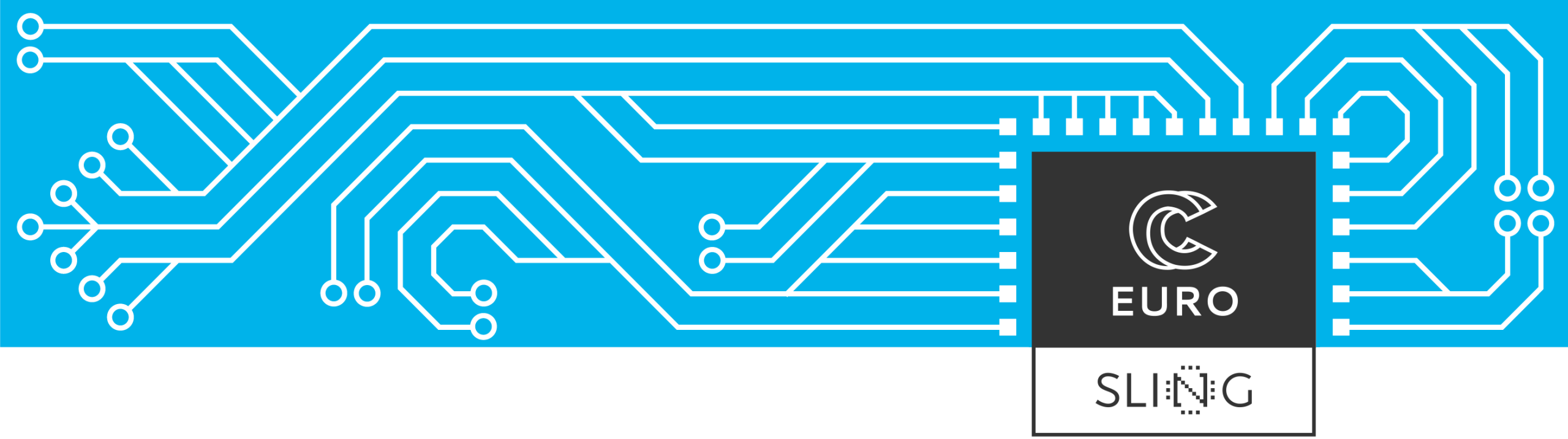


Working with supercomputer HPCFS

Grega Tomažin, Matjaž Šubelj, Leon Kos, Pavel Tomšič

Description

Introducing to the participants the usage of the HPCFS supercomputer. The operation of the supercomputer, access to HPCFS, and a list of available software will be presented.

The supercomputer available at UL FS (we call it HPCFS) consists of two computing nodes at the time.

The partition named haswell from 2016: 20 compute nodes, 1 GPU compute node, 1 login node, with a total of 480 processor cores.

The partition named rome from 2021: 20 compute nodes, 1 login node, with a total of 960 processor cores.

Further upgrades are planned for 2025.

Agenda

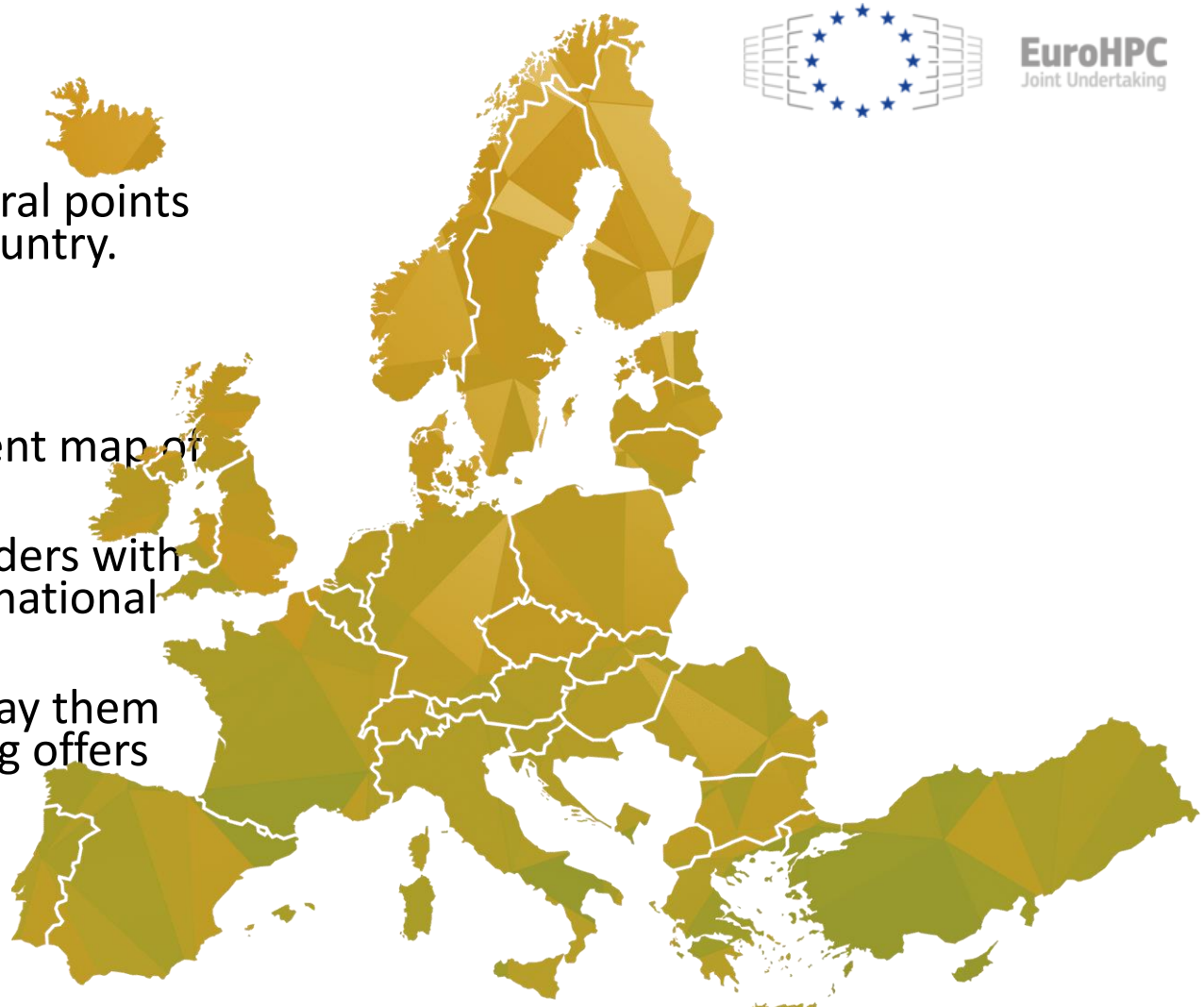
27 March 2025		
Beginning	**End**	**Description**
14:00	**14:10**	**Introduction to supercomputing** (*10 min*) what are supercomputers, / benefits of use / presentation of the HPCFS architecture
14:10	**14:40**	**Accessing and usage of HPCFS** (*30 min*) How to get a user account, useful information, necessary software to connect (10 min) Connection to HPCFS via SSH (Putty) or via virtual desktop (NX client) / data transfer to HPCFS (20 min)
14:40	**15:00**	**Overview of Software on HPCFS** (*20 min*) Overview of installed software / basic startup of some installed software (Ansys, Matlab, Mathematica, OpenFoam / presentation of parallel computing).
15:00	**15:20**	**Startup and first steps using software** (*20 min*) Matlab startup and usage (10 min) Ansys startup and usage (10 min)
15:20	**15:30**	**Current projects on HPCFS** (*10 min*) Brief presentation of projects running on HPCFS.

About EuroHPC

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

Their mission:

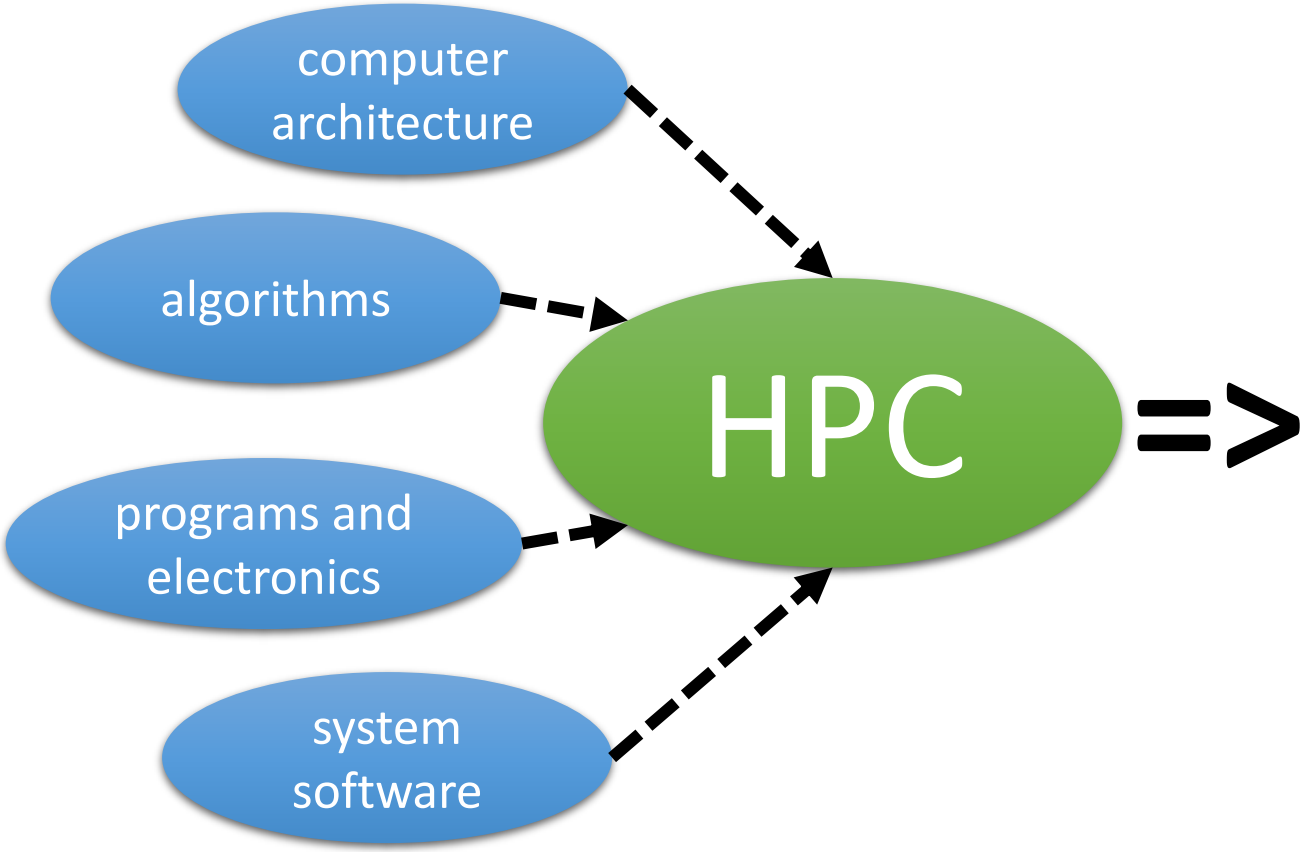
- Develop and display a comprehensive and transparent map of HPC competences and institutions in their country
- Act as a gateway for industry and academia to providers with suitable expertise or relevant projects, may that be national or international
- Collect HPC training offers in their country and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

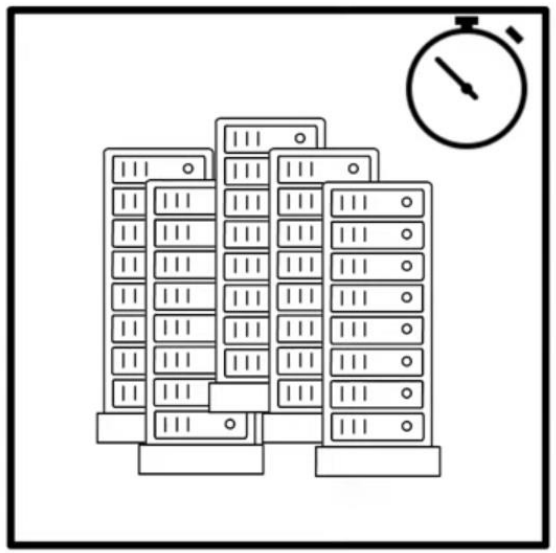


National Competence Centres for HPC in 32 countries

Introduction to supercomputing

What is High Performance Computing (HPC)?



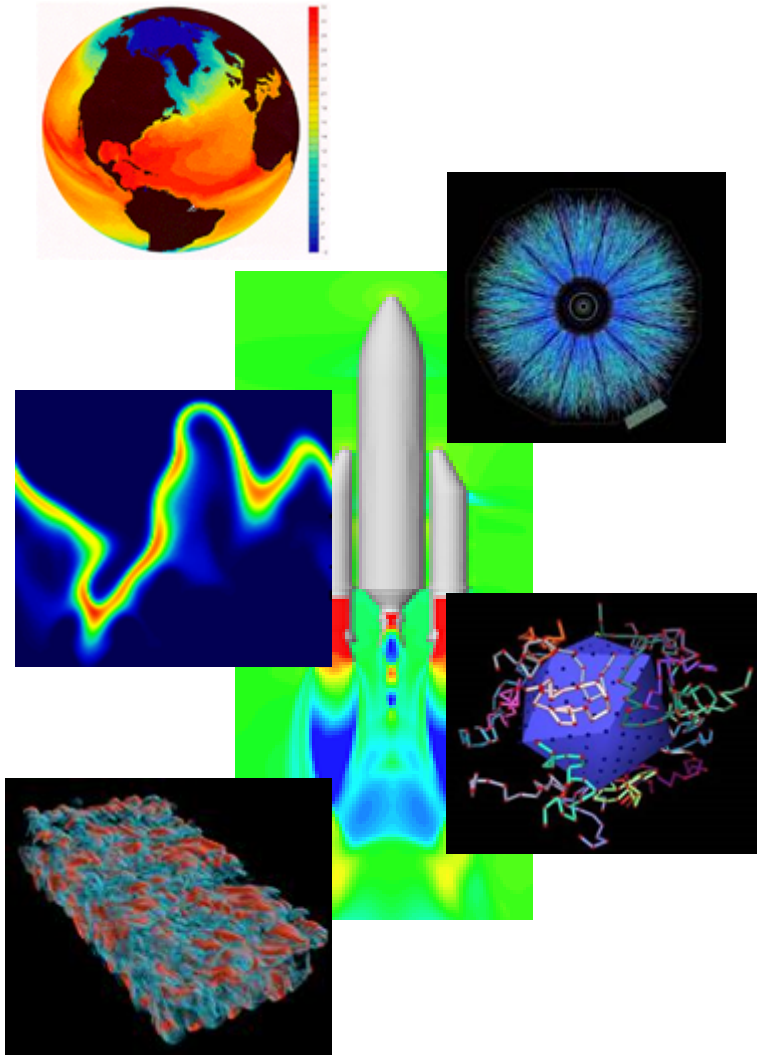


10101010101010101010100011
10101010100001011100010111000
10101010101010101010100011
10101010100001011100010111000

Introduction to supercomputing

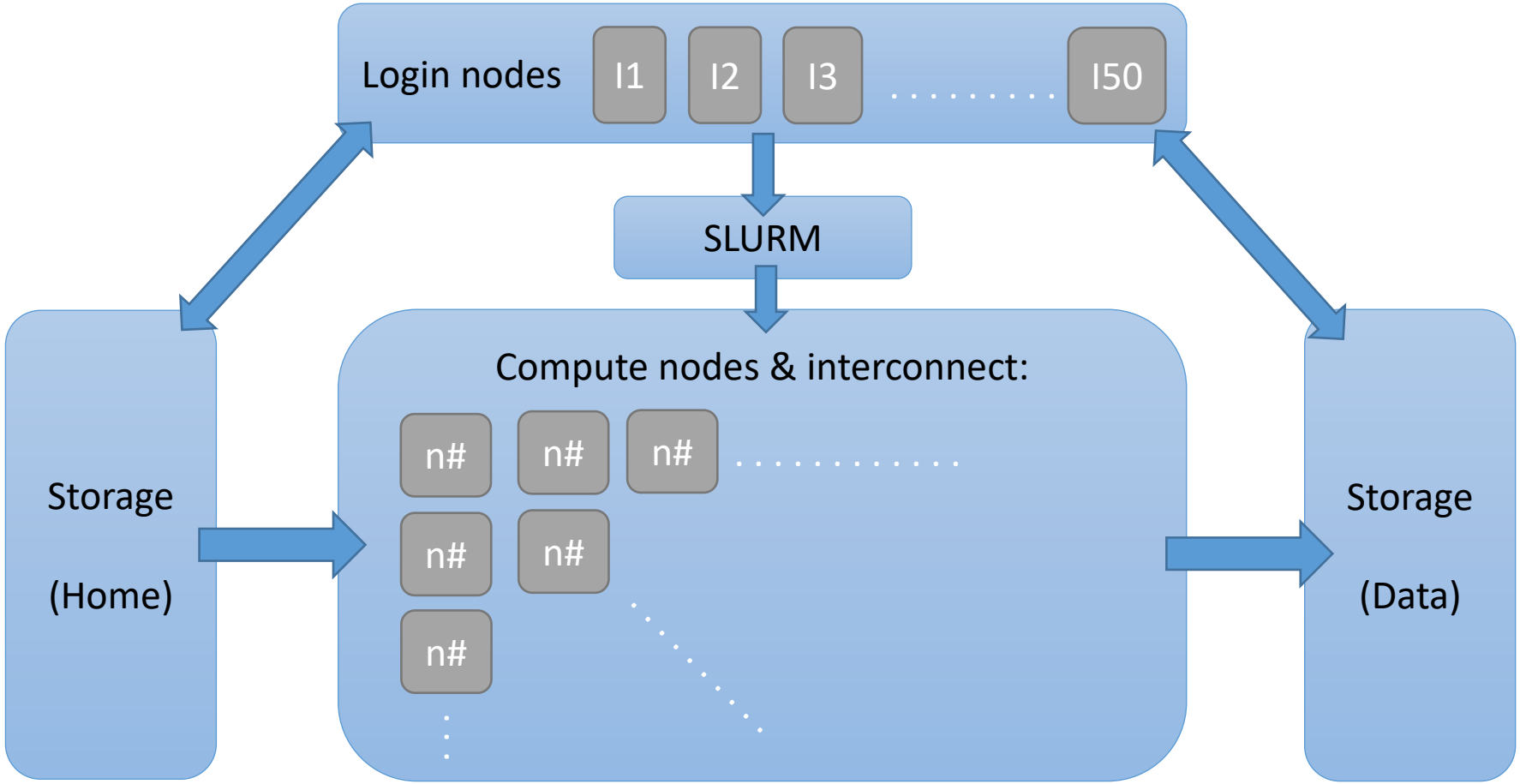
Area of usage

- **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



Introduction to supercomputing

Components of a HPC cluster



Login nodes

↕

Compute nodes

Shared (login, storage)

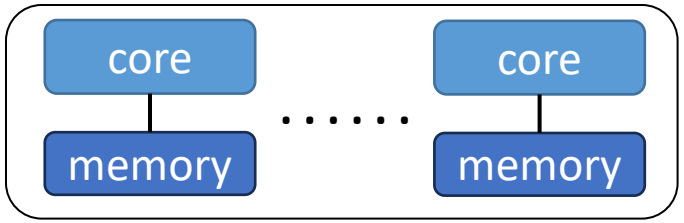
↕

Exclusive (compute nodes)

Introduction to supercomputing

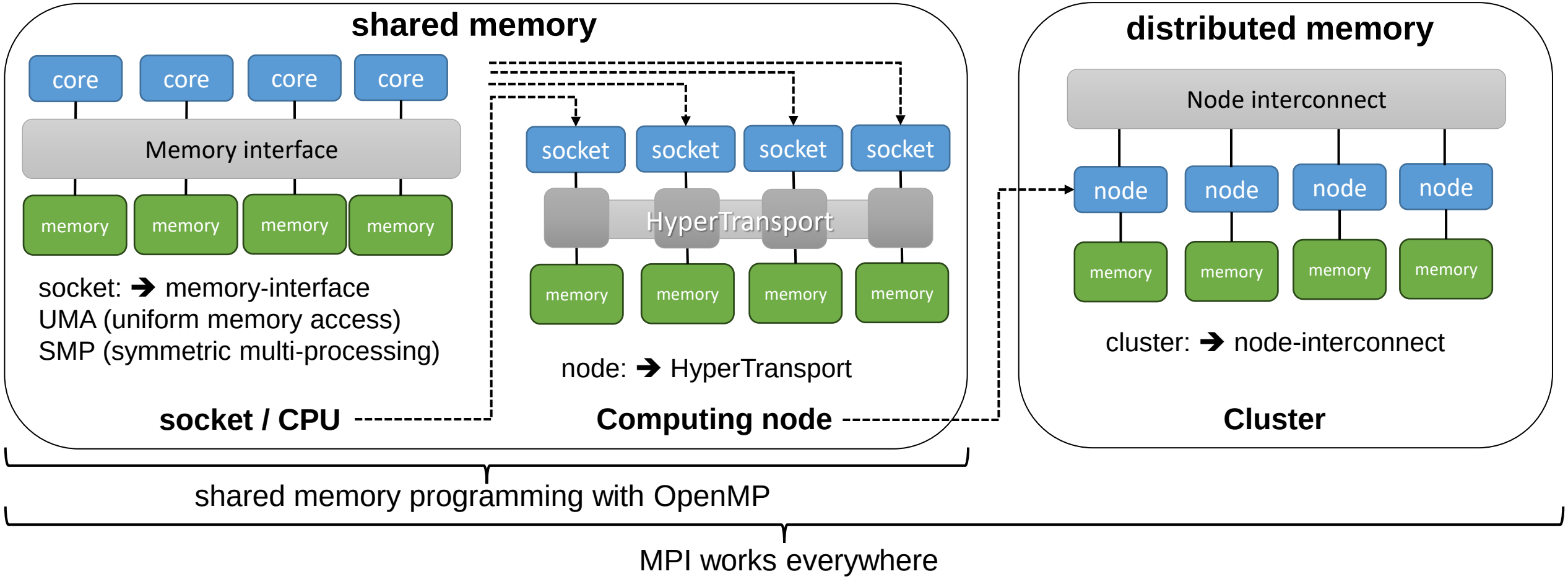
Parallel hardware

Serial computing



Introduction to supercomputing

Parallel hardware

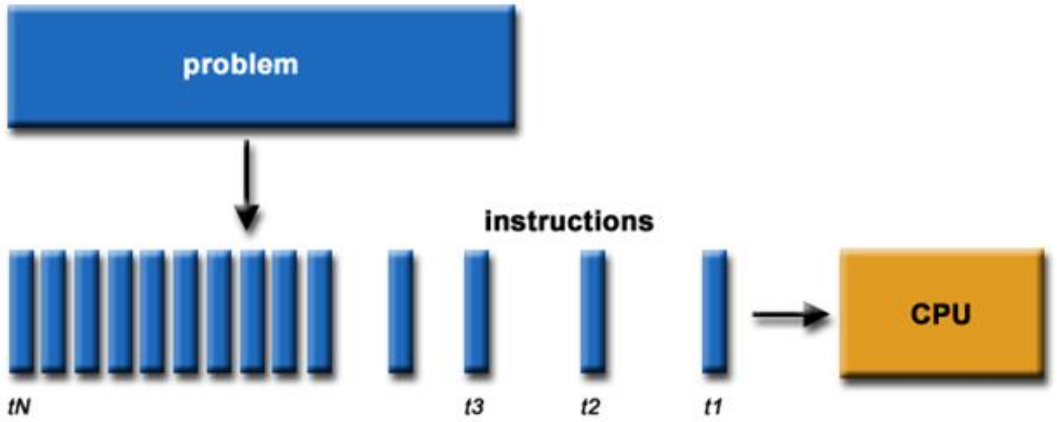


Introduction to supercomputing

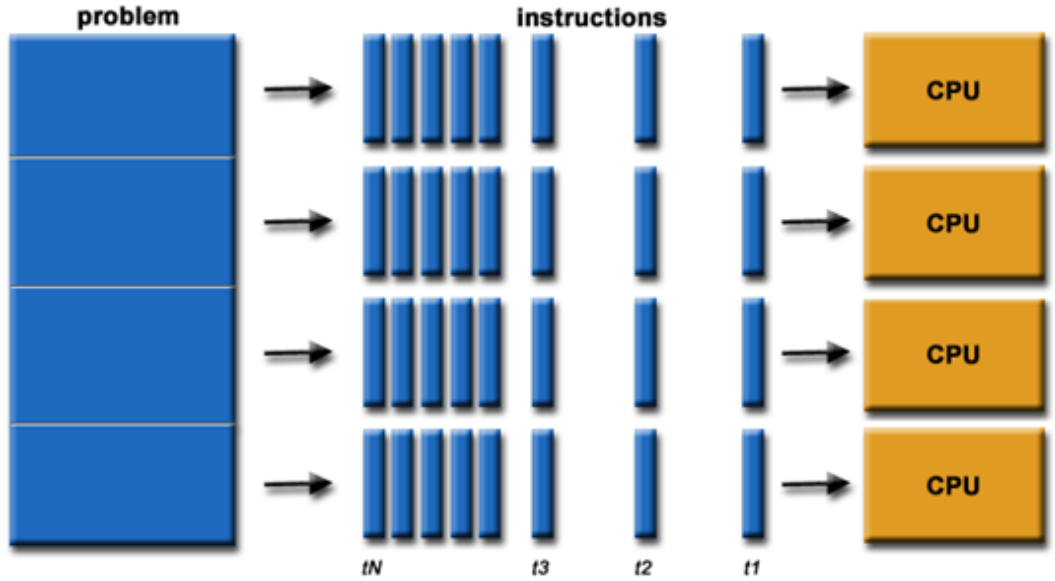
A simple parallel process

Problem discretisation

Serial computing



Problem paralelisation



Introduction to supercomputing

Problem scalability

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

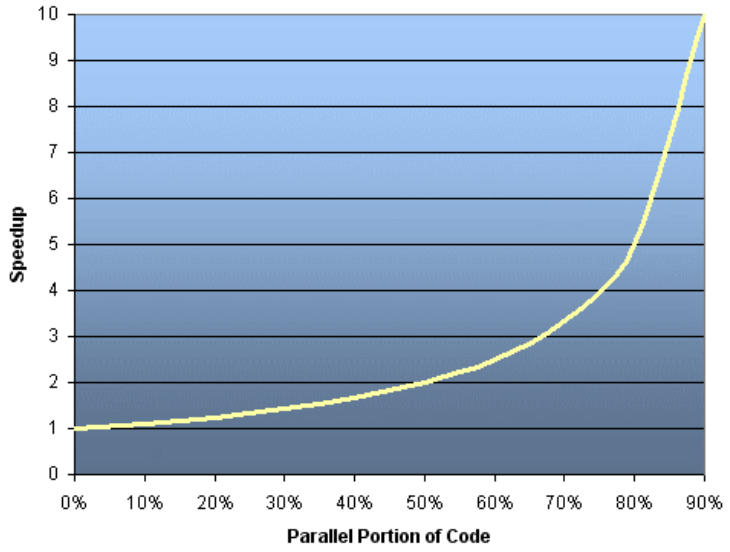
neglecting time for communication
neglecting load imbalance

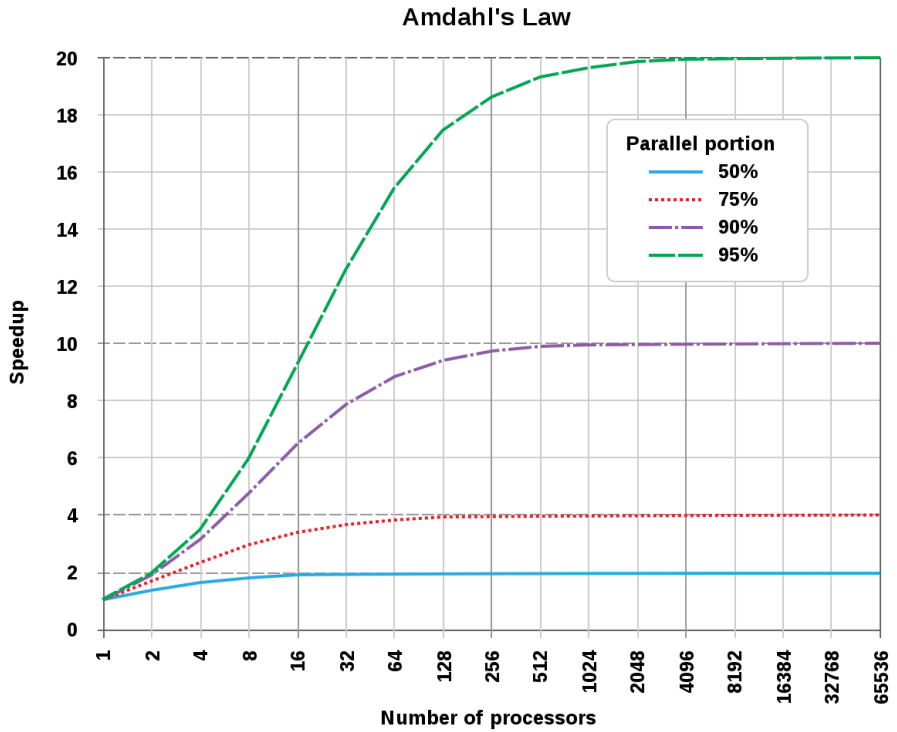
$$S_p = T_{serial} / T_{parallel,p} = 1 / (f + (1-f) / p)$$

Speedup is limited:

$$S_p < 1 / f$$

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





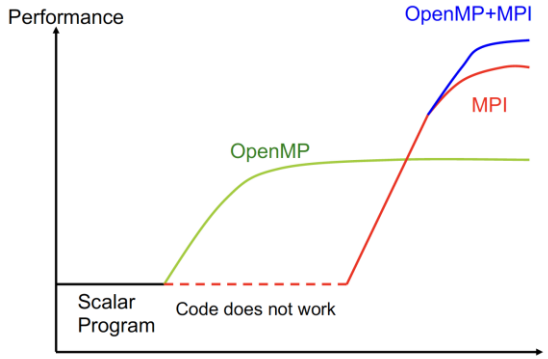
Source: Wikipedia

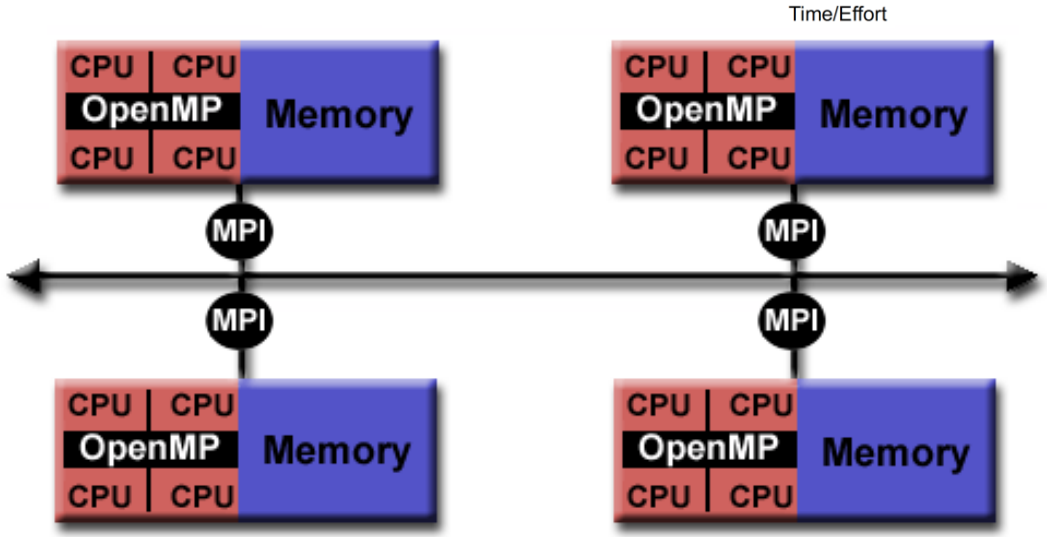
https://en.wikipedia.org/wiki/Amdahl%27s_law

Supercomputing arhitecture

Parallel programming models

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





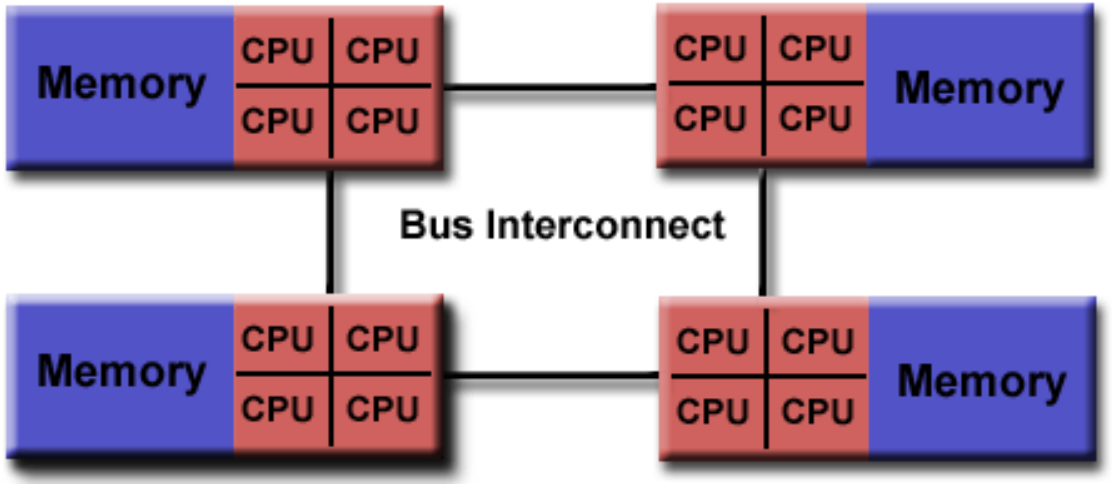
Multi-threading is needed to exploit modern hardware platforms:

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

Supercomputing arhitecture

Logical view of a computing node

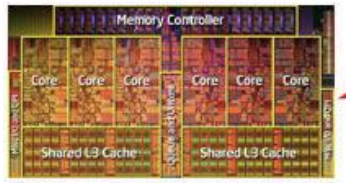
- Need to know computing arhitecture
- Interconnect bus for sharing memory between processors (NUMA interconnect)



Supercomputer - each blue light is a node

Node - standalone Von Neumann computer

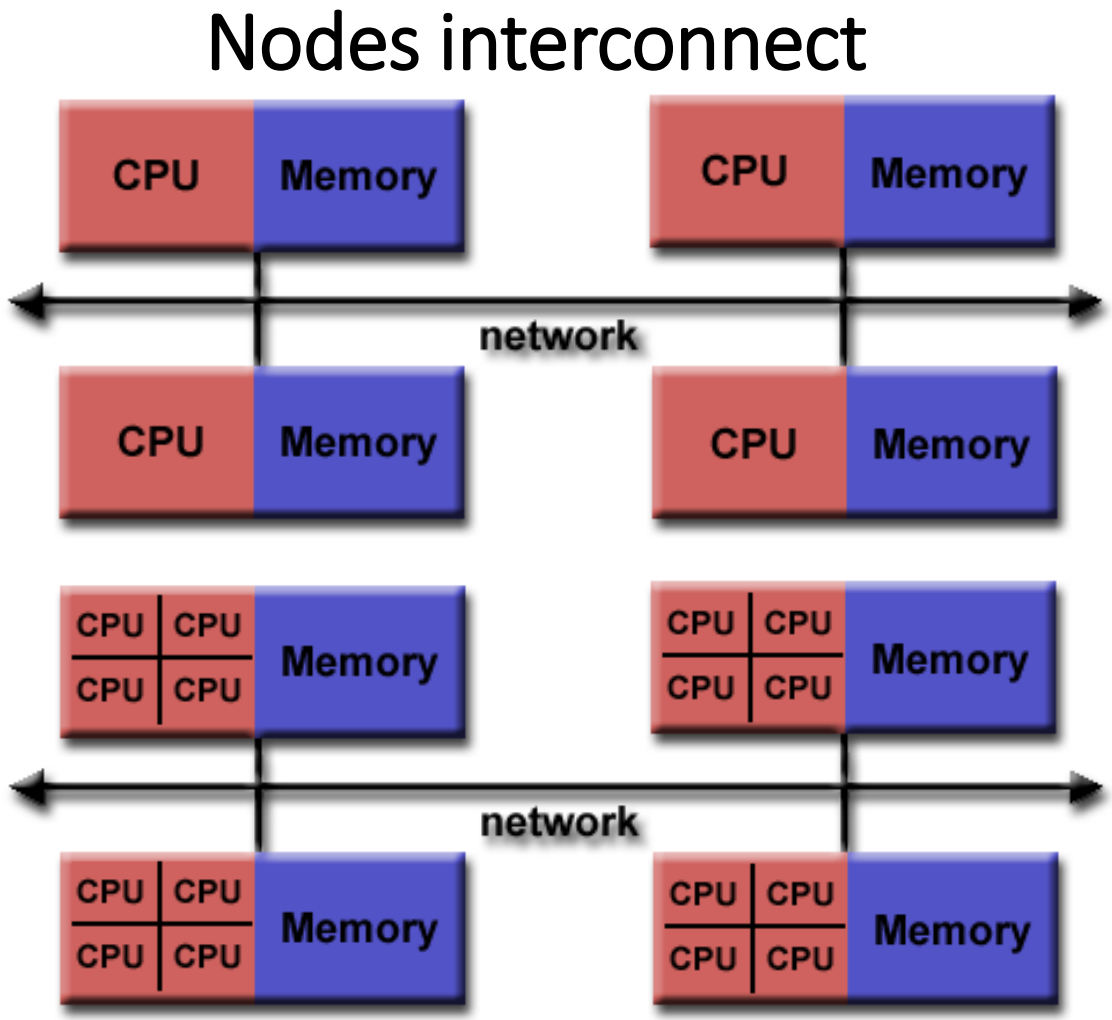
CPU / Processor / Socket - each has multiple cores / processors.



Supercomputing arhitecture

- 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

HPCFS - about

haswell

20 compute nodes, 1 GPU compute node, 1 login node

Haswell compute nodes:

- 2x Intel Xeon E5-2680V3 12-core processor, 2.5 GHz base clock
- 64 GB DDR4-2133 ram
- QDR Infiniband
- 250 GB SATA HDD
- 2x 1 Gbps Ethernet

Haswell GPU compute node:

- 2x Intel Xeon E5-2680V3 12-core processor, 2.5 GHz base clock
- 256 GB DDR4-2133 ram
- QDR Infiniband
- 3x NVIDIA K80
- 250 GB SATA HDD
- 10 Gbps Ethernet

Haswell login node:

- 2x Intel Xeon E5-2680V3 12-core processor, 2.5 GHz base clock
- 256 GB DDR4-2133 ram
- QDR Infiniband
- NVIDIA K40
- 2x 1TB SATA SSD
- 10 Gbps Ethernet

rome

20 compute nodes, 1 login node

Rome compute nodes:

- 2x AMD EPYC 7402 24-core processor, 2.8 GHz base clock
- 128 GB DDR4-3200 ram
- HDR100 Infiniband
- 1 TB NVMe SSD
- 2x 1 Gbps Ethernet

Rome login node:

- 2x AMD EPYC 7302 16-core processor, 3 GHz base clock
- 512 GB DDR4-3200 ram
- HDR100 Infiniband
- NVIDIA A100
- 2x 1 TB NVMe SSD
- 2x 10 Gbps Ethernet

storage

Lustre filesystem:

- 4x 24 TB Object Storage Targets
- 96 TB total capacity

Network filesystem:

- 191 TB total capacity

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://wiki.hpc.fs.uni-lj.si/*

Accessing HPCFS

You can access the supercomputer HPCFS by **username and password** OR by **ssh key**. In both cases, you first need **approval** by computing center of UL FME.

To get it, you need to send to hpc@fs.uni-lj.si **two things:**

1. a completed form (fill it out, sign it, provide a signature from the head of research unit (laboratory) whose computing resources you will spend);
2. encrypted password **OR** SSH key:
   - **password:** click on the password to obtain the encrypted password. Send this password to hpc@fs.uni-lj.si;
   - **SSH key:** you can create SSH key pair and send public key to us

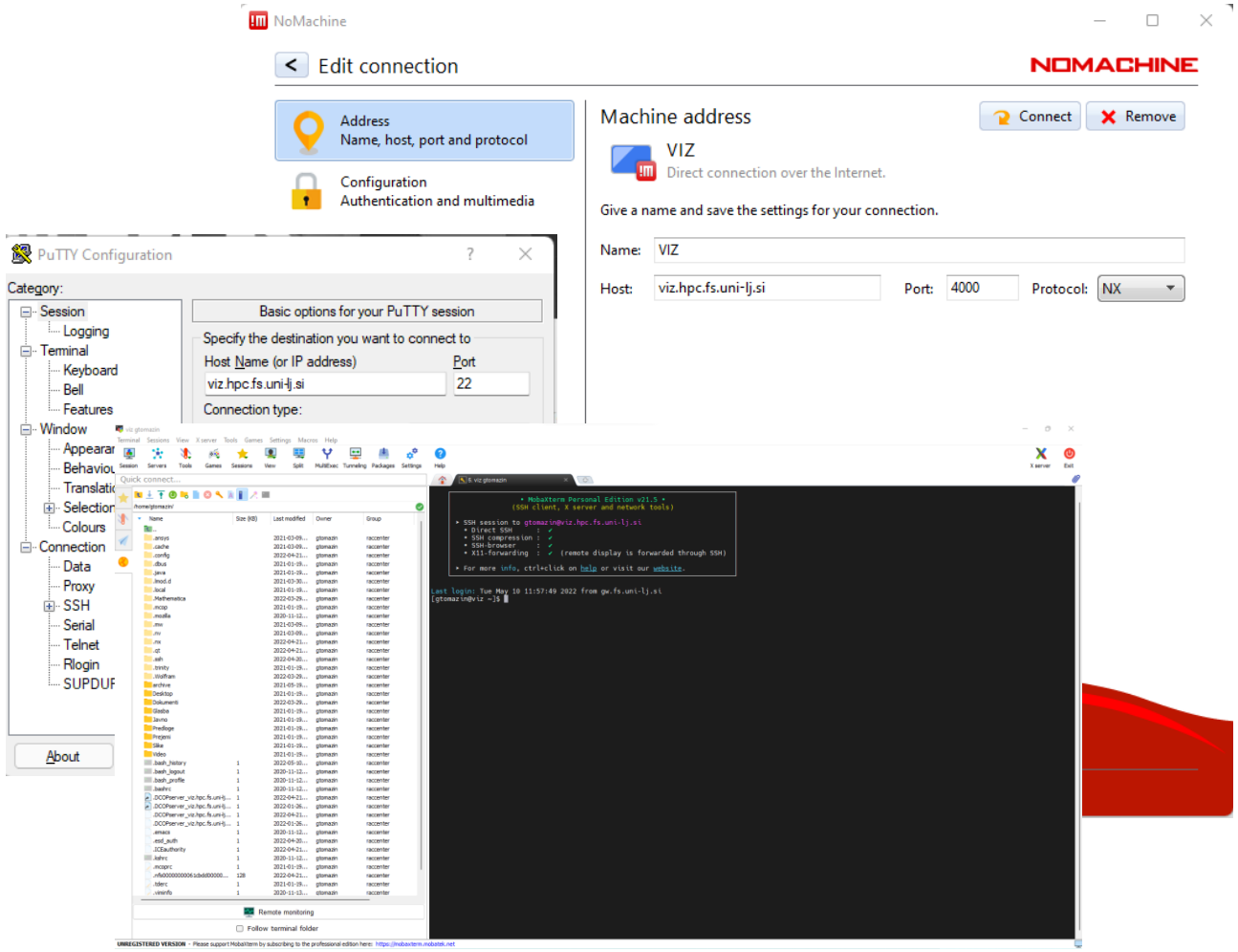
Accessing HPCFS

- **Accessing HPC software**

1. NoMachine NX or SSH
2. Putty (Windows only)
3. MobaXterm (Windows only; console and files)
4. Terminal / console

- **login.hpc.fs.uni-lj.si**

SSH:22, NX:4000



Accessing HPCFS

Basic SLURM commands

- srun, sbatch, sinfo, squeue, scancel <jobid>
- Interactive
  - env --unset=LD_PRELOAD TMOUT=600 srun --mem=0 -p haswell --pty bash -i
  - env --unset=LD_PRELOAD TMOUT=600 srun --mem=100G --time=4:0:0 -p rome --x11 --pty bash –i
  - alias rome='env --unset LD_PRELOAD --unset SESSION_MANAGER TMOUT=600 srun --partition=rome --mem=0 --time=2:00:00 ${DISPLAY:+--x11} --pty bash -i'

- **More about SLURM usage you can read here:**

  https://wiki.hpc.fs.uni-lj.si

  https://hackmd.io/@hpcfs/wiki/%2F%40hpcfs%2Faccess%3Fview

  https://slurm.schedmd.com/documentation.html

WIKI
HPC FS

SLURM
DOC

Accessing HPCFS

Useful SLURM commands

- sbatch myjob.slurm

```
#!/bin/bash
#SBATCH -J "MyFEM 1" # Ime naloge
#SBATCH -n 4 # Stevilo procesov
#SBATCH -N 2 # Stevilo vozlisc
#SBATCH -p rome # particija
#SBATCH --mem=0 # ves spomin vozlisca
#SBATCH --time=0:10:0 # najvec 10 minut
#SBATCH -o output.log # Zaslonski izpis
#SBATCH -e error.log # Mozne napake programa

module purge
module load Eigen/3.4.0-GCCcore-12.2.0
module load VTK/9.2.6-foss-2022b
module load OpenMPI/4.1.5-GCC-12.2.0
```

- sinfo or sinfo -s
- squeue or squeue -u $USER
- scancel <jobid>
- scontrol show --details job <jobid>
- squeue --user=$USER --start --iterate=60

HPCFS – software overview

avaliable software

- ANSYS Multiphysics
- ANSYS CFX, Fluent, …
- OpenFOAM CFD, Elmer Multiphysics
- ParaView postprocessor, SALOME
- Compilers: GNU, Intel F90, CC
- R, Mathematica, MATLAB, Python
- Parallel environments: OpenMP, OpenMPI, IntelMPI
- Libraries: BLAS, BLACS, FFTW, GOTO, MUMPS, NetCDF, HDF5, Sparsekit, Scalapack, Tensorflow

HPCFS – software overview

Module environment

- Module concept is available on most supercomputers
- It simplifies the use of different software (versions) in a controlled way.
- Settings for each SW (version) are encapsulated into "environment modules" maintained by the module system that updates environment variables such as search paths ($PATH), dynamic loader paths ($LD_LIBRARY_PATH).

HPCFS – software overview

Basic module commands

- To get list currently loaded modules on the node

[johndoe@cn62 ~]$ **module list**

No modules loaded

- To list all available modules on HPC

[johndoe@cn62 ~]$ **module avail**

```
------------------------------------------------------------------------------ /opt/pkg/modules/all ------------------------------------------------------------------------------
   ANSYS/20.1                                                                  Tcl/8.6.9-GCCcore-8.2.0
   ANSYS/21.R1                                                (D)              Tcl/8.6.9-GCCcore-8.3.0
   ANSYS/21.1                                                                  Tcl/8.6.10-GCCcore-9.3.0
   ANSYS/2021R2                                                                Tcl/8.6.10-GCCcore-10.2.0                                                  (D)
   ATK/2.22.0-foss-2016b                                                       TensorFlow/1.10.1-foss-2018b-Python-3.6.6
   ATK/2.28.1-fosscuda-2018b                                                   TensorFlow/1.10.1-fosscuda-2018b-Python-2.7.15
   ATK/2.32.0-GCCcore-8.2.0                                   (D)              TensorFlow/1.13.1-foss-2019a-Python-3.7.2
   ATLAS/3.10.2-GCC-5.4.0-2.26-LAPACK-3.6.1                                    TensorFlow/1.13.1-fosscuda-2019a-Python-3.7.2
   Arrow/0.17.1-foss-2020a-Python-3.8.2                                        TensorFlow/2.0.0-foss-2019a-Python-3.7.2
   Autoconf/2.69-foss-2016b                                                    TensorFlow/2.0.0-fosscuda-2019b-Python-3.7.4                               (D)
```

HPCFS – software overview commands - keyword

- To search for a module starting with some **keyword**

[johndoe@cn62 ~]$ **module keyword Ansy**

The following modules match your search criteria: "Ansy"

ANSYS: ANSYS/2021R2, ANSYS/2022R1, ANSYS/2022R2, ANSYS/2023R1, ANSYS/2024R1, ANSYS/2024R2

Ansys offers a comprehensive software suite that spans the entire range of physics, providing access to virtually any field of engineering simulation that a design process requires.

HPCFS – software overview

commands - whatis

- To see details about some module

[johndoe@cn62 ~]$ **module whatis** MATLAB

MATLAB/2023a : Description: MATLAB is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran.

MATLAB/2023a : Homepage: https://www.mathworks.com/products/matlab

MATLAB/2023a : URL: https://www.mathworks.com/products/matlab

HPCFS – software overview commands - spider

- For slightly more details about given module

[johndoe@cn62 ~]$ **module spider MATLAB**

MATLAB:

Description:

MATLAB is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran.

Versions:

MATLAB/2019a

MATLAB/2022b

MATLAB/2023a

HPCFS – software overview commands - show

- For even more details about given module

  [johndoe@cn62 ~]$ **module show MATLAB**

```
   /opt/pkg/modules/all/MATLAB/2023a.lua:
-------------------------------------------------------------------------------------------------------
help([[
Description
===========
MATLAB is a high-level language and interactive environment
 that enables you to perform computationally intensive tasks faster than with
 traditional programming languages such as C, C++, and Fortran.


More information
================
 - Homepage: https://www.mathworks.com/products/matlab
]])
whatis("Description: MATLAB is a high-level language and interactive environment
 that enables you to perform computationally intensive tasks faster than with
 traditional programming languages such as C, C++, and Fortran.")
whatis("Homepage: https://www.mathworks.com/products/matlab")
whatis("URL: https://www.mathworks.com/products/matlab")
conflict("MATLAB")
prepend_path("CMAKE_PREFIX_PATH","/opt/pkg/software/MATLAB/2023a")
prepend_path("PATH","/opt/pkg/software/MATLAB/2023a/bin")
setenv("EBROOTMATLAB","/opt/pkg/software/MATLAB/2023a")
setenv("EBVERSIONMATLAB","2023a")
setenv("EBDEVELMATLAB","/opt/pkg/software/MATLAB/2023a/easybuild/MATLAB-2023a-easybuild-devel")
prepend_path("PATH","/opt/pkg/software/MATLAB/2023a")
prepend_path("LD_LIBRARY_PATH","/opt/pkg/software/MATLAB/2023a/runtime/glnxa64")
prepend_path("LD_LIBRARY_PATH","/opt/pkg/software/MATLAB/2023a/bin/glnxa64")
prepend_path("LD_LIBRARY_PATH","/opt/pkg/software/MATLAB/2023a/sys/os/glnxa64")
setenv(" JAVA OPTIONS","-Xmx2048m")
```

HPCFS – software overview

Most important commands

[johndoe@cn62 ~]$ **module available MATLAB**

---------- /opt/pkg/modules/all ----------

MATLAB/2019a MATLAB/2022b MATLAB/2023a (D)

[johndoe@cn62 ~]$ **module load MATLAB/2023a**

[johndoe@cn62 ~]$ **module list**

Currently Loaded Modules:

1) MATLAB/2023a

[johndoe@cn62 ~]$ **module unload MATLAB/2023a**

If you want to remove all the currently loaded modules and clean the environment variables:

[johndoe@cn62 ~]$ **module purge**

HPCFS – software overview

Some other modules

- Intel compiler

  [johndoe@cn62 ~]$ module load intel

  [johndoe@cn62 ~]$ module load iimpi # with Intel MPI

- GCC compiler and MPI Free and Open Source Software environment

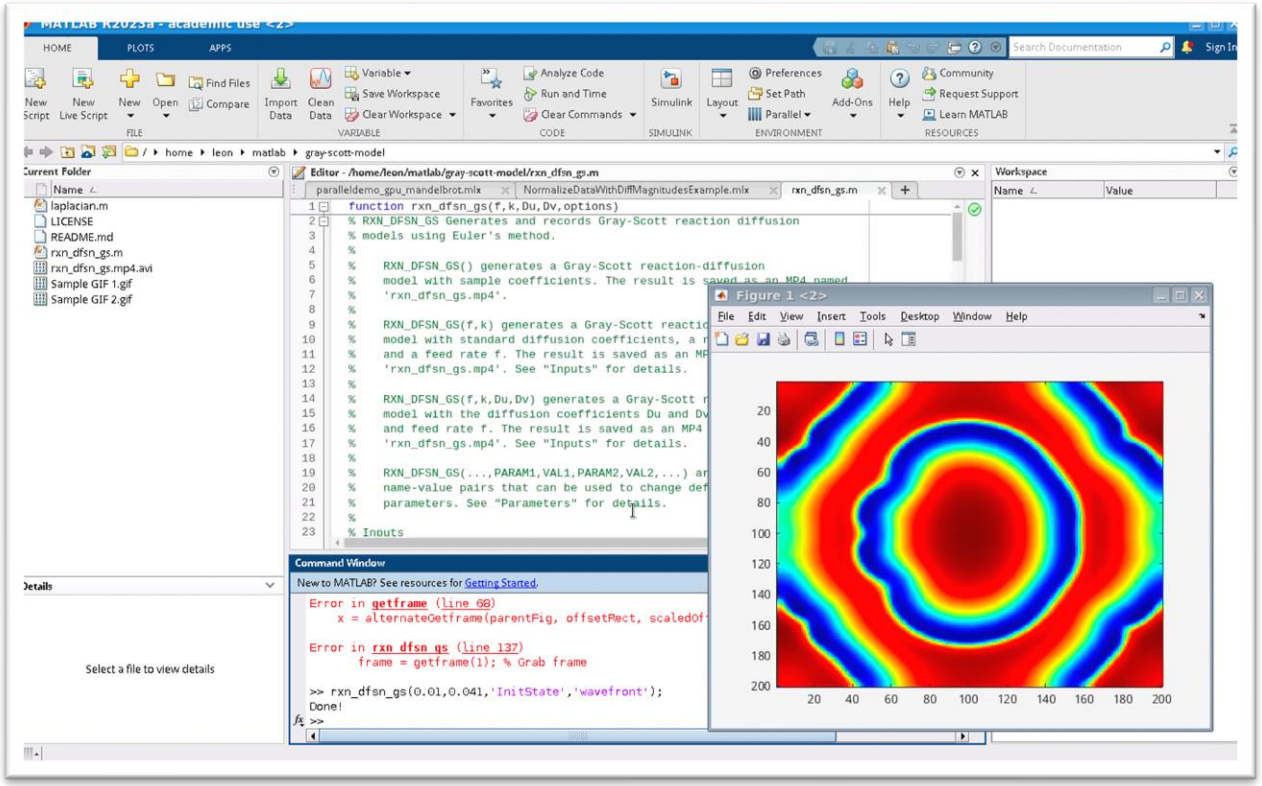
  [johndoe@cn62 ~]$ module available foss/

  [johndoe@cn62 ~]$ module load foss/2023b

Avaliable software: Matlab

module load MATLAB

- env --unset LD_PRELOAD TMOUT=600 srun --time=2:00:00 --partition=haswell ${DISPLAY:+--x11} --pty matlab



Avaliable software: Mathematica

ml Mathematica

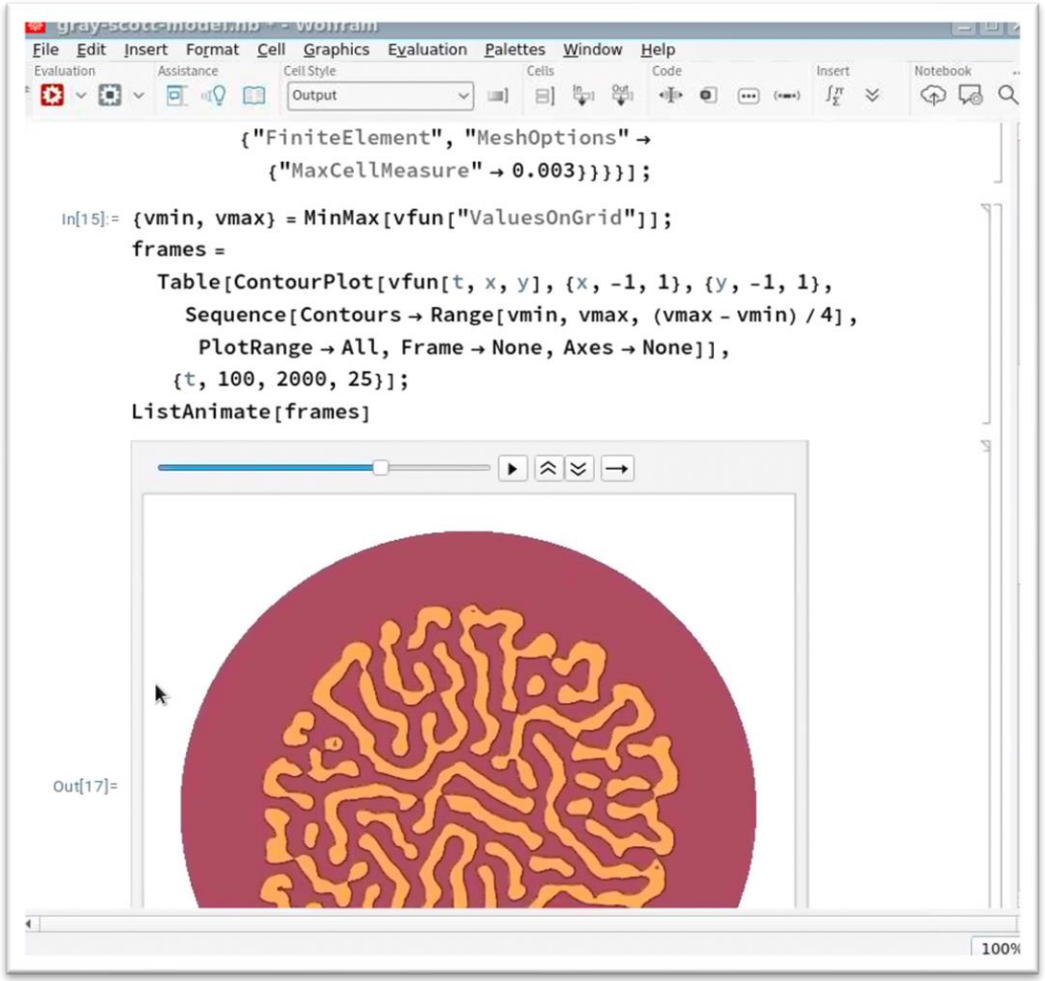
- Start with
  mathematica
  wolframnb # 14.2
- For the first time an activation window pops out suggesting activation through web. Please find a button **Other ways to activate** in the bottom of the dialog and thenselect **Activate through a Wolfram network license server**and for server enter **flex.hpc** and press **Activate**.

Parallel execution on a compute node(s) [version 14.2]

Use *Edit->Preferences->Kernels->Kernel Configuration Editor* and modify example.com to Class *SshKernels* and rename it to the compute node allocated within terminal with

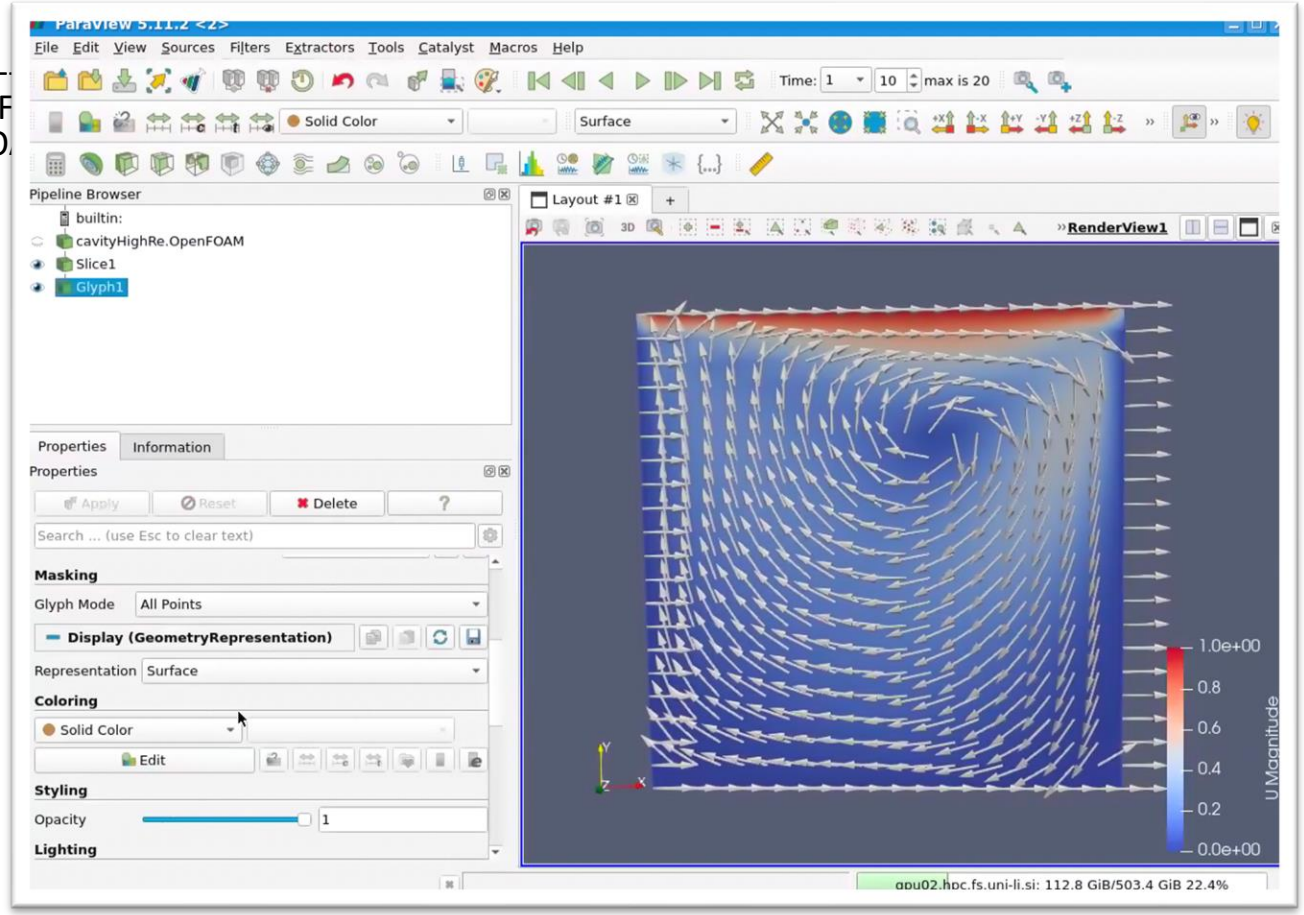
salloc -N 1

For *Default Kernels* select that configuration as a *Defult kernel for RemoteEvaluate*



Avaliable software: OpenFOAM

- $ ml av OpenFOAM
  ---------- /opt/pkg/modules/all ----------
  OpenFOAM/v2206-foss-2022a OpenFOAM/v2412-foss-2023a OpenF
  OpenFOAM/v2406-foss-2023a OpenFOAM/10-foss-2022a OpenFO
  (D)
- ml OpenFOAM
- ml show OpenFOAM
- source $FOAM_BASH
- env|grep FOAM|cut -c -40|head
- ls $FOAM_TUTORIALS/incompressibleFluid
- cp -r $FOAM_TUTORIALS/incompressibleFluid/cavity $HOME
- cd cavity
- ./Allrun
- cd cavityHighRe
- env LD_PRELOAD=/lib64/libcrypto.so:$LD_PRELOAD paraFoam



Avaliable software: Ansys

- Documentation available online:
  - https://ansyshelp.ansys.com/public
  - access loaded version from command line

    $ module load ANSYS/2024R2

    $ anshelp

- Two ways to run: **interactive** or **batch mode**
  - intractive with GUI interface on (login) node
  - batch can be scheduled – preferred use on clusters

Avaliable software: Ansys Workbench

- command:
  - runwb2
  - interactive job on compute node*:

    $ haswell runwb2

    $ rome runwb2

- batch script:
  - runwb2 -b -F file.wbpj -?...

*very version dependent, some don't work

Avaliable software: Ansys CFX

- commands:
  - cfx5
  - cfx5pre
  - cfx5solve
  - cfx5post
- batch script:

```
#!/bin/bash
#SBATCH --export=ALL,LD_PRELOAD=
#SBATCH --ntasks-per-core=1
#SBATCH --ntasks=12  # total number of cores requested
#SBATCH --nodes=2  # number of nodes
#SBATCH --job-name=cfx-test
#SBATCH --error=sbatch.err
#SBATCH --output=sbatch.out
#SBATCH --time=0-00:05  # time limit days-hh:mm
#SBATCH --mem=0G
module purge
module load ANSYS/2024R2

MACHINES=$(srun -l hostname -s | sort -n | uniq -f 1 -c | awk '{print $3 "*" $1}' | paste -sd ',')
DEF=StaticMixer.def

###  single node run
#cfx5solve -def $DEF -part $SLURM_NTASKS -start-method "Open MPI Local Parallel"

###  distributed run
cfx5solve -def $DEF -par-dist $MACHINES -start-method "Open MPI Distributed Parallel"
```

Avaliable software: Ansys Fluent

- commands:
  - fluent
- batch script:

```
#!/bin/bash
#SBATCH --export=ALL,LD_PRELOAD=
....
#SBATCH --time=0-00:05
module purge
module load ANSYS/2024R2

###  checkpointing fluent:
###  'touch check-fluent' ...save result and continue
###  'touch exit-fluent' ...save result and exit

JOURNAL=test.jou

###  distributed run
MACHINES=$(srun -l hostname -s | sort -n | uniq -f 1 -c | awk '{print $3 ":" $1}' |
paste -sd ',')

fluent 3ddp -g -t$SLURM_NTASKS -cnf=$MACHINES -mpi=intelmpi -i $JOURNAL

###  single node run
#fluent 3ddp -g -t$SLURM_NTASKS -i $JOURNAL
```

Avaliable software: Ansys MAPDL

- commands:
  - mapdl
  - ansys
  - ansys242
- batch script:

```
#!/bin/bash
#SBATCH --export=ALL,LD_PRELOAD=
#SBATCH --ntasks-per-core=1
#SBATCH --ntasks=12  # total number of cores requested
#SBATCH --nodes=2  # number of nodes
#SBATCH --job-name=mapdl-test
#SBATCH --error=sbatch.err
#SBATCH --output=sbatch.out
#SBATCH --time=0-00:05  # time limit days-hh:mm
#SBATCH --mem=0G

module purge
module load ANSYS/2024R2

mapdl -dis -b -m 10000 -np $SLURM_NTASKS -mpi intelmpi2018 -i bench07.mac -o bench07.out
```

Avaliable software: Ansys LS-Dyna

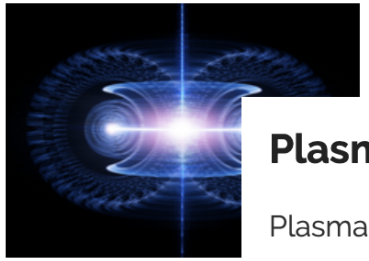
- commands:
  - mapdl
  - ansys
  - ansys242
- batch script:

```
#!/bin/bash
#SBATCH --export=ALL,LD_PRELOAD=
...
#SBATCH --signal=B:USR1  # default @60s before time limit

module load ANSYS/2024R2
export LSTC_LICENSE=ansys

FILENAME=000_yaris_dynamic_roof_crush_01

ansysbin=$(dirname $(which lsdyna))
lsdyna_e=$ansysbin/linx64/lsdyna_dp_mpp.e
.....

srun --mpi=pmi2 $lsdyna_e memory=250M memory2=100M \
                          i=$FILENAME.k \
                          o=$FILENAME.out \
                          g=$FILENAME.d3plot \
                          f=$FILENAME.d3thdt &
```

HPC – current projects @LeCAD

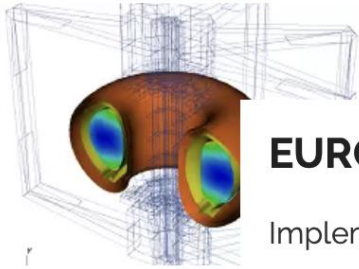
EuroCC 2

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

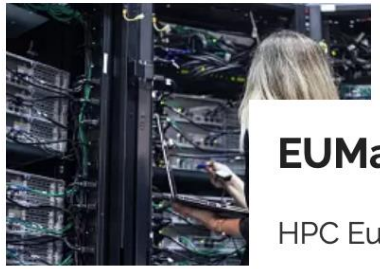
Implementation of activities described in the Roadmap to Fusion during Horizon Europe through a Joint Programme of the members of the

EXCELLERAT P2

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

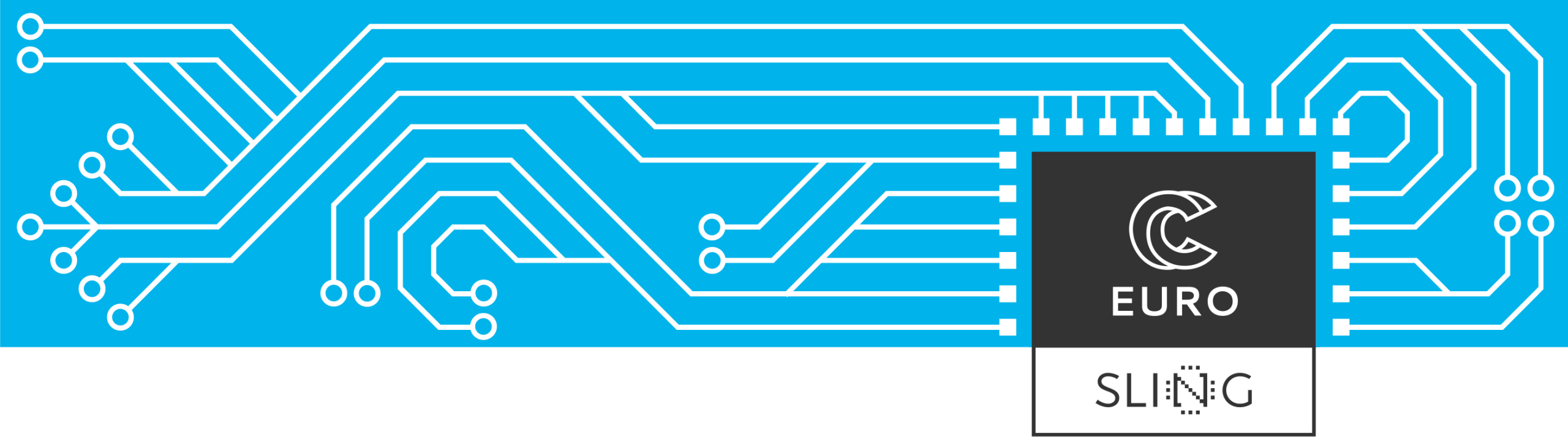
P2-0425

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



EUMaster4HPC

HPC European Consortium Leading Education Activities



Thank you for your attention :)

EuroHPC Joint Undertaking

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.