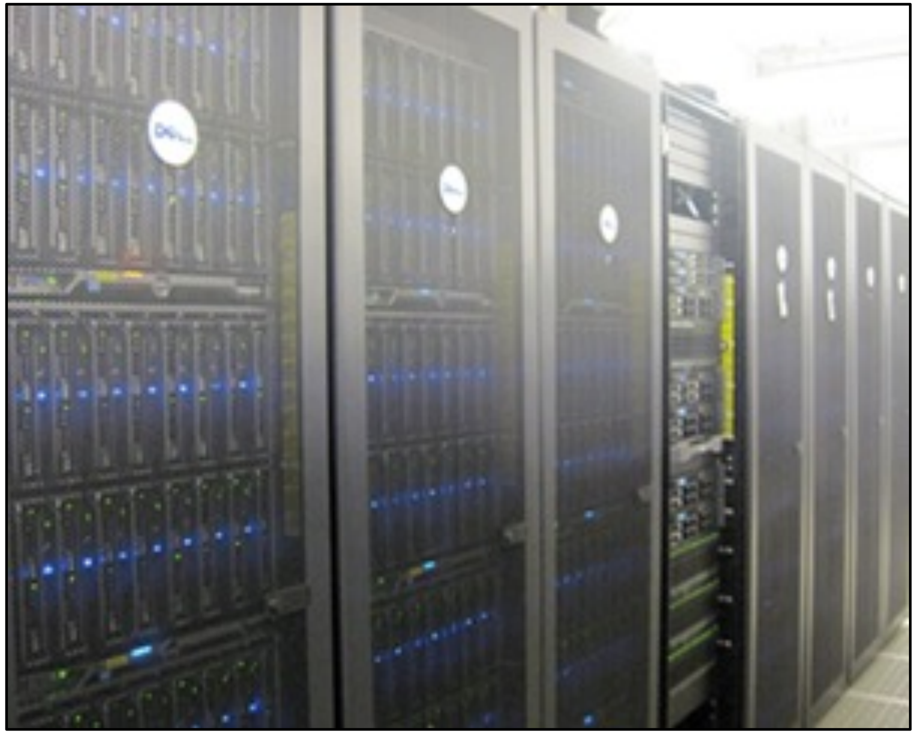
Parallel Computers

HPC Cluster: the vast majority of HPC systems

- A collection of servers connected to form a single entity
    - Each is essentially a self-contained computer
    - OS with CPU, RAM, hard drive, etc.
- Stored in racks in a dedicated machine room
- Networked together a via low-latency interconnect
    - Ethernet < InfiniBand, Omni-Path (Intel)
- Interconnect extends to shared storage

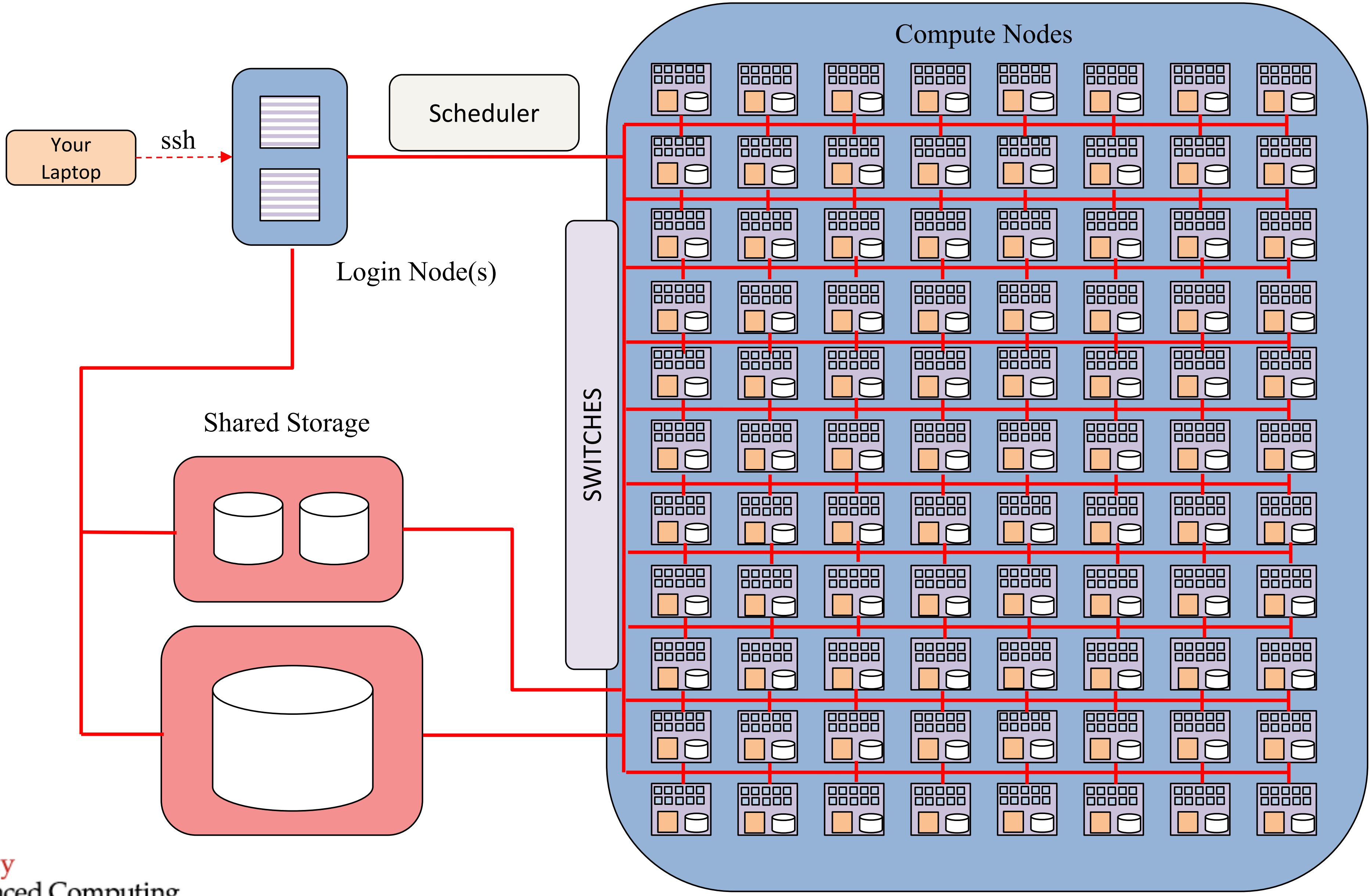
SMP System: Symmetric Multi-Processing

- CPUs all share memory – essentially one computer
- Expensive, likely needed for a unique purpose
    - Huge memory, huge OpenMP jobs



Princeton’s Della Cluster

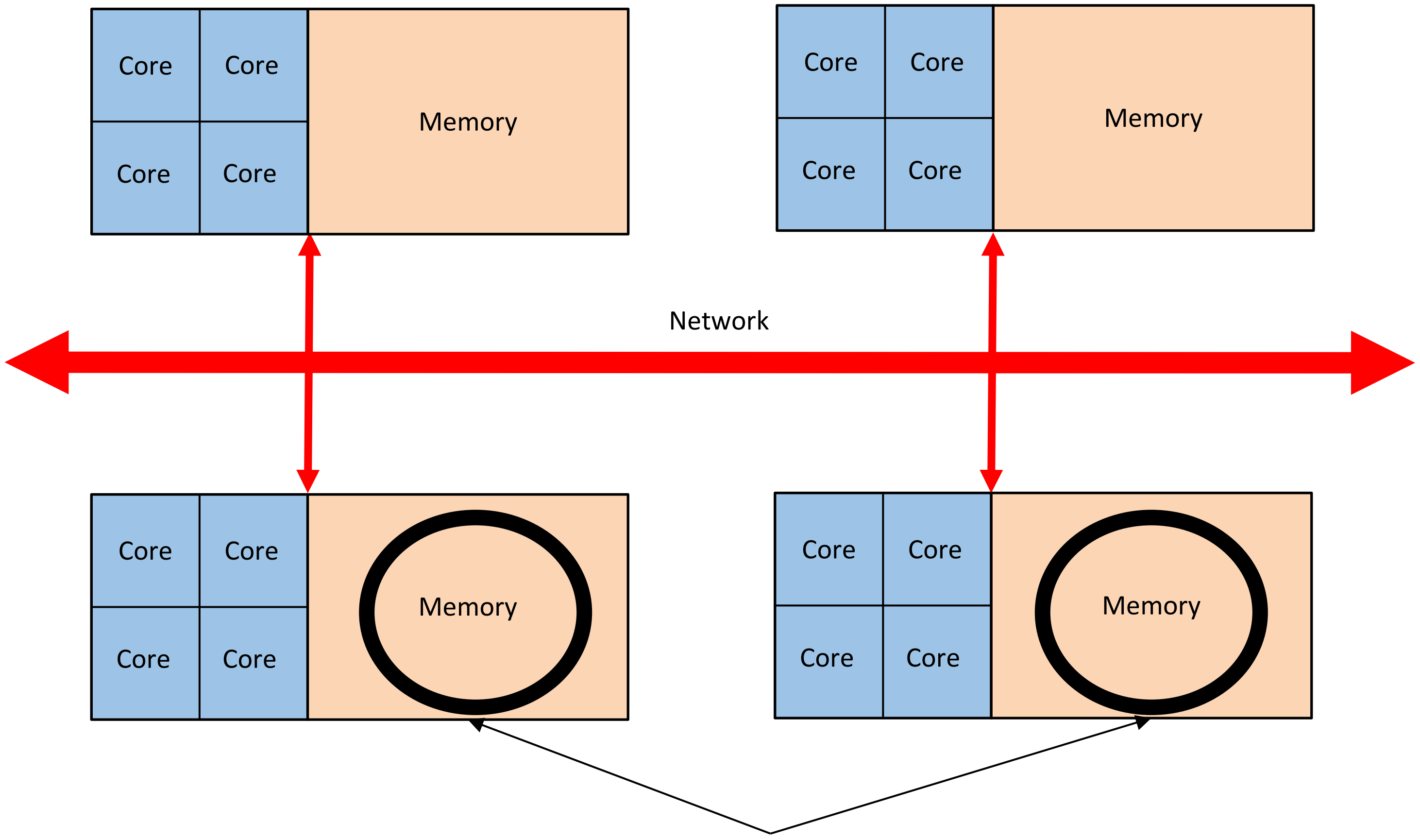
Basic Cluster Layout



Cornell University
Center for Advanced Computing

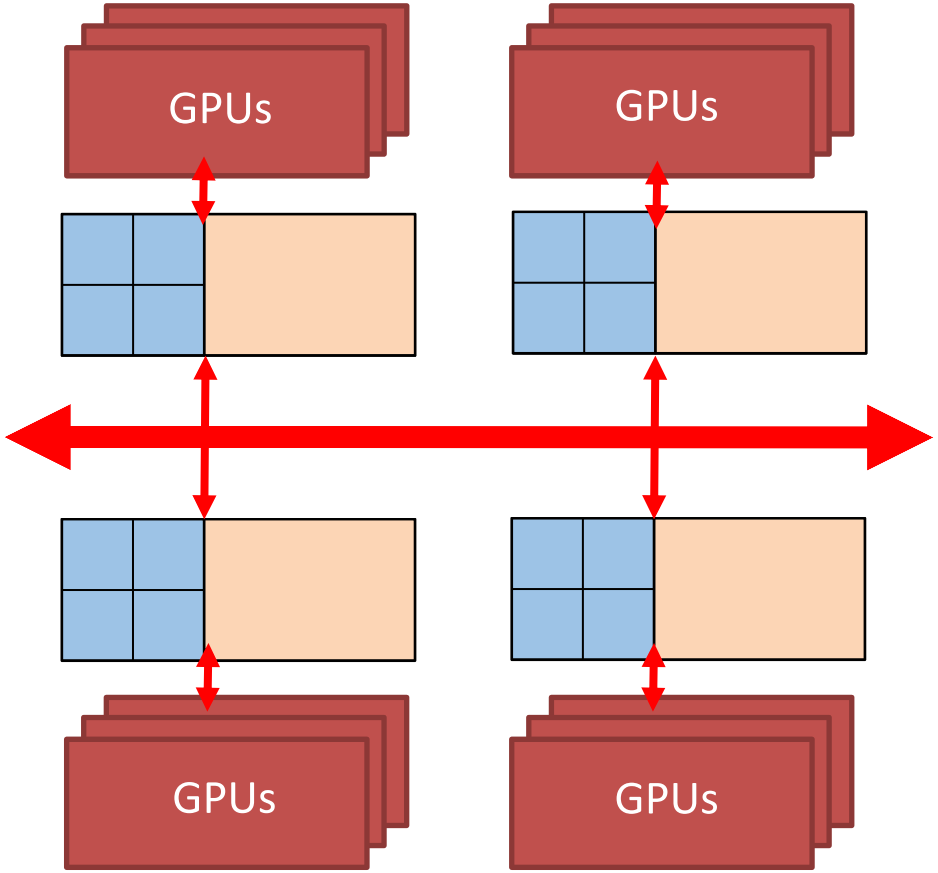
Nodes Are Discrete Systems

- Memory address space
  - Range of unique addresses
- Separate for each node
- Can’t just access all memory in cluster



Each node has a separate address space

Clusters With Accelerators



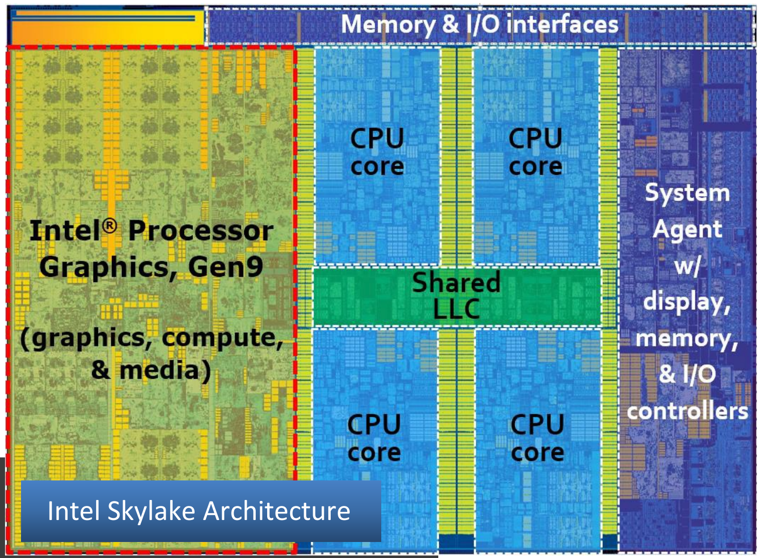
- "Heterogeneous" system architecture
- Accelerators: GPUs, typically
  - Or FPGAs, etc.
  - Node can have multiple accelerators
- Programmable with MPI + x
  - Where x = OpenMP, OpenACC, CUDA, …
- Increases computational power
  - Increases FLOPS / Watt
- Trending in Top500.org
  - Strong shift toward systems with accelerators started ~10 years ago

Outline

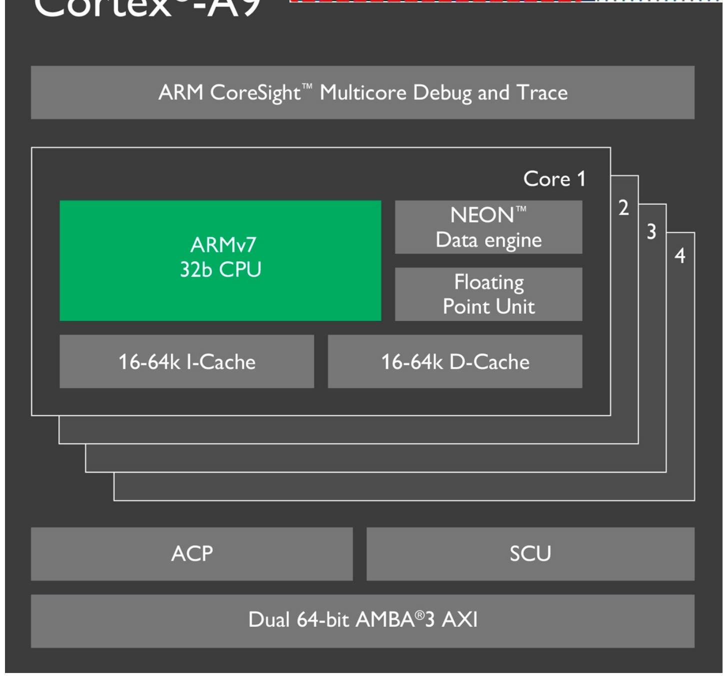
- Motivation
- HPC Cluster
- Compute
- Memory
- Disk
- Other Architectures

Multicore Processors

- Nearly all modern processors are multicores
  - In servers: Intel Xeon, AMD EPYC, IBM Power
  - In laptops: Intel, AMD, ARM
  - In GPUs, phones, and mobile devices as well
- Most have vector units, multiple cache levels
- Require special care to program
  - Multi-threading, vectorization, …
  - The free lunch is over!
- Instruction sets can vary
  - Intel and AMD: x86_64 base, not identical
  - IBM Power, ARM, NVIDIA, etc. all differ

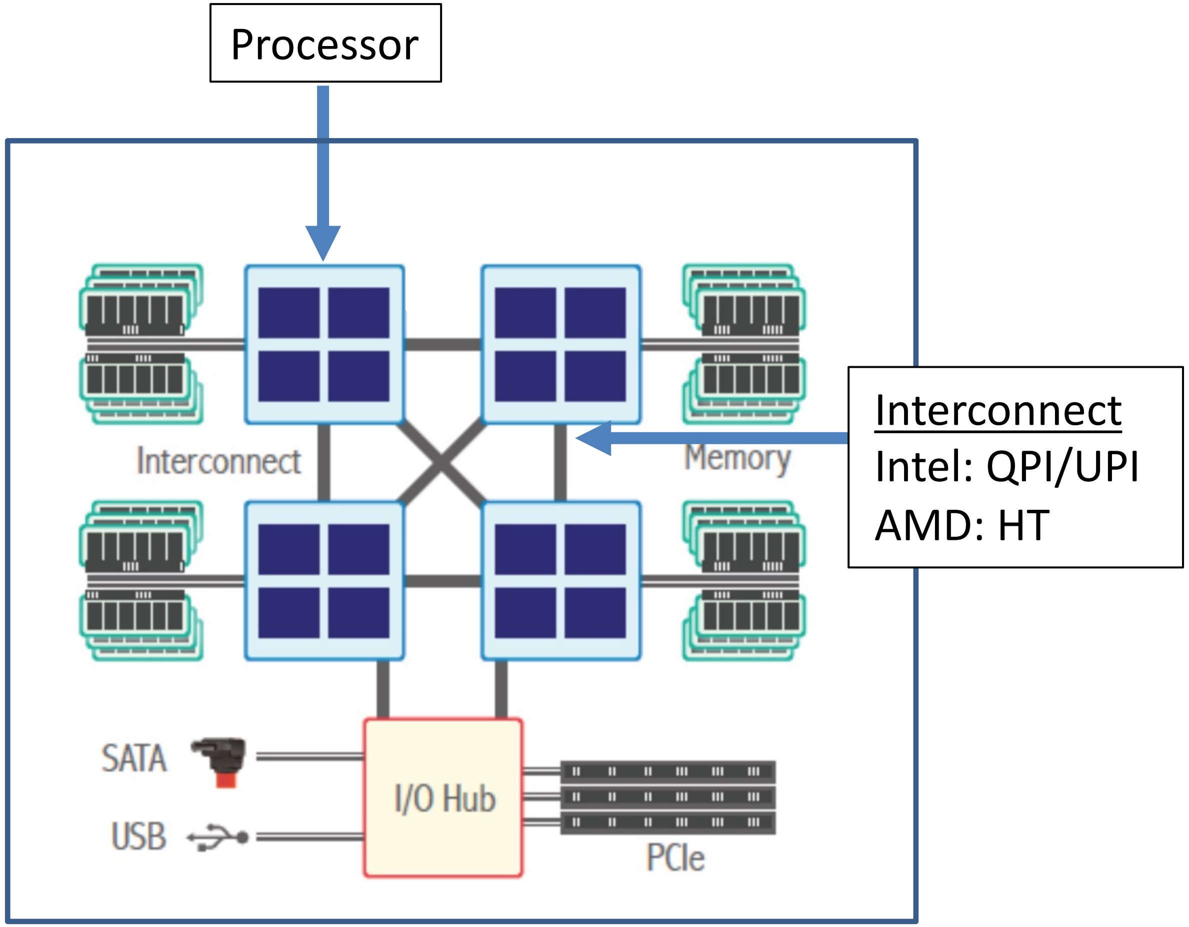

Memory & I/O interfaces
Intel® Processor Graphics, Gen9
(graphics, compute, & media)
CPU core
CPU core
Shared LLC
CPU core
CPU core
System Agent w/ display, memory, & I/O controllers

Intel Skylake Architecture



Motherboard Layout

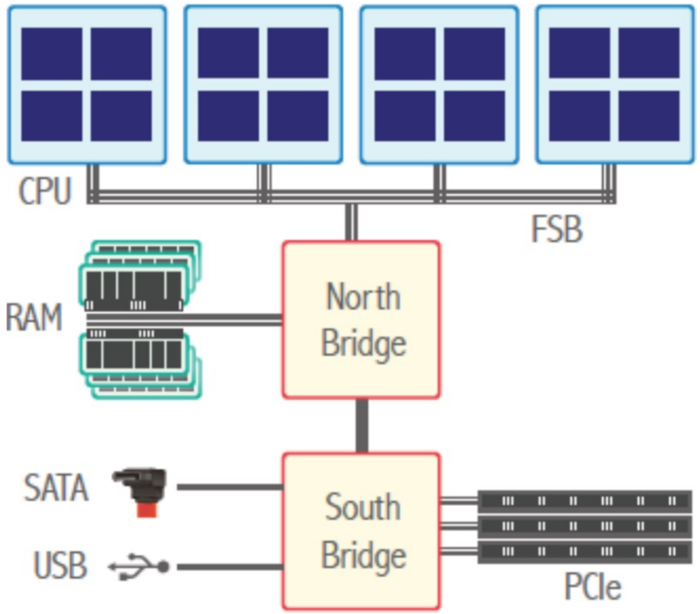
- Typical nodes have 1, 2, or 4 processors
  - 1: laptop/desktops/servers
  - 2: servers/clusters
  - 4 (or more): high end special use
- Each processor is attached to a socket
- Each processor has multiple cores
- CPU can refer to a core... or a processor
- Processors connected via interconnect
- Memory attached to processor/socket
  - Programmer sees single node memory address space

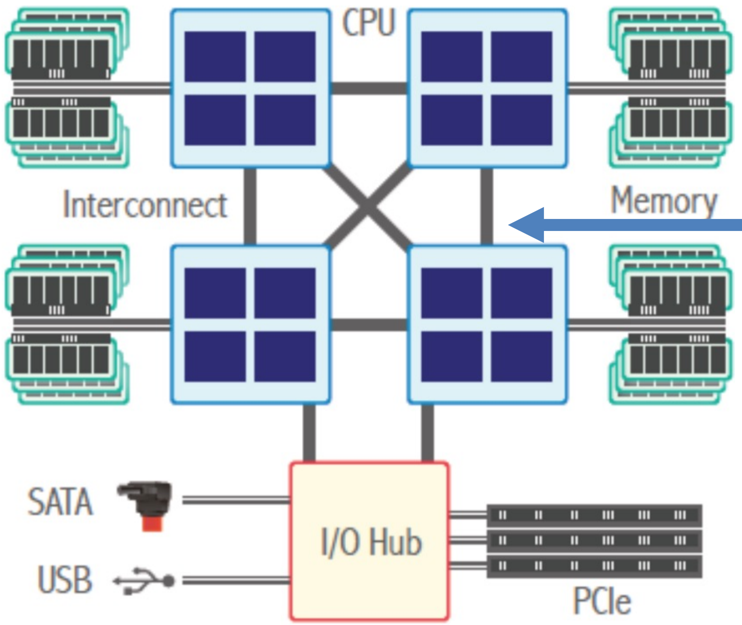


Processor

Interconnect
Intel: QPI/UPI
AMD: HT

Motherboard Layouts, Past and Present



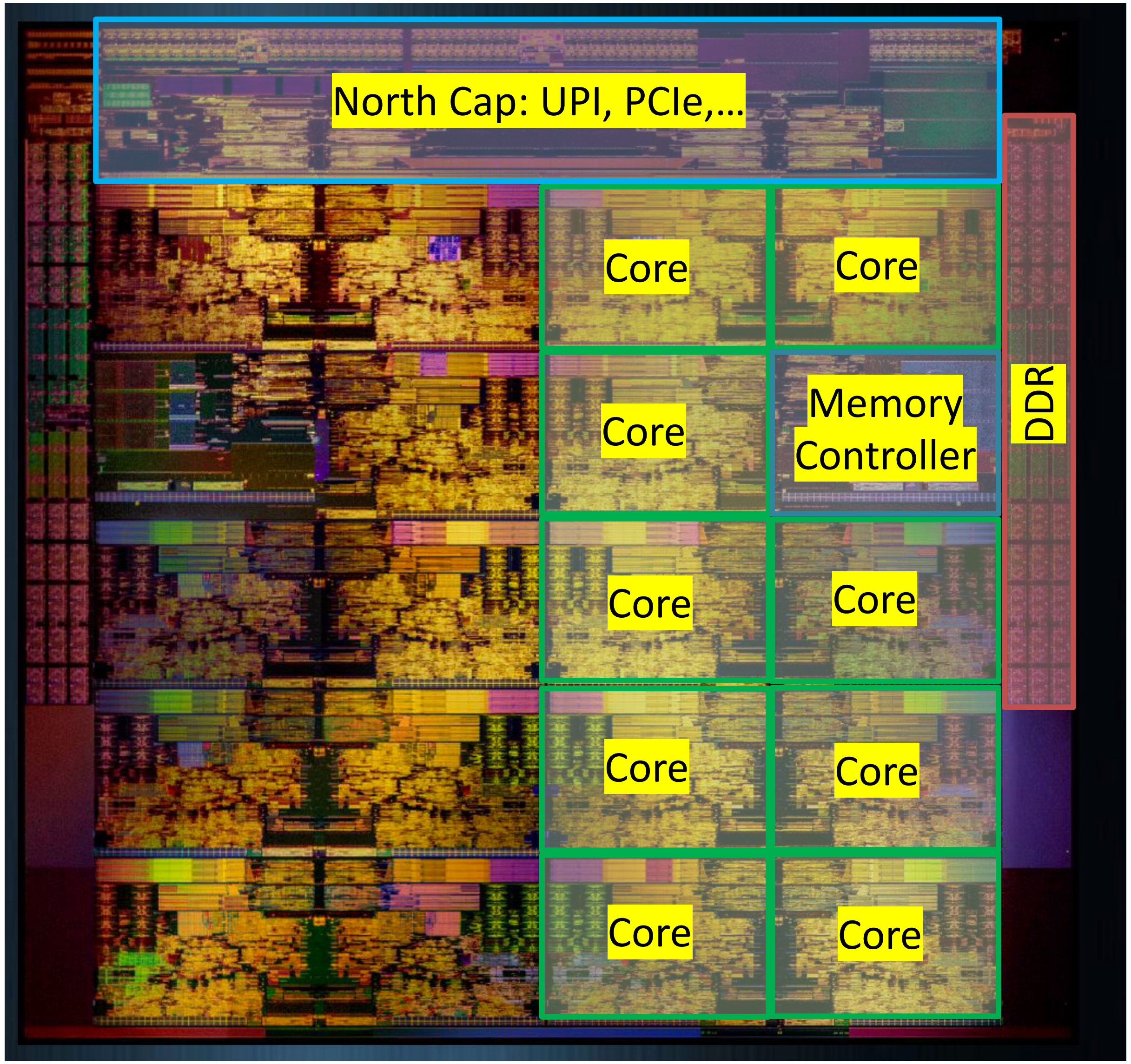


Intel QPI/UPI,
AMD HT

Symmetric MultiProcessing (SMP) –
Memory access is uniform, as all traffic goes via north bridge, but is bottlenecked by front side bus (FSB) and north-bridge-to-RAM bandwidth limitations. Current motherboards have no north bridge; the south bridge is now referred to as the I/O hub or just the chipset.

Non-Uniform Memory Access (NUMA) –
Physical memory is partitioned per CPU, with a fast interconnect to link CPUs to each other and to I/O. This removes bottlenecks but memory is no longer uniform – there may be 30-50% extra latency to access remote memory, more in multi-hop systems.

Example: Intel Skylake Scalable Processor



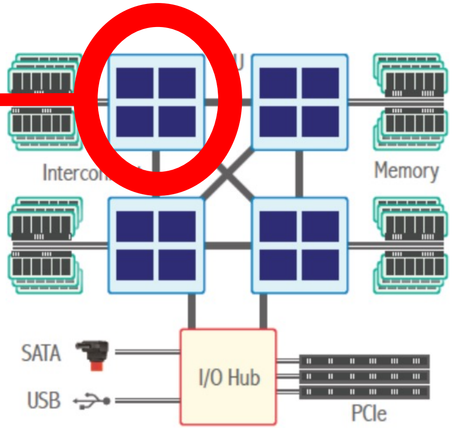
“System on a Chip” (SoC)

- Core: “A complete ensemble of execution logic, and cache storage as well as register files plus instruction counter (IC) for executing a software process or thread.” (Jarp)

UPI – Intel Ultra Path Interconnect

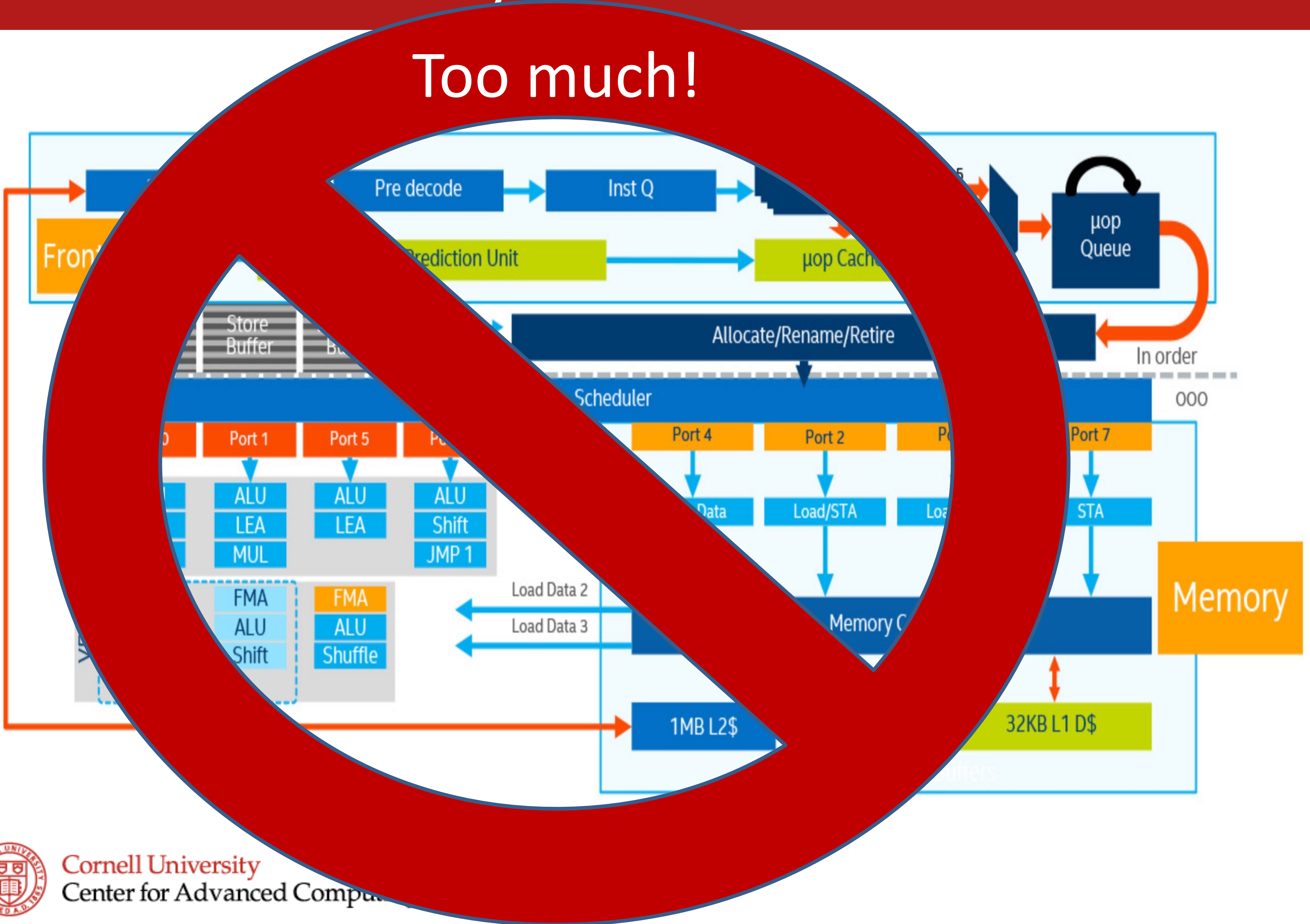
PCIe – Peripheral Component Interconnect Express

DDR – Double Data Rate



https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server)#High_Core_Count_.28HCC.29

Skylake Core Microarchitecture



Key Components:

Control logic
Register file
Functional Units

- ALU (arithmetic and logic unit)
- FPU (floating point unit)

- Data Transfer
- Load / Store

Vectorization Terminology

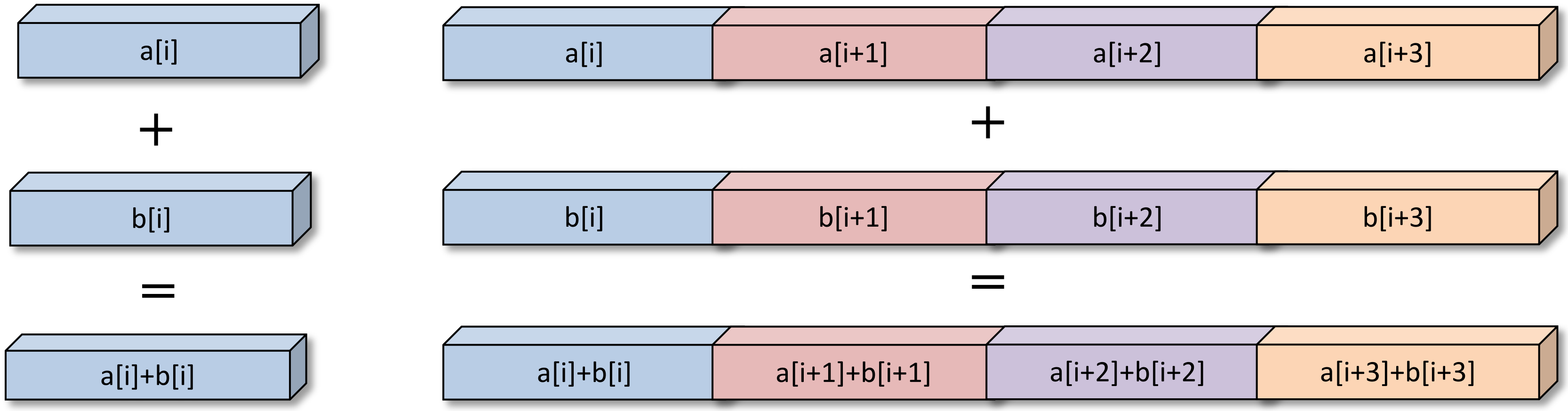
- New generations of CPUs add capabilities and instructions
  - Each core can have one or more vector processing units (VPUs) and vector registers
  - Machine (assembly) commands are often added to take advantage of new hardware
- Vector instructions are also called SIMD = Single Instruction Multiple Data
  - x86_64 has Streaming SIMD Extensions (SSE), Advanced Vector Extensions (AVX)

At inner-most loop level

```
for (int i=0; i<N; i++) {
    c[i]=a[i]+b[i];   }
```

Scalar (SISD)

Vector (SIMD)

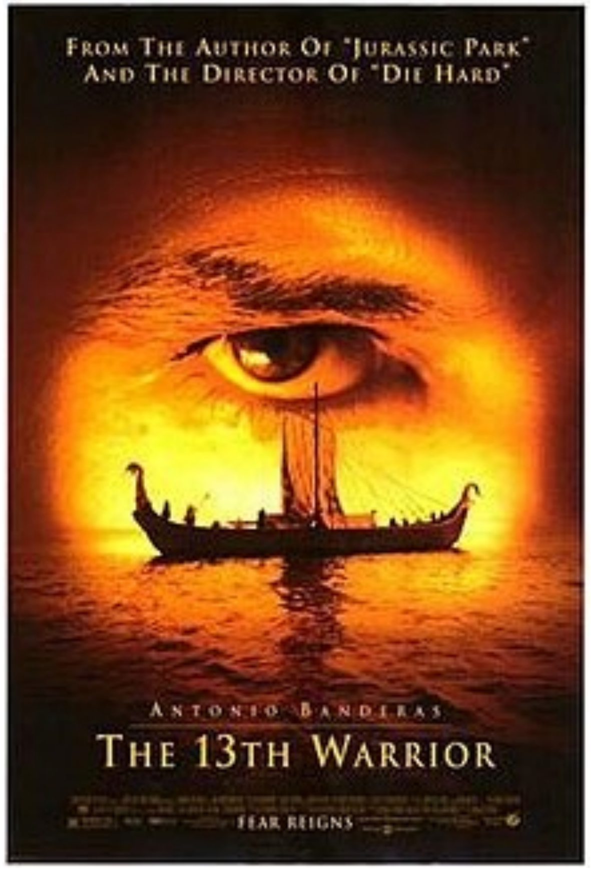


One operation
One result

One operation
Multiple results

HPC Performance Metrics

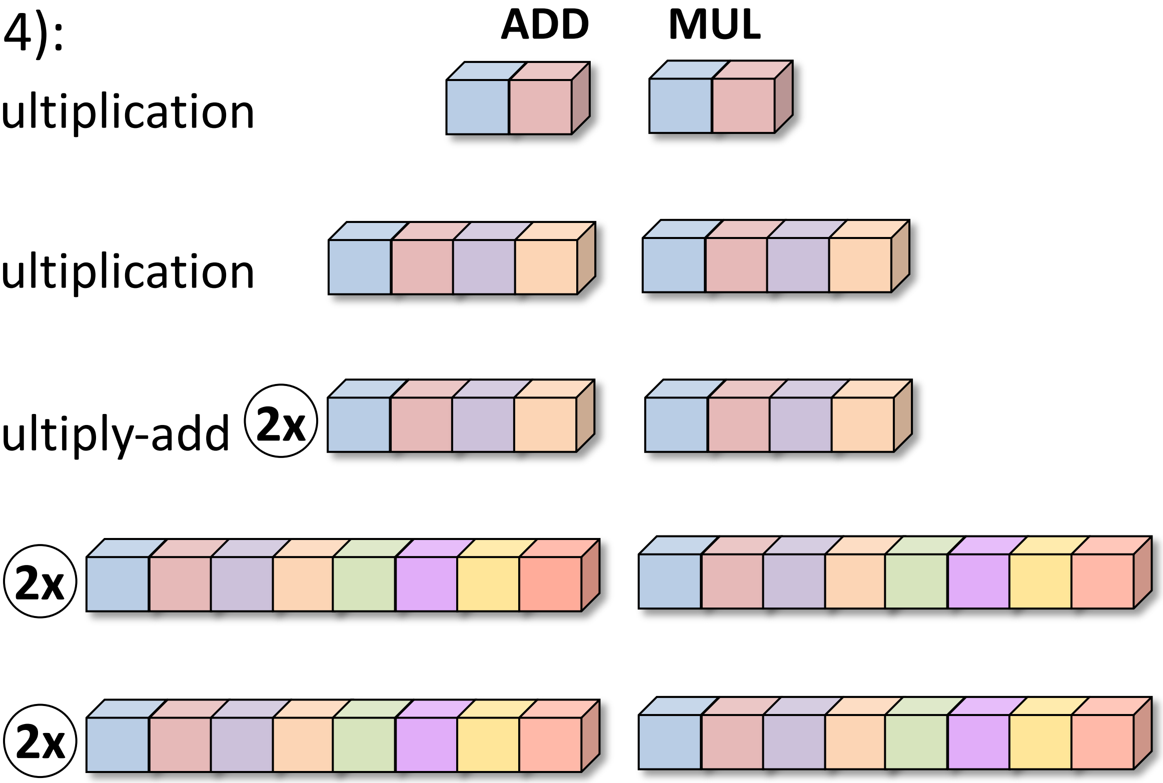
- FLOP = Floating-point Operation
  - FLOPs = Floating-point Operations
  - FLOPS = Floating-point Operations Per Second
  - Often write FLOP/s or FLOP/sec to avoid ambiguity
- Calculation for single processor
  - Typically Double Precision (64-bit)
  - FLOPS = (Clock Speed)*(Cores)*(FLOPs/cycle)
- FLOPS/$, FLOPS/Watt are useful metrics too
- Bandwidth metrics (“giga” usually means 10^9)
  - GB/s = gigabytes/sec
  - Gb/s = gigabits/sec


From The Author Of "Jurassic Park"
And The Director Of "Die Hard"
Antonio Banderas
The 13th Warrior
Fear Reigns

N.B. - not to be confused with a Hollywood flop!

Intel Xeon (Server) Architecture Codenames

- Number of FLOP/s depends on width, number, and abilities of VPUs in a core
- Per-core rates, double precision (64-bit doubles – twice this for 32-bit floats)
  - Pentium 4*… Nehalem/Westmere (SSE2… SSE4):
    - 4 DP FLOPs/cycle: 128-bit addition + 128-bit multiplication
  - Sandy Bridge/Ivy Bridge (AVX)
    - 8 DP FLOPs/cycle: 256-bit addition + 256-bit multiplication
  - Haswell/Broadwell (AVX2)
    - 16 DP FLOPs/cycle: **two**, 256-bit **FMA**, fused multiply-add
  - KNL/Skylake (AVX-512)
    - 32 DP FLOPs/cycle: **two****, 512-bit **FMA**
  - Cascade Lake (AVX-512 VNNI with INT8)
    - 32 DP FLOPs/cycle: **two**, 512-bit **FMA**



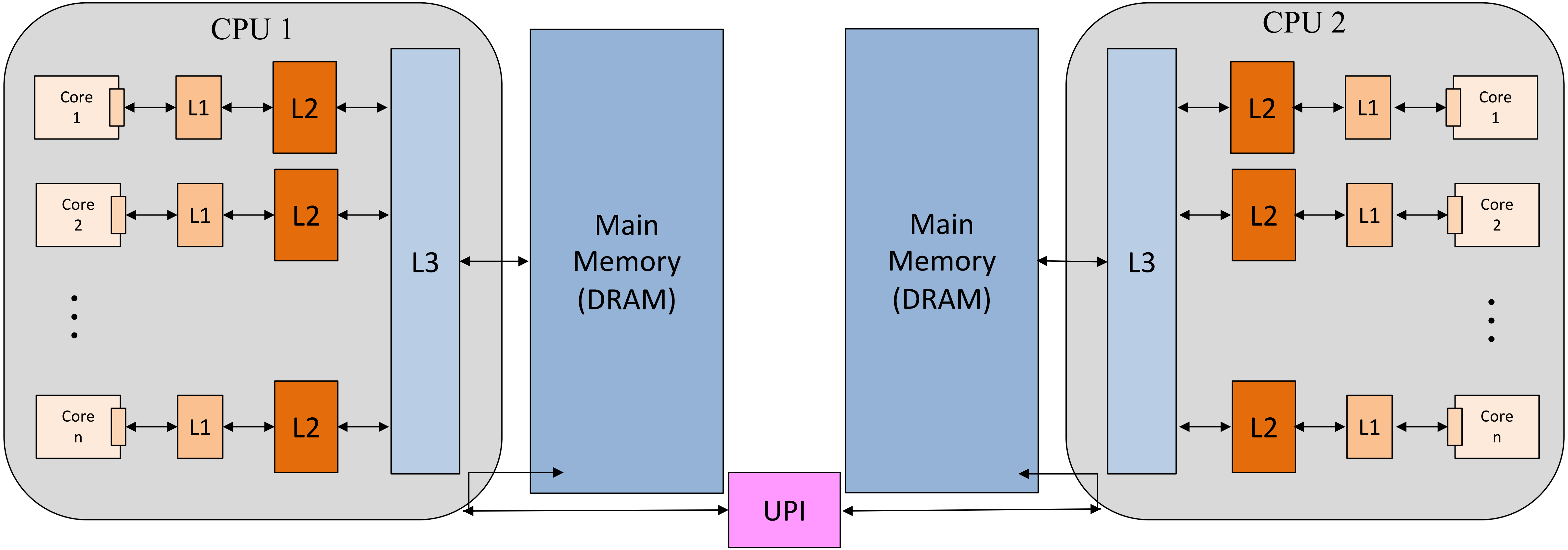
*Pentium III (1999) introduced 128-bit SSE for 4 single-precision floats
**Some Skylake-SPs can do only one 512-bit FMA = (a × b + c), per core

Outline

- Motivation
- HPC Cluster
- Compute
- Memory
  - Layout
  - Cache
  - Performance
- Disk
- Other Architectures

Memory Hierarchy

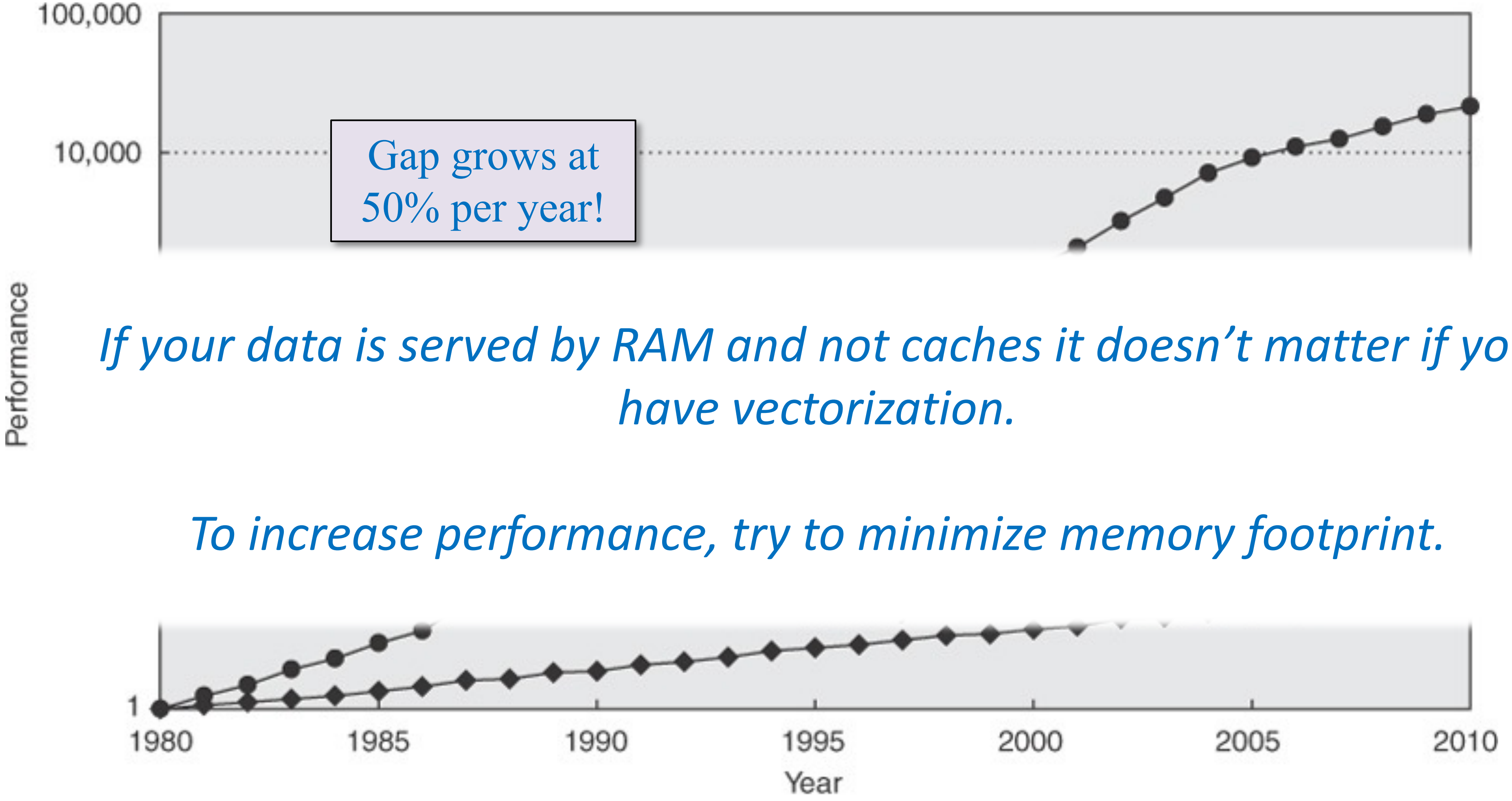
Dual Socket Intel Xeon CPU



	Registers	L1 Cache	L2 Cache	L3 Cache	DRAM	Disk
Speed	1 cycle	~4 cycles	~10 cycles	~30 cycles	~200 cycles	10ms
Size	< KB per core	~32 KB per core	~256 KB per core	~35 MB per socket	~100 GB per socket	TB

Cornell
Center for Advanced Computing

Growth in Memory Performance Lags CPU



Gap grows at 50% per year!

If your data is served by RAM and not caches it doesn’t matter if you have vectorization.

To increase performance, try to minimize memory footprint.

http://web.sfc.keio.ac.jp/~rdv/keio/sfc/teaching/architecture/architecture-2008/hennessy-patterson/Ch5-fig02.jpg

Outline

- Motivation
- HPC Cluster
- Compute
- Memory
- Disk
  - Types
  - Filesystems
- Other Architectures

File I/O

“A supercomputer is a device for turning compute-bound problems into I/O-bound problems.”

-- *Ken Batcher, Emeritus Professor of Computer Science at Kent State University*

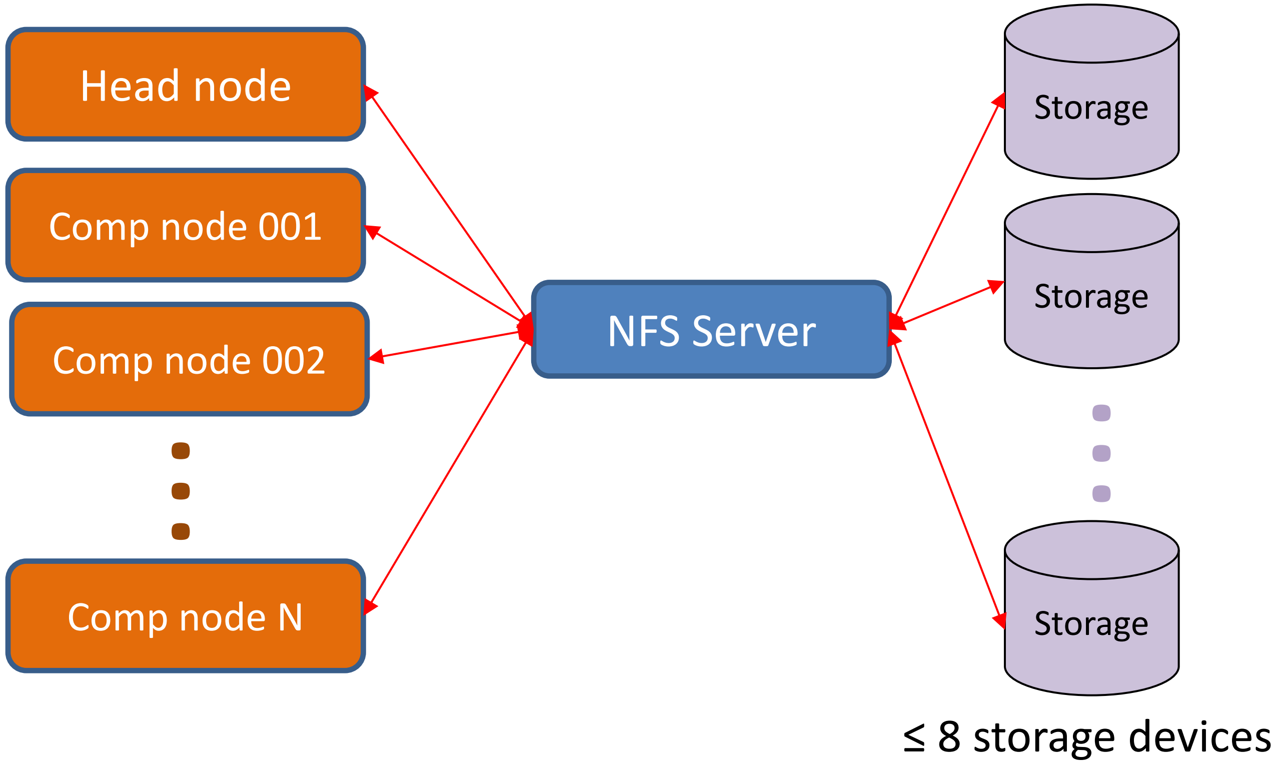
Types of Disk

- Hard Disk Drive (HDD)
  - Traditional Spinning Disk
- Solid State Drives (SSD)
  - ~5x faster than HDD
- Non-Volatile Memory Express (NVMe)
  - ~5x faster than SSD

Photo Credit: Maximum PC

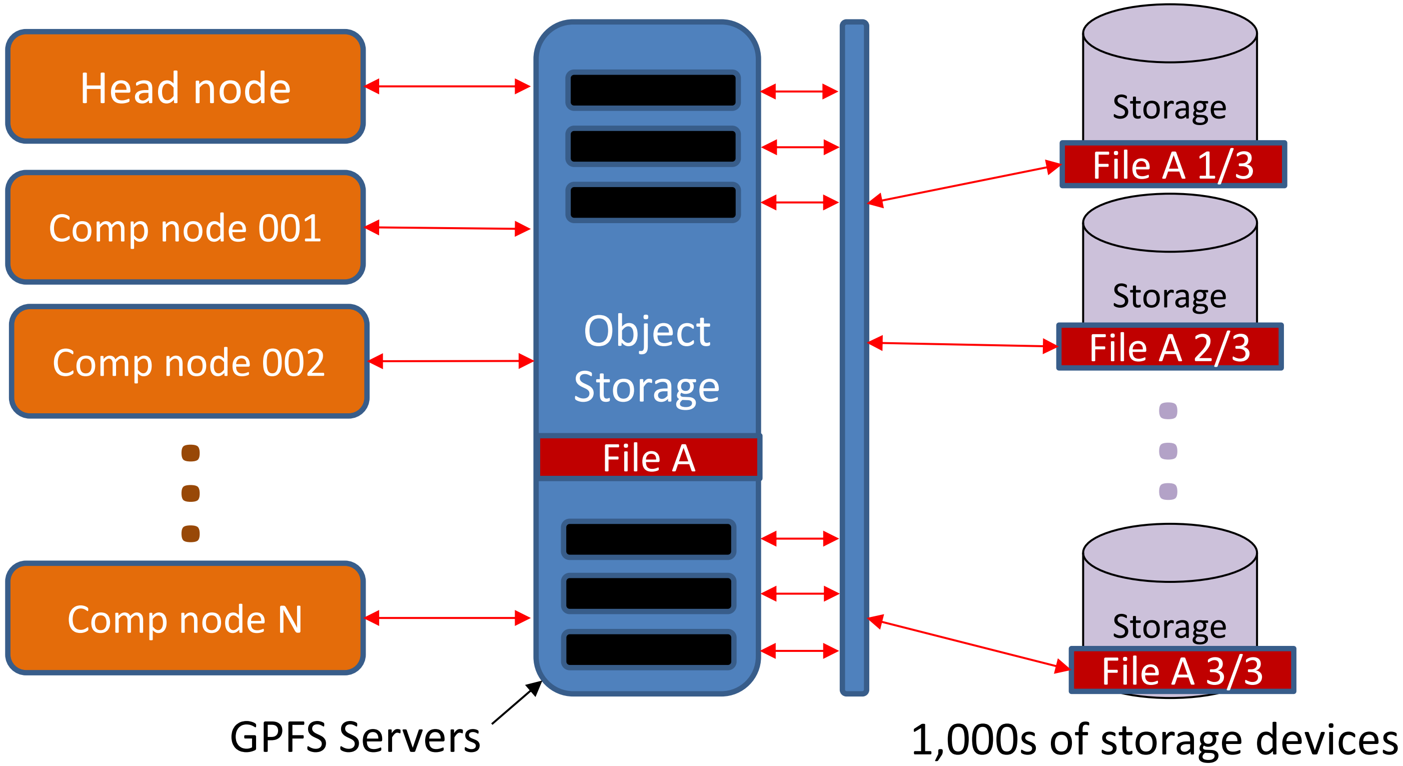
Types of Filesystems - NFS

- NFS – Network File System
  - Simple, cheap, and common
  - Single filesystem across network
  - Not well suited for large throughput



Parallel Filesystems

- GPFS (General Parallel Filesystem – IBM)
  - Designed for parallel read/writes
  - Large files spread over multiple storage devices
  - Allows concurrent access
  - Significantly increase throughput
- Lustre
  - Similar idea
  - Different implementation
- Parallel I/O library in software
  - Necessary for performance realization



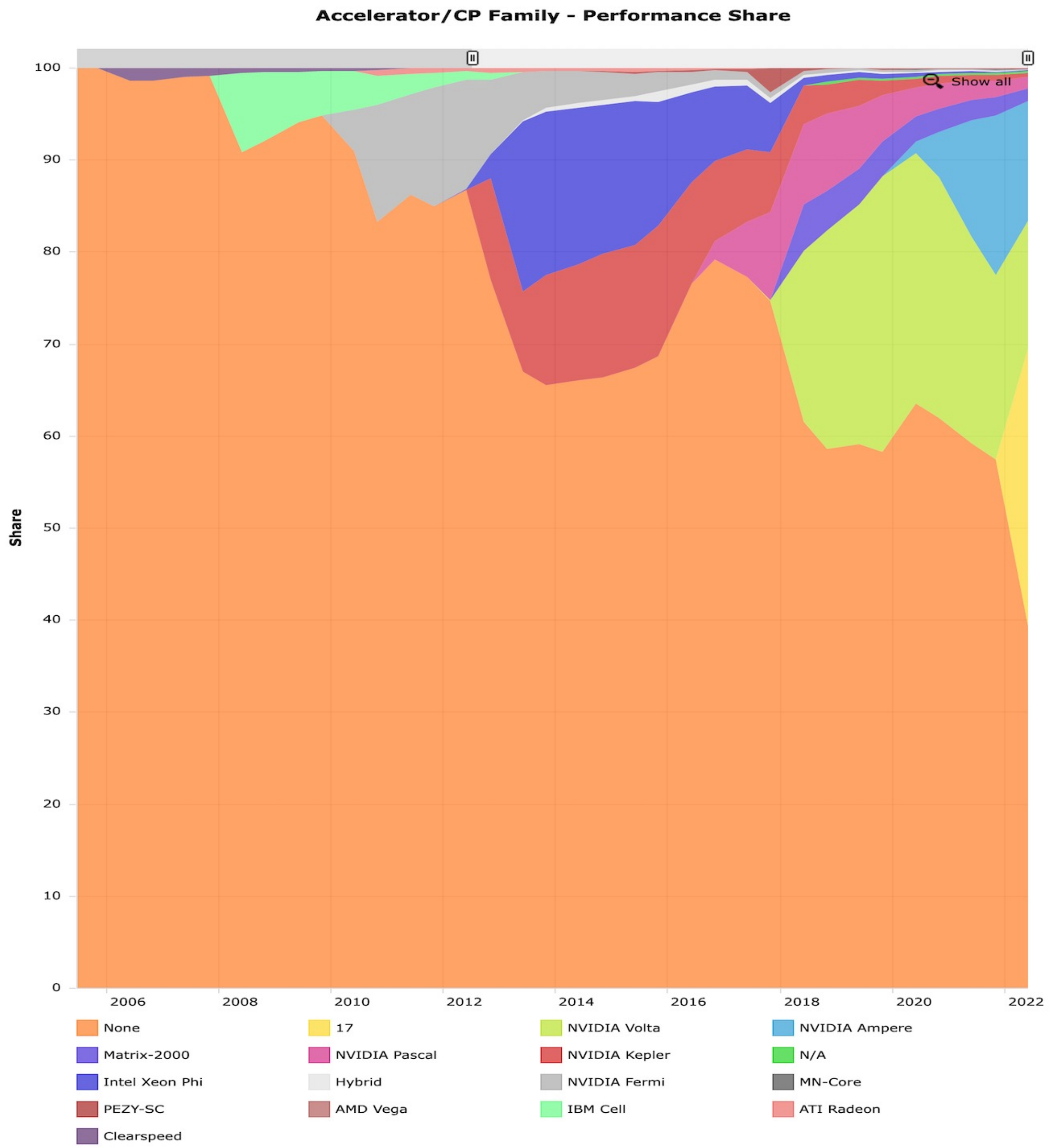
Cornell University
Center for Advanced Computing

Outline

- Motivation
- HPC Cluster
- Compute
- Memory
- Disk
- Other Architectures
  - GPUs
  - Xeon Phi
  - Cloud

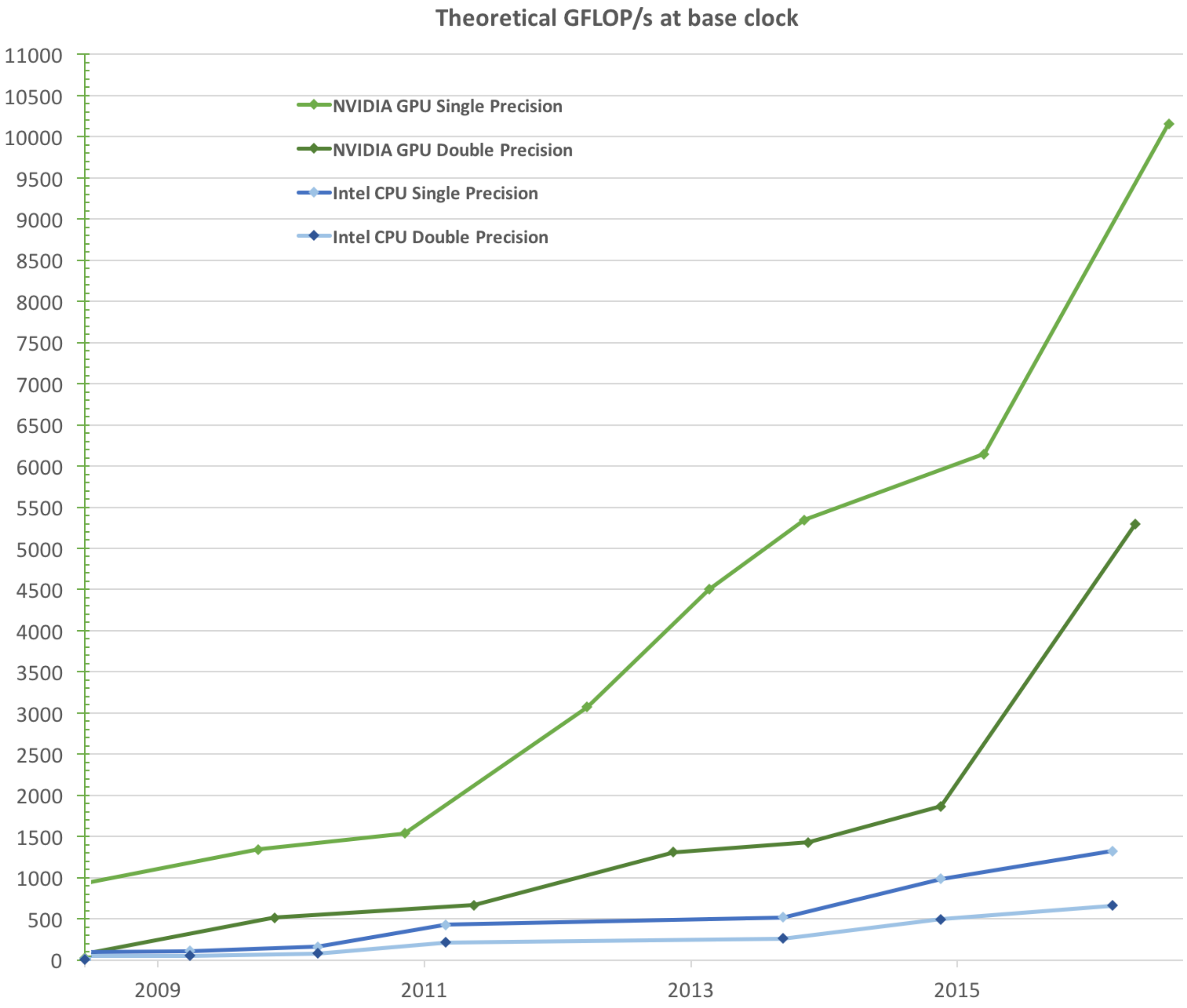
Heterogeneous Systems on the Rise

- The computing landscape changes rapidly
- 60% of TOP500 systems now have accelerators (June 2022)
- Graphics processing units (GPUs) are by far the most common type of accelerator



https://www.top500.org/statistics

GPU and CPU Performance Trends



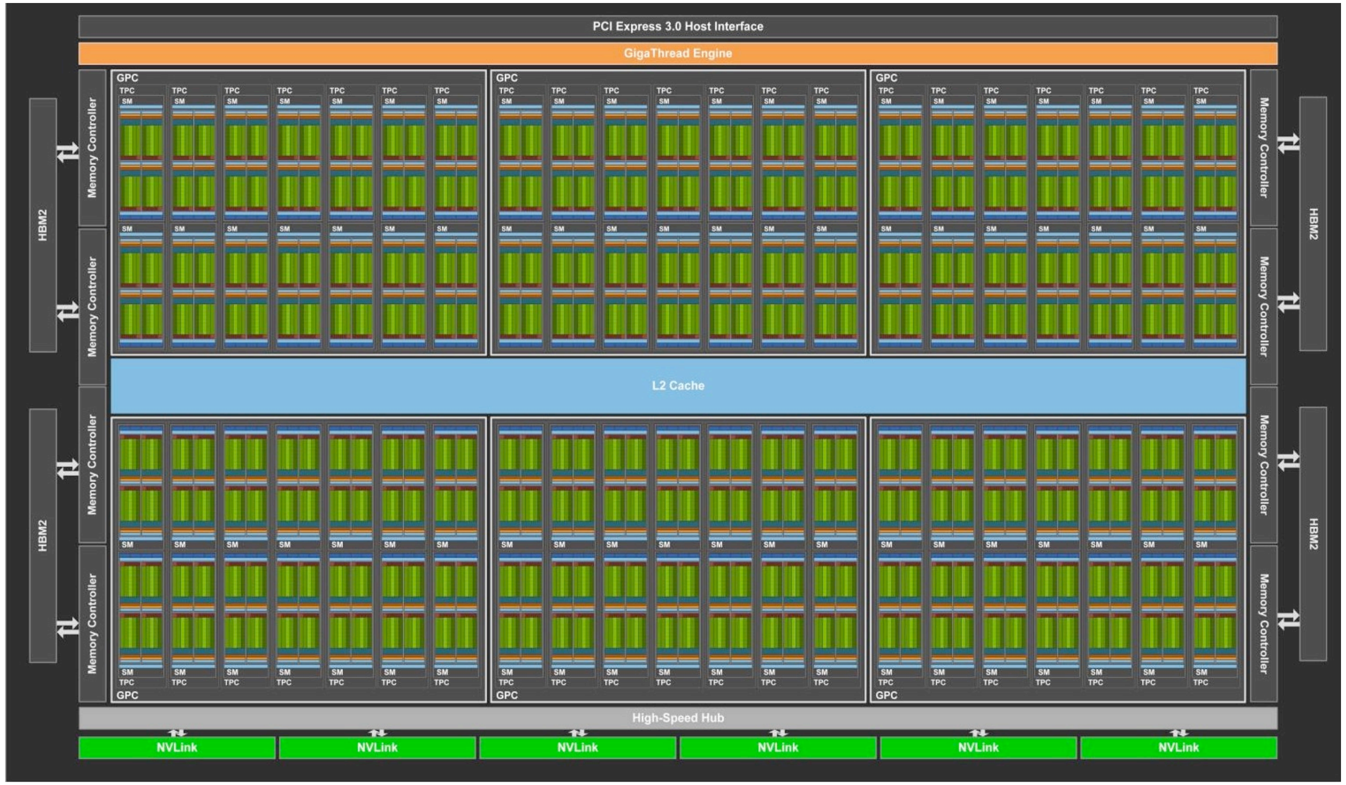
Cornell University
Center for Advanced Computing

https://docs.nvidia.com/cuda/archive/9.1/pdf/CUDA_C_Programming_Guide.pdf

General Purpose GPUs

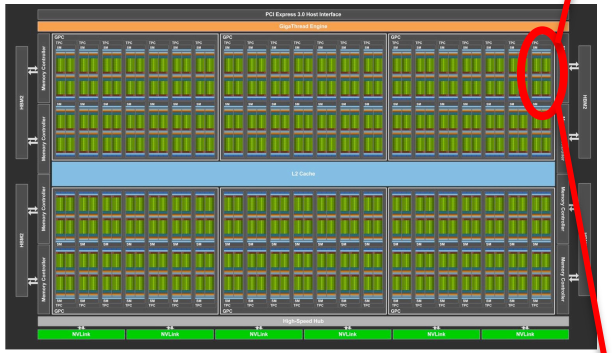
- General Purpose GPU (GPGPU)
  - NVIDIA Pascal/Volta/Ampere…
  - AMD Radeon Instinct, others
- Always attached to a CPU host
  - Data travels to device over PCIe
  - NVIDIA NVLink has higher speed
- Not compatible with x86
- Programmable sort-of-like CPUs
  - NVIDIA CUDA: extensions to C++
  - OpenCL, OpenACC, OpenMP, …
  - Need compatible compiler, libraries

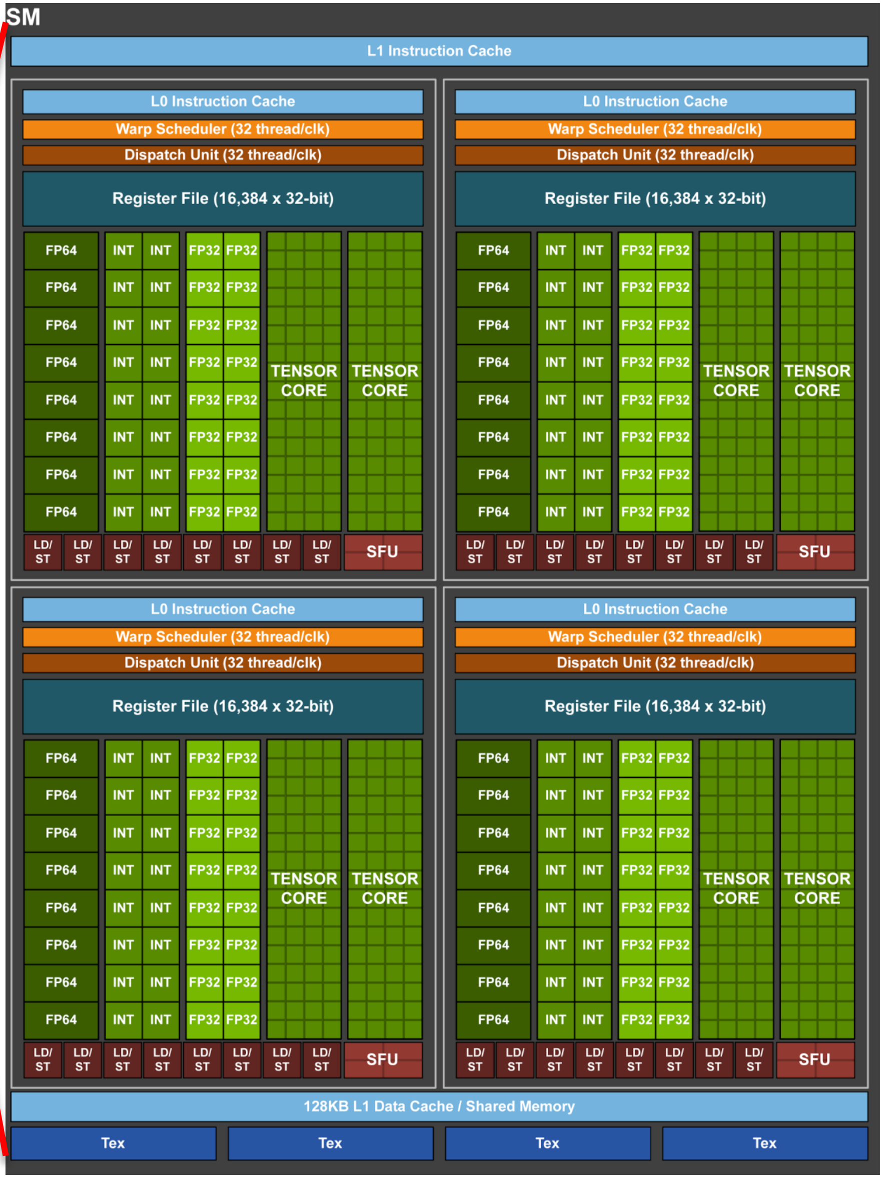
NVIDIA Volta V100



NVIDIA Tesla V100 (V for “Volta”)

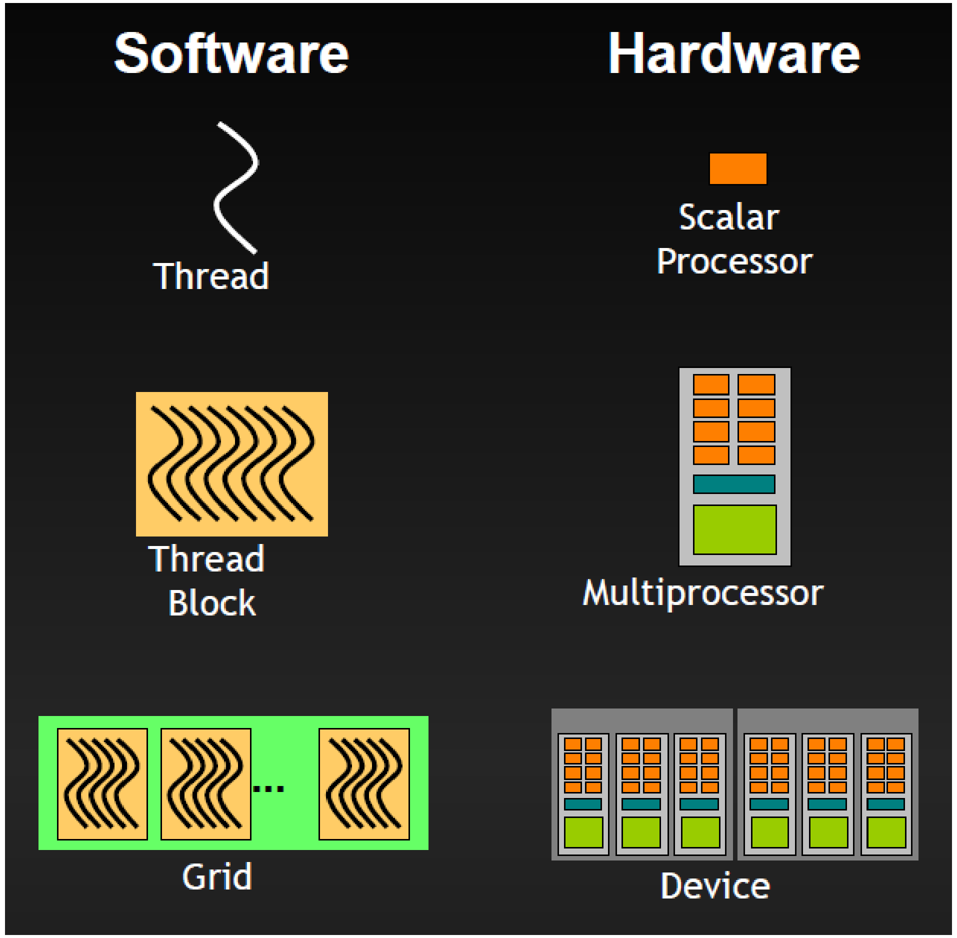
- 80 Streaming Multiprocessors (SMs)
  - Analogous to CPU cores
  - Execute *only* SIMD-type instructions
- One SM holds many “CUDA cores”
  - Analogous to vector lanes
  - 64 FP32 CUDA cores
  - 64 INT32 CUDA cores
  - 32 FP64 CUDA cores
- One SM also has 8 tensor cores
  - Fast 4x4 matrix multiplications in FP16
- Device memory: 32 GB of HBM2




SM
L1 Instruction Cache
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
FP64
INT
FP32
TENSOR CORE
LD/ ST
SFU
128KB L1 Data Cache / Shared Memory
Tex

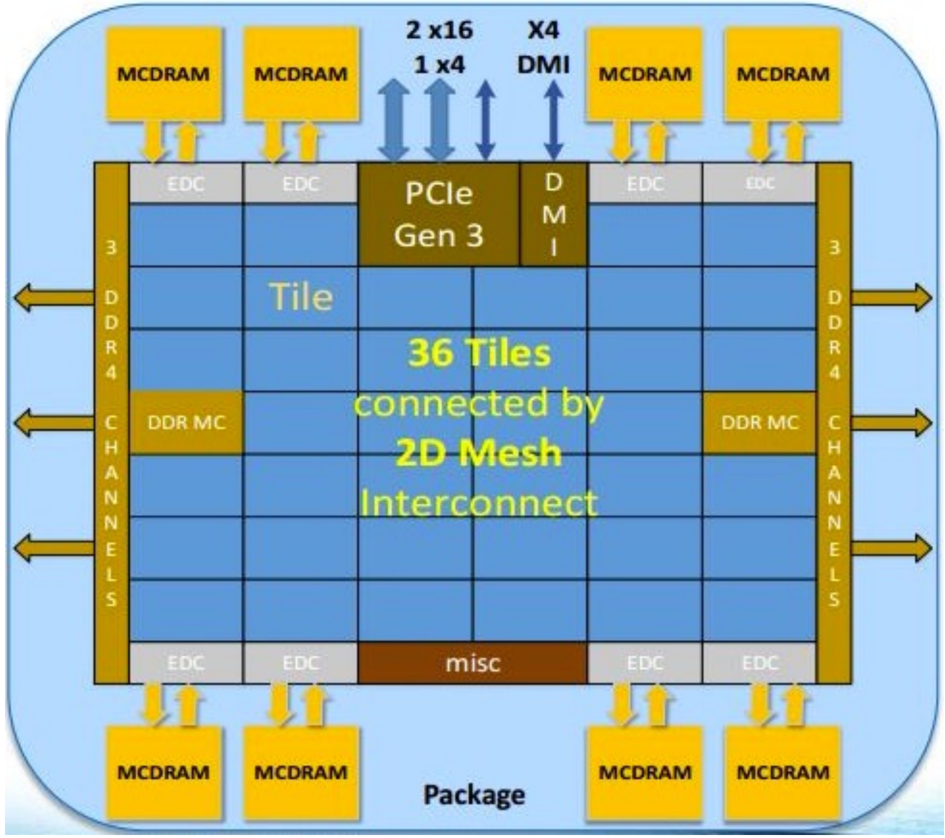
NVIDIA Execution Model

- SIMT = Single Instruction Multiple Thread
  - Operations apply to “warps” of 32 threads
  - A warp of threads is like a CPU vector
  - Not all that different from SIMD
- CUDA core = scalar (thread) processor
  - Instructions only go to groups of 32
  - A thread block must be split into warps of 32
  - Thus, an SM is like 1 or 2 vector units in a CPU
- One thread block maps to one SM
- A grid of thread blocks can be spread over the entire device



R.I.P. Intel Xeon Phi, a.k.a. Intel MIC

- MIC: Many Integrated Cores
  - x86-compatible multiprocessor architecture
- Programmable just CPUs
  - MPI, OpenMP, OpenCL, …
  - Same compilers as CPUs
- But many cores, fast MCDRAM like GPUs
  - Intel's "answer" to the GPU challenge
- Started out as a coprocessor
  - Data travels over PCIe
  - Became main processor in 2nd generation
- Eventually merged into Xeon line



Processors vs. PCIe Devices, Circa 2019

M. Lefebvre

	Xeon Gold 5122 (2017)	Xeon Platinum 8280M (2019)	Xeon Phi 7290F (2017)	NVIDIA V100 (2017)
Cores	4	28	72	84 SMs
Logical Cores	8	56	288	5120 cores
Clock rate	3.6 – 3.7 GHz	2.5 – 3.8 GHz	1.5-1.7 GHz	1530 MHz
Theoretical GFLOPS (double)	460.8	2240	3456	7450
SIMD width	512 bit	512 bit	512 bit	Warp of 32 threads
Memory	--	--	16 GB MCDRAM 384 GB DDR4	32 GB
Memory B/W	127.8 GB/s	127.8 GB/s	400+ GB/s MCDRAM 115.2 GB/s RAM	900 GB/s
Approx. Unit Price	$1,200	$12,000	$3,400	$9,000

Demystifying the Cloud

- “Regular” computers, just somewhere else
- Provide users with remote virtual machines or containers
- Can be used for anything:
  - Mobile services, website hosting, business applications, …
  - **Data analysis, high performance computing**
- Providers
  - Amazon Web Services (AWS)
  - Microsoft Azure
  - Google Cloud Platform (GCP)
  - Oracle Cloud
  - Lots of others

Cloud Computing Pros and Cons

- Advantages:
  - Potentially lower cost
    - Pay as you go
    - Save on sysadmins and infrastructure
    - Economy of scale
  - Scaling up or down as needed
    - Can be used to run overflow from a regular data center
  - Access to a wide range of hardware
- Challenges:
  - Data movement
    - Expensive and time consuming
  - Security, privacy, …

What We Didn’t Talk About

- Microarchitecture details (ALU, FPU, pipelining…)
- Memory bandwidth and latency
- Cache lines and cache coherence
- IBM (Power) or AMD processors
- FPGAs, Google TPUs, NPUs, …

Resources & References

- Very nice glossary: https://cvw.cac.cornell.edu/main/glossary
- J. Hennessy, D. Patterson, Computer Architecture: A Quantitative Approach, 6th edition (2017), ISBN 978-0128119051
- U. Drepper, What Every Programmer Should Know About Memory, http://people.redhat.com/drepper/cpumemory.pdf
- NVIDIA Volta Architecture Whitepaper https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf