What is parallelization

Overview

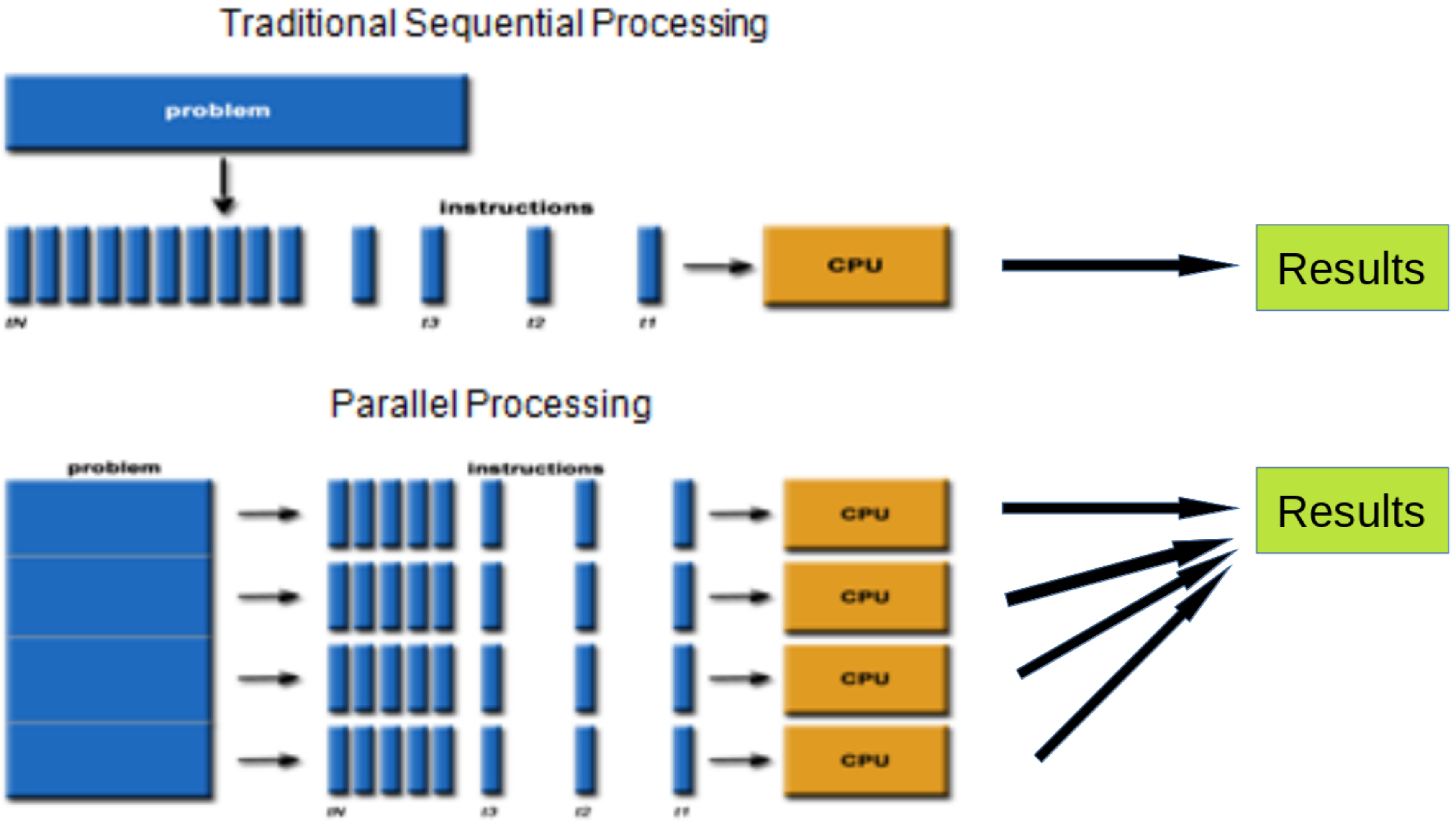
1. Introduction to supercomputers
2. Shared memory computing
3. Distributed memory computing
4. Scaling

Introduction to supercomputers

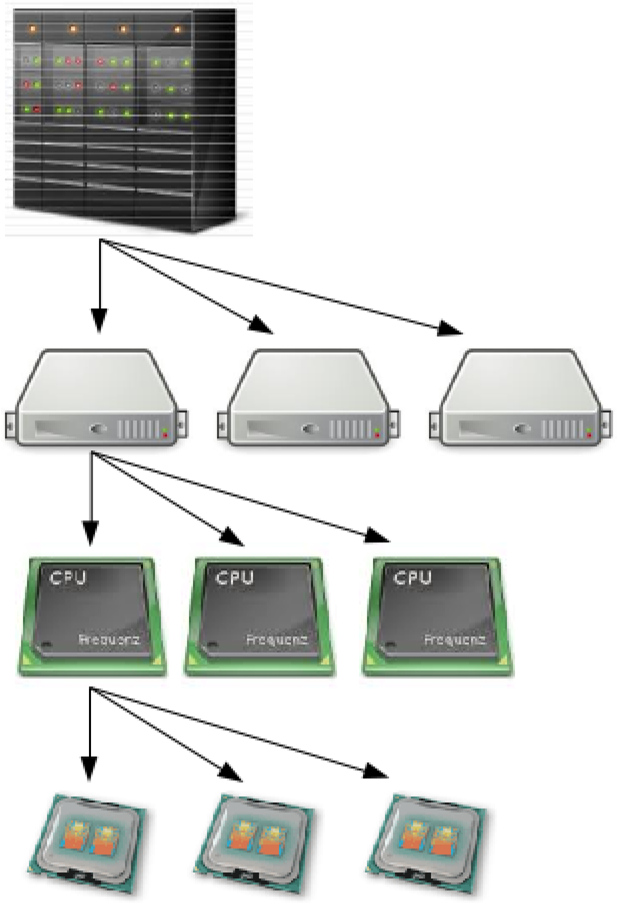
Supercomputers can be useful when...

1. Your problem is too large to be solved in a single computer
2. The solution to your problem is requiring too much time to be solved

Parallel methodology



What is a cluster



Cluster

Nodes

CPUs

Cores

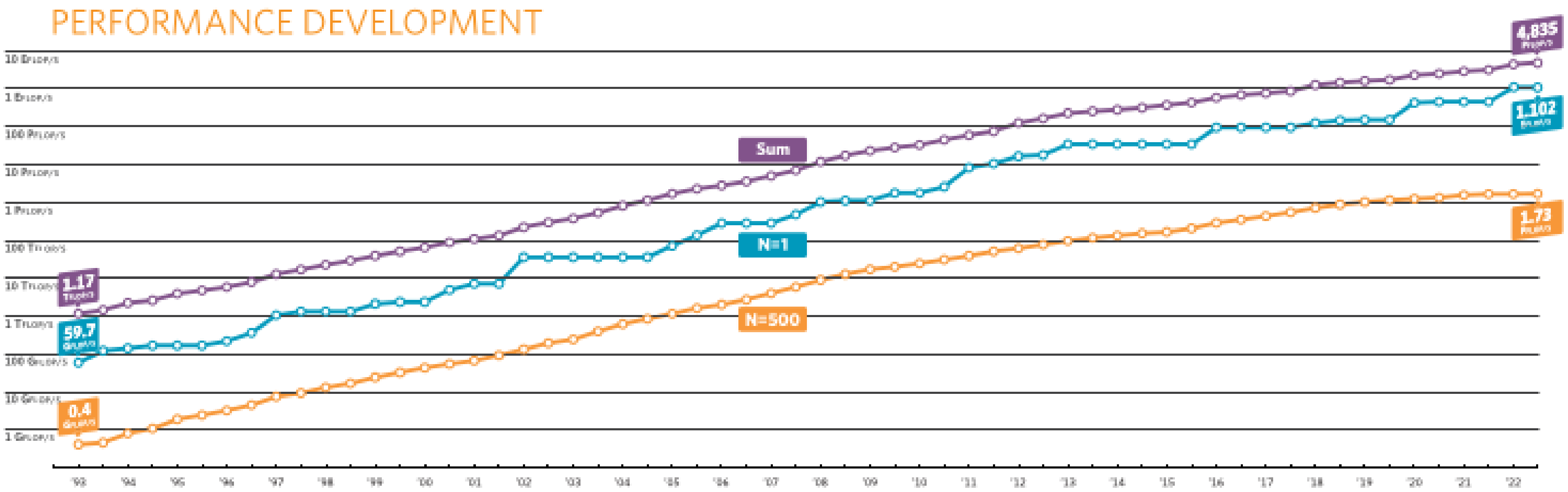
HPC cluster

- HPC cluster is a collection of normal computers connected together

Performance is measured in *floating point operations per second*, **flop/s, FLOPS**

- Fast interconnect to make possible inter-node communications
- Software to manage communications between the nodes
- Workload manager

HPC development



Top 500 list

NOVEMBER 2022

			SITE	COUNTRY	CORES	RMAX PFLOP/S	POWER MW
1	**Frontier**	HPE Cray EX235a, AMD Opt 3rd Gen EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-10	DOE/SC/ORNL	USA	8,730,112	1,102.0	21.1
2	**Fugaku**	Fujitsu A64FX (48C, 2.2GHz), Tofu Interconnect D	RIKEN R-CCS	Japan	7,630,848	442.0	29.9
3	**LUMI**	HPE Cray EX235a, AMD Opt 3rd Gen EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-10	EuroHPC/CSC	Finland	2,174,976	304.2	5.82
4	**Leonardo**	Atos Bullsequana intelXeon (32C, 2.6 GHz), NVIDIA A100 quad-rail NVIDIA HDR100 Infiniband	EuroHPC/CINEC	Italy	1,463,616	174.7	5.61
5	**Summit**	IBM POWER9 (22C, 3.07GHz), NVIDIA Volta GV100 (80C), Dual-Rail Mellanox EDR Infiniband	DOE/SC/ORNL	USA	2,414,592	148.6	10.1

https://www.top500.org (Nov. 2022)

Kebnekaise

Nodes: 603

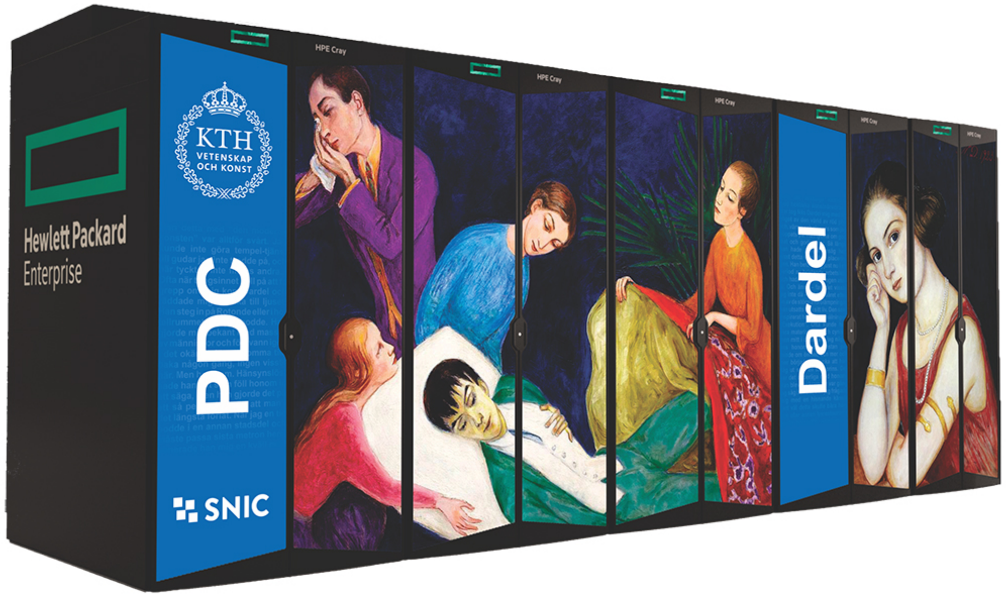
Cores: 19288

Peak performance: 984 TFlops/s

Node configuration (standard)

- Intel Xeon E5-2690v4 (Broadwell)
  - 432 nodes, 128 GB/node
  - 32 nodes, 2xK80 GPU
  - 4 nodes, 4xK80 GPU
  - 20 nodes, 3072 GB/node

Dardel



Nodes: 794

Cores: 101632

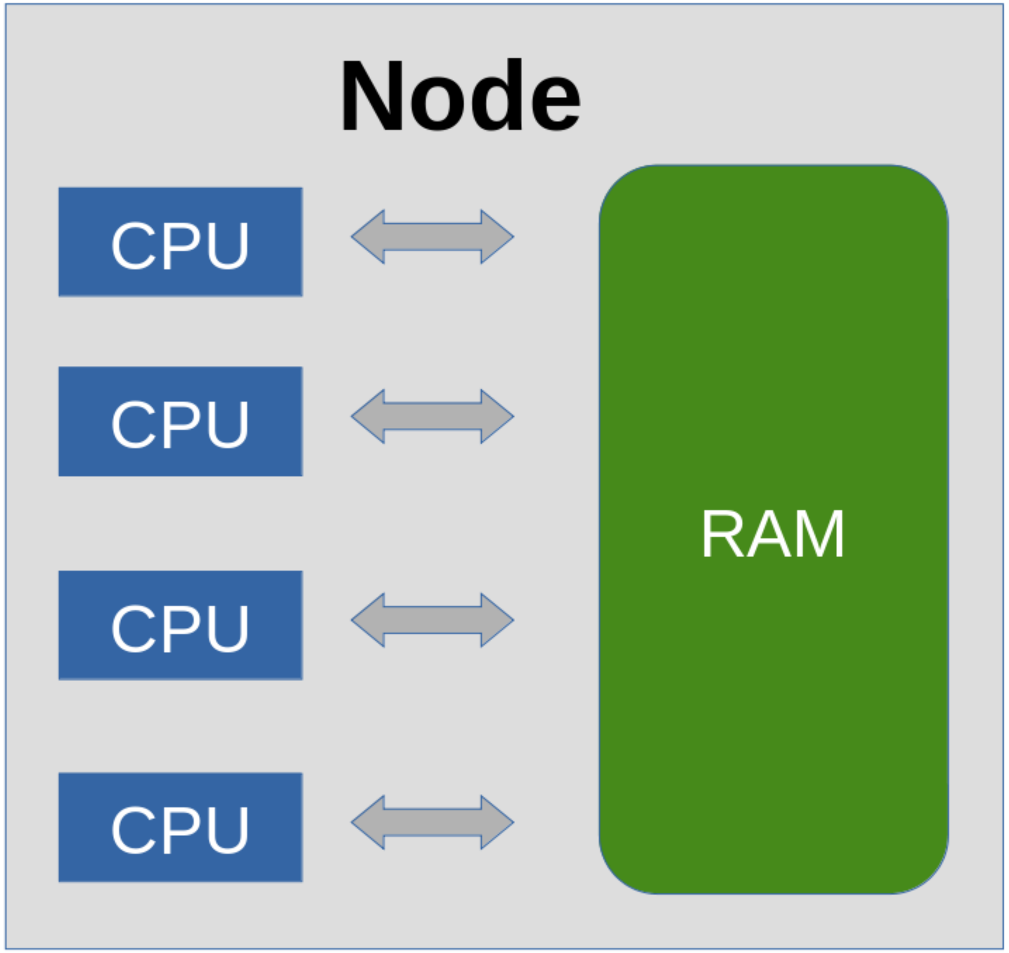
Peak performance: 13.5 PFLOPS

Node configuration

- 2xAMD EPYC™ 2.25 GHz CPU with 64 cores each
- RAM
  - 524 nodes, 256 GB
  - 256 nodes, 512 GB RAM
  - 8 nodes, 1024 GB RAM
  - 6 nodes, 2048 GB RAM
- 4xAMD Instinct™ MI250X GPUs

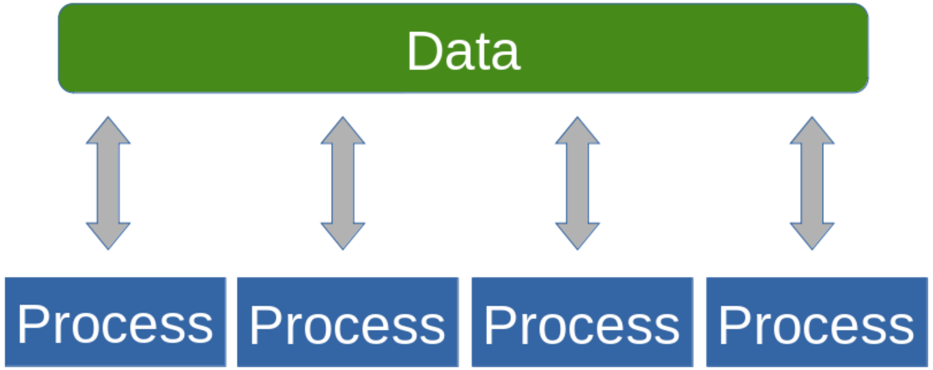
Shared memory computing

Shared memory hardware



Can be executed on a standard computer

Pros and cons of shared memory

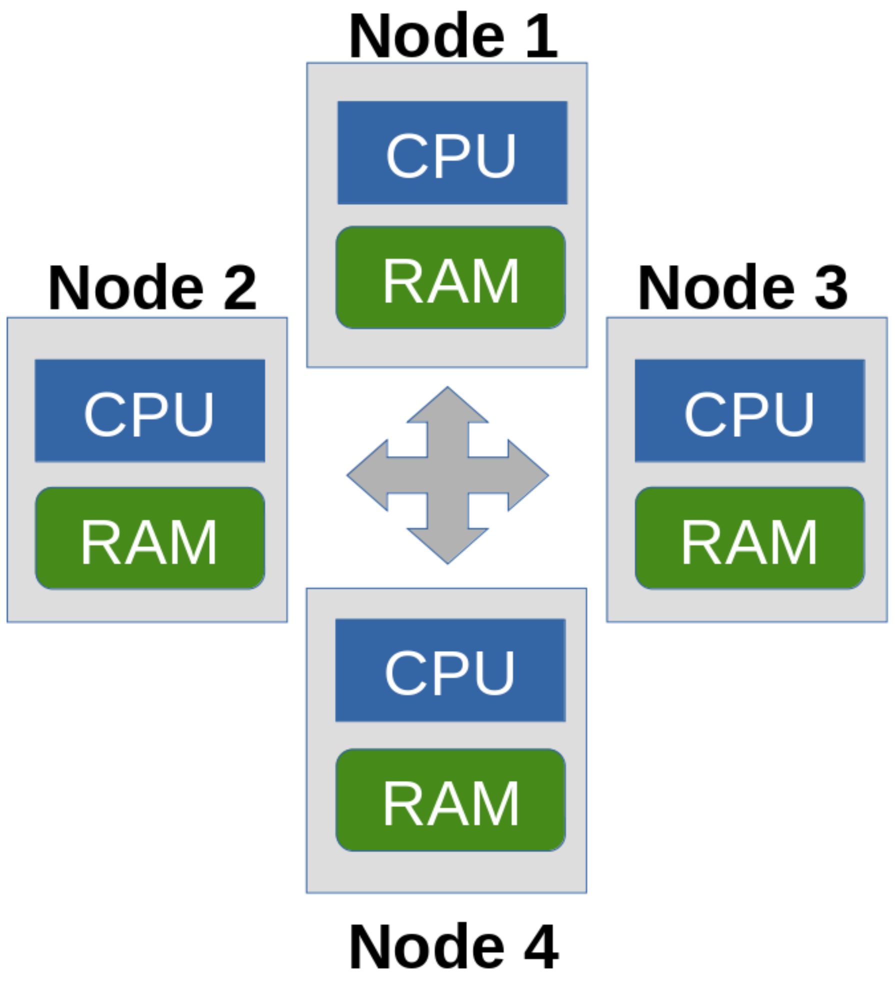


Data is available in same hardware location

1. No need to move data
2. All processing units share the same memory address space
3. It can be difficult to manage concurrent access to the memory (race conditions)

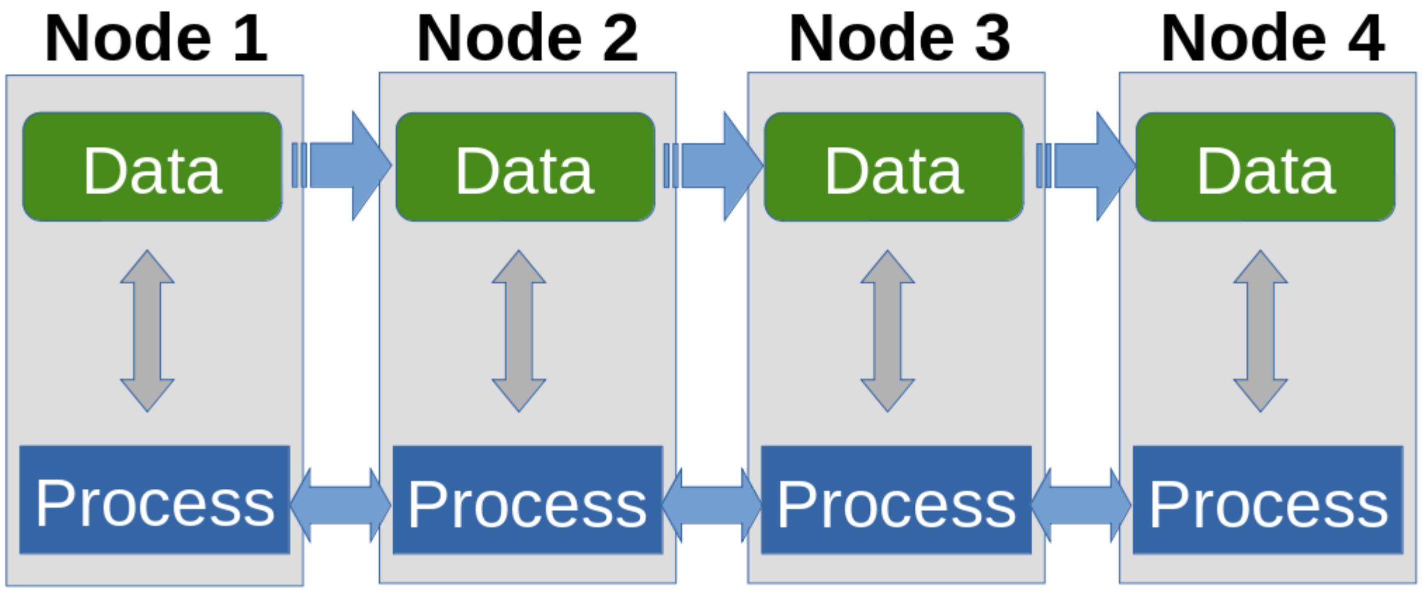
Distributed memory computing

Distributed memory computing



Can be executed on a supercomputer

Distributed memory computing



1. Analysis can be expanded across several nodes
2. Each CPU has access to only its own memory
3. In order to access data in another memory it needs to be transferred
4. Communications needs to be explicitly managed by the programmer

Hybrid parallelism

1. Best of both worlds
2. Communications between nodes (Distributed memory)
3. Computation within each node (Shared memory)

Scaling

"More processes does not always mean faster performance"

- Code cannot be parallelized
- Overhead of creating parallel process take more time than the parallel process itself

The computational dilemma

Easibility, Flexibility

Performance

High level programming

Python, matlab, **R**, JavaScript

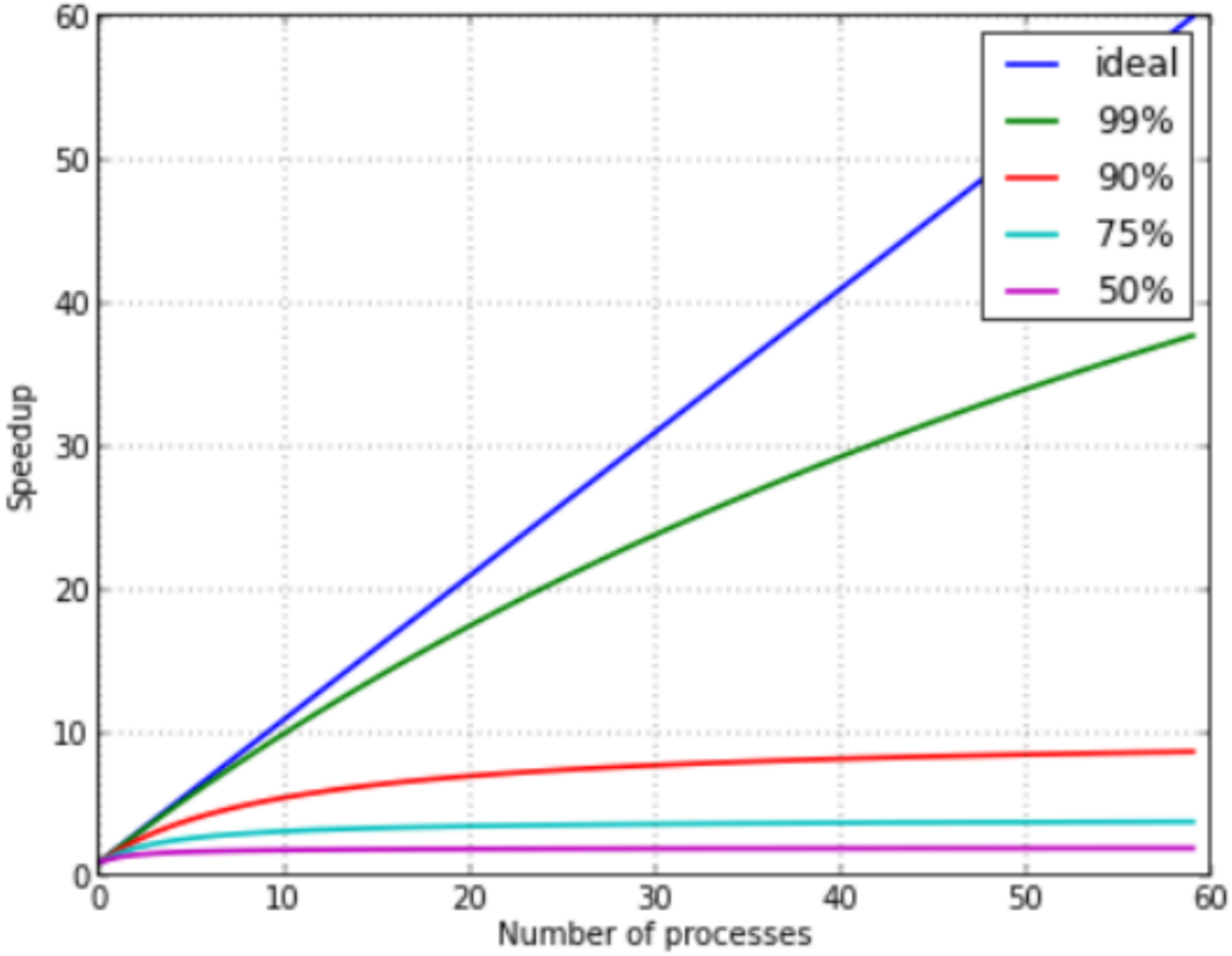
Modular programming

Python+C/C++

Low level programming

C/C++, Fortran, Assembler

Scaling



Speedup
Number of processes
ideal
99%
90%
75%
50%
0
10
20
30
40
50
60

Ahmdal's law

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

- S is the theoretical speedup of the task
- *s* is the number of processes
- *p* is the proportion of executation time that benefits from improved resources

Example

Executing a code takes 10 h, where 8 h can be parallelised. How much speedup do we get using 15 parallel processes?

$p=8/10=0.8$, $s=15$

$S=1/((1-0.8)+0.8/15)=\mathbf{3.9}$