HPC.NRW

HPC.NRW in a Nutshell: HPC Architectures Basics

What is a supercomputer and how can I use it?

Tim Cramer

GREAT COMPUTING COMES WITH GREAT SUPPORT.

Agenda

HPC.NRW

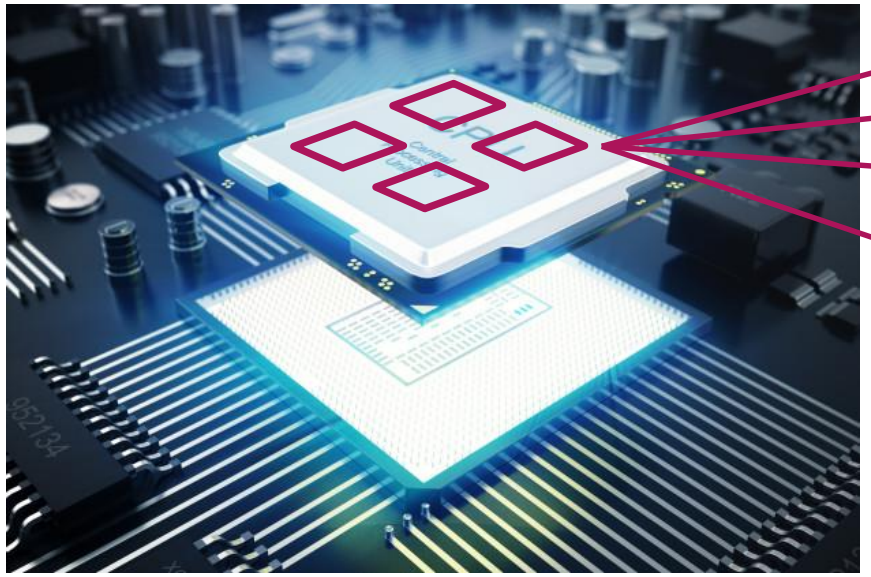
Key question: What is a supercomputer?

AND

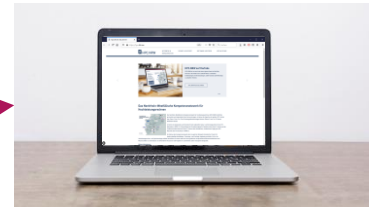
What makes it different from your laptop?

What is a core?

What is a core?



Today’s CPUs are multi-core CPUs

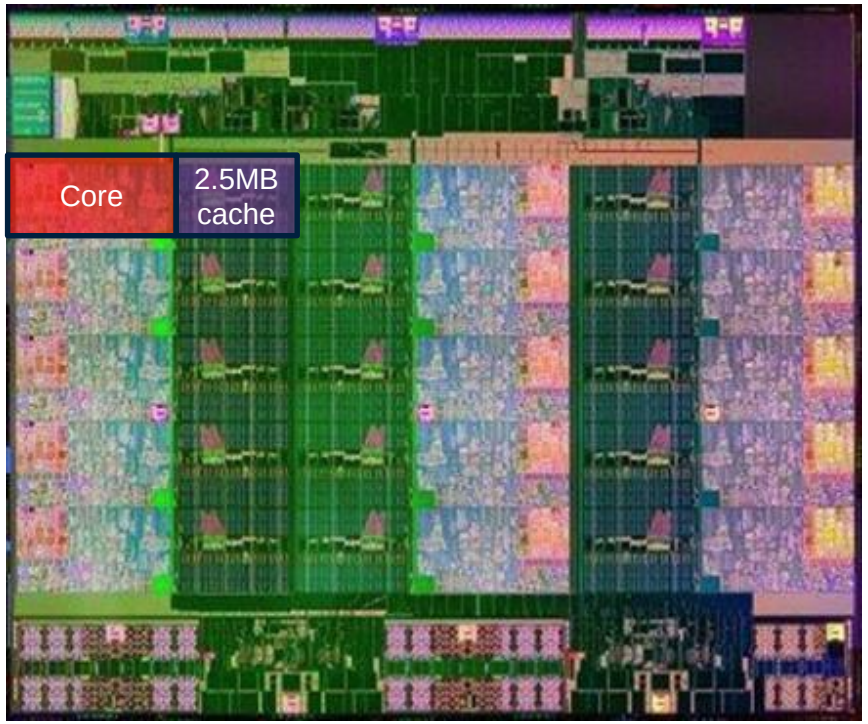


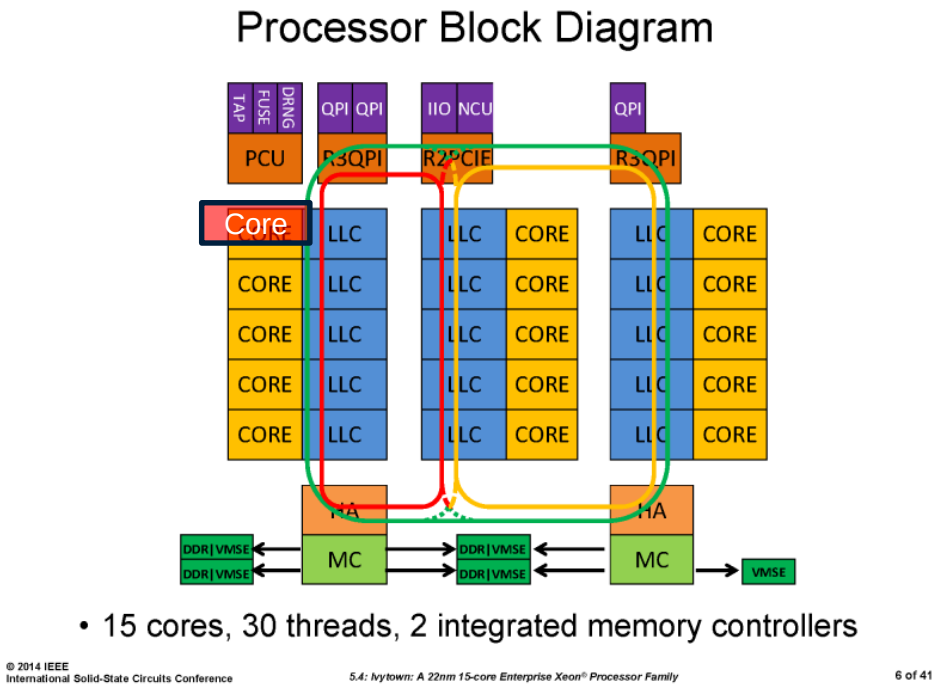
Can be part of different “devices”

Picture sources: FREEPIK

What is a core?

(a more technical perspective)





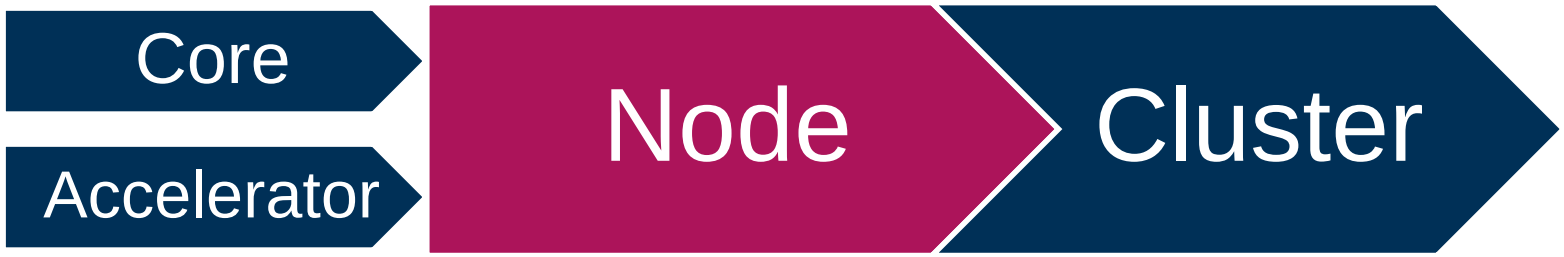
Processor level parallelism

HPC.NRW

- Parallelism at processor/ instruction level
  - Pipelining (overlap in execution: fetch, decode, execute)
  - Superscalar (redundant arithmetical units: Multiplication, Addition, …)
  - SIMD execution (e.g. 512 bit registers, AVX-512)

- Programming techniques
  - Code modifications: Unrolling, Cache reuse
  - Compiler optimizations

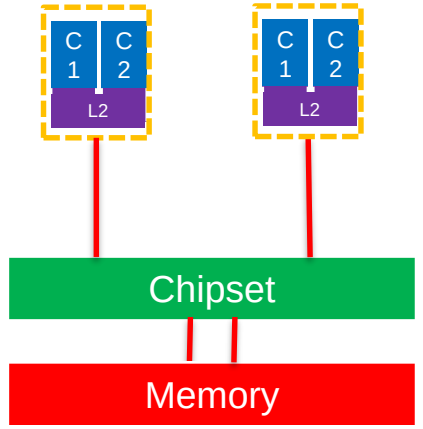
What is a node?



What is a node?

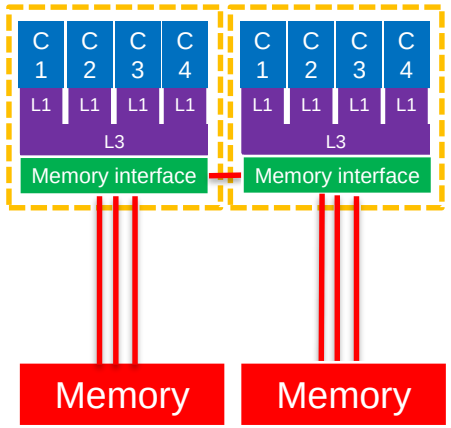
Node in a supercomputer

- A node may contain
  - One or more (multi-core) processors
  - Memory hierarchy (caches, disks, etc.)
  - Interconnects, power supply, fans, …
  - Accelerators



UMA

- Multicore Designs
  - Early multicore design
    - Uniform Memory Architecture (UMA)
    - Flat Memory design
  - Recent multicore design
    - ccNUMA (Cache Coherent Non-Uniform Memory Architecture)
    - Memory Interface + HT/QPI provides inter-socket connectivity



NUMA

What is a cluster?

What is a compute cluster (supercomputer)?

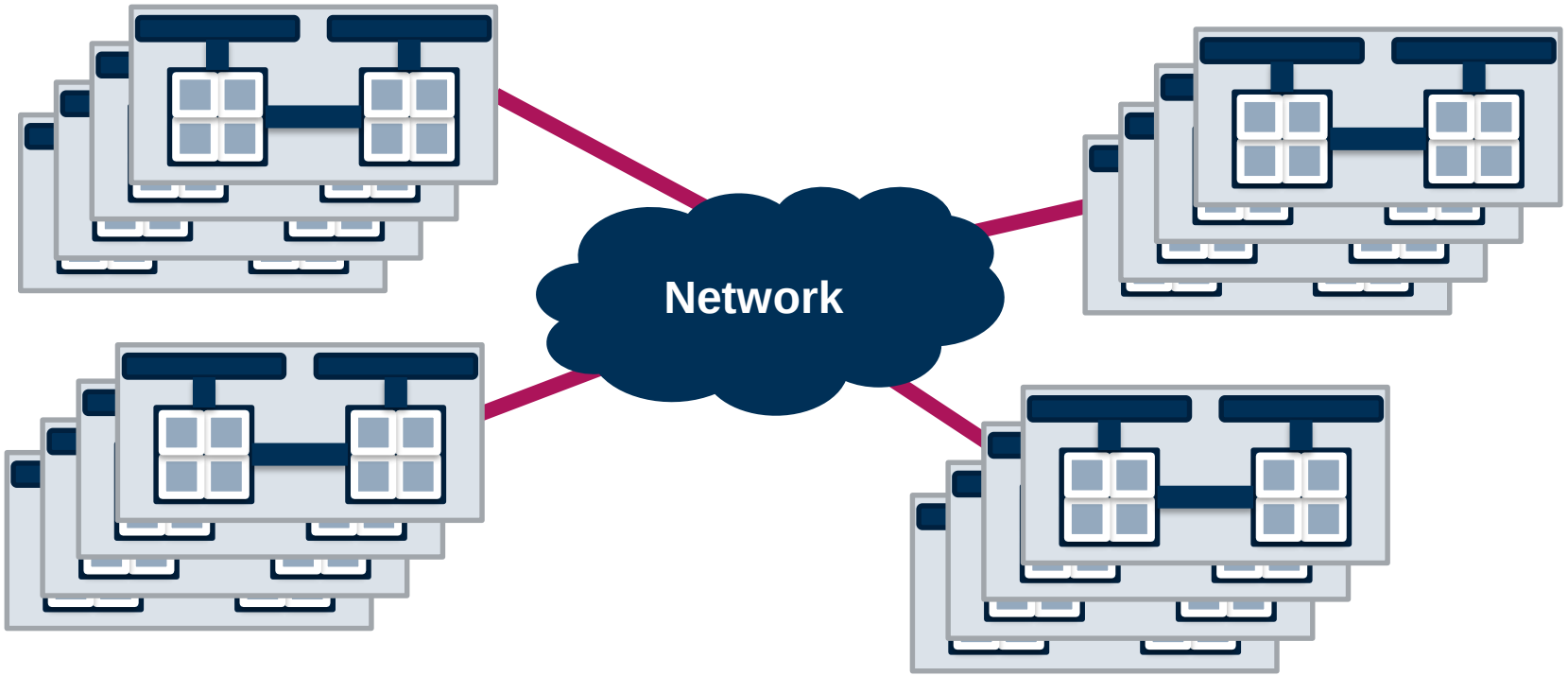
Cray I, 1974

CLAIX (RWTH Aachen University), 2019

Frontier (Oak Ridge National Lab), First official exascale system, 2022

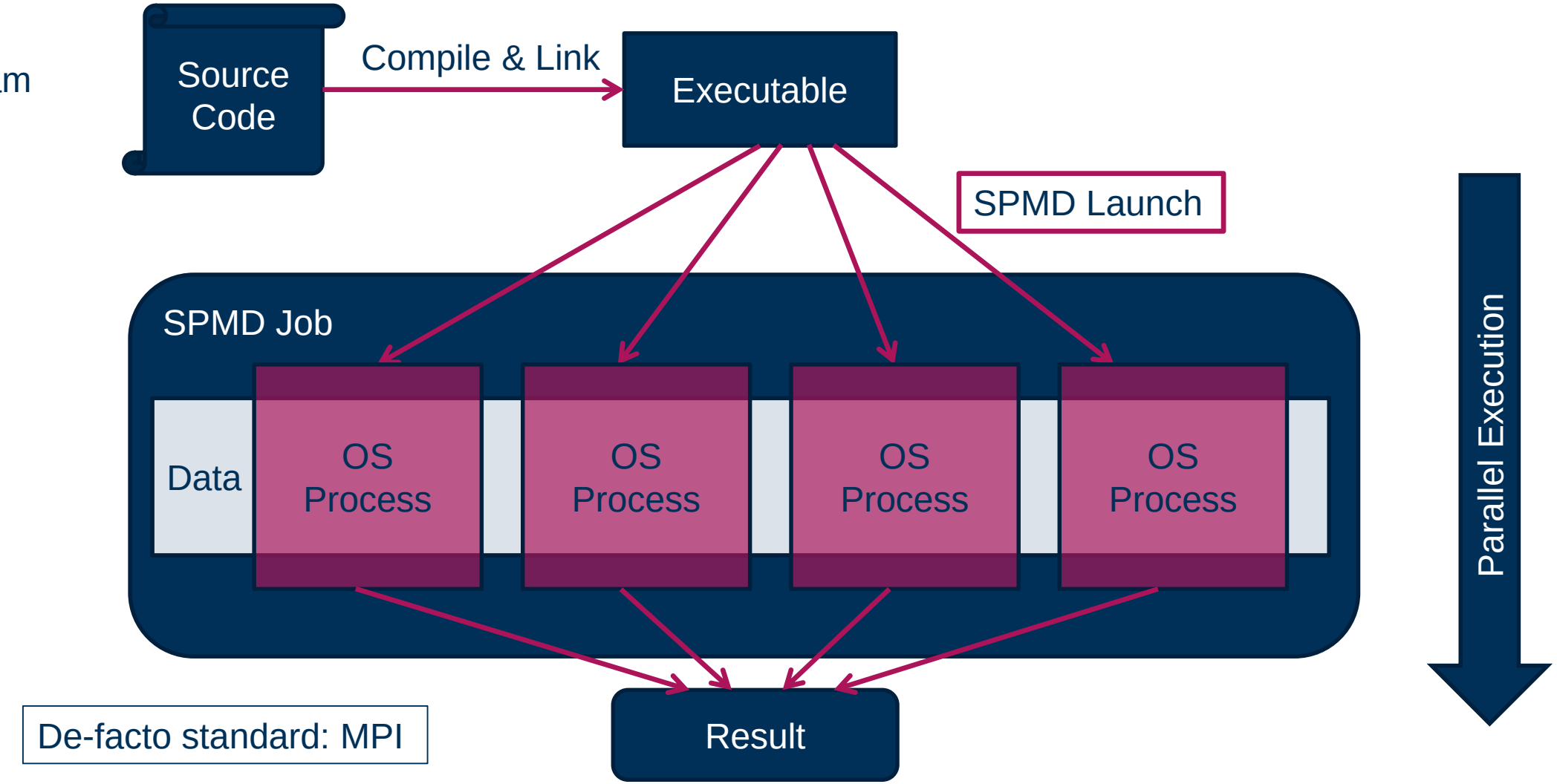
Cluster

- HPC market is dominated by distributed memory multicomputers (clusters)
- Many nodes with no direct access to other nodes' memory



How can I program for a complete cluster?

SPMD:
Single Program
Multiple Data

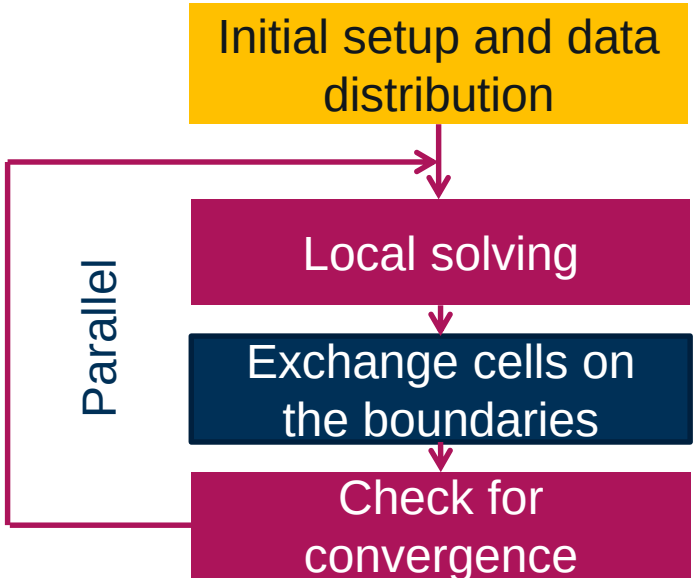


De-facto standard: MPI

System level parallelism

- Example: Domain decomposition in CFD: Mapping of 3D mesh to the processors

- Programming techniques
  - Data parallel approach
    - Distribute data structures
  - Parallel algorithms
  - Explicit data exchange (MPI)

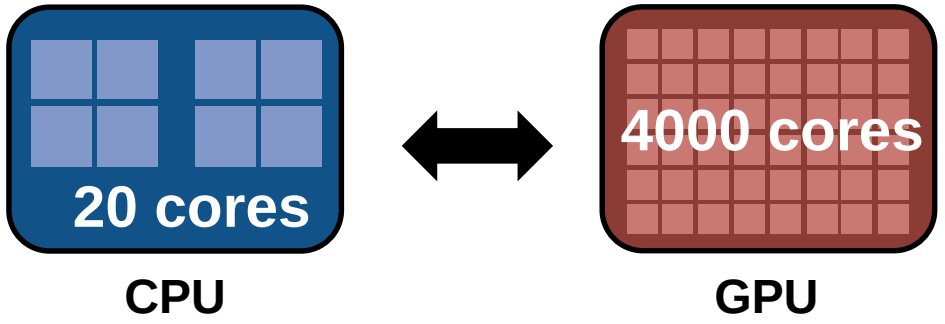


What is an accelerator?

HPC.NRW

What is an accelerator?

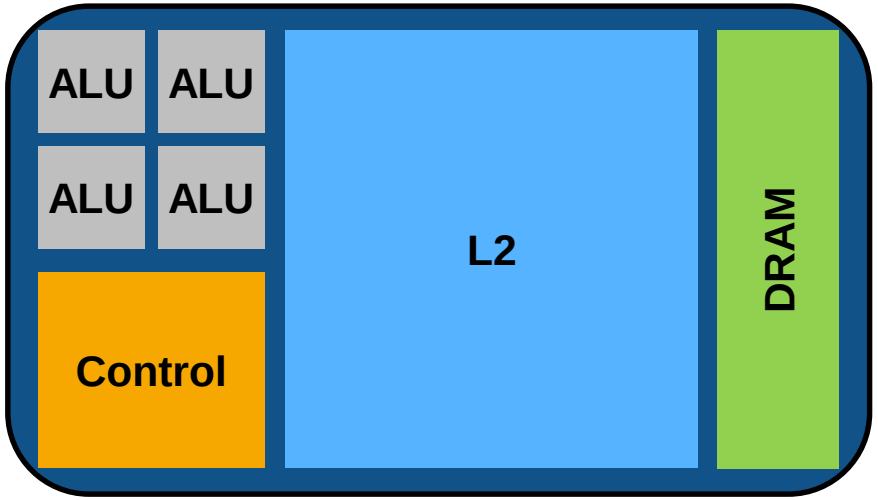
Comparision CPU ⇔ GPU



- GPU-Threads
  - Thousands (“few” on CPU)
  - Light-weight, little creation overhead
  - Fast switching
- Lots of parallelism needed on GPU to get good performance!

Comparision CPU ⇔ GPU /2

– Different designs



CPU

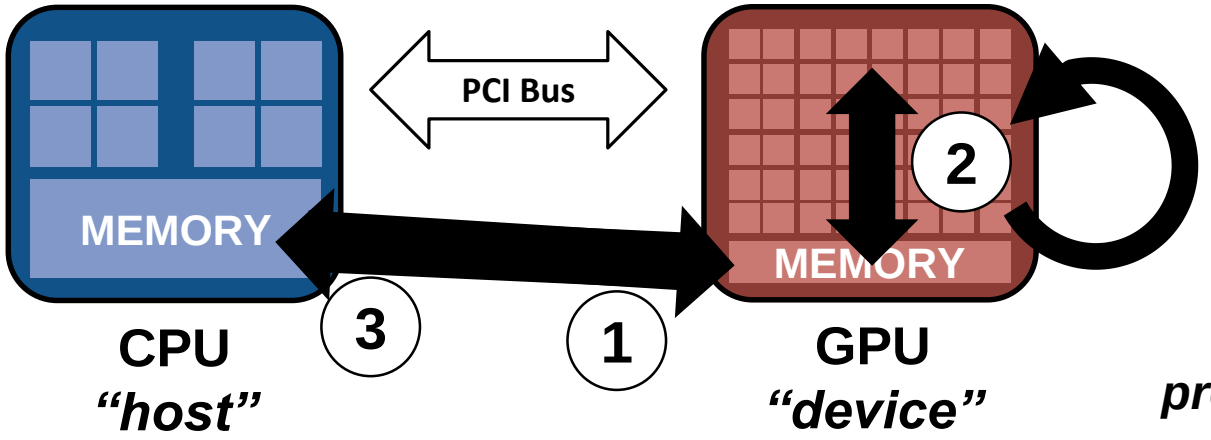
- Optimized for low latencies
- Huge caches
- Control logic for out-of-order and speculative execution
- **Targets on general-purpose applications**

GPU

- Optimized for data-parallel throughput
- Architecture tolerant of memory latency
- More transistors dedicated to computation
- **Suited for special kind of apps**

How can I program a GPU?

– Example:
  Offloading approach



We refer to "discrete GPUs" here.

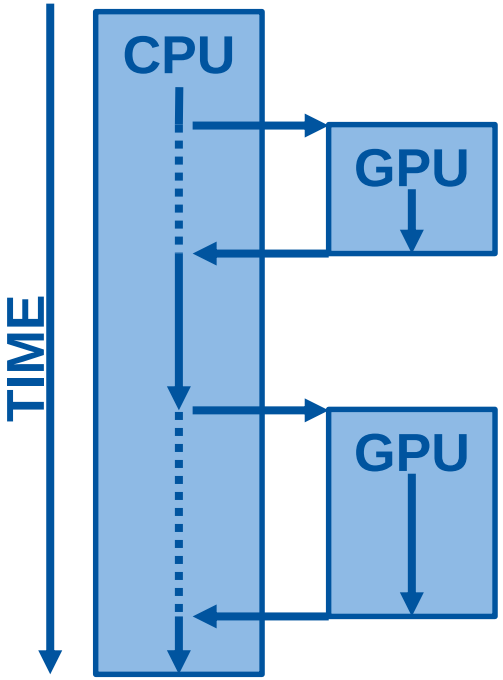
Weak memory model

– Host + device memory = separate entities
– No coherence between host + device
  – Data transfers needed

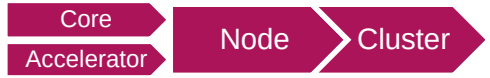
Host-directed execution model

– Copy input data from CPU mem. to device mem.
– Execute the device program
– Copy results from device mem. to CPU mem.

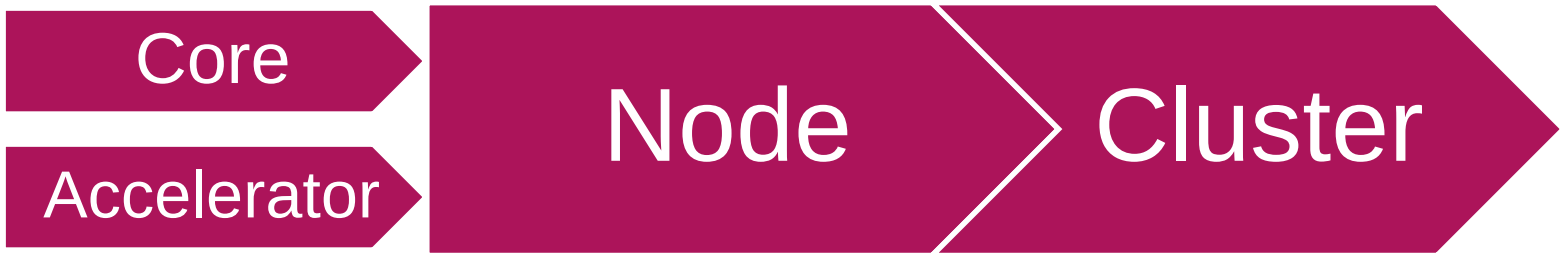
processing flow *(simplified)*



How to use the complete supercomputer

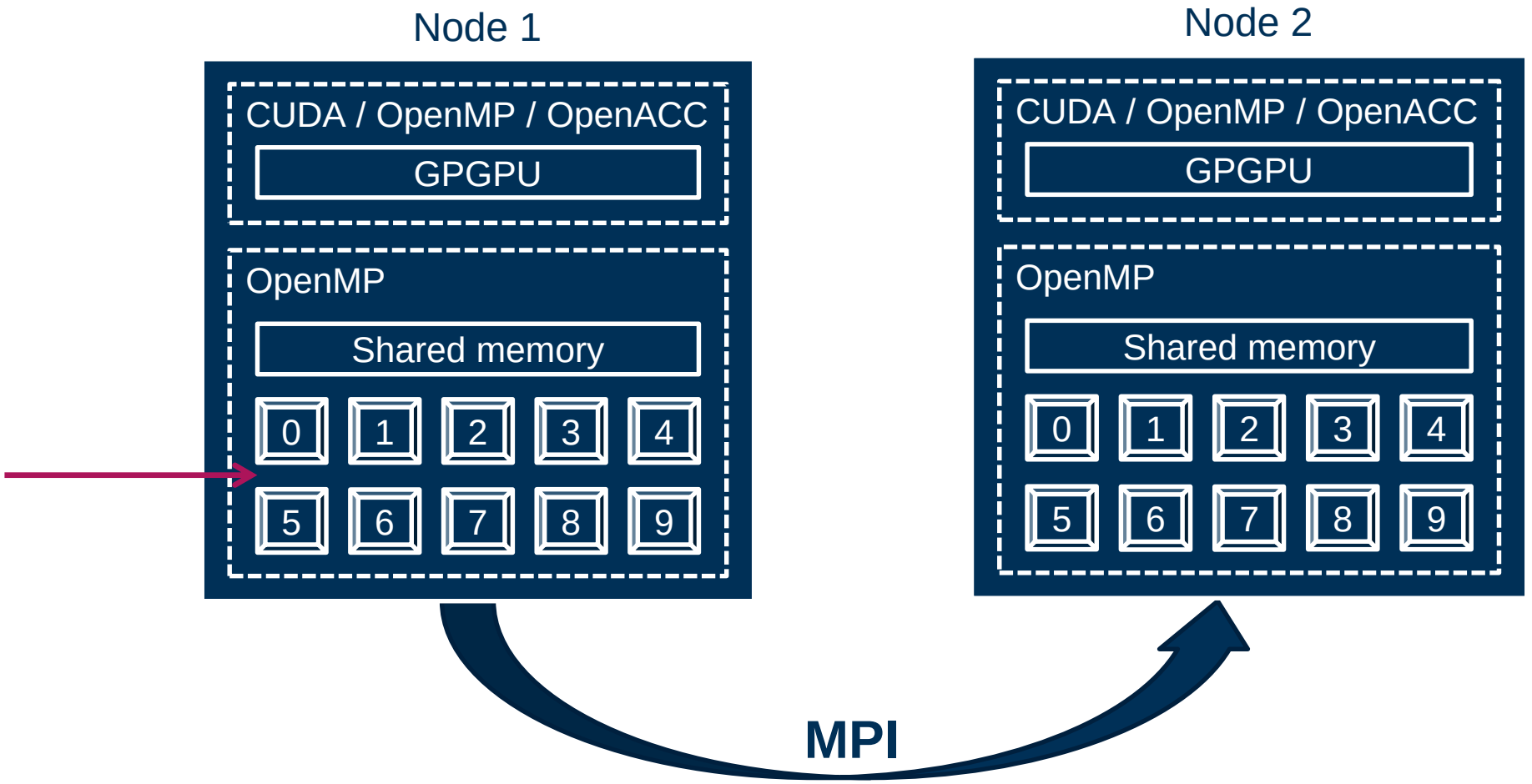


HPC.NRW



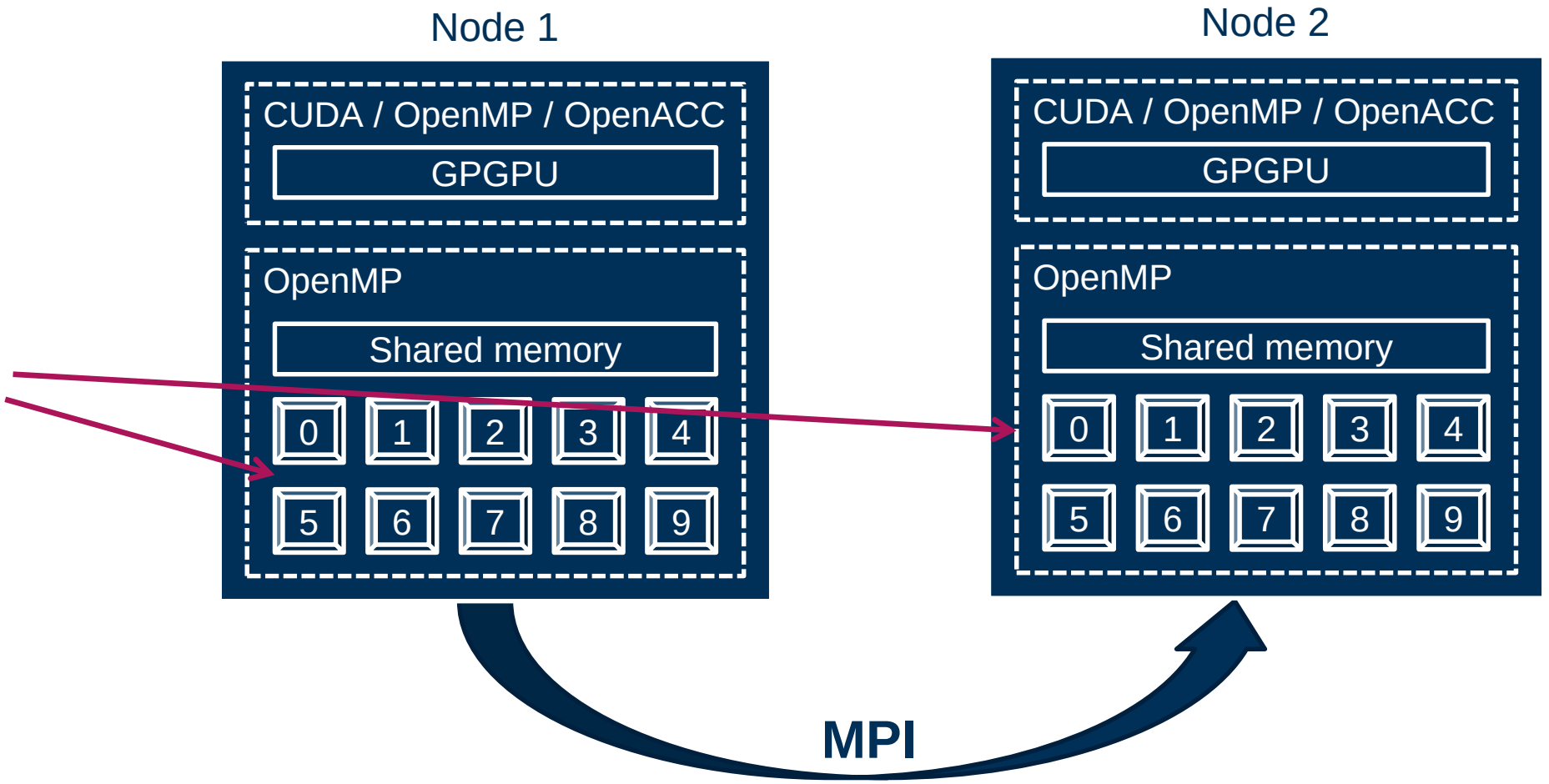
Threads vs. Processes vs. Accelerators

Cores *within* a node: You can use **threads** or **processes** to use all cores. **Threads** working on the same address space, process on their own (virtual) address space.



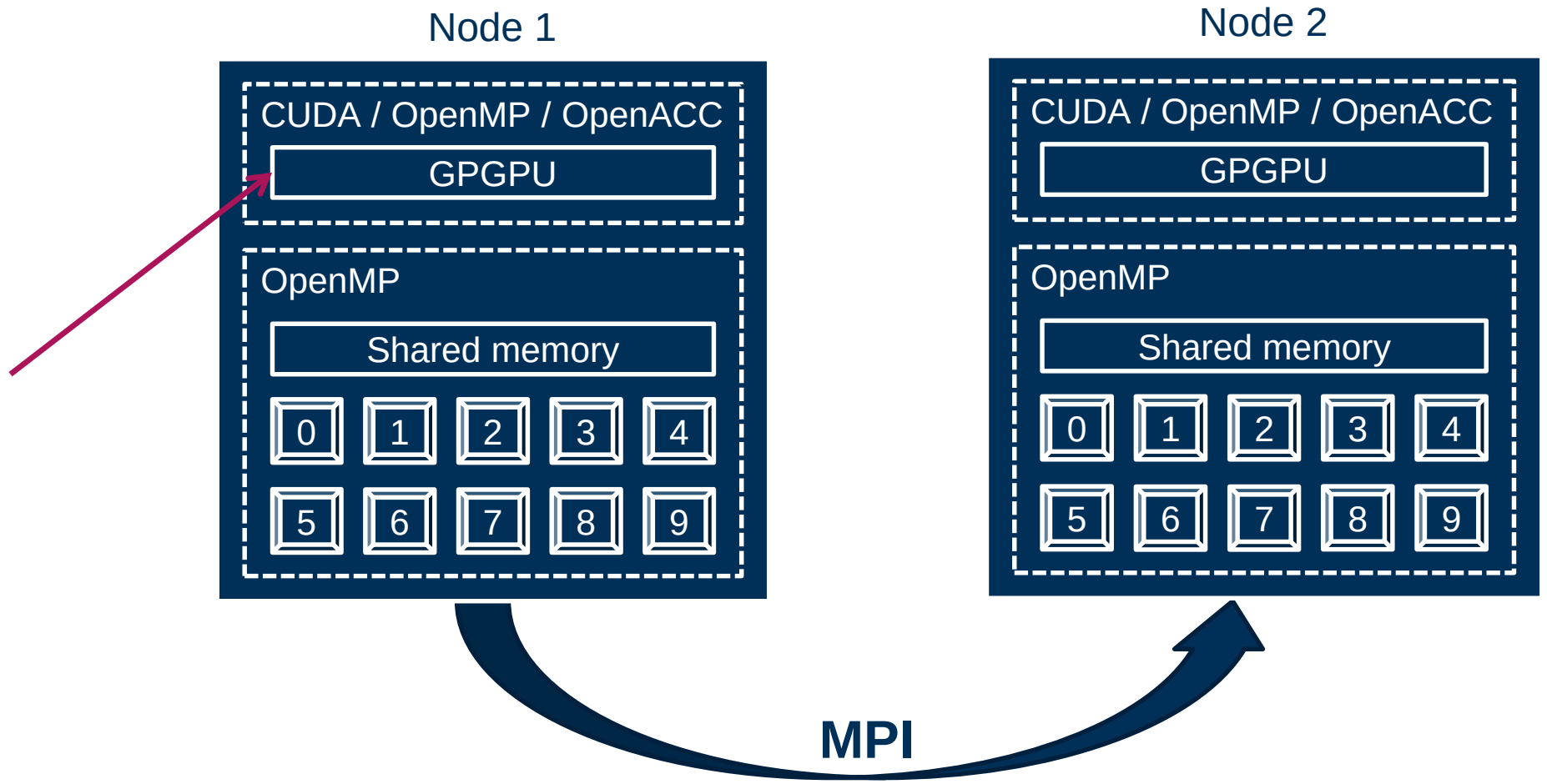
Threads vs. Processes vs. Accelerators

Cores *across* nodes: You can use **processes** to use all cores. Combination with inner node threads possible (“hybrid programming”).



Threads vs. Processes vs. Accelerators

Accelerators: “**Threads**[1]” can be used within an GPGPU. These threads do not use the same memory as the host system.



[1] Depending on vendor: Concept of “thread” is different to CPUs.

HPC.NRW

Questions?