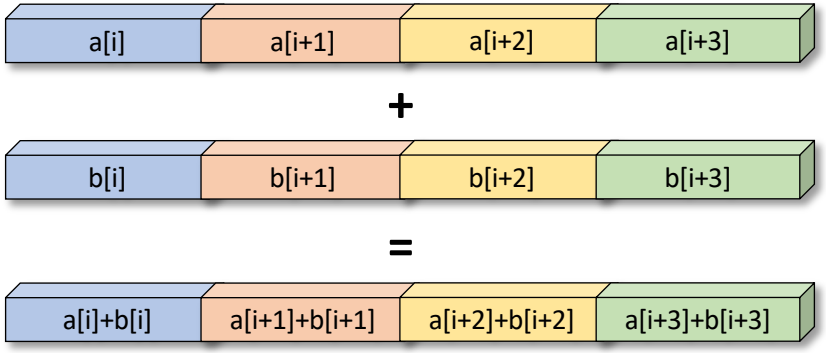
# What Every Computational Physicist Should Know About Computer Architecture*




*Plus some other useful technology

Steve Lantz, Cornell University

*CoDaS-HEP Summer School, July 22, 2024*

Cornell University
Center for Advanced Computing

*with thanks to Ian Cosden, Princeton Univ.*

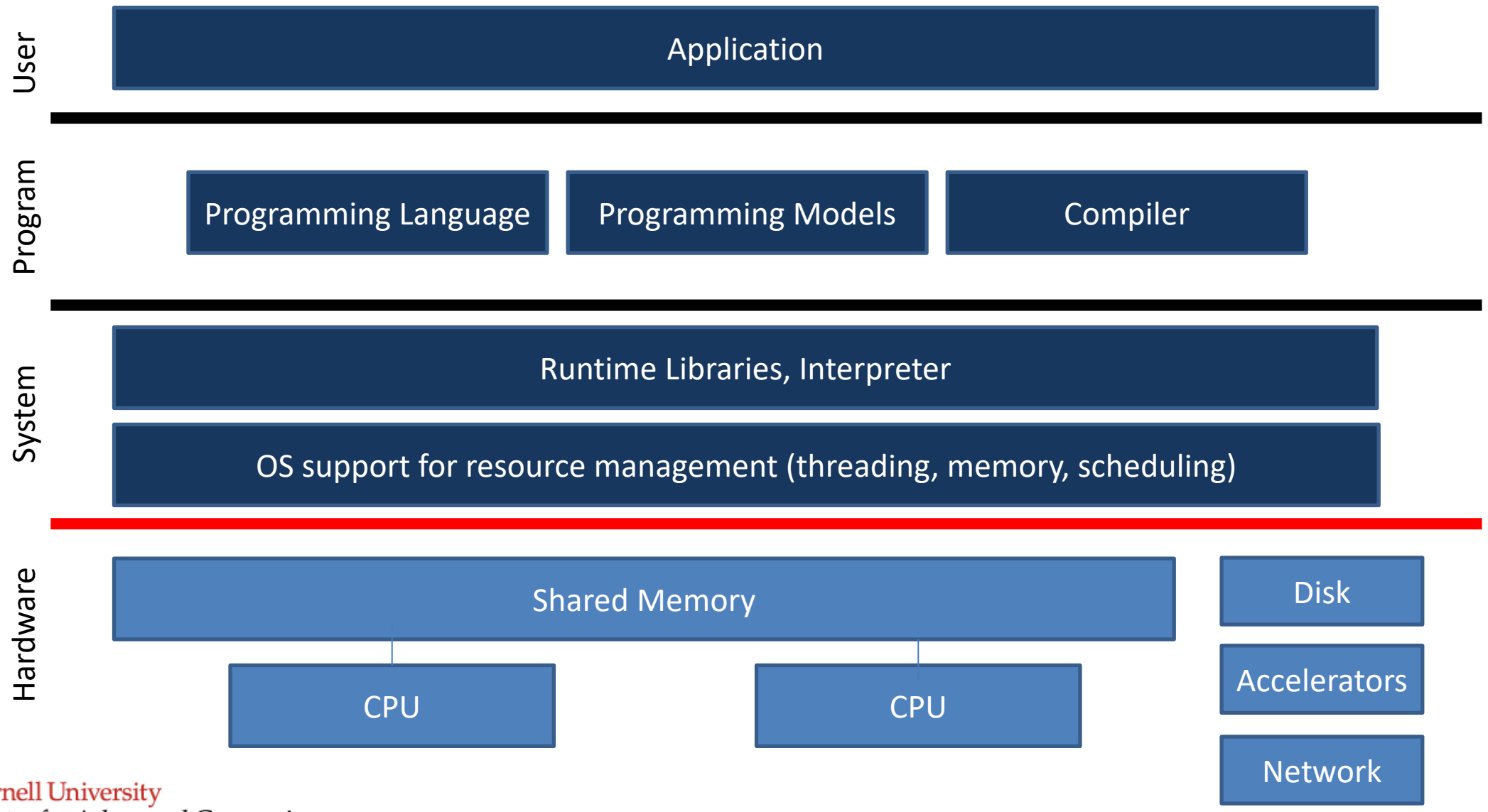
# Outline

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

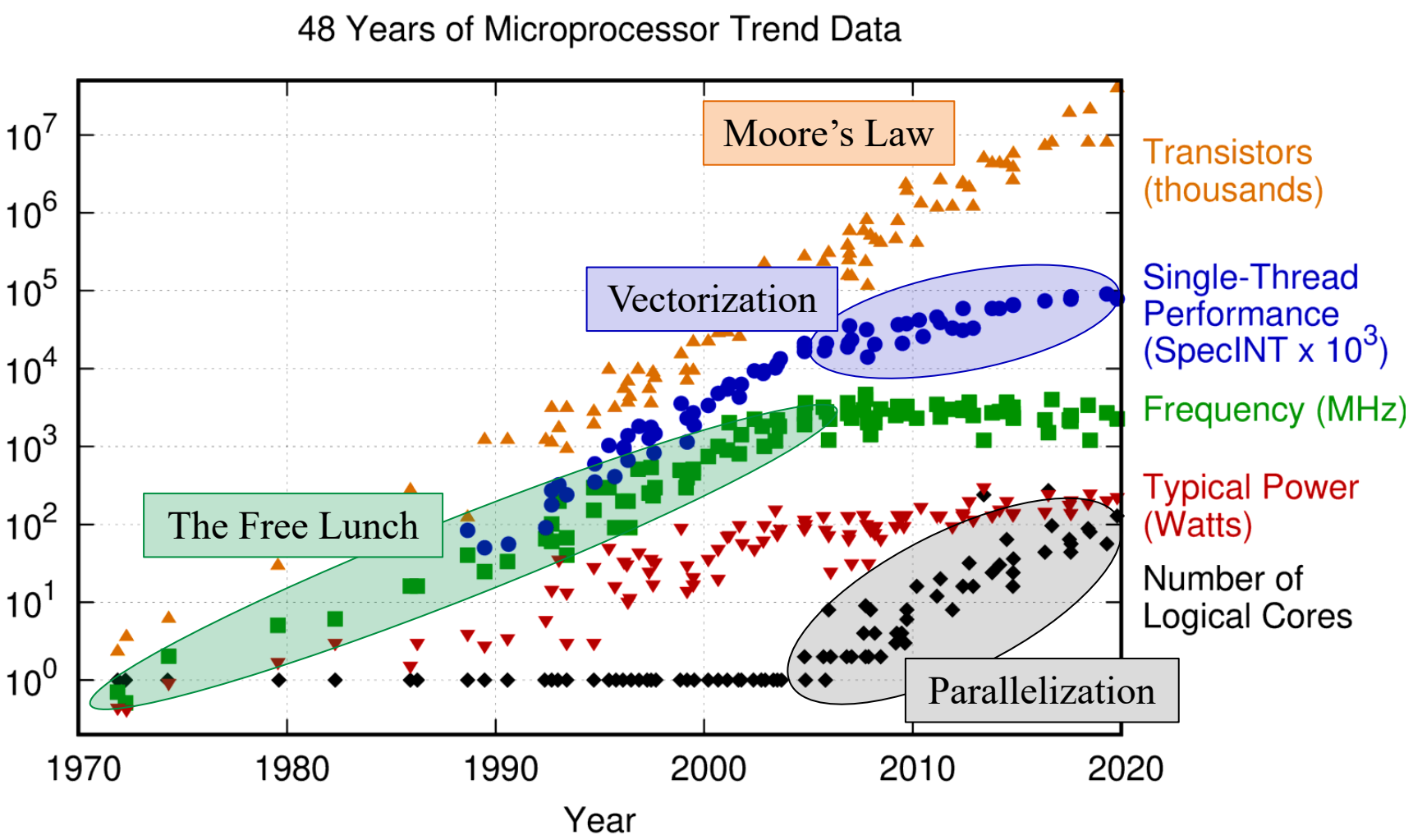
# Why Care About the Hardware?

- 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



# Intra-Processor Parallelism: The Free Lunch Is Over…

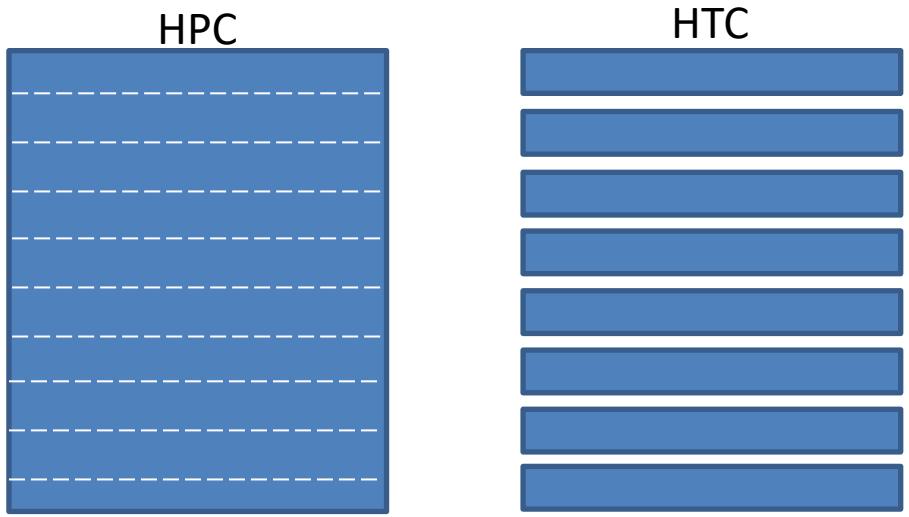



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

https://github.com/karlrupp/microprocessor-trend-data

# 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

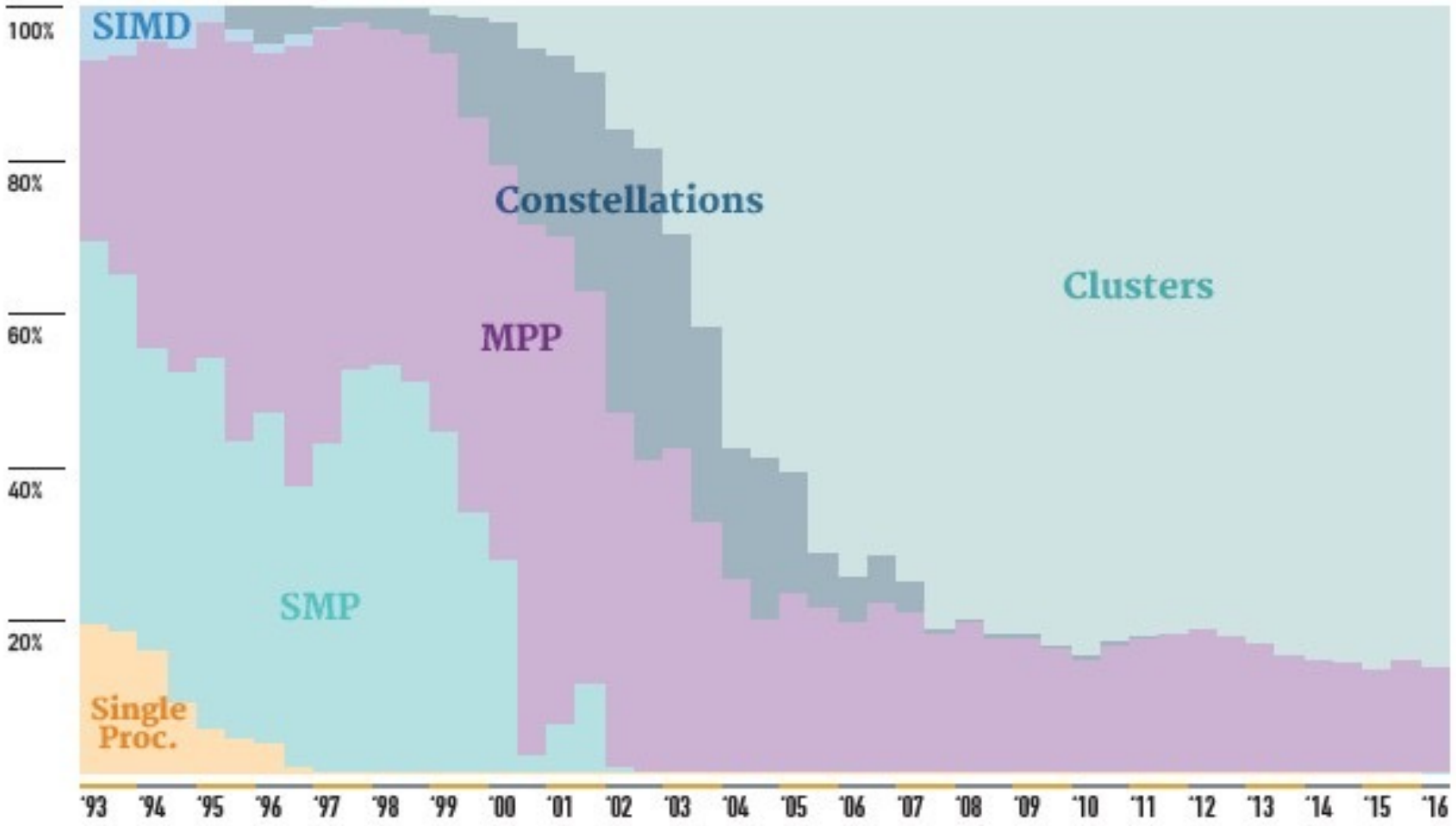


# Outline

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

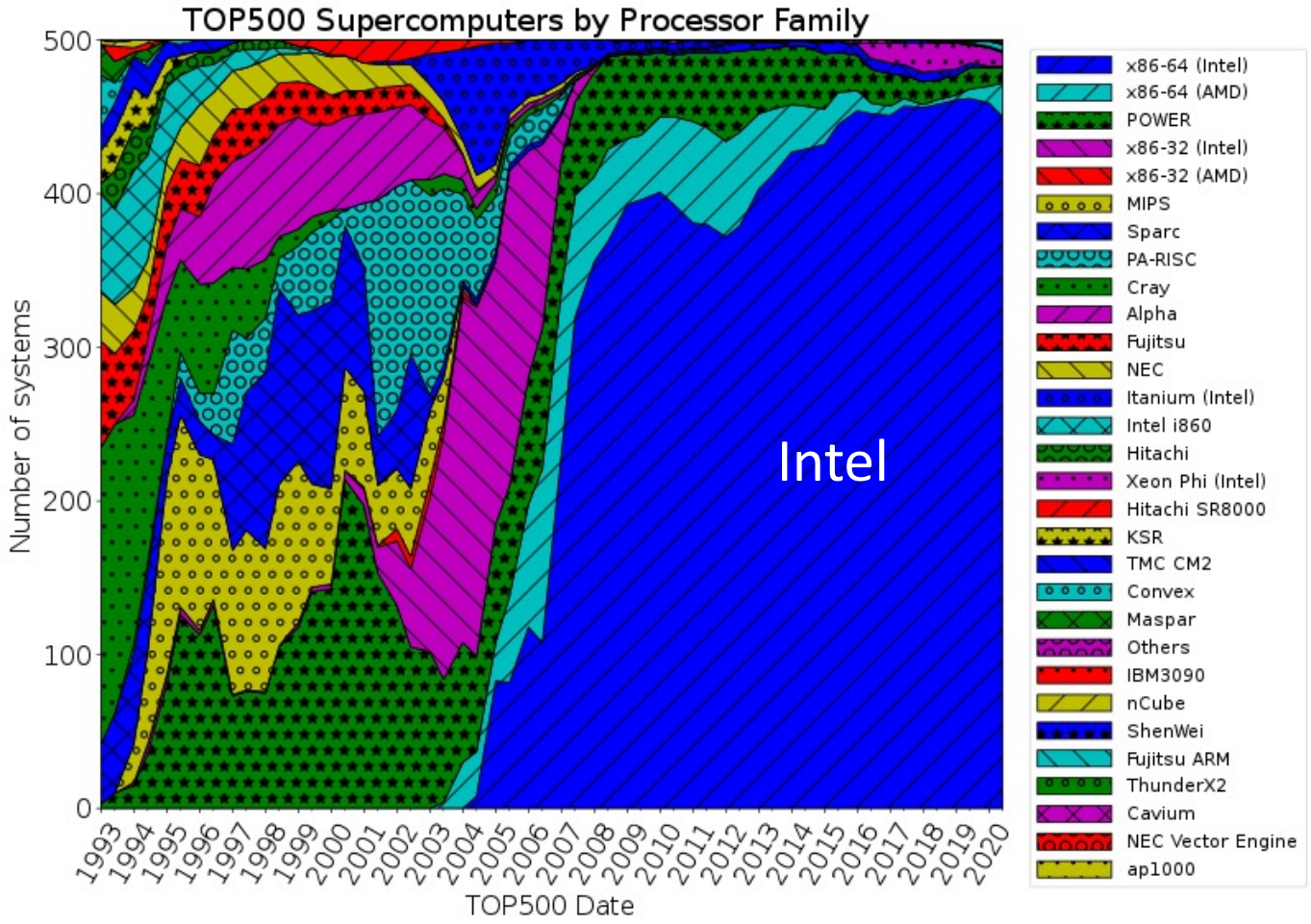
# Top 500 Supercomputers: Nearly All Clusters

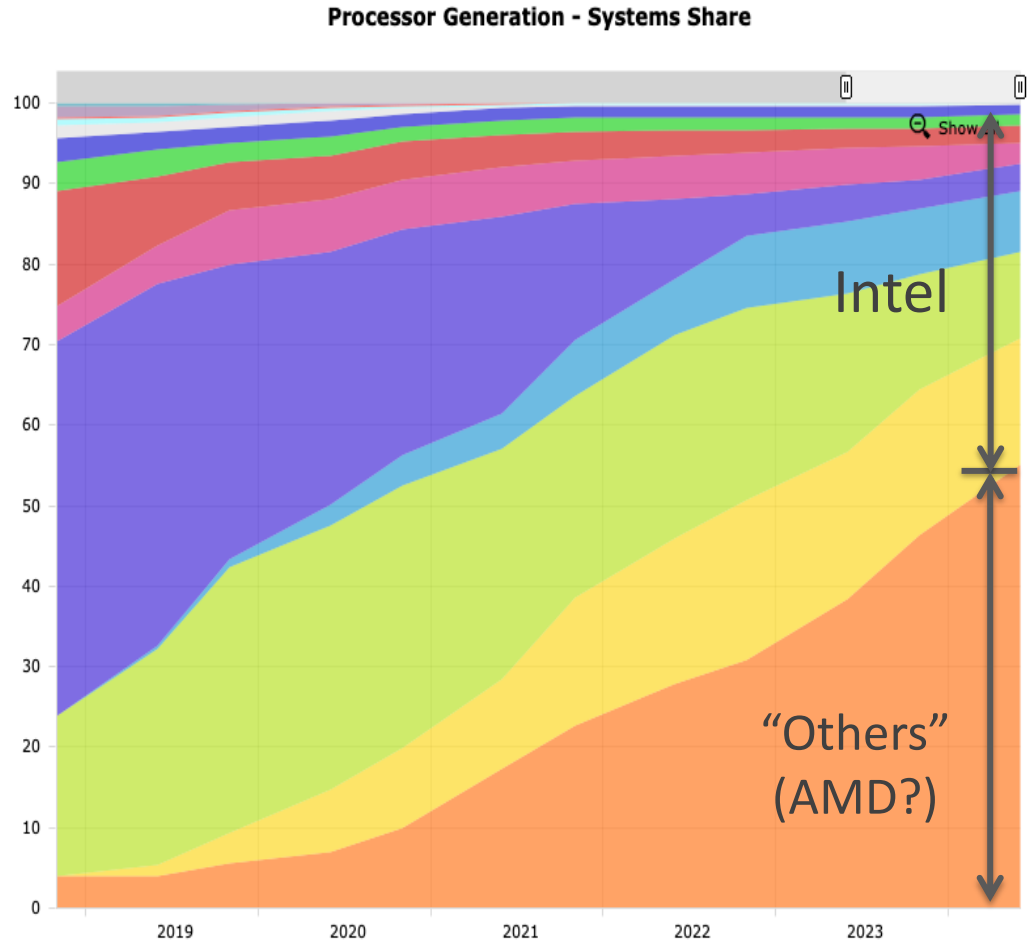
https://www.top500.org




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

# The Increasing Role of x86-64 Architecture



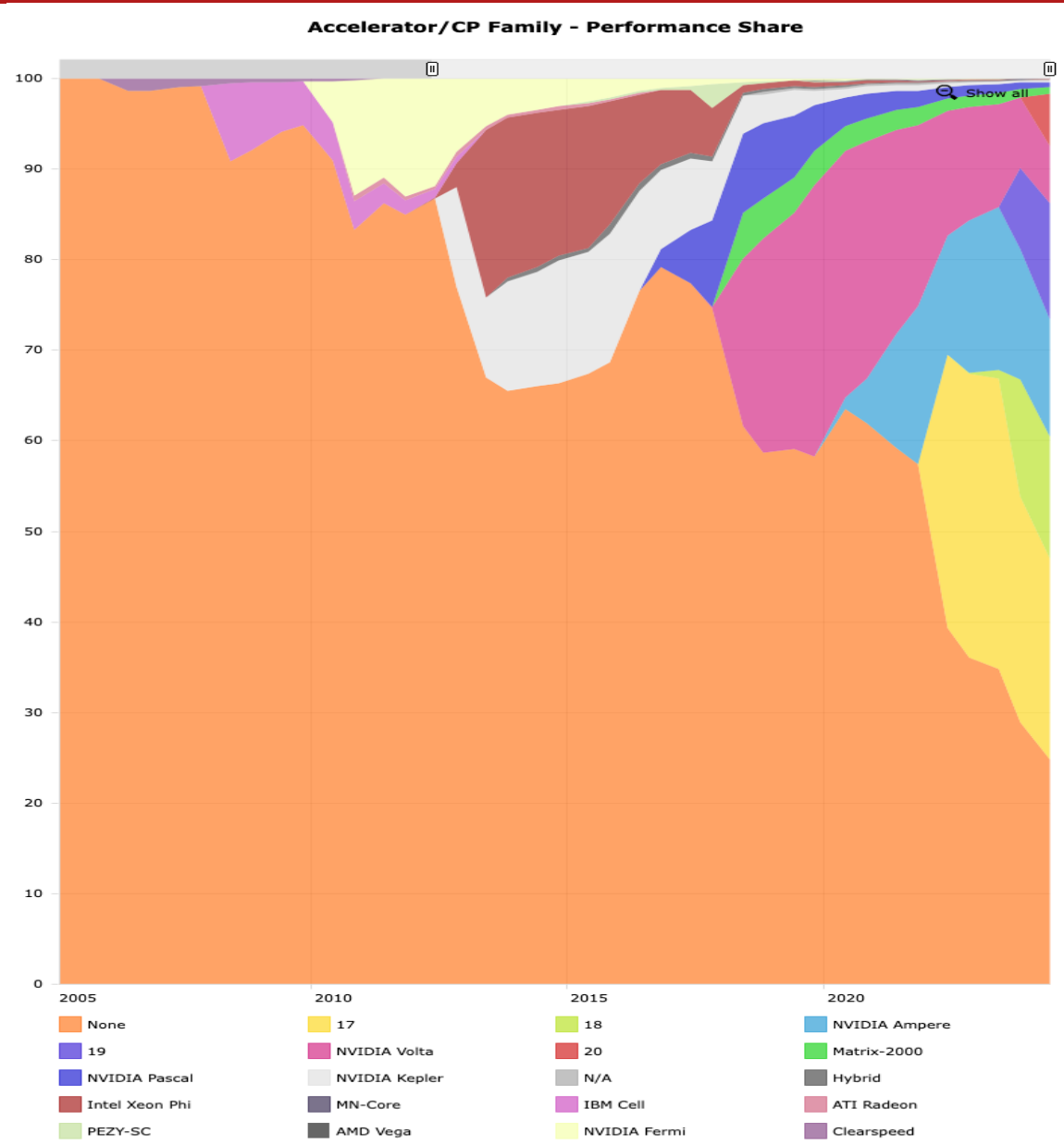



https://en.wikipedia.org/wiki/TOP500

https://www.top500.org/statistics/overtime/

# Heterogeneous Systems on the Rise

- The computing landscape changes rapidly
- 75% of TOP500 systems have accelerators as of June 2024
- Graphics processing units (GPUs) are by far the most common type of accelerator




https://www.top500.org/statistics

# Why Linux Dominates HPC

- Free and open-source OS
  - The largest open-source software project in the world! (according to Red Hat)
  - Therefore, a natural companion for open-source software by and for researchers
- Well established
  - Has its roots in Unix (1969); opened up by Linus Torvalds' "Linux" kernel (1991)
  - Relies heavily on Richard Stallman's "GNU's Not Unix" = GNU Project (1983)
- Powerful, yet compact
  - Linux runs all the TOP500 supercomputers, all the major stock exchanges…
  - Yet, it's small (fast!) enough to embed in thermostats, refrigerators, televisions…
- Perhaps the biggest reason: clusters need a network!
  - Unix was the first OS to include TCP/IP; today, Linux basically runs the Internet

# 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

- Note, cluster computing is not an emphasis of CoDaS-HEP…
  - MPI (Message Passing Interface) is the most common API for inter-node communication
  - At CoDaS-HEP, OpenMP and multithreading will be used to illustrate parallel methods

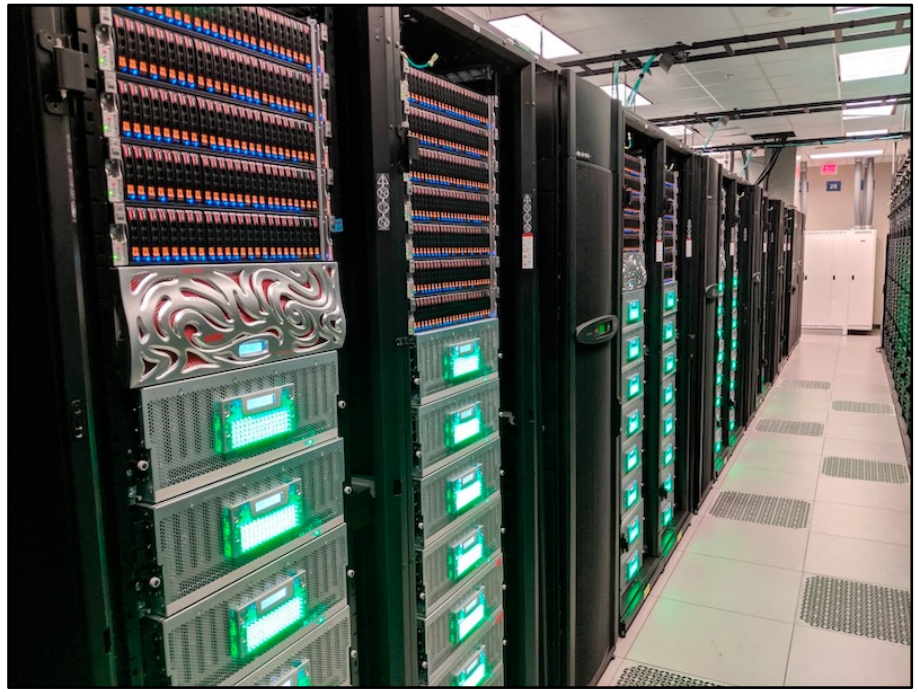
# Parallel Computers

HPC cluster: a collection of servers connected to form a single entity

- Each node 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

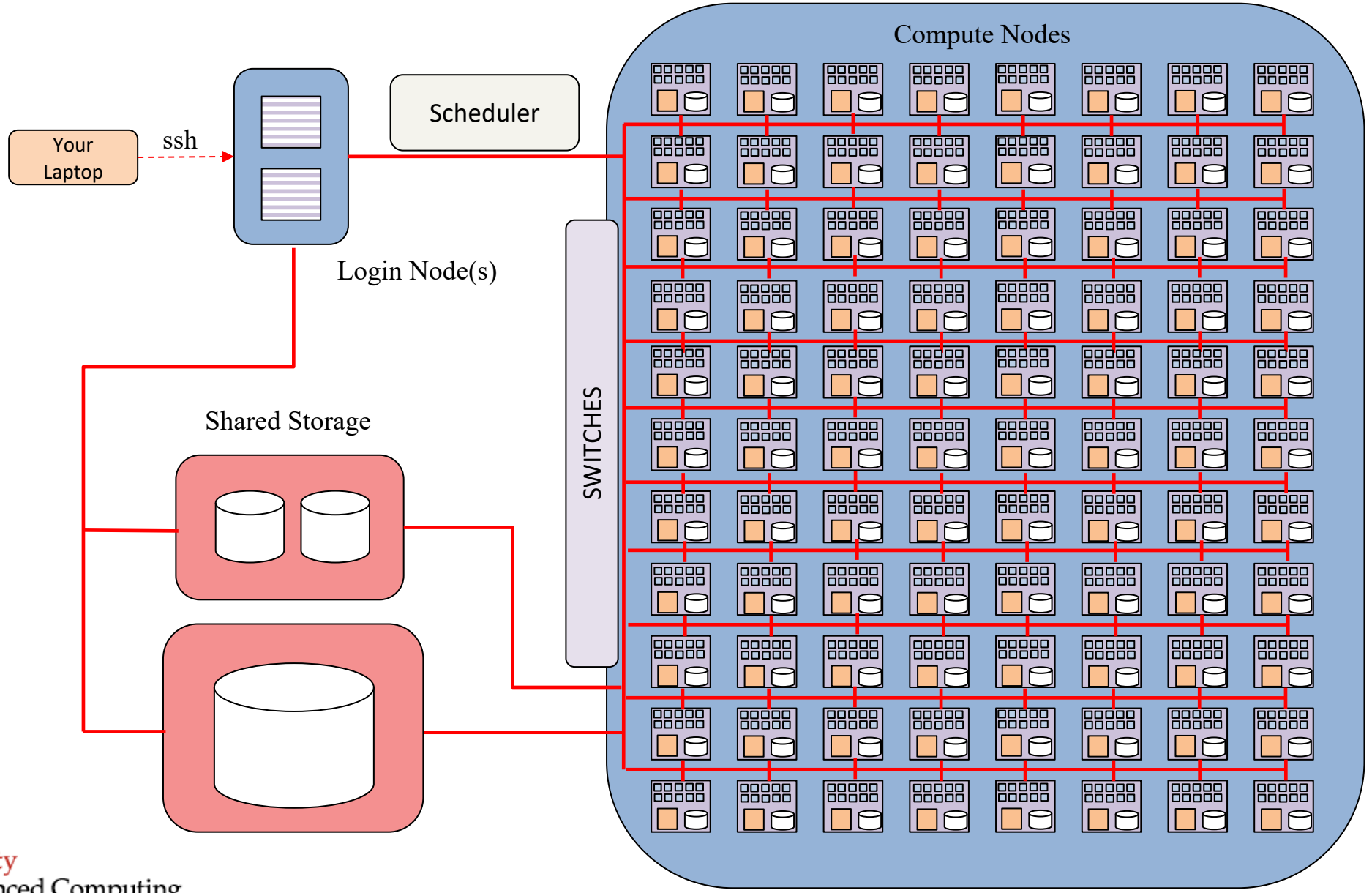
SMP System: Symmetric Multi-Processing

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



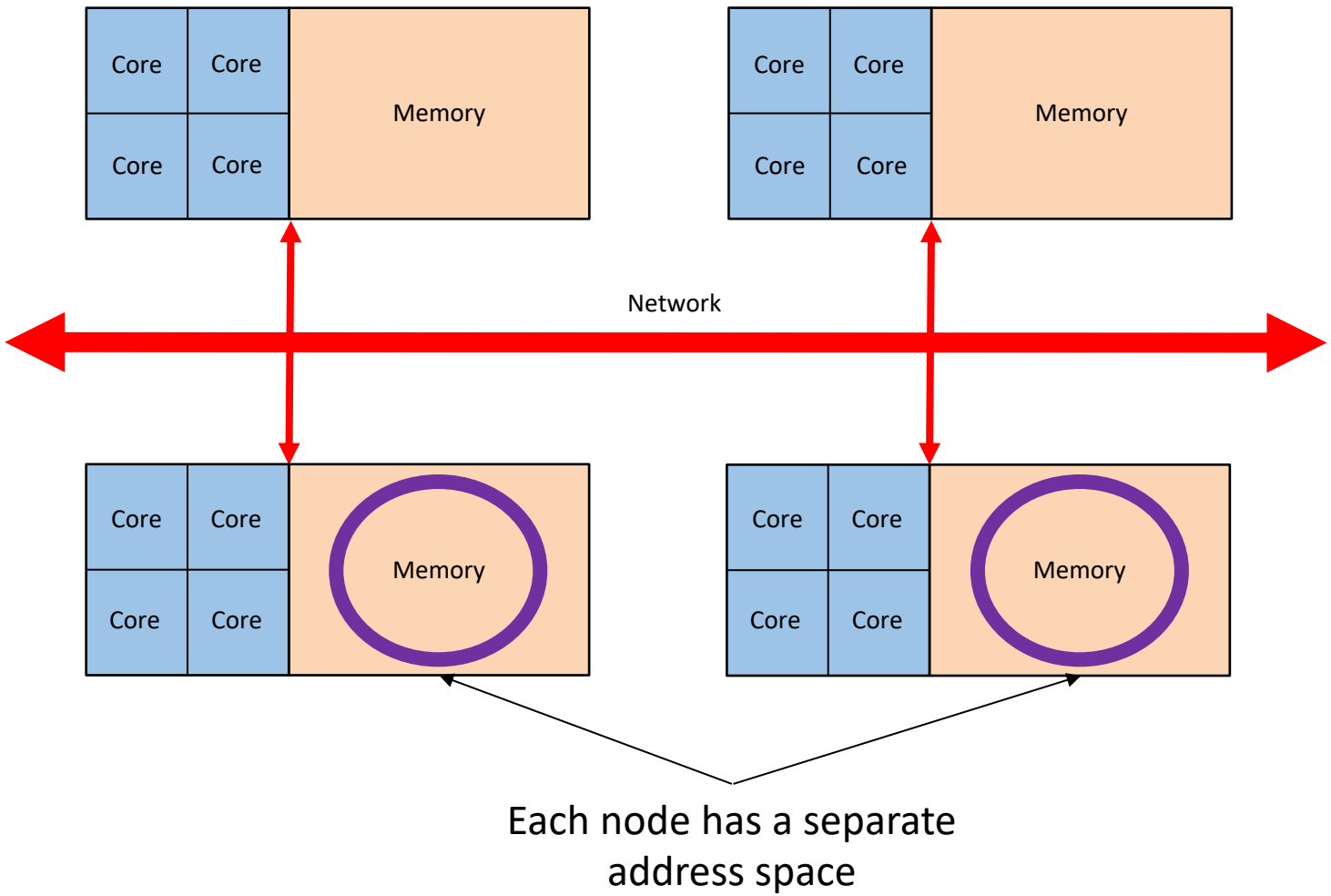
TACC's Frontera Cluster

# Basic Cluster Layout

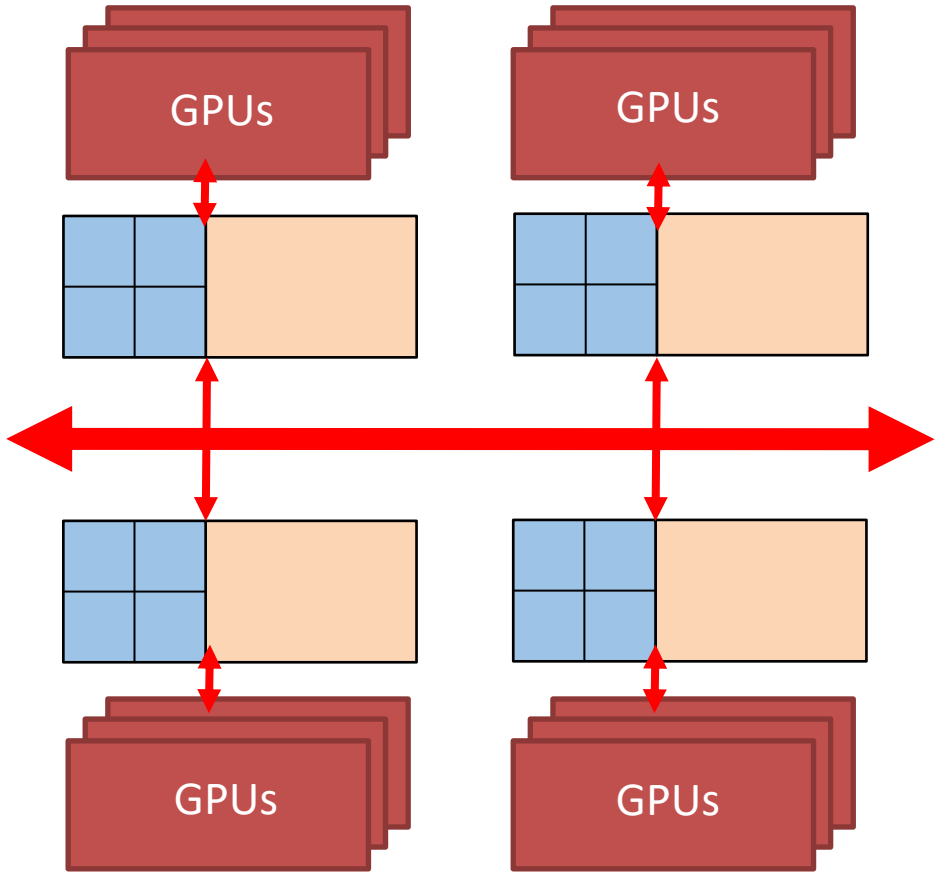


# Nodes Are Discrete Systems

- Memory address space
  - Range of unique addresses
- Separate for each node
- Can't just access all memory in cluster
  - Cores within a node can access all memory on that node



# Clusters With Accelerators



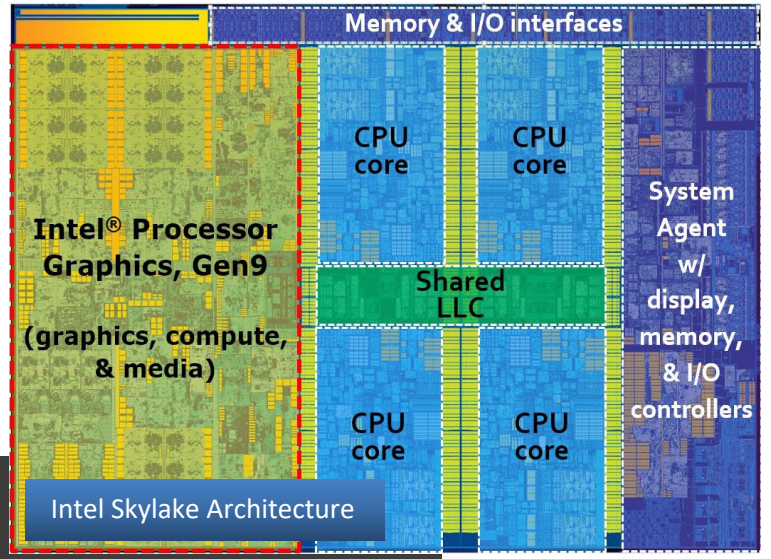

- "Heterogeneous" system architecture
- Accelerators: GPUs, typically
  - Or FPGAs, etc.
  - Node can have multiple accelerators
- Programmable with MPI + x
  - Where x = OpenMP, OpenACC, CUDA, …
- Increases computational power
  - Increases FLOPS / Watt
- Trending in Top500.org
  - Strong shift toward systems with accelerators started ~15 years ago

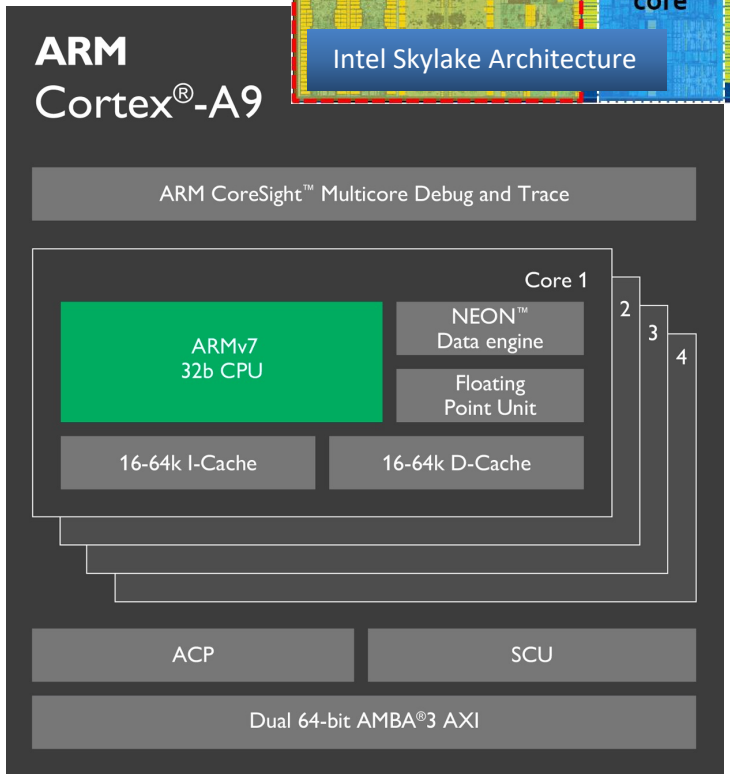
# Outline

- Motivation
- HPC Cluster
- Compute
  - Multicore
  - Microarchitecture
  - Vector Processing
- Memory
- Disk
- Other Architectures

# Multicore Processors

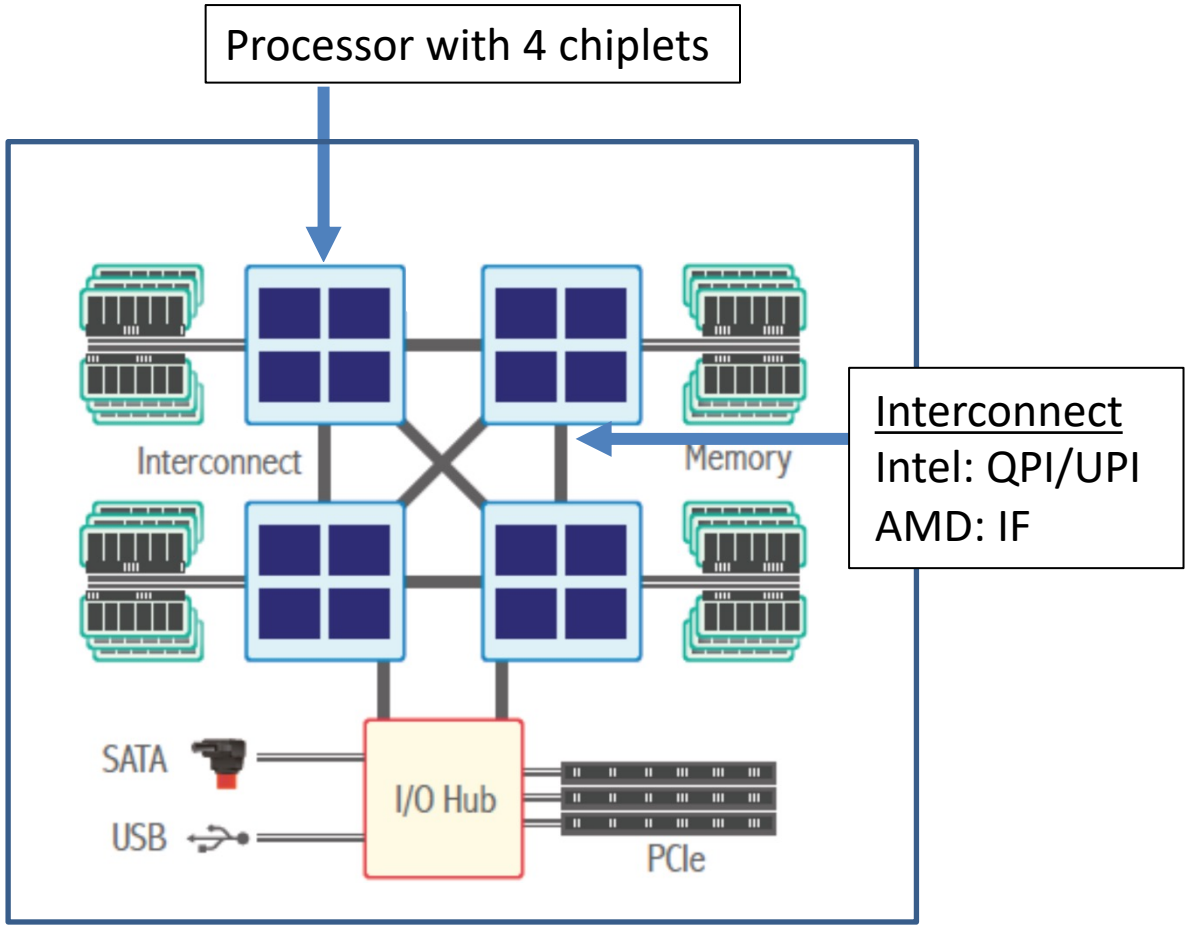
- Nearly all modern processors are multicores
  - In servers: Intel Xeon, AMD EPYC, IBM Power
  - In laptops: Intel, AMD, ARM (Apple M1, M2, M3)
  - New P and E core types, performance vs. efficiency
  - In GPUs, phones, and mobile devices as well
- Most have vector units, multiple cache levels
- Require special care to program
  - Multi-threading, vectorization, …
- Instruction sets can vary
  - Intel and AMD: x86_64 base, not identical
  - IBM Power, ARM, NVIDIA, etc. all differ



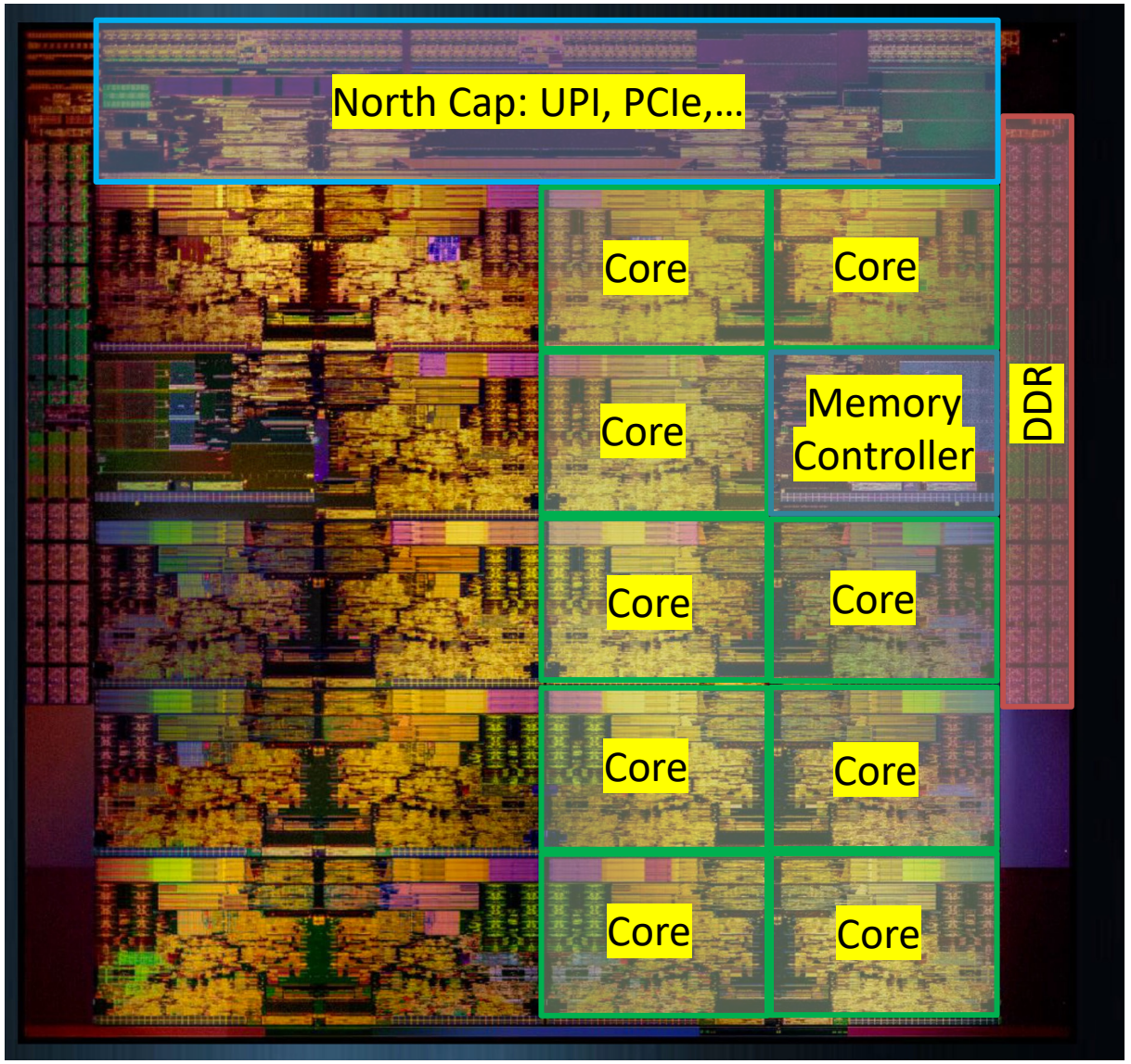



# Motherboard Layout

- 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 has multiple cores (on chiplets)
  - CPU can refer to a core... or a processor
- Processors connected via interconnect
- Memory attached to processor/socket
  - Memory address space spans entire node
  - But: nonuniform memory access (NUMA)

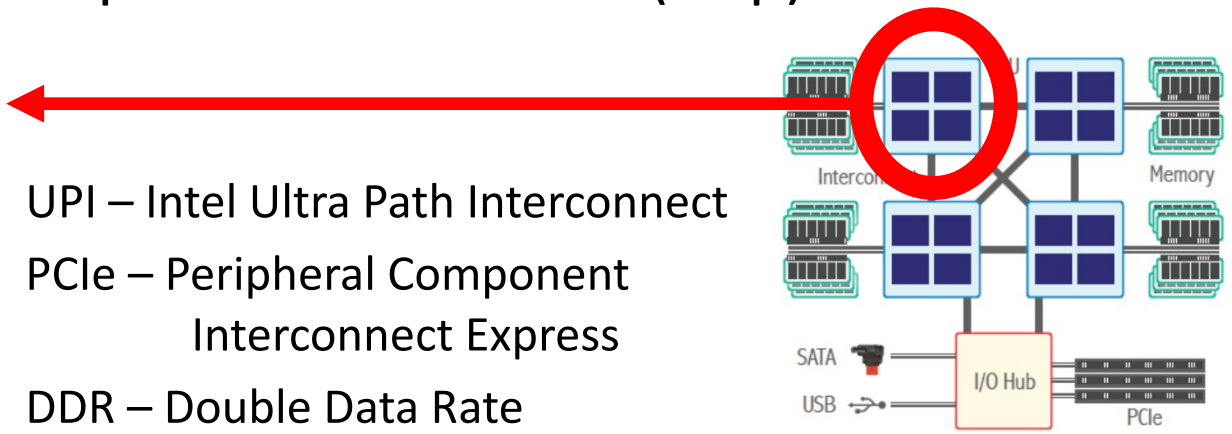


# Example: Intel Skylake Scalable Processor




"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

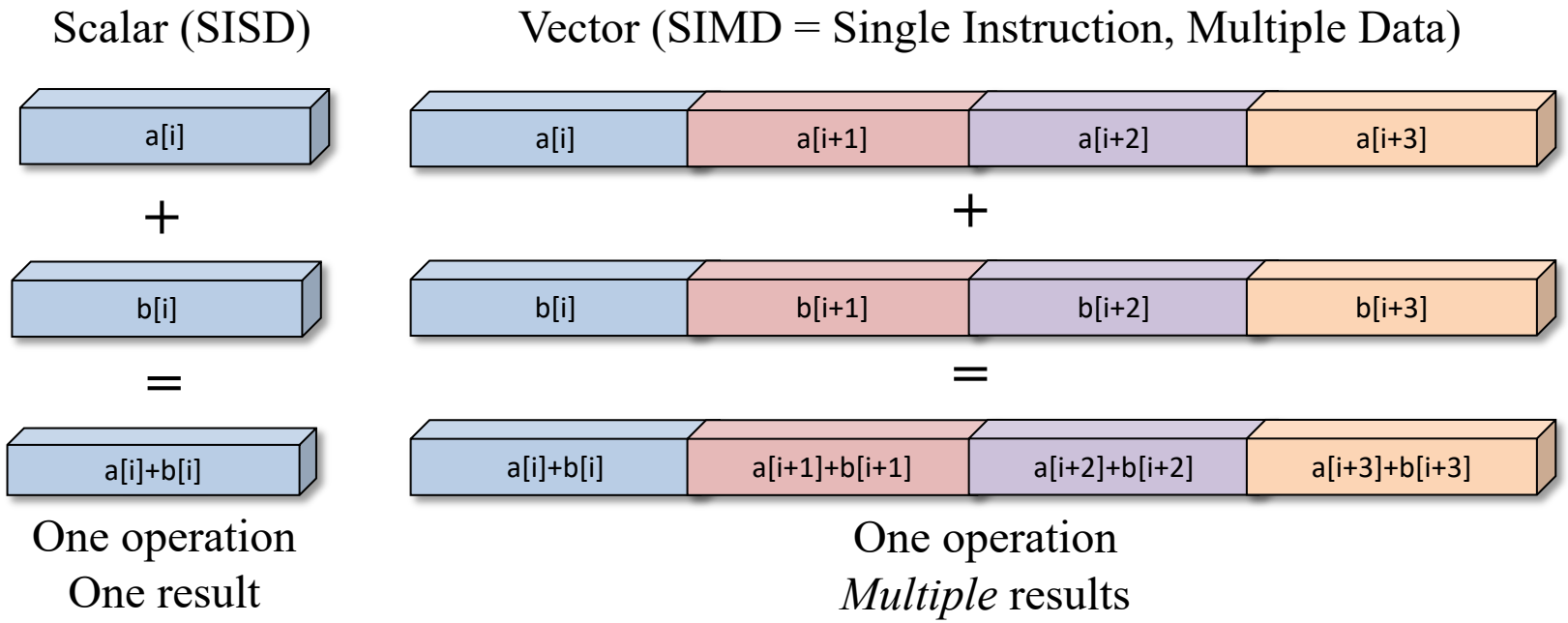
- ALU (arithmetic and logic unit)
- FPU (floating point unit)

- Data Transfer
- Load / Store

# Vectorization Explained in a Picture

At inner-most loop level

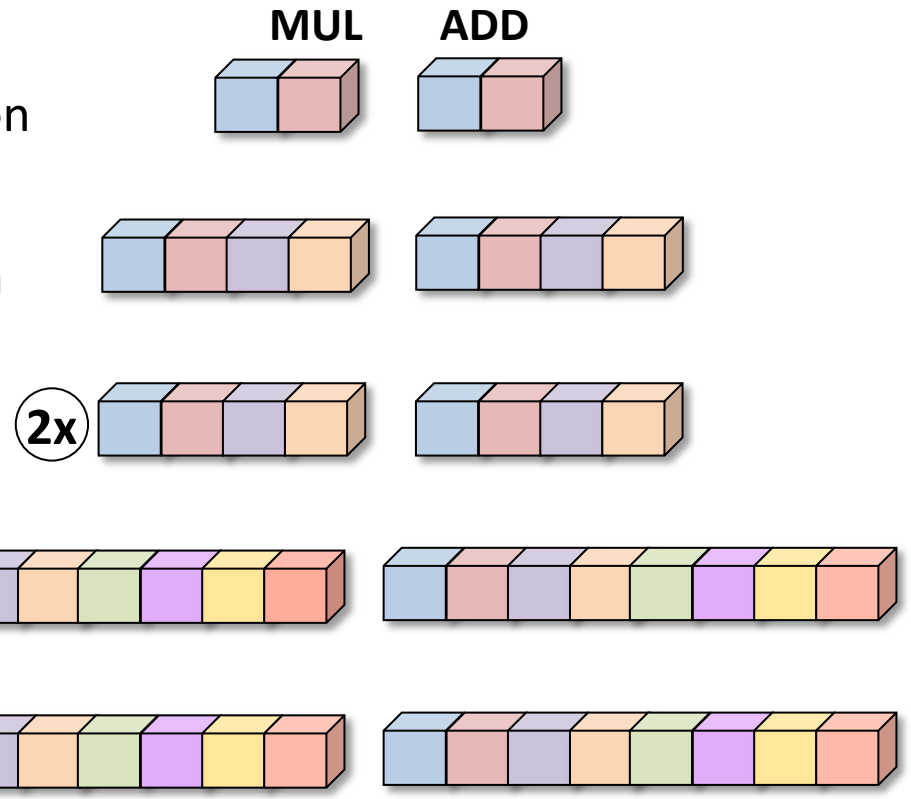
```
for (int i=0; i<N; i++) {
    c[i]=a[i]+b[i];  }
```




- A core has 1 or 2 vector processing units (VPUs), many vector registers
  - New generations of CPUs add capabilities and instructions
  - Machine (assembly) commands are added to take advantage of new hardware

# Vector Growth: Intel Xeon Codenames, Speeds per Core

- FLOPs = Floating Point Operations; DP = Double Precision = 8 bytes
  - Pentium 4… Nehalem/Westmere (SSE2… SSE4):
    - 4 DP FLOPs/cycle*: 128-bit multiplication, 128-bit addition
  - Sandy Bridge/Ivy Bridge (AVX)
    - 8 DP FLOPs/cycle: 256-bit multiplication, 256-bit addition
  - Haswell/Broadwell (AVX2)
    - 16 DP FLOPs/cycle: **two**, 256-bit **FMA**, fused mul-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**

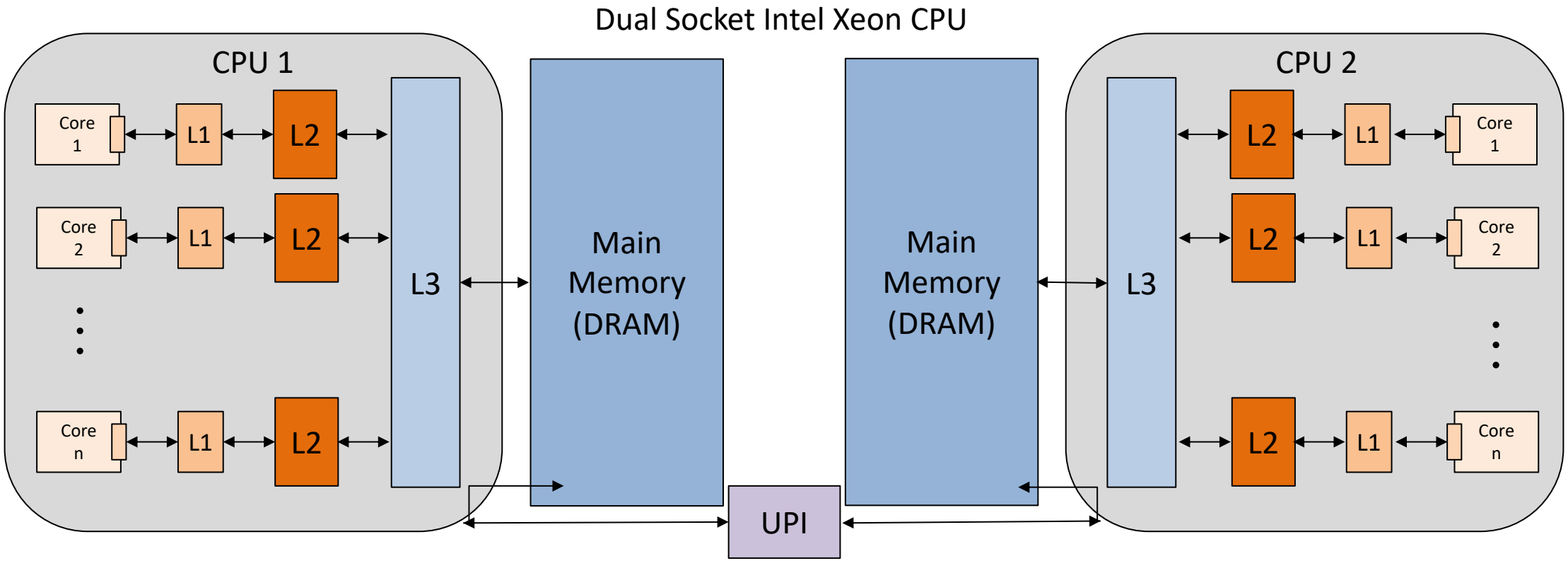



* Single Precision (SP) rate = 2 × DP; Pentium III (1999) only did 4 SP FLOPs/cycle
**Some Skylake-SPs can do only one 512-bit FMA = (a × b + c), per core

# Outline

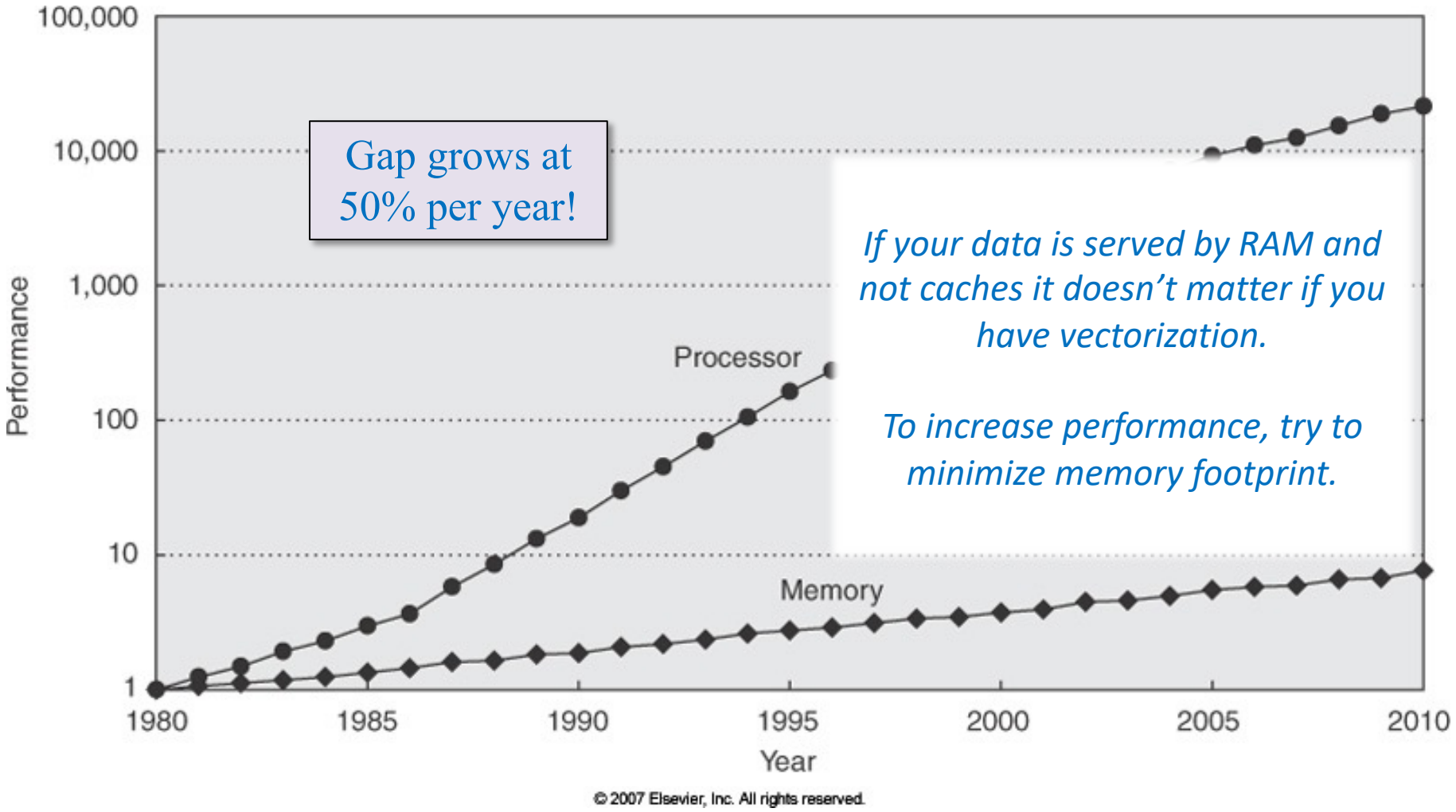
- Motivation
- HPC Cluster
- Compute
- Memory
  - Layout
  - Cache
  - Performance
- Disk
- Other Architectures

# Memory Hierarchy




| | 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 |

# Growth in Memory Performance Lags CPU




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*

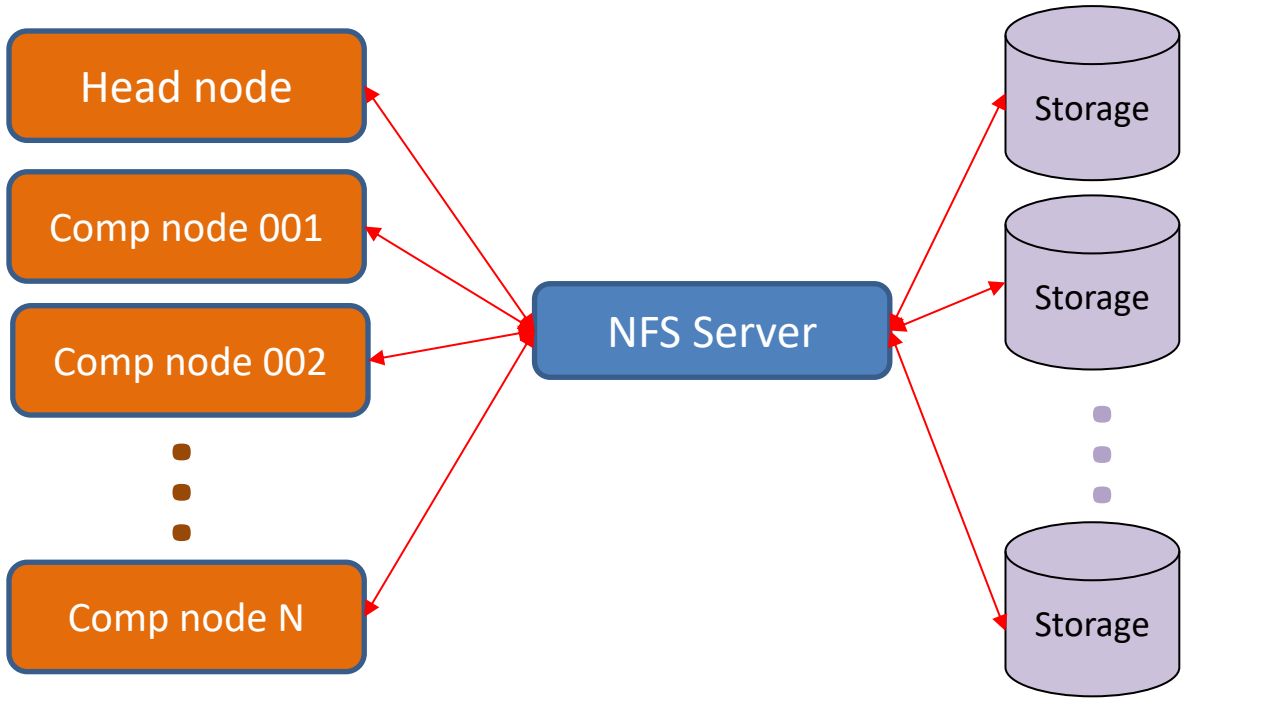
# Types of Disk

- 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

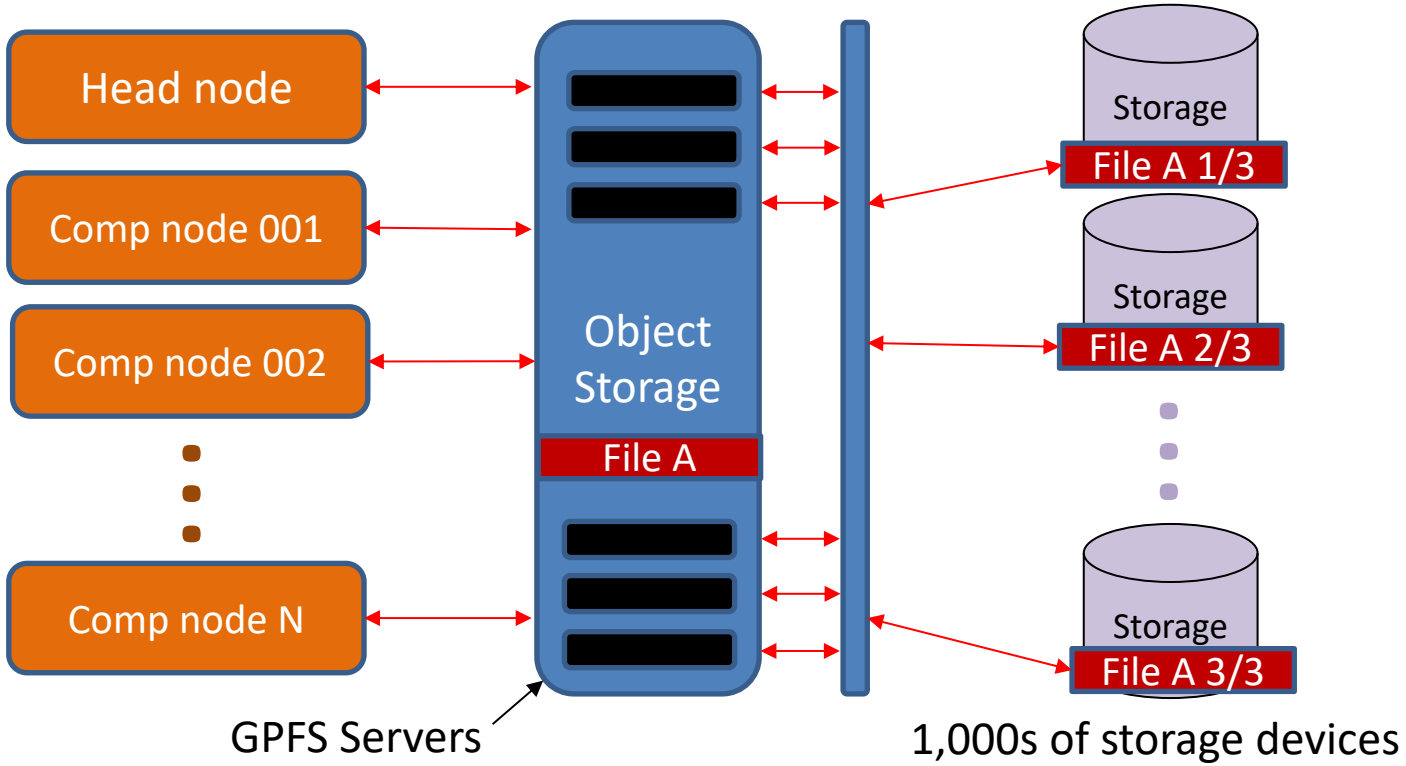
# Types of Filesystems - NFS

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



# Parallel Filesystems

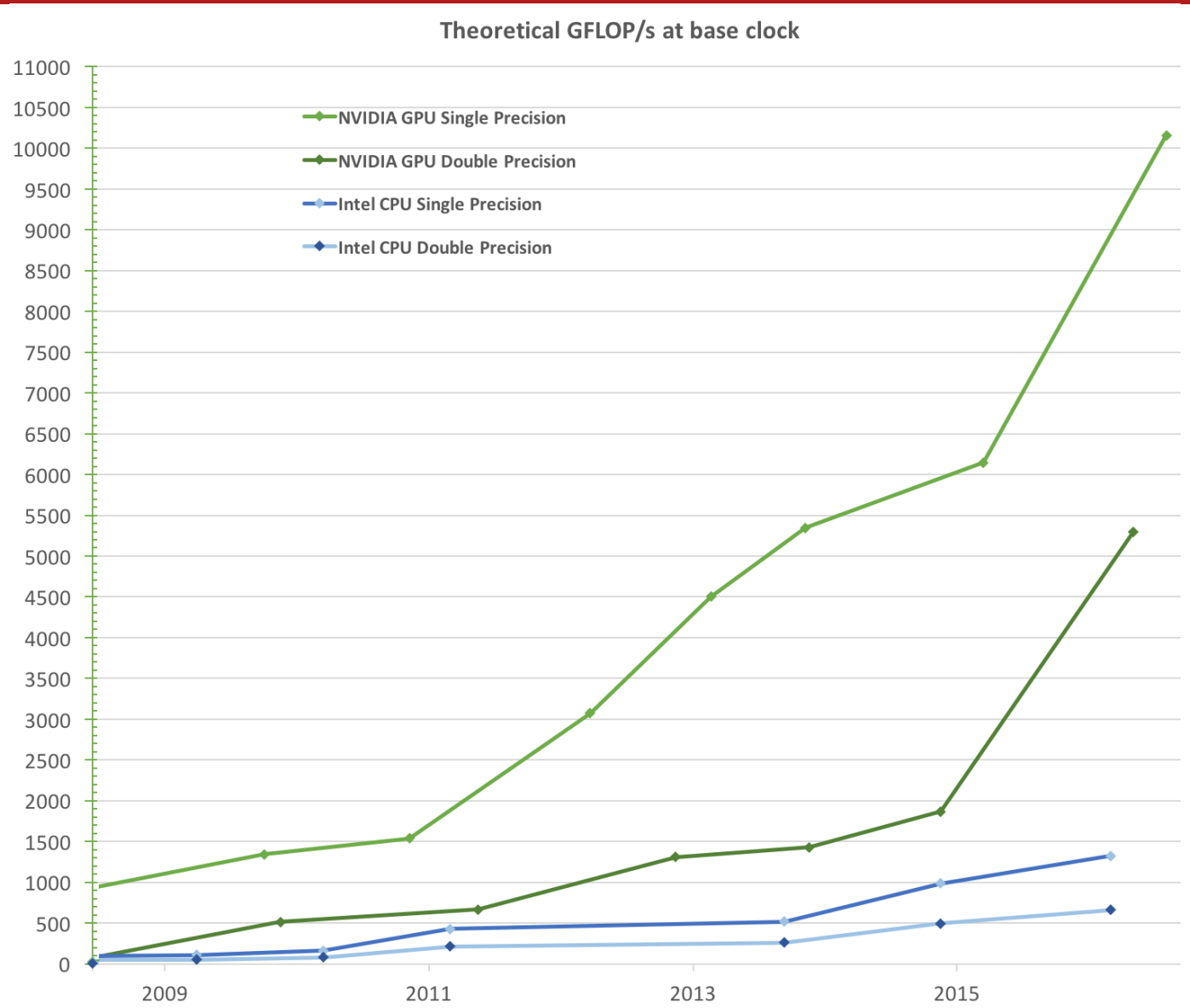
- GPFS (General Parallel Filesystem – IBM)
  - Designed for parallel read/writes
  - Large files spread over multiple storage devices
  - Allows concurrent access
  - Significantly increase throughput
- Lustre
  - Similar idea
  - Different implementation
- Parallel I/O library in software
  - Necessary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



# Outline

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

# GPU and CPU Performance Trends




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

# General Purpose GPUs

- General Purpose GPU (GPGPU)
  - NVIDIA Volta/Ampere/Hopper…
  - AMD 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++
  - SYCL, OpenACC, OpenMP, …
  - Need compatible compiler, libraries

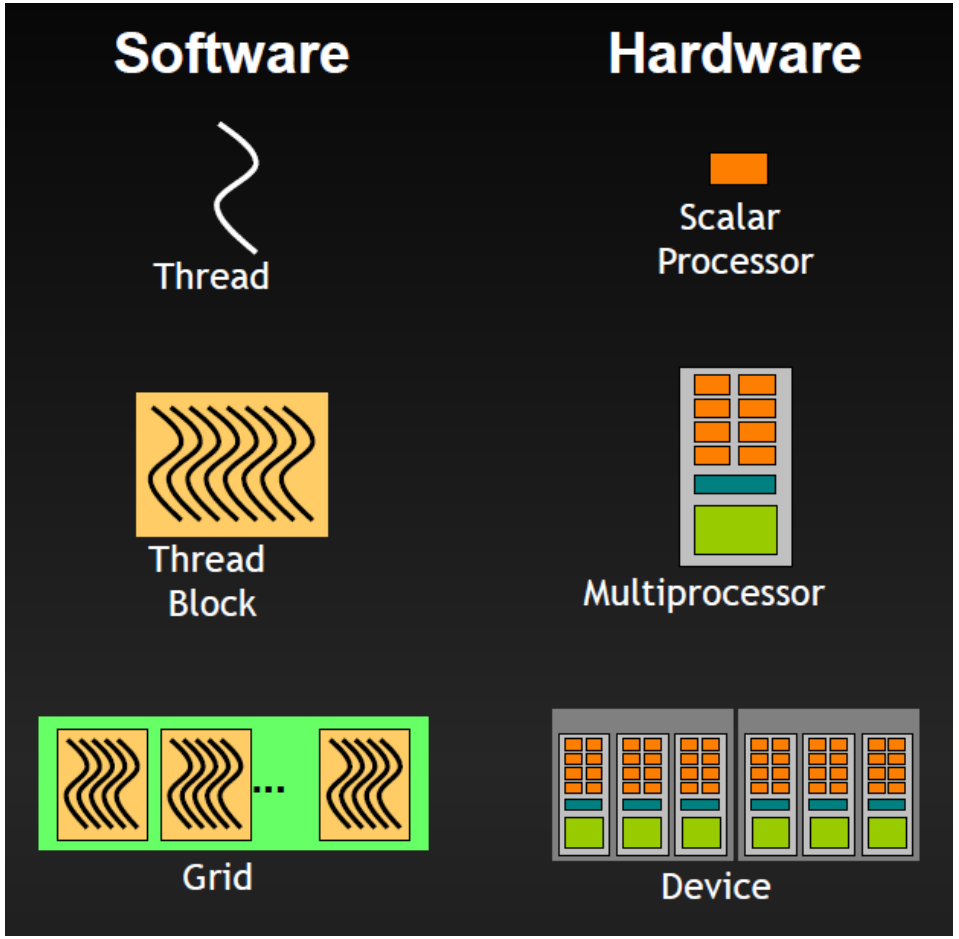
NVIDIA Volta V100

# 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

# 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



# When You Look at the Cloud, What Do You See?

- "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
    - "Burst" from a regular data center
  - Access to a wide range of hardware
    - Try before you buy (or not)
- Challenges:
  - Data movement
    - Expensive and time consuming
  - Support
  - Security, privacy, …

# What We Didn't Talk About

- Microarchitecture details (ALU, FPU, pipelining…)
- Memory bandwidth and latency
- Cache lines and cache coherence
- IBM (Power), AMD, or ARM 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