CFD on HPC - OpenFOAM

Introduction to supercomputers, supercomputing world, benefits for the research and industrial needs.

Description

The three-day workshop includes a presentation of the basic and advanced OpenFOAM usage, with explanation of programming of new sets within the framework. The basic part includes steps of making a mesh, setting boundary / initial conditions, running the problem on the HPC system and reviewing the results. The advanced part covers more complex cases with advanced application usage. What is not covered in the OpenFOAM can be upgraded within the software environment with the implementation of your own code. Showcases and hands-on cases will be presented.

Gained skills

- Be able to connect to HPC@ULFS with NoMachine client and work in HPC Linux environment
- Understand the theoretical background of the Computational Fluid Mechanics (CFD), especially of the Finite Volume Method (FVM)
- Be able to set up CFD mesh using different open source programs for CFD mesh design (OF – Block Mesh, GMSH)
- Be able to setup complete OF case (mesh, pysical model, inital and boundary conditions, ...)
- Be able to setup and run various OF cases in parallel on an HPC cluster
- Be able to preview and post-process OF results

Lecturer

Aleksander Grm

Aleksander Grm graduated with a Bachelor's degree in Physics from the Faculty of Mathematics and Physics at the University of Ljubljana. He then completed a Master's degree in Applied Mathematics at ICTP/SISA in Trieste, Italy. After the MSc, he continued his studies at the University of Kaiserslautern in Germany and obtained a PhD in Industrial Mathematics. After the PhD, he worked partly in academia and fully in industry. In 2014, he moved to the University of Ljubljana to work in basic and applied research and to teach young people mechanics and mathematics at the engineering level.

Agenda

Day 01, 09 Dec 2024		
Beginning	**End**	**Description**
9:00	**10:30**	Introduction to supercomputers, supercomputing world, benefits for the research and industrial needs.
10:30	**12:00**	Introduction to Computational fluid dynamics (CFD)
12:00	**13:00**	Setting up environment for OpenFOAM, description of the basic program environment and running of basic programs.

Agenda

Day 02, 10 Dec 2024		
Beginning	**End**	**Description**
9:00	10:30	Basic usage of OpenFOAM I
10:30	12:00	Hands-on with OpenFOAM I

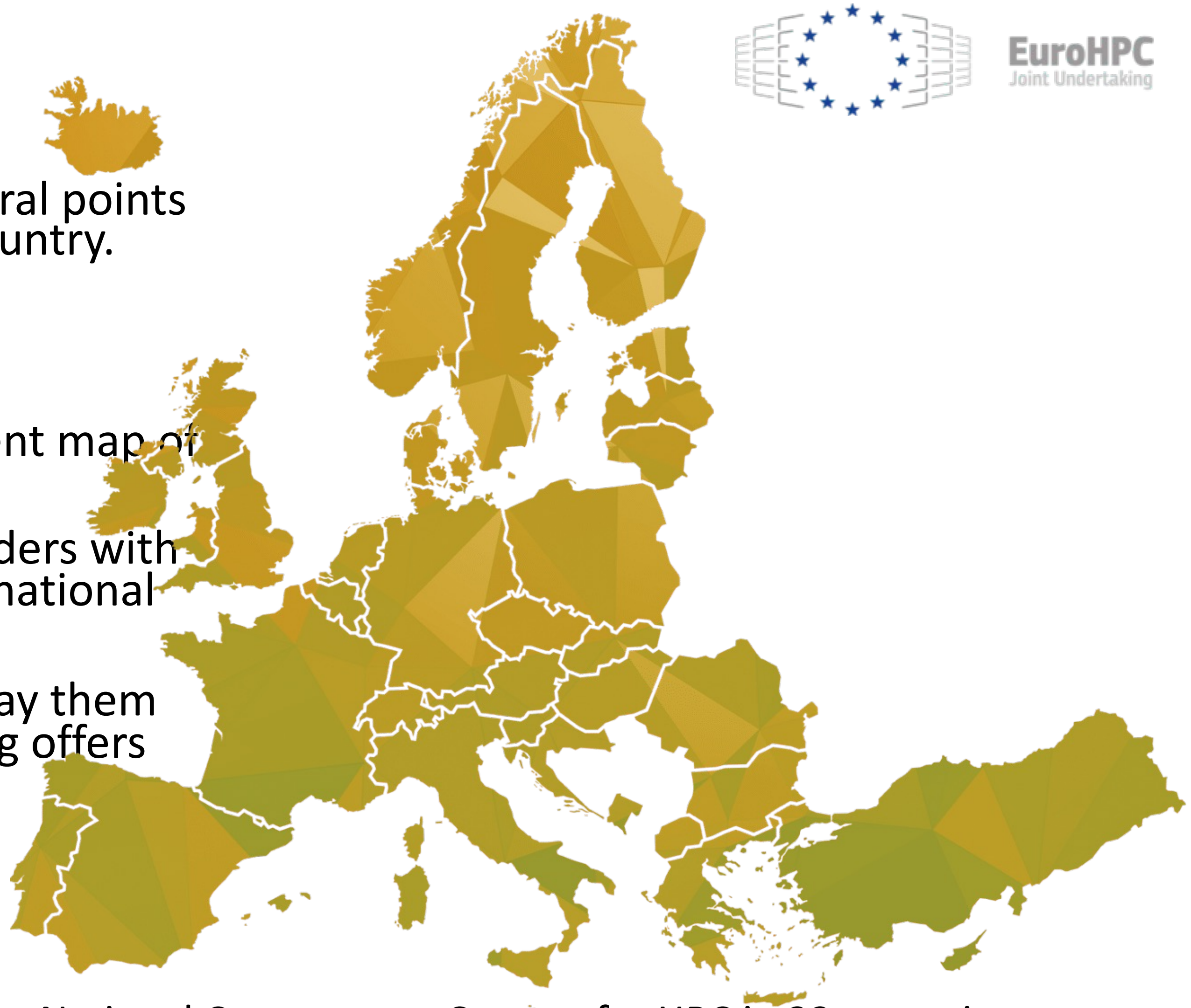
Day 03, 11 Dec 2024		
Beginning	**End**	**Description**
9:00	10:30	Basic usage of OpenFOAM II
10:30	12:00	Hands-on with OpenFOAM II

About EuroHPC

The National Competence Centres (NCCs) are the central points of contact for HPC and related technologies in their country.

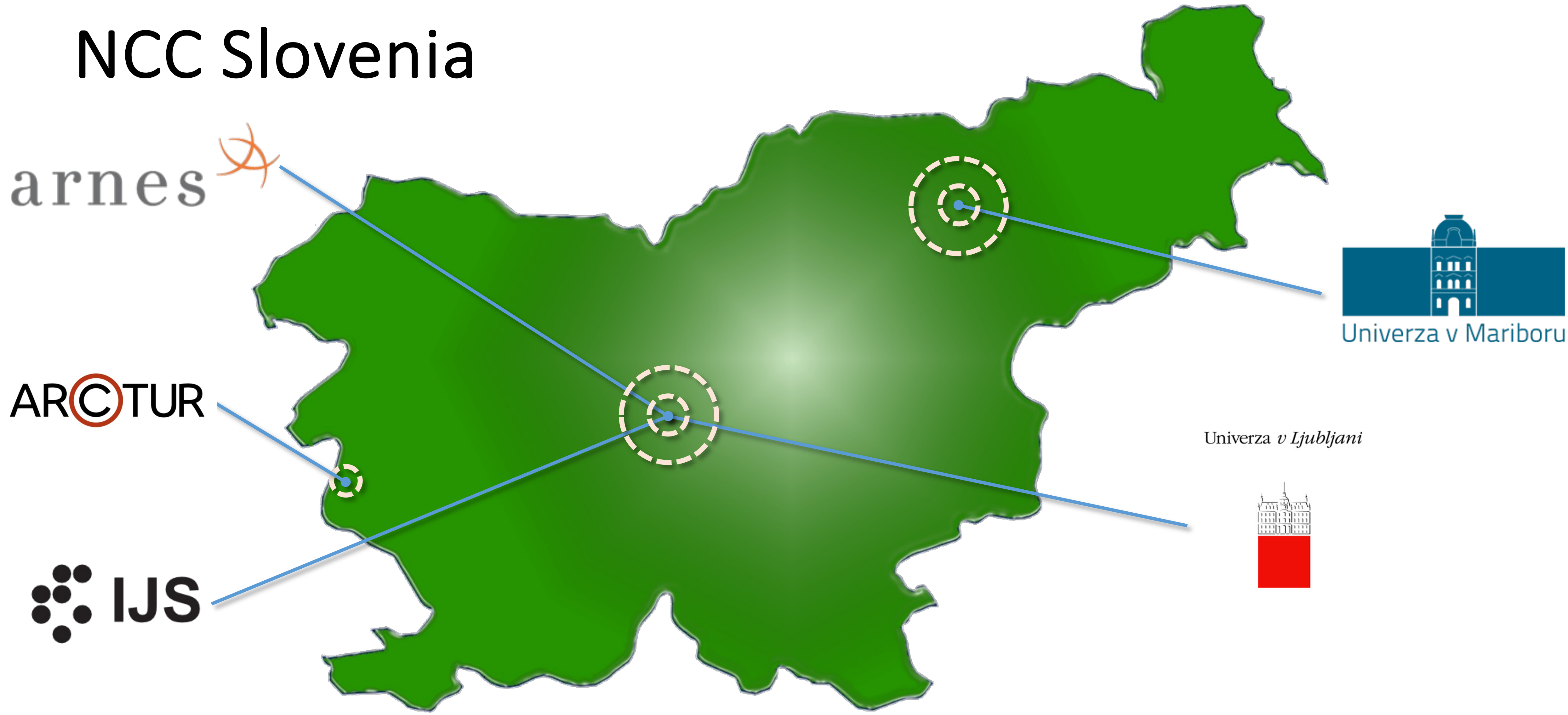
Their mission:

- Develop and display a comprehensive and transparent map of HPC competences and institutions in their country
- Act as a gateway for industry and academia to providers with suitable expertise or relevant projects, may that be national or international
- Collect HPC training offers in their country and display them in a central place together with international training offers collected by other NCCs
- Foster the industrial uptake of HPC

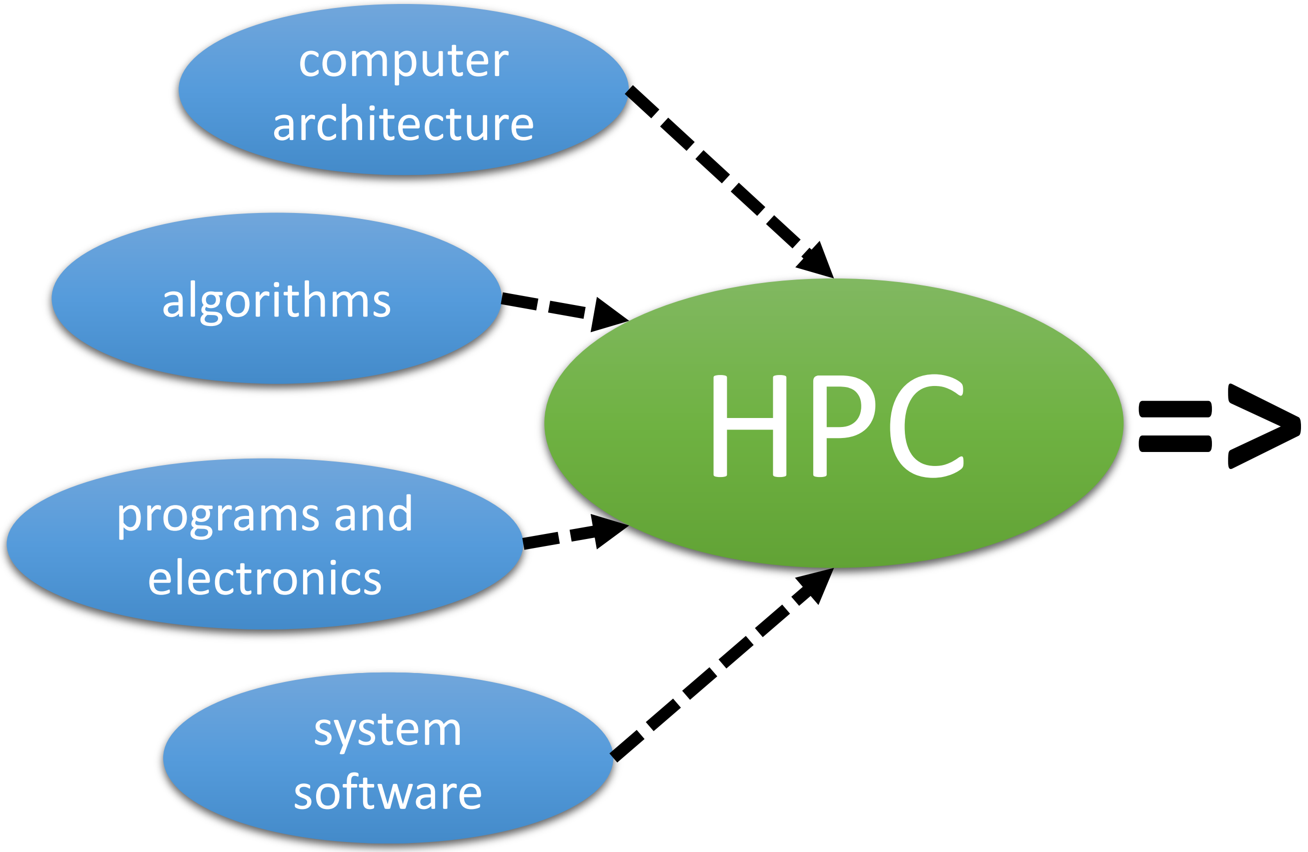


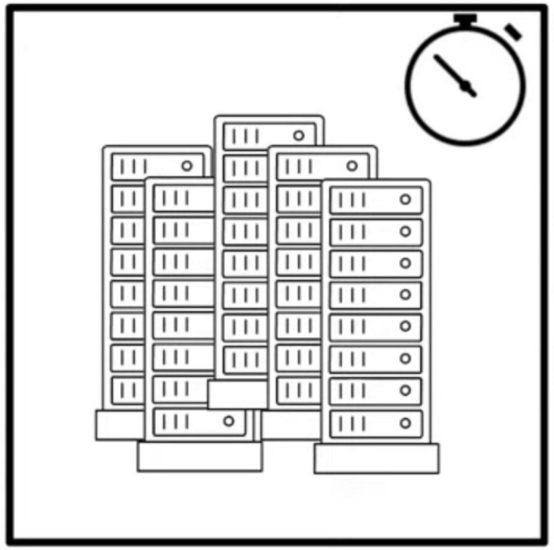
National Competence Centres for HPC in 32 countries

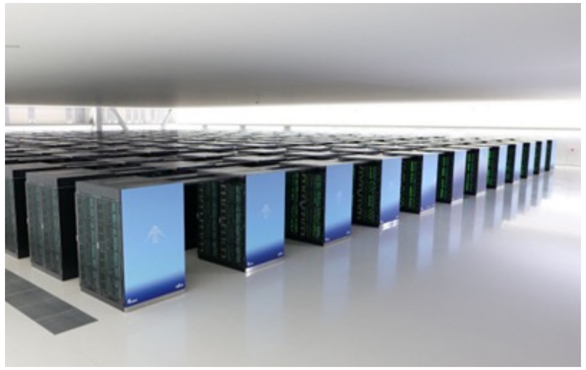
NCC Slovenia



What is High Performance Computing (HPC)



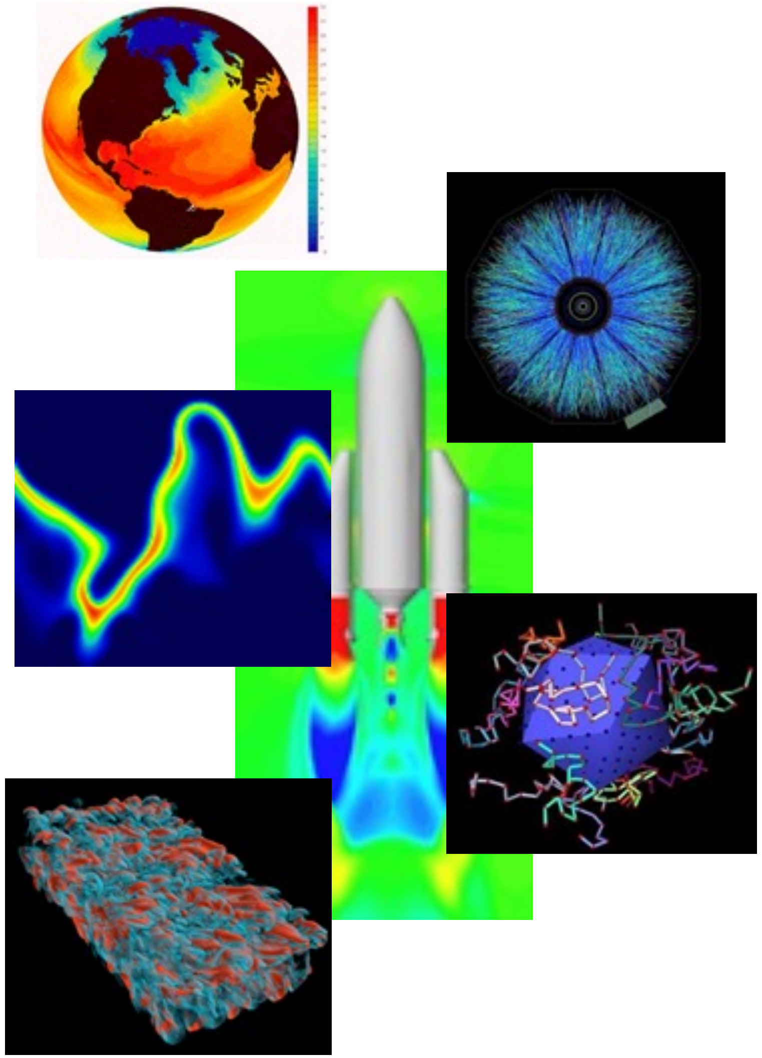




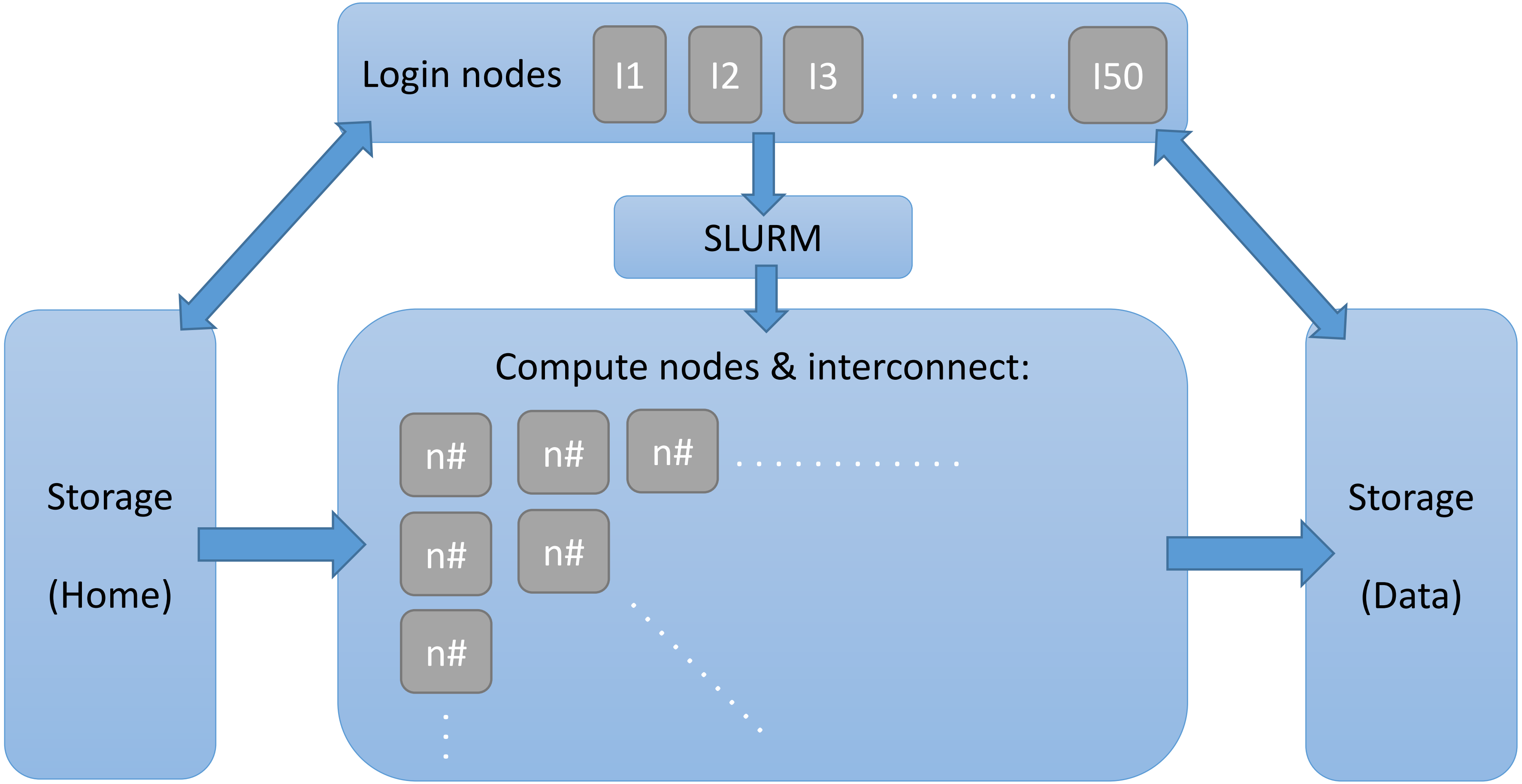
1010101010101010101010100011
10101010100001011100010111000
1010101010101010101010100011
10101010100001011100010111000

Area of usage

- **Weather, Climatology, Earth Science**
  - degree of warming, scenarios for our future climate.
  - understand and predict ocean properties and variations
  - weather and flood events
- **Astrophysics, Elementary particle physics, Plasma physics**
  - systems, structures which span a large range of different length and time scales
  - quantum field theories like QCD, ITER
- **Material Science, Chemistry, Nanoscience**
  - understanding complex materials, complex chemistry, nanoscience
  - the determination of electronic and transport properties
- **Life Science**
  - system biology, chromatin dynamics, large scale protein dynamics, protein association and aggregation, supramolecular systems, medicine
- **Engineering**
  - complex helicopter simulation, biomedical flows, gas turbines and internal combustion engines, forest fires, green aircraft,
  - virtual power plant



Components of a HPC cluster

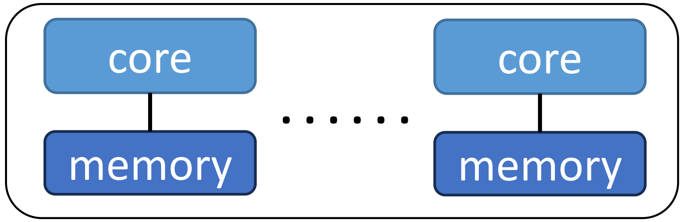


Login nodes ↕ **Compute** nodes

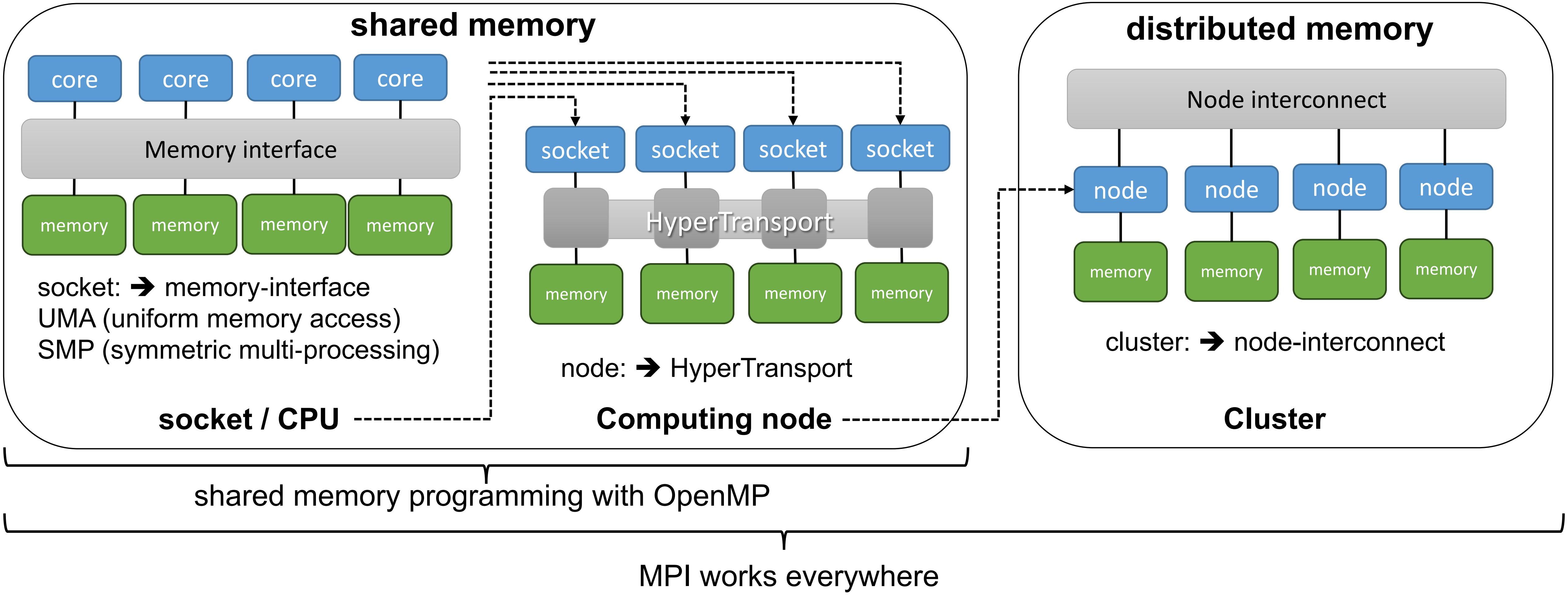
Shared (login, storage) ↕ **Exclusive** (compute nodes)

Parallel hardware

Serial computing



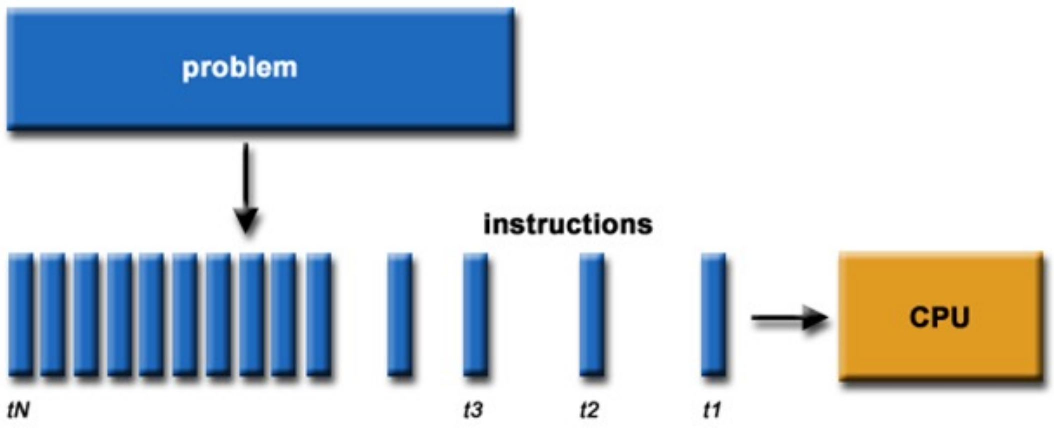
Parallel hardware



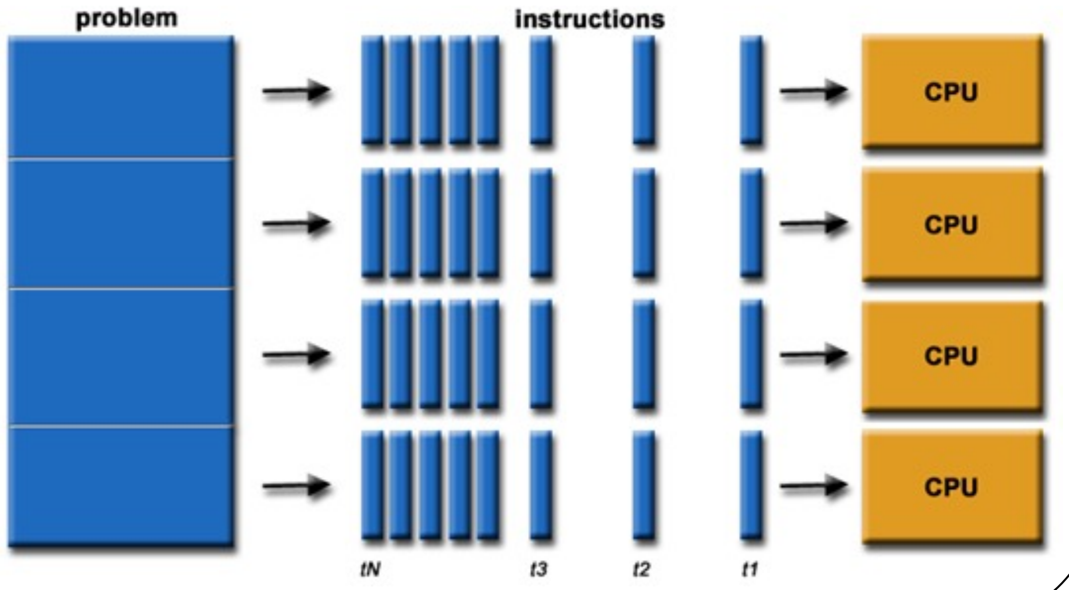
A simple parallel process

Problem discretisation

Serial computing



Problem paralelisation



Problem scalability

- Parallel portion of the code determines code scalability
- Amdahl's law:

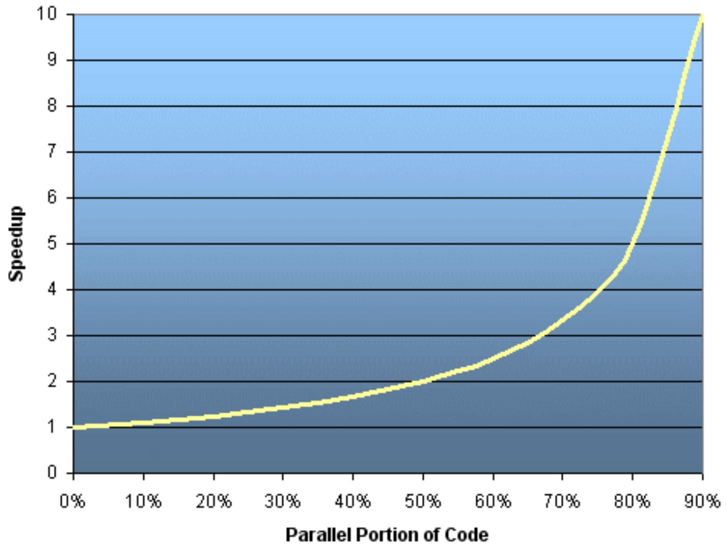
neglecting time for communication
neglecting load imbalance

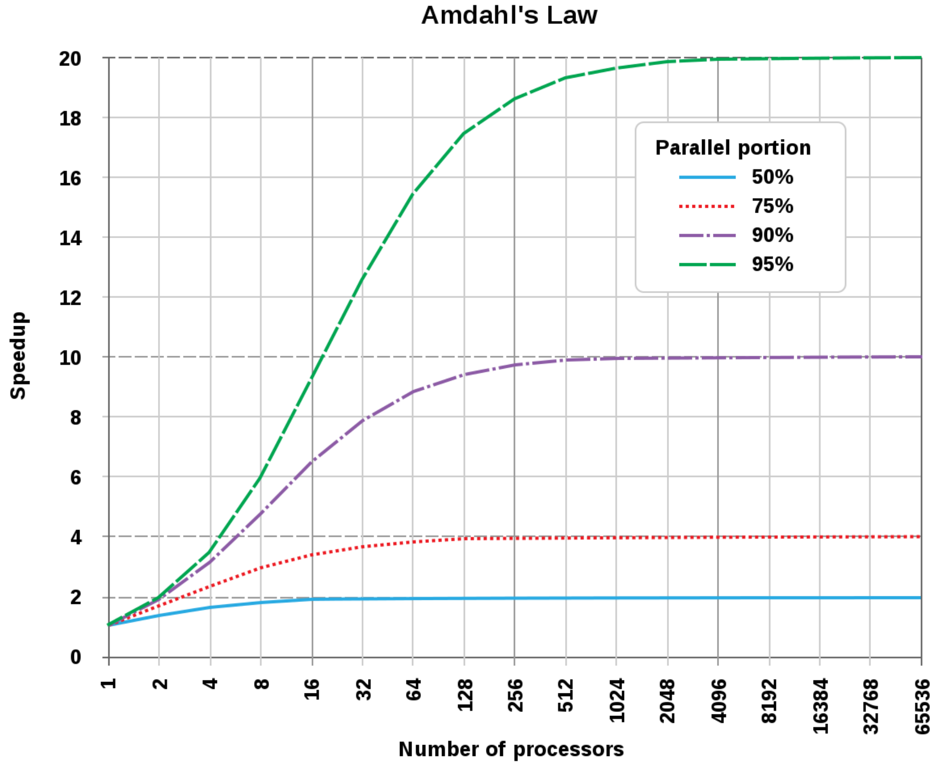
$$S_p = T_{serial} / T_{parallel,p} = 1 / (f + (1-f) / p)$$

Speedup is limited:

$$S_p < 1 / f$$

f ... sequential part of code
p … #processors

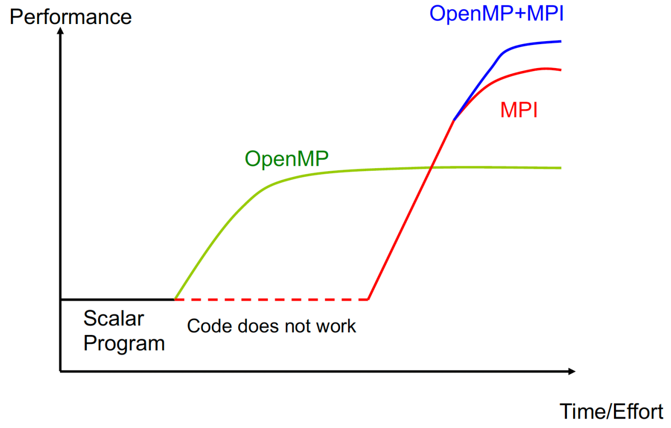




Source: Wikipedia
https://en.wikipedia.org/wiki/Amdahl%27s_law

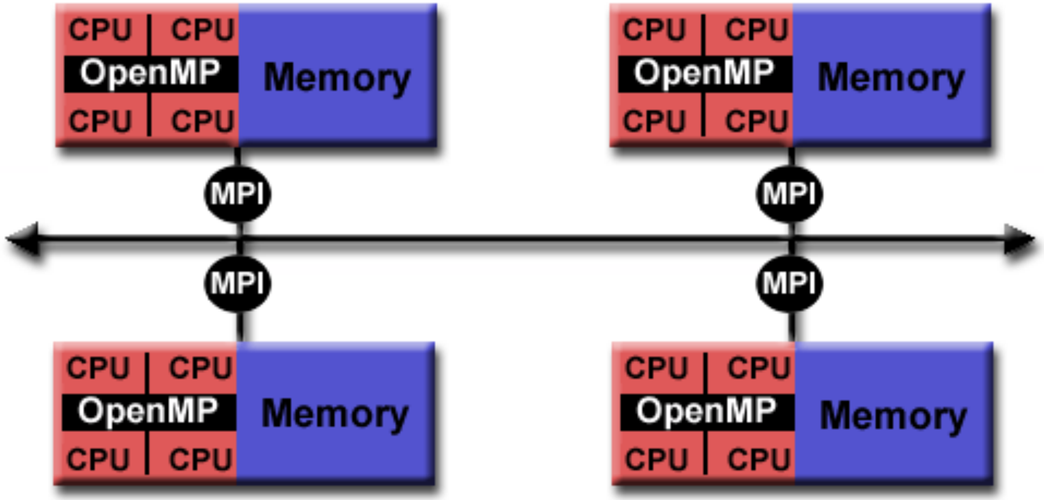
Parallel programming models

- **OpenMP** – automatic paralelization
- Distributed memory model = **Message parsing Interface (MPI)** – manual parallelization needed
- **Hybrid model OpenMP/MPI**



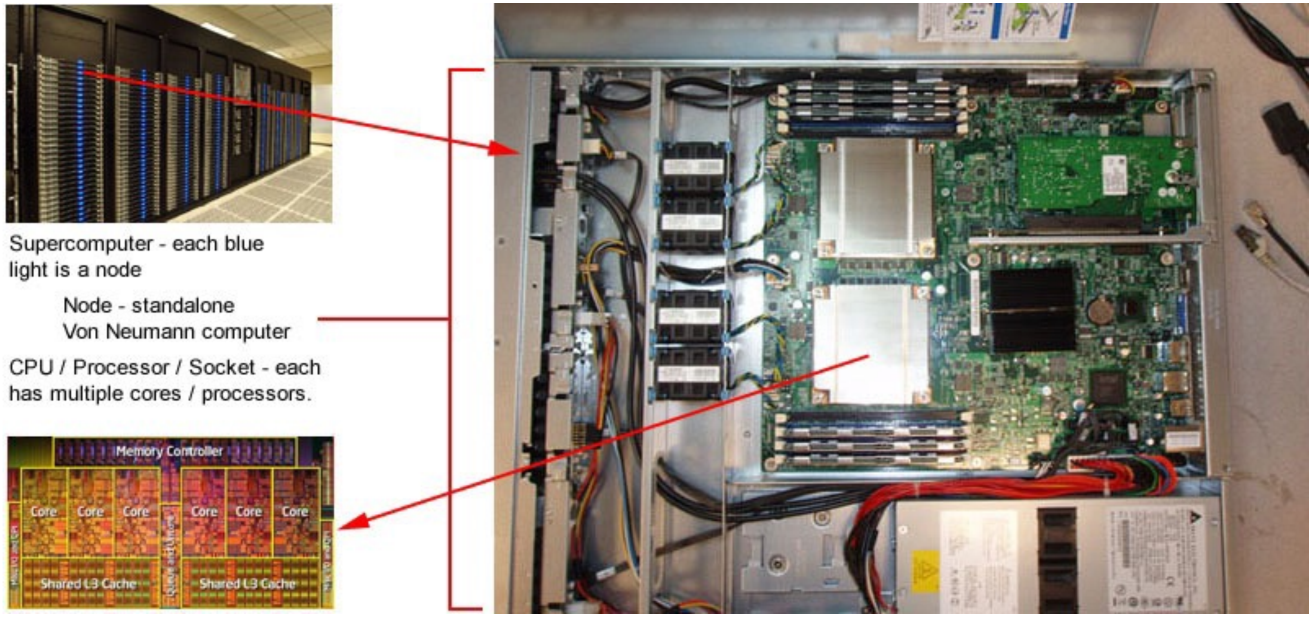
Multi-threading is needed to exploit modern hardware platforms:

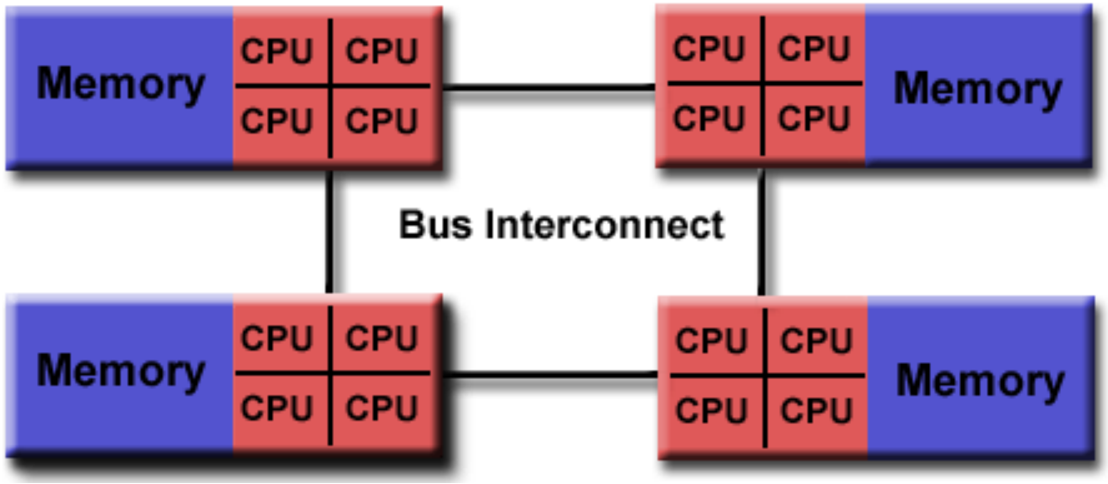
- Several CPUs together within one ccNUMA node
- Several cores per CPU
- Hyperthreading



Logical view of a computing node

- Need to know computing arhitecture
- Interconnect bus for sharing memory between processors (NUMA interconnect)

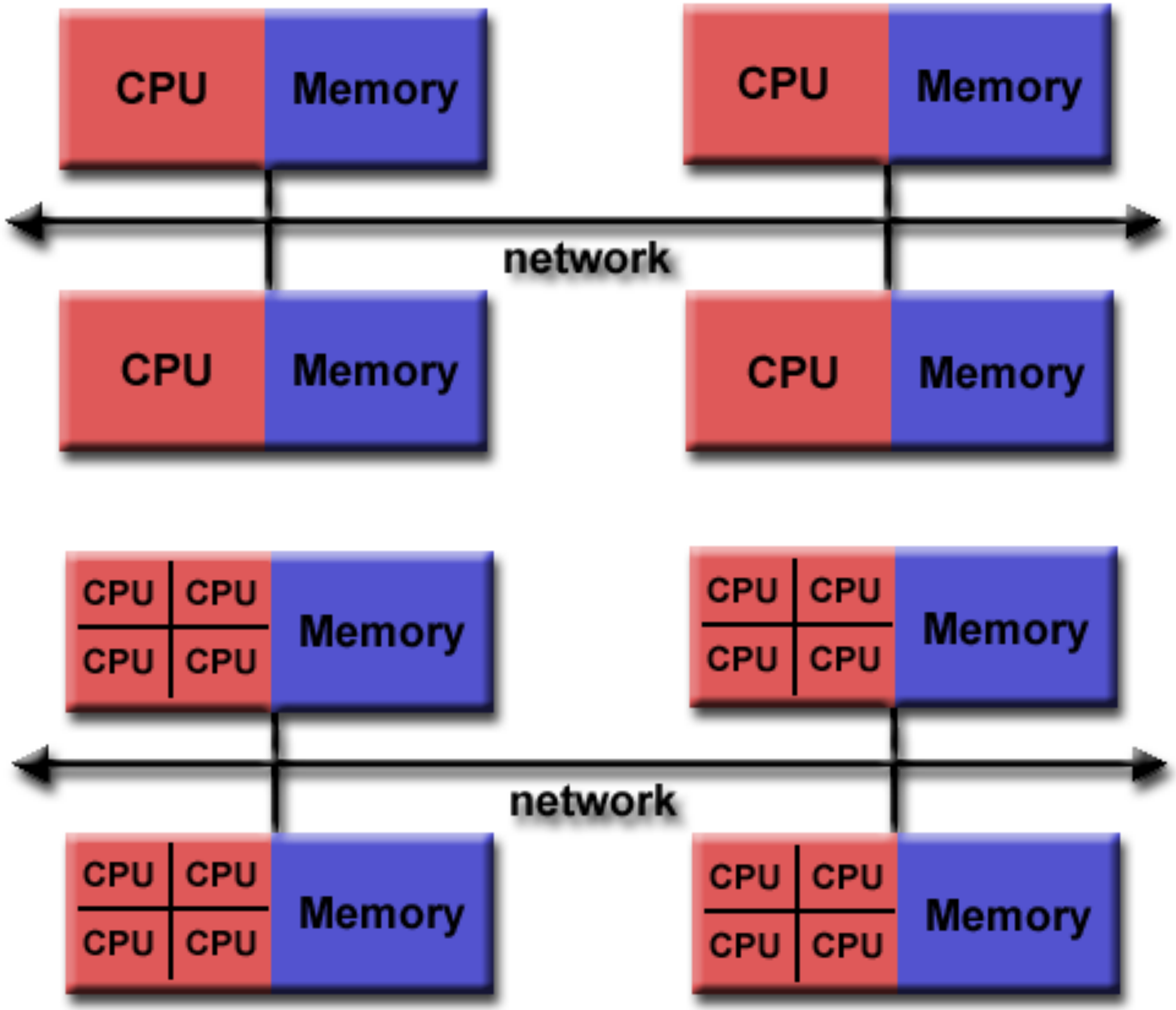




Nodes interconnect

- Distributed computing
- Many nodes exchange messages on
  - High speed
  - Low latency interconnect

Such as **Infiniband**



Development of parallel codes

- Good understanding of the problem being solved
- How much of the problem can be run in parallel
- Bottleneck analysis and profiling
- We optimize and parallelize parts that consume most of the computing time
- Problem needs to be dissected into parts functionally and logically

Interprocess communications

- *Having little an infrequent communication between processes is the best*
- *Determining the largest block of code that can run in parallel and still provides scalability*
- *Basic properties*
  - *response time*
  - *transfer speed - bandwidth*
  - *interconnect capabilities*

Direct or iterative solver

- *We are solving a set of matrix equations of the form [K]{u} = {f}. Here [K] is referred to as the stiffness matrix; {f} as the force vector and {u} as the set of unknowns.*
  - *Several milions of unknowns*
  - *Lot of zeros in K*
- *Direct solvers: Multfront, MUMPS, and LDLT, Pardiso, ...*
- *Iterative solvers: PETSc and GCPC, ...*

Computer Aided Engineering open source tools:

- *CAD/CAM:*

  Salome, Freecad, OpenSCAD, LibreCad, Pycam, Camotics, dxf2gcode & Cura

- *FEA, CFD & multiphysic simulation:*

  *Salome-Meca / Code-Aster, SalomeCFD/Code-Saturne, HelyxOs/**OpenFOAM**, Elmer FEM, Calculix with Launcher & CAE GUI, Impact FEM, MBDyn, FreeFEM, MFEM, Sparselizard*

- *Meshing, pre-post, & visualization:*

  Salome, Paraview, Helyx-OS, Elmer GUI, VoxelMesher, Tetgen, CGX, GMSH

HPCFS - about

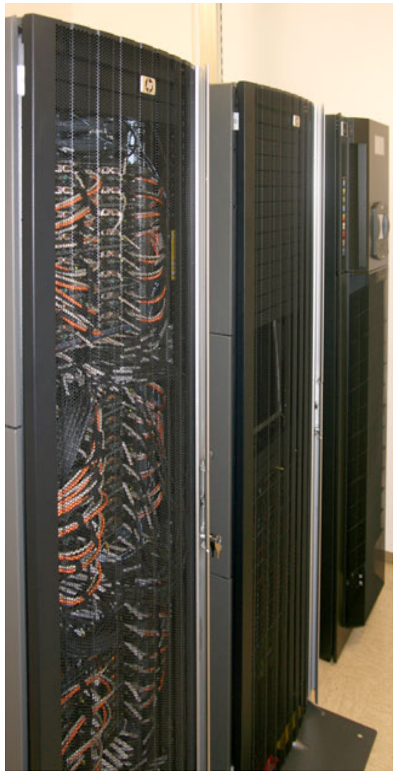
Performance: = 24.3 TFlops excluding graphics accelerators

haswell ↔ ***rome***

Computing nodes
- 24 processor cores (2x 12-core Intel Xeon E5-2680V3 processor)
- 64 GB of DDR4 ram at 2133 MHz
- hard disk 250 GB

GPU Computing node
- 24 processor cores (2x 12-core Intel Xeon E5-2680V3 processor)
- 256 GB of DDR4 ram at 2133 MHz
- 3x NVIDIA Tesla K80
- hard disk 250 GB

Login node (depricated)
- 24 processor cores (2x 12-core Intel Xeon E5-2680V3 processor)
- 256 GB of DDR4 ram at 2133 MHz
- 1x NVIDIA Quadro K40M
- 2x 1 TB SATA SDD



Computing nodes
- 48 processor cores (2x AMD EPYC 7402 24-core processor)
- 128 GB DDR4-3200 ram
- 1 TB NVMe SSD

Login node
- 32 processor cores (2x AMD EPYC 7302 16-core processor)
- 512 GB DDR4-3200 ram
- 2x 1 TB NVMe SSD
- NVIDIA A100

HPCFS – software overview

- Ansys Multiphysics
- Ansys CFX, Fluent, Maxwell, HFSS
- OpenFOAM CFD + extend
- VisIt in ParaView postprocesor
- Intel F90, CC
- TotalView, Allinea DDT
- Siemens NX
- Octave, R, Mathematica, Matlab
- OpenMP, OpenMPI, HPMPI, IntelMPI
- ATLAS, BLAS, BLACS, FFTW, GOTO, MUMPS, NetCDF, HDF5, Sparsekit, Scalapack

Working with HPCFS cluster

- *Demonstration of the work on the cluster by repeating*
- *Access with NX client*
- *Learning basic Linux commands*
- *SLURM scheduler commands*
- *Modules*
- *Development with OpenMP and OpenMPI parallel paradigms*
- *Excercises and extensions of basic ideas*
- *Instructions available at http://hpc.fs.uni-lj.si/*

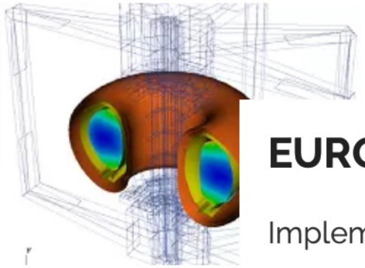
HPC – current projects @LeCAD

EuroCC 2

National Competence Centres in the framework of EuroHPC Phase 2

Plasma-PEPSC

Plasma Exascale-Performance Simulations CoE - Pushing flagship plasma simulations codes to tackle exascale-enabled Grand Challenges via



EUROfusion

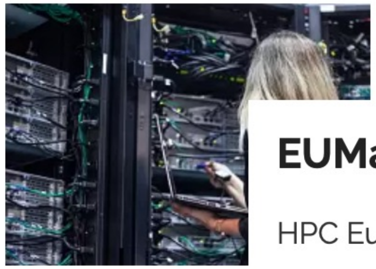
Implementation of activities described in the Roadmap to Fusion during Horizon Europe through a Joint Programme of the members of the

EXCELLERAT P2

European Centre of Excellence for Engineering Applications on HPC and associated technologies

P2-0425

Decentralized solutions for the digitalization of industry and smart cities and communities



EUMaster4HPC

HPC European Consortium Leading Education Activities

Thank you for attention!

REPUBLIC OF SLOVENIA
MINISTRY OF HIGHER EDUCATION, SCIENCE AND INNOVATION

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia.