

# HPC Induction

## Part I: Hardware

Jascha Schewtschenko

Royal Observatory of Edinburgh, University of Edinburgh

May 14, 2025



# Outline

## 1 What is HPC?

## 2 Building an HPC system

- Processors
- Memory / Data Storage
- Data Transfer / Connectivity

## 3 ARTEMIS Hardware



# What is HPC?



# What is HPC?

## Definition

**High Performance Computing** (HPC) most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.

# What is HPC?

## Definition

**High Performance Computing** (HPC) most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.

- Requires large-scale machines and clusters (aka supercomputers) or huge distributed computing networks (cf. e.g. Einstein@Home / BOINC)



# What is HPC?

## Definition

**High Performance Computing** (HPC) most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.

- Requires large-scale machines and clusters (aka supercomputers) or huge distributed computing networks (cf. e.g. Einstein@Home / BOINC)
- But also requires special software to allow the components to work efficiently together (scheduling, distributed data, parallelisation, etc.)



# Supercomputers & Frameworks

## Definition

A supercomputer is a computer with a high level of performance compared to a general-purpose computer. The performance of a supercomputer is commonly measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS).



# Supercomputers & Frameworks

## Definition

A supercomputer is a computer with a high level of performance compared to a general-purpose computer. The performance of a supercomputer is commonly measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS).

- Originally, supercomputers were big machines (SMP), often with highly specialised hardware. Nowadays, most supercomputers consists of closely inter-connected clusters (MPP) of (semi-)independent *nodes* of "off-the-shelf" hardware



# Supercomputers & Frameworks

## Definition

A supercomputer is a computer with a high level of performance compared to a general-purpose computer. The performance of a supercomputer is commonly measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS).

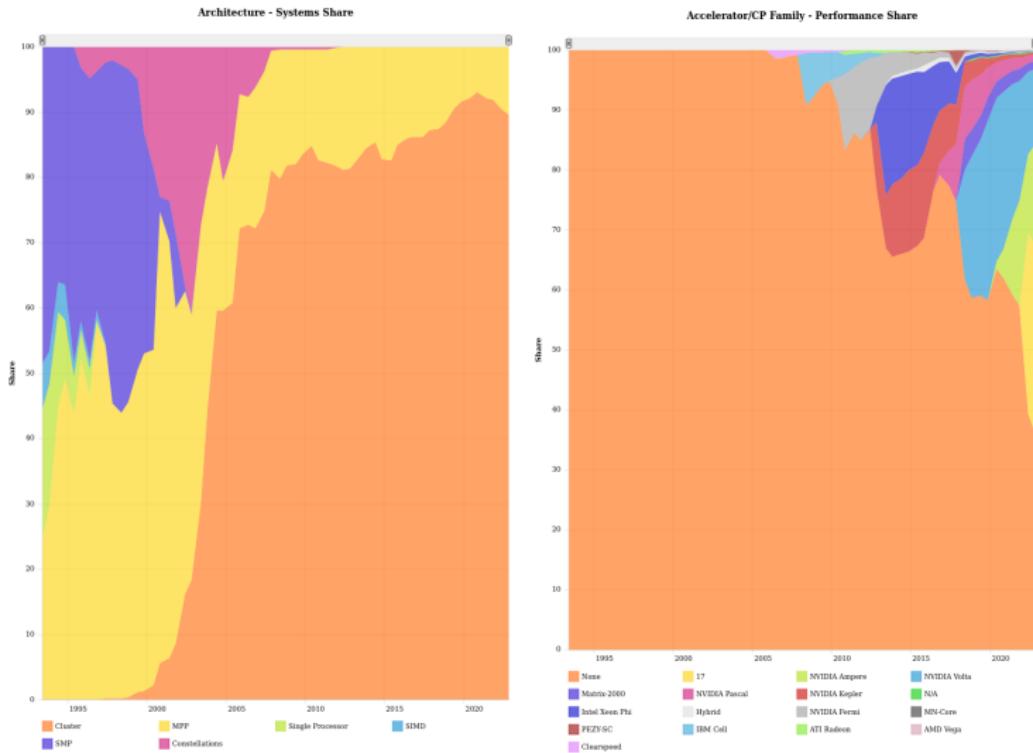
- Originally, supercomputers were big machines (SMP), often with highly specialised hardware. Nowadays, most supercomputers consists of closely inter-connected clusters (MPP) of (semi-)independent *nodes* of "off-the-shelf" hardware
- These 'servers' are hosted in so-called *data centres*



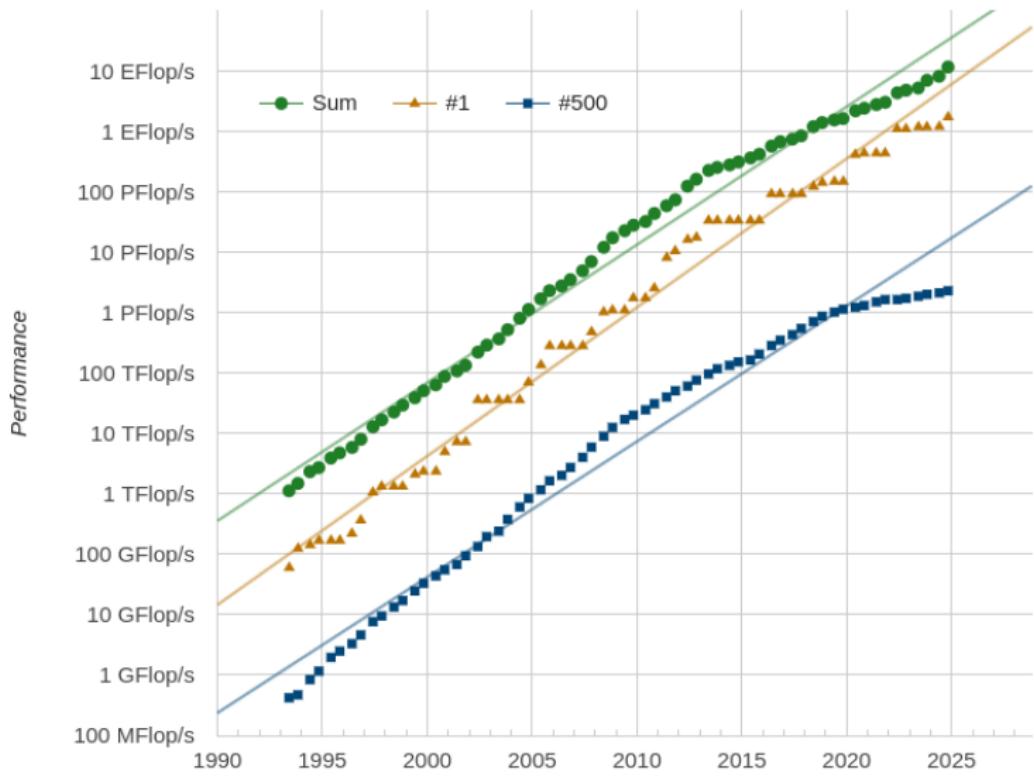
# Supercomputers & Frameworks



# Supercomputers & Frameworks



# Supercomputers & Frameworks (cont.)



## Supercomputers & Frameworks (cont.)

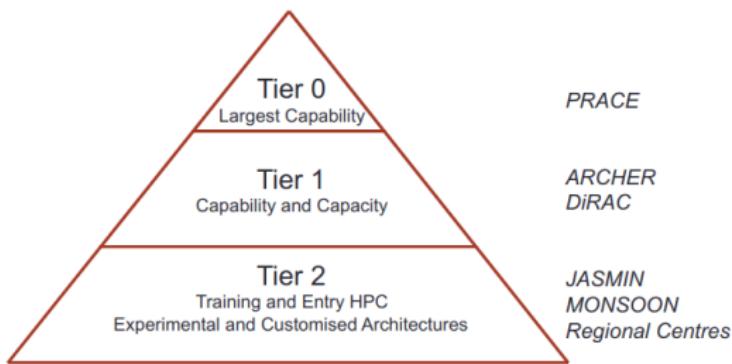
- Besides the Top500, thousands of smaller supercomputer exist - for both commercial and research purposes, both dedicated or shared

## Supercomputers & Frameworks (cont.)

- Besides the Top500, thousands of smaller supercomputer exist - for both commercial and research purposes, both dedicated or shared
- Commercial, shared HPC frameworks/services → e.g. Google Cloud Services, AWS

# Supercomputers & Frameworks (cont.)

- Besides the Top500, thousands of smaller supercomputer exist - for both commercial and research purposes, both dedicated or shared
- Commercial, shared HPC frameworks/services → e.g. Google Cloud Services, AWS
- Non-commercial/research shared HPC infrastructure → e.g. DiRAC & Archer (UK), EuroHPC (EU), usually classified in so-called Tiers



# Building an HPC system

## Summit Overview



### Components

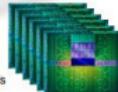
#### IBM POWER9

- 22 Cores
- 4 Threads/core
- NVLink



#### NVIDIA GV100

- 7 TF
- 16 GB @ 0.9 TB/s
- NVLink



### Compute Node

- 2 x POWER9
- 6 x NVIDIA GV100
- NVMe-compatible PCIe 1600 GB SSD



- 25 GB/s EDR IB- (2 ports)
- 512 GB DRAM- (DDR4)
- 96 GB HBM- (3D Stacked)
- Coherent Shared Memory

### Compute Rack

- 18 Compute Servers
- Warm water (70°F direct-cooled components)
- RDHX for air-cooled components



- 39.7 TB Memory/rack
- 55 KW max power/rack

### Compute System

- 10.2 PB Total Memory
- 256 compute racks
- 4,608 compute nodes
- Mellanox EDR IB fabric
- 200 PFLOPS
- ~13 MW



### GPFS File System

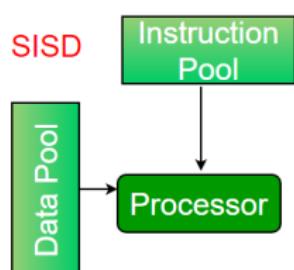
- 250 PB storage
- 2.5 TB/s read, 2.5 TB/s write



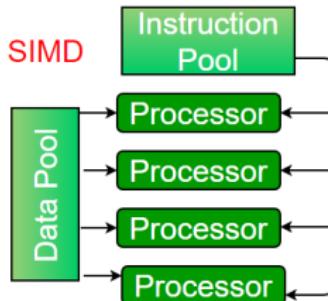
OAK RIDGE  
National Laboratory | LEADERSHIP COMPUTING FACILITY



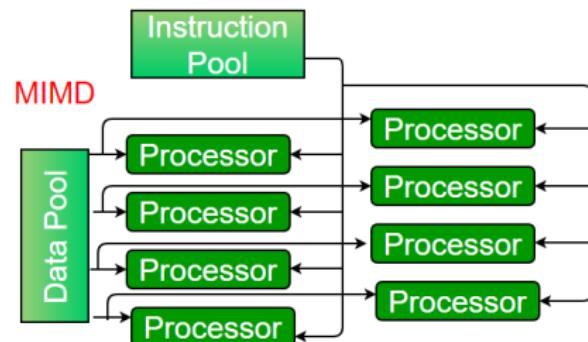
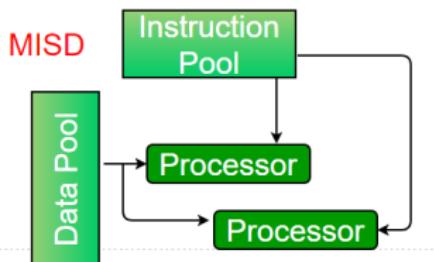
# Computer Architectures - Flynn's taxonomy



classical computer



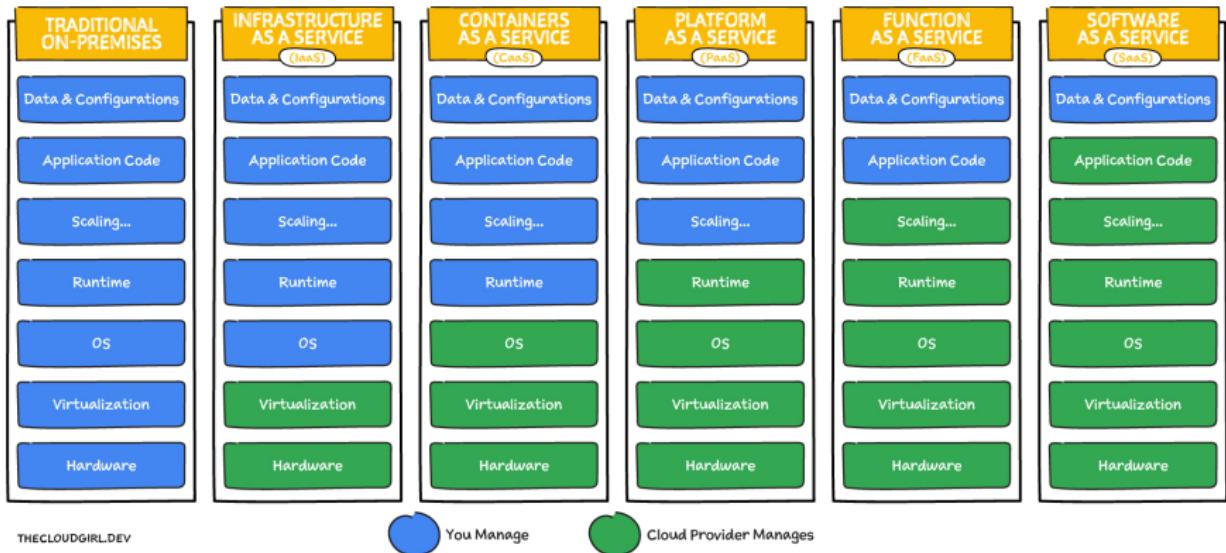
vector processing, GPUs



Multi-processing / multi-computing



# Teaser: HPC as a service



# HPC stack

## SOFTWARE

Environments & Applications

## SYSTEM SOFTWARE

Resource & Job Management

Runtime System Interprocess Comm

Operating System

## VIRTUALISATION

Cloud computing / OpenStack

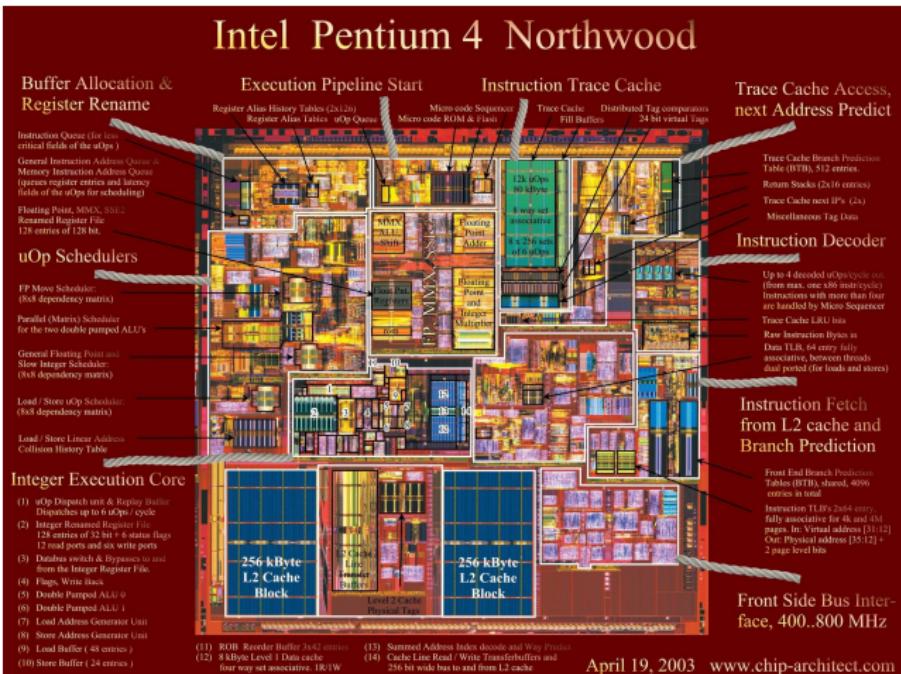
## HARDWARE

Network Interconnects

Memory & Data Storage

Processors & Accelerators

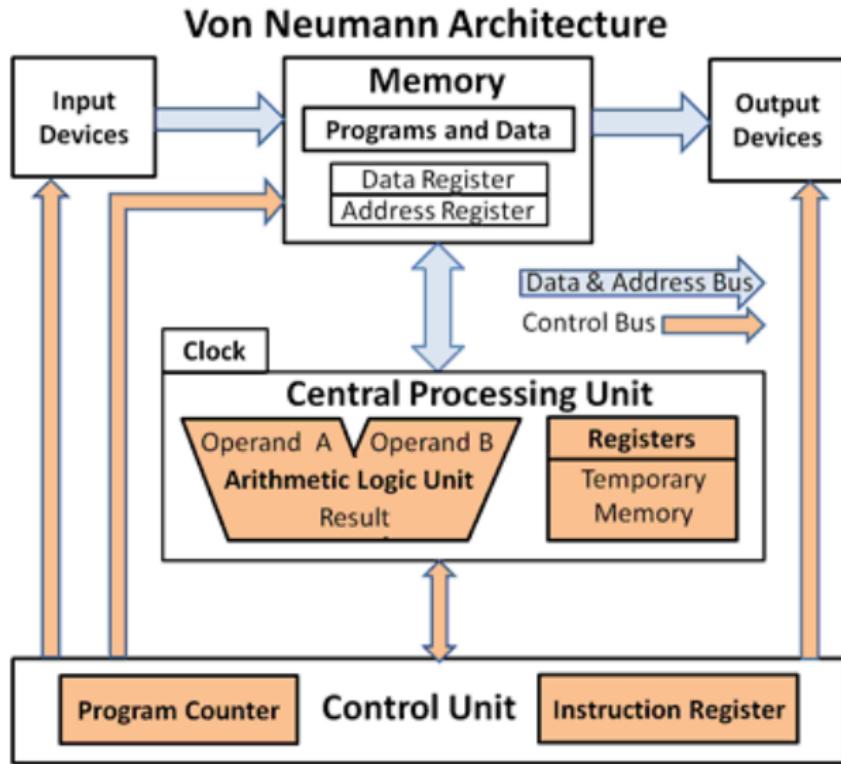
# Processors



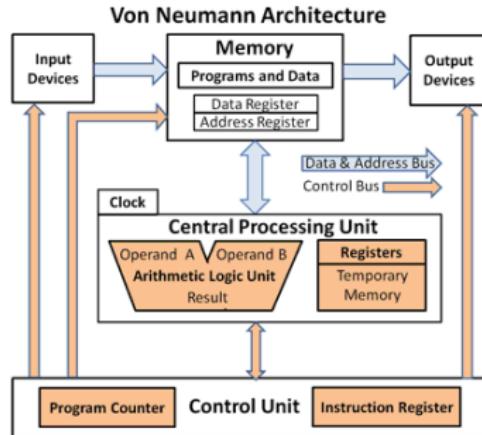
April 19, 2003 [www.chip-architect.com](http://www.chip-architect.com)



# Processors: Basics



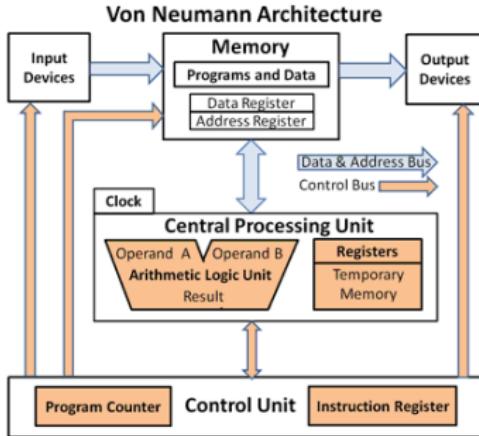
# Processors: Basics



Control Unit manages the flow and timing of data and instructions through the computer as well as the operations performed by the CPU



# Processors: Basics

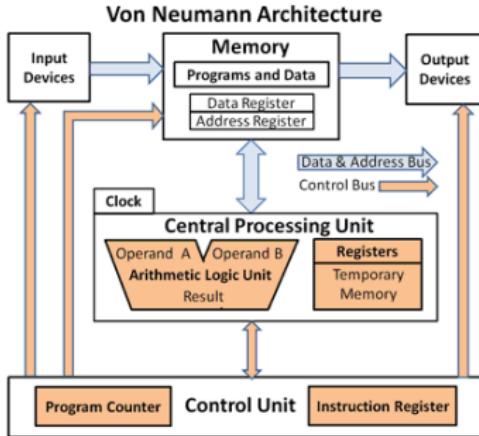


**Control Unit** manages the flow and timing of data and instructions through the computer as well as the operations performed by the CPU

**ALU** digital circuit within the processor that performs integer arithmetic and bitwise logic operations



# Processors: Basics



**Control Unit** manages the flow and timing of data and instructions through the computer as well as the operations performed by the CPU

**ALU** digital circuit within the processor that performs integer arithmetic and bitwise logic operations

**Registers** (fastest) memory to store instructions/data used by CPU in this cycle

# Processors: Supplementary instruction sets (SIMD/Vectorization)

- processors provide special hardware extensions to perform certain operations faster/more efficiently



# Processors: Supplementary instruction sets (SIMD/Vectorization)

- processors provide special hardware extensions to perform certain operations faster/more efficiently
- History/Overview:

FPU (pre-Pentium) special co-processor to provide additional FP operations, uses additional registers

# Processors: Supplementary instruction sets (SIMD/Vectorization)

- processors provide special hardware extensions to perform certain operations faster/more efficiently
- History/Overview:

FPU (pre-Pentium) special co-processor to provide additional FP operations, uses additional registers

MMX (for Pentium I/II) SIMD instruction set for integers; using FP registers

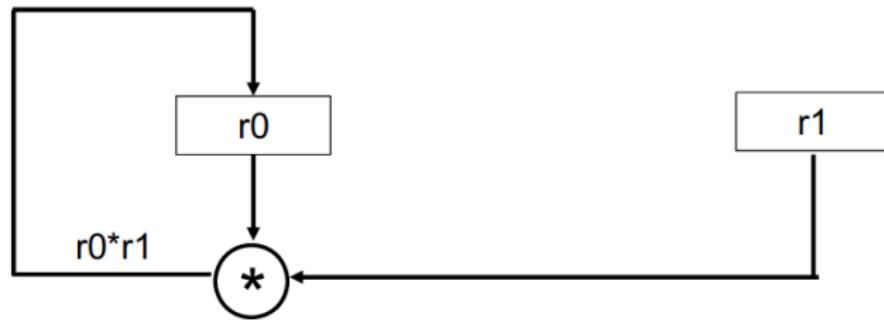
# Processors: Supplementary instruction sets (SIMD/Vectorization)

- processors provide special hardware extensions to perform certain operations faster/more efficiently
- History/Overview:

FPU (pre-Pentium) special co-processor to provide additional FP operations, uses additional registers

MMX (for Pentium I/II) SIMD instruction set for integers; using FP registers

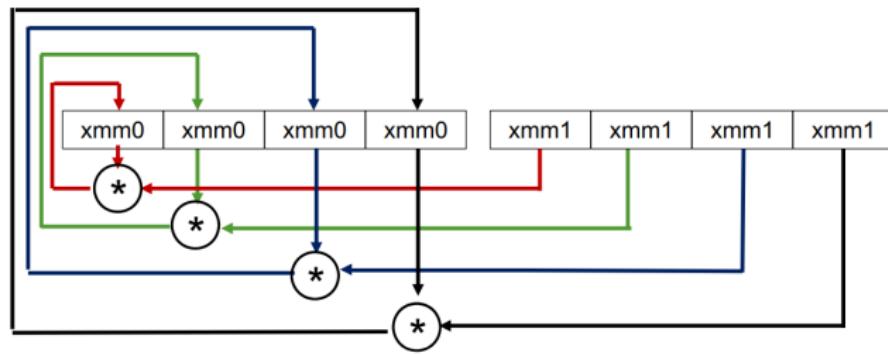
**mulss r1, r0**



# Processors: Supplementary instruction sets (SIMD/Vectorization)

- processors provide special hardware extensions to perform certain operations faster/more efficiently
- History/Overview:
  - FPU (pre-Pentium) special co-processor to provide additional FP operations, uses additional registers
  - MMX (for Pentium I/II) SIMD instruction set for integers; using FP registers

`mulps xmm1, xmm0`



## Processors: Supplementary instruction sets (cont.)

SSE (since Pentium 3) SIMD instruction set for SP floats;  
additional registers

SSE2/SSE3/SSE4.x (since Pentium 4 / Xeon) SIMD instruction set for  
SP/DP floats, long/standard/short integers, chars;  
additional registers

## Processors: Supplementary instruction sets (cont.)

SSE (since Pentium 3) SIMD instruction set for SP floats;  
additional registers

SSE2/SSE3/SSE4.x (since Pentium 4 / Xeon) SIMD instruction set for  
SP/DP floats, long/standard/short integers, chars;  
additional registers

AVX extension to SSE extensions

AVX2 extension of SSE/AVX operations to 256 bits

AVX-512 extension of SSE/AVX operations to 512 bits (Artemis)

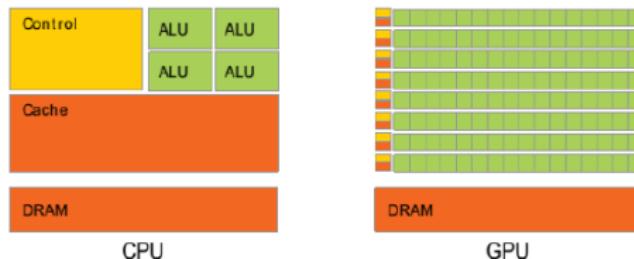
# Processors: GPGPUs / SIMT

- processors with reduced instruction set (RISC) for fast execution



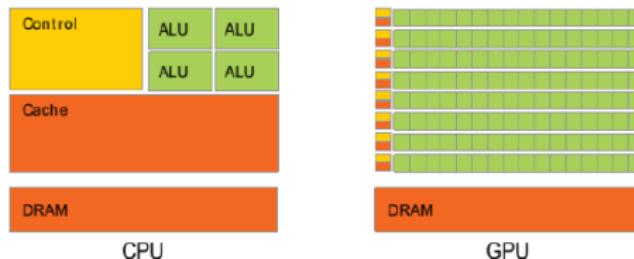
# Processors: GPGPUs / SIMD

- processors with reduced instruction set (RISC) for fast execution
- contains thousands of ALUs organised in batches/(slim-lined) SIMD cores, each of them able to run a single “warp” of threads at the same time (SIMT)



# Processors: GPGPUs / SIMD

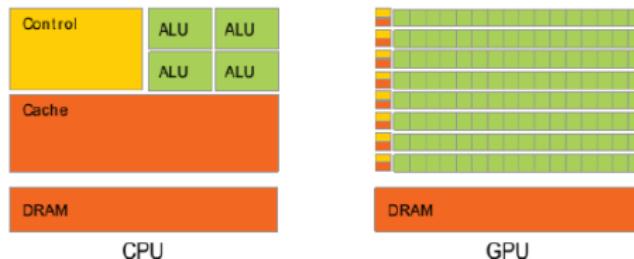
- processors with reduced instruction set (RISC) for fast execution
- contains thousands of ALUs organised in batches/(slim-lined) SIMD cores, each of them able to run a single “warp” of threads at the same time (SIMT)



- perfect for performing same operation on large datasets (e.g. filters on images)

# Processors: GPGPUs / SIMD

- processors with reduced instruction set (RISC) for fast execution
- contains thousands of ALUs organised in batches/(slim-lined) SIMD cores, each of them able to run a single “warp” of threads at the same time (SIMT)



- perfect for performing same operation on large datasets (e.g. filters on images)
- horrible for serial code



# Processors: GPGPUs / SIMD

- processors with reduced instruction set (RISC) for fast execution
- contains thousands of ALUs organised in batches/(slim-lined) SIMD cores, each of them able to run a single “warp” of threads at the same time (SIMT)



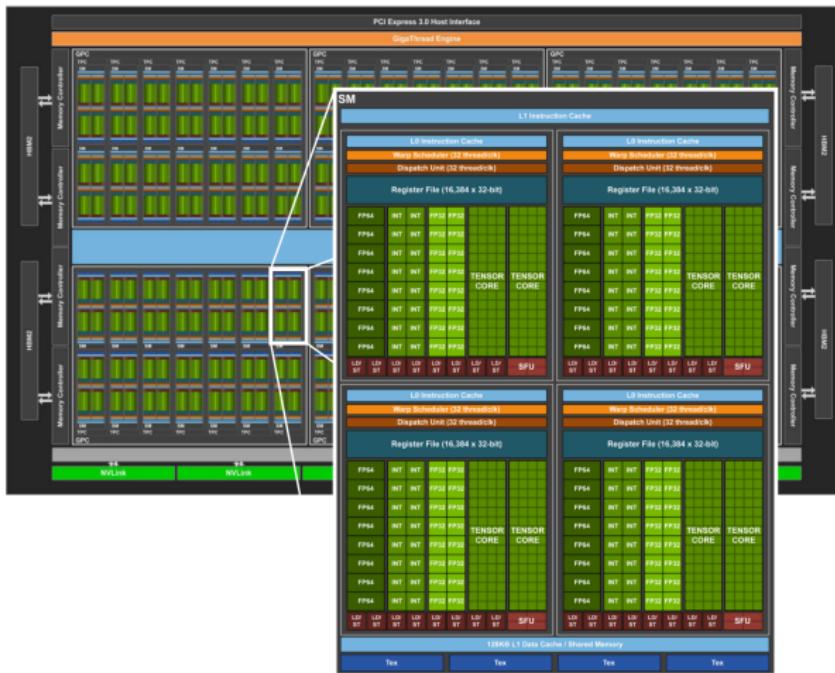
- perfect for performing same operation on large datasets (e.g. filters on images)
- horrible for serial code
- require special frameworks for development: e.g. CUDA, OpenCL

# Processors: GPGPUs / SIMT



- here: NVIDIA Volta
- latest gen: NVIDIA Blackwell (Q4 2024)

# Processors: GPGPUs / SIMT



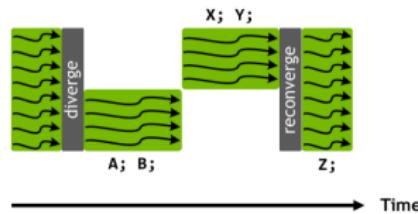
- here: NVIDIA Volta
- latest gen: NVIDIA Blackwell (Q4 2024)
- 192 SMs (12 GPCs)  
x 128 cores per SM  
= 24576 cuda cores



# Processors: GPGPUs / SIMT

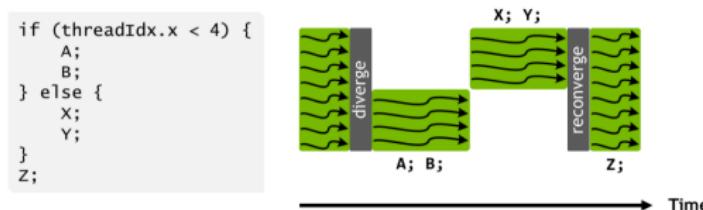
- an SIMT *thread* is a single, independent unit of execution (similar to software multithreading) and grouped into *thread blocks* of up to 1024 threads sharing the same resources of SM
- a thread block is subdivided into *warps* of 32 threads each (to be handled by processing blocks)
- one command executed at same time per SM (other threads waiting)

```
if (threadIdx.x < 4) {  
    A;  
    B;  
} else {  
    X;  
    Y;  
}  
Z;
```

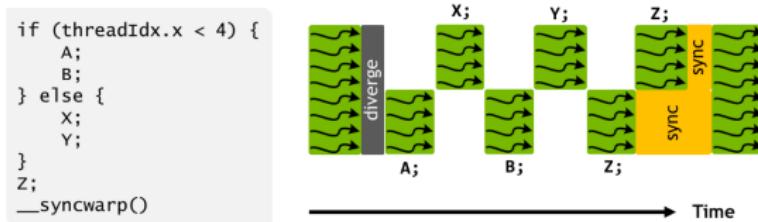


# Processors: GPGPUs / SIMT

- an SIMT *thread* is a single, independent unit of execution (similar to software multithreading) and grouped into *thread blocks* of up to 1024 threads sharing the same resources of SM
- a thread block is subdivided into *warps* of 32 threads each (to be handled by processing blocks)
- one command executed at same time per SM (other threads waiting)



- newer architecture ( $\geq$  Volta) allows for more complex 'divergence'



# Processors: ASIC / TPUs

- there are application-specific integrated circuits (ASIC) used for a particular rather than general-purpose use

# Processors: ASIC / TPUs

- there are application-specific integrated circuits (ASIC) used for a particular rather than general-purpose use
- Historical example: GRA(vity)P(ip)E for direct summation of gravity forces between bodies

# Processors: ASIC / TPUs

- there are application-specific integrated circuits (ASIC) used for a particular rather than general-purpose use
- Historical example: GRA(vity)P(ip)E for direct summation of gravity forces between bodies
- Today: e.g. T(ensor)P(rocessing)U(nit)s for hardware acceleration of machine learning

# Processors: ASIC / TPUs

- there are application-specific integrated circuits (ASIC) used for a particular rather than general-purpose use
- Historical example: GRA(vity)P(ip)E for direct summation of gravity forces between bodies
- Today: e.g. T(ensor)P(rocessing)U(nit)s for hardware acceleration of machine learning
- require special frameworks: e.g. TensorFlow for TPUs



# Processors: MIMD / Multi-Core CPUs and Multi-CPU

- Today, CPUs consists of multiple cores (for Multiple-Instructions Multiple-Data architecture)



# Processors: MIMD / Multi-Core CPUs and Multi-CPU

- Today, CPUs consists of multiple cores (for Multiple-Instructions Multiple-Data architecture)
- usually 4-8 cores for consumer PCs/ up to 56 cores for server-grade CPUs



# Processors: MIMD / Multi-Core CPUs and Multi-CPU

- Today, CPUs consists of multiple cores (for Multiple-Instructions Multiple-Data architecture)
- usually 4-8 cores for consumer PCs/ up to 56 cores for server-grade CPUs
- Each core has its own control unit, logical unit, registers and low-level cache

# Processors: MIMD / Multi-Core CPUs and Multi-CPU

- Today, CPUs consists of multiple cores (for Multiple-Instructions Multiple-Data architecture)
- usually 4-8 cores for consumer PCs/ up to 56 cores for server-grade CPUs
- Each core has its own control unit, logical unit, registers and low-level cache
- Alternatively, systems can also have multiple CPUs

# Processors: MIMD / Multi-Core CPUs and Multi-CPU

- Today, CPUs consists of multiple cores (for Multiple-Instructions Multiple-Data architecture)
- usually 4-8 cores for consumer PCs/ up to 56 cores for server-grade CPUs
- Each core has its own control unit, logical unit, registers and low-level cache
- Alternatively, systems can also have multiple CPUs
- 'higher-level' memory and I/O interfaces are shared between cores / CPUs

# Processors: MIMD / Multi-Core CPUs and Multi-CPU

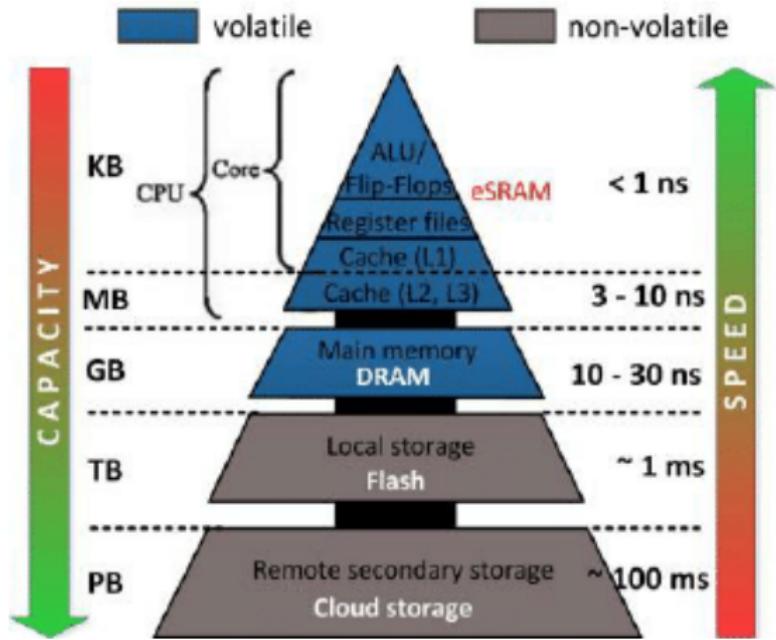
- Today, CPUs consists of multiple cores (for Multiple-Instructions Multiple-Data architecture)
- usually 4-8 cores for consumer PCs/ up to 56 cores for server-grade CPUs
- Each core has its own control unit, logical unit, registers and low-level cache
- Alternatively, systems can also have multiple CPUs
- 'higher-level' memory and I/O interfaces are shared between cores / CPUs
- on OS/software level, multi-core and multi-CPU systems are equivalent



# Memory / Data Storage



# Memory: Hierarchy



# Memory: Registers & Cache

**Registers** Fastest memory

(latency: 1 CPU cycle; size:  $\mathcal{O}(1 \text{ kB})$ )

**Level 0** (some arch.) Micro operations cache

(latency: few CPU cycles; size:  $\mathcal{O}(1 \text{ kB})$ )

**Level 1** Data & instruction caches

(latency: few CPU cycles; size:  $\mathcal{O}(100 \text{ kB})$ ; transfer: 700 GB/s)

# Memory: Registers & Cache

**Registers** Fastest memory

(latency: 1 CPU cycle; size:  $\mathcal{O}(1 \text{ kB})$ )

**Level 0** (some arch.) Micro operations cache

(latency: few CPU cycles; size:  $\mathcal{O}(1 \text{ kB})$ )

**Level 1** Data & instruction caches

(latency: few CPU cycles; size:  $\mathcal{O}(100 \text{ kB})$ ; transfer: 700 GB/s)

---

**Level 2** Shared data & instruction cache

(latency: few CPU cycles; size:  $\mathcal{O}(1 \text{ MB})$ ; transfer: 200 GB/s)

**Level 3** Shared cache (also with GFX)

(latency: 3-10 ns; size: few MB/core; transfer: 100 GB/s)

**Level 4** (some arch.) Shared cache

(latency: 3-10 ns; size:  $\mathcal{O}(100 \text{ MB})$ ; transfer: 40 GB/s)



# Memory: Main Memory / DRAM

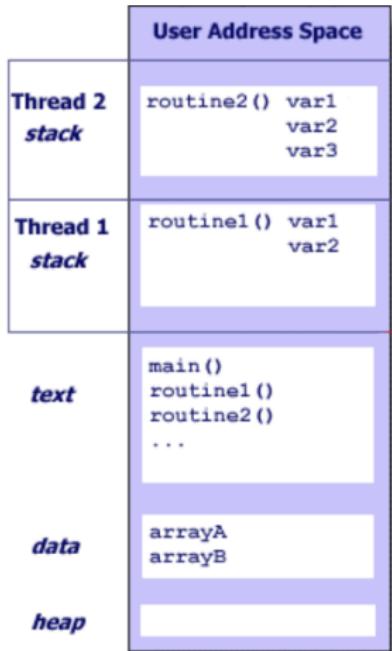
- Fast, volatile memory to hold (all) data and program code for OS and whole software application  
(latency:  $\mathcal{O}(10 \text{ ns})$ ; size:  $\mathcal{O}(1 \text{ GB} - 1 \text{ TB})$ ;  
transfer: 10 GB/s)

# Memory: Main Memory / DRAM

- Fast, volatile memory to hold (all) data and program code for OS and whole software application  
(latency:  $\mathcal{O}(10 \text{ ns})$ ; size:  $\mathcal{O}(1 \text{ GB} - 1 \text{ TB})$ ;  
transfer: 10 GB/s)
- Organized by hardware into pages & segments; MMU translates from physical to logical address (page/segment & offset)

# Memory: Main Memory / DRAM

- Fast, volatile memory to hold (all) data and program code for OS and whole software application  
(latency:  $\mathcal{O}(10 \text{ ns})$ ; size:  $\mathcal{O}(1 \text{ GB} - 1 \text{ TB})$ ; transfer: 10 GB/s)
- Organized by hardware into pages & segments; MMU translates from physical to logical address (page/segment & offset)
- Accessible/manageable via low-level programming languages (within user address space of application)



# Memory: Virtual Memory / Paging

- If memory requirement of system exceeds size of main memory, currently unused memory pages are exported to secondary storage aka swap space with higher latency and lower transfer speeds (see data storage).

# Memory: Virtual Memory / Paging

- If memory requirement of system exceeds size of main memory, currently unused memory pages are exported to secondary storage aka swap space with higher latency and lower transfer speeds (see data storage).
- Works great for parts of memory data that is not often used (i.e. not part of the working set)

# Memory: Virtual Memory / Paging

- If memory requirement of system exceeds size of main memory, currently unused memory pages are exported to secondary storage aka swap space with higher latency and lower transfer speeds (see data storage).
- Works great for parts of memory data that is not often used (i.e. not part of the working set)
- But may lead to horrible *thrashing* (i.e. constant swapping of pages) if e.g. a working set exceeds the main memory size



# Data Storage: Types of Persistent Memory

- There are various types of persistent storage:

# Data Storage: Types of Persistent Memory

- There are various types of persistent storage:

Hard Disk Drives (HDD) based on magnetic storage: cheap (\$18/TB), but slow(er) (both latency and transfer)

# Data Storage: Types of Persistent Memory

- There are various types of persistent storage:

Hard Disk Drives (HDD) based on magnetic storage: cheap (\$18/TB), but slow(er) (both latency and transfer)

Solid-State Drives (SSD) based on solid-state/NAND flash memory: faster, but more expensive (\$40/TB)

# Data Storage: Types of Persistent Memory

- There are various types of persistent storage:

Hard Disk Drives (HDD) based on magnetic storage: cheap (\$18/TB), but slow(er) (both latency and transfer)

Solid-State Drives (SSD) based on solid-state/NAND flash memory: faster, but more expensive (\$40/TB)

Tape based on magnetic storage: very cheap (\$10/TB), but very slow

# Data Storage: Types of Persistent Memory

- There are various types of persistent storage:

Hard Disk Drives (HDD) based on magnetic storage: cheap (\$18/TB), but slow(er) (both latency and transfer)

Solid-State Drives (SSD) based on solid-state/NAND flash memory: faster, but more expensive (\$40/TB)

Tape based on magnetic storage: very cheap (\$10/TB), but very slow

Optical e.g. CDs,DVDs,Bluray,etc. (currently obsolete, but in future, holographic with high data densities)

# Data Storage: Types of Persistent Memory

- There are various types of persistent storage:

Hard Disk Drives (HDD) based on magnetic storage: cheap (\$18/TB), but slow(er) (both latency and transfer)

Solid-State Drives (SSD) based on solid-state/NAND flash memory: faster, but more expensive (\$40/TB)

Tape based on magnetic storage: very cheap (\$10/TB), but very slow

Optical e.g. CDs,DVDs,Bluray,etc. (currently obsolete, but in future, holographic with high data densities)

- Usually, data centres use a combination: SSDs for e.g. core OS files, swap space), HDDs as main data storage and tapes for backups and "cold" data



# Data Storage: Local vs Network vs Remote/Cloud

- Independently of its type, storage can be located/accessible in three different ways:

# Data Storage: Local vs Network vs Remote/Cloud

- Independently of its type, storage can be located/accessible in three different ways:
  - Local Each node has usually its own HDD/SSD that contains the core OS and temporary user data

# Data Storage: Local vs Network vs Remote/Cloud

- Independently of its type, storage can be located/accessible in three different ways:
  - Local** Each node has usually its own HDD/SSD that contains the core OS and temporary user data
  - Network** Big/shared data is usually stored on specialised storage servers and provided to the compute nodes via network file systems (NFS,LUSTRE)

# Data Storage: Local vs Network vs Remote/Cloud

- Independently of its type, storage can be located/accessible in three different ways:
  - Local** Each node has usually its own HDD/SSD that contains the core OS and temporary user data
  - Network** Big/shared data is usually stored on specialised storage servers and provided to the compute nodes via network file systems (NFS,LUSTRE)
  - Remote/Cloud** Similar to network storage, but off-site (usually much lower latency / transfer).



# Data Storage: Local vs Network vs Remote/Cloud

- Independently of its type, storage can be located/accessible in three different ways:
  - Local** Each node has usually its own HDD/SSD that contains the core OS and temporary user data
  - Network** Big/shared data is usually stored on specialised storage servers and provided to the compute nodes via network file systems (NFS,LUSTRE)
  - Remote/Cloud** Similar to network storage, but off-site (usually much lower latency / transfer).
- Usually, data centres use a combination of all three (remote/cloud storage mostly for backup)



# Data Storage: Redundancy / RAID

- Hard drives have a very limited lifespan (disc failures are the norm, not the exception in big data centres)

# Data Storage: Redundancy / RAID

- Hard drives have a very limited lifespan (disc failures are the norm, not the exception in big data centres)
- In order to avoid data loss, copies of data is usually stored in multiple places



# Data Storage: Redundancy / RAID

- Hard drives have a very limited lifespan (disc failures are the norm, not the exception in big data centres)
- In order to avoid data loss, copies of data is usually stored in multiple places
- Solutions:

# Data Storage: Redundancy / RAID

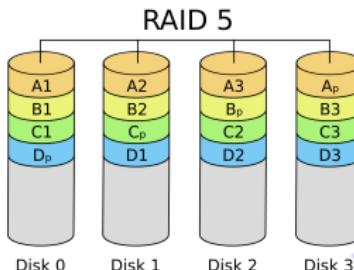
- Hard drives have a very limited lifespan (disc failures are the norm, not the exception in big data centres)
- In order to avoid data loss, copies of data is usually stored in multiple places
- Solutions:
  - Software e.g. via distributed file systems (cf. GFS, HDFS)

# Data Storage: Redundancy / RAID

- Hard drives have a very limited lifespan (disc failures are the norm, not the exception in big data centres)
- In order to avoid data loss, copies of data is usually stored in multiple places
- Solutions:

Software e.g. via distributed file systems (cf. GFS, HDFS)

Hardware RAID - controller combines multiple physical disk drive components into one or more logical units. Depending on level used, by either mirroring whole disks (RAID 1) or spreading over multiple disk with add. parity blocks to ensure operability with (or more) discs failing (RAID 5/6)



# Data Transfer / Connectivity



# Data Transfer / Connectivity: LAN

- There are multiple technologies available for communication/data transfer between nodes of a supercluster:



# Data Transfer / Connectivity: LAN

- There are multiple technologies available for communication/data transfer between nodes of a supercluster:

**Ethernet** Widely-used consumer-grade networking standard (speed up to 1Gbit/s)



# Data Transfer / Connectivity: LAN

- There are multiple technologies available for communication/data transfer between nodes of a supercluster:

**Ethernet** Widely-used consumer-grade networking standard (speed up to 1Gbit/s)

**InfiniBand** special networking standard used in HPC with very low latencies and high throughput (transfer speeds up to 100Gbit/s per link (NDR))



# Data Transfer / Connectivity: LAN

- There are multiple technologies available for communication/data transfer between nodes of a supercluster:

**Ethernet** Widely-used consumer-grade networking standard (speed up to 1Gbit/s)

**InfiniBand** special networking standard used in HPC with very low latencies and high throughput (transfer speeds up to 100Gbit/s per link (NDR))



- Usually data centres employ both in parallel: Ethernet for low-priority/non-critical communication (e.g. logins, remote shells) and InfiniBand for data transfers (from/to file servers or multi-processing message passing (cf. MPI) between nodes)



## Data Transfer / Connectivity: JANET

- many university data centres in the UK are connected to JANET, a high-speed (up to 600 Gbit/s) network for the UK research and education community



## Data Transfer / Connectivity: JANET

- many university data centres in the UK are connected to JANET, a high-speed (up to 600 GBit/s) network for the UK research and education community
  - it is the busiest National Research and Education Network in Europe by volume of data carried (6 PB/day)



## Data Transfer / Connectivity: JANET

- many university data centres in the UK are connected to JANET, a high-speed (up to 600 GBit/s) network for the UK research and education community
  - it is the busiest National Research and Education Network in Europe by volume of data carried (6 PB/day)
  - linked to other European and worldwide national research and education networks as well as big company networks like Google, AWS, etc.



## Data Transfer / Connectivity: JANET

- many university data centres in the UK are connected to JANET, a high-speed (up to 600 GBit/s) network for the UK research and education community
  - it is the busiest National Research and Education Network in Europe by volume of data carried (6 PB/day)
  - linked to other European and worldwide national research and education networks as well as big company networks like Google, AWS, etc.
  - allows for (relatively) fast data transfer between HPC data centres



# Artemis Hardware

# Artemis Hardware: Overview

- Newest iteration of US HPC infrastructure (deployed in Dec 2024 as successor to Apollo2)

# Artemis Hardware: Overview

- Newest iteration of US HPC infrastructure (deployed in Dec 2024 as successor to Apollo2)
  - ▶ 4 x 128c AMD EPYC, 1TB RAM, 2 A40 Gpu Nodes
  - ▶ 3 x 128c AMD EPYC, 512TB RAM, 2 A40 Gpu Nodes

# Artemis Hardware: Storage

- Artemis has two mass storage servers:

# Artemis Hardware: Storage

- Artemis has two mass storage servers:  
NFS server mainly hosts users' home directories and software

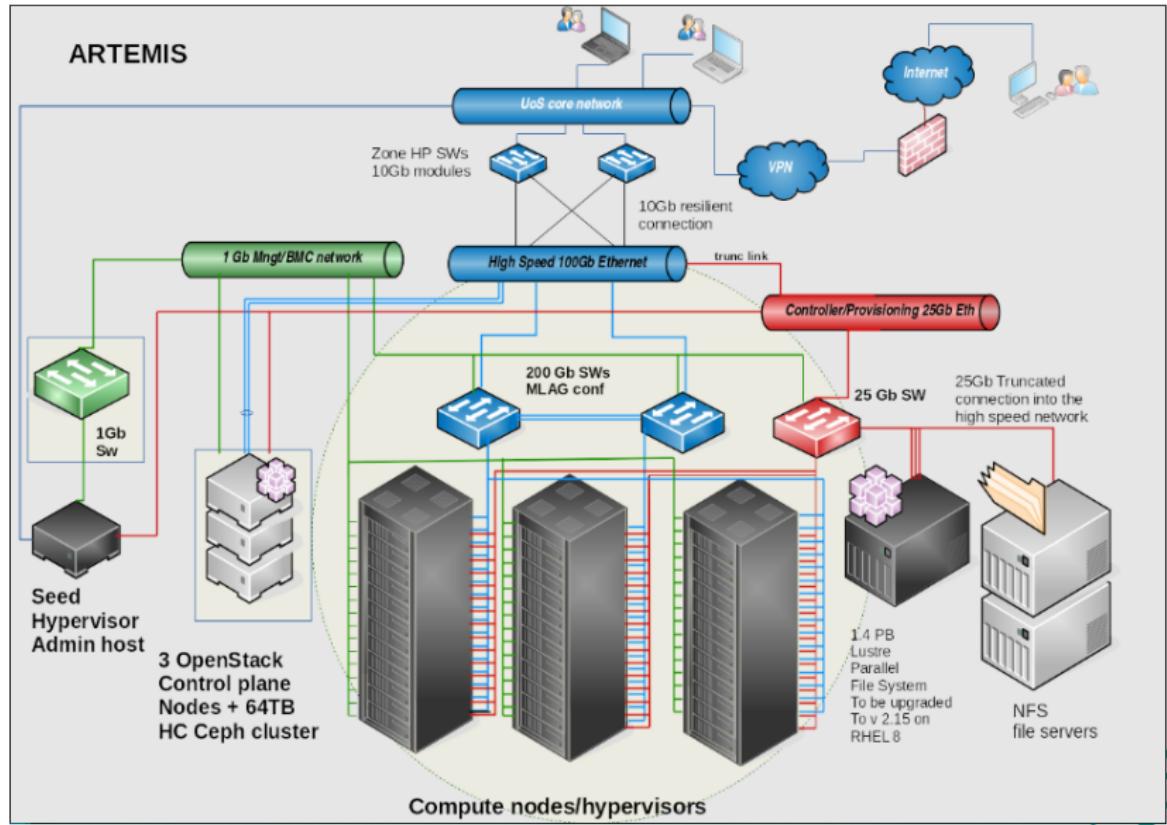
# Artemis Hardware: Storage

- Artemis has two mass storage servers:
  - NFS server mainly hosts users' home directories and software
  - Lustre server size: 1.4 PB; scratch, project & user space

# Artemis Hardware: Storage

- Artemis has two mass storage servers:
  - NFS server mainly hosts users' home directories and software
  - Lustre server size: 1.4 PB; scratch, project & user space
- additionally, experimental support for google drive cloud storage

# Artemis Hardware: Network



# HPC stack

## SOFTWARE

Environments & Applications

## SYSTEM SOFTWARE

Resource & Job Management

Runtime System Interprocess Comm

Operating System

## VIRTUALISATION

Cloud computing / OpenStack

## HARDWARE

Network Interconnects

Memory & Data Storage

Processors & Accelerators