Academic Computer Centre
CYFRONET AGH

Maciej Czuchry, Klemens Noga

Introduction to scientific High Performance Computing

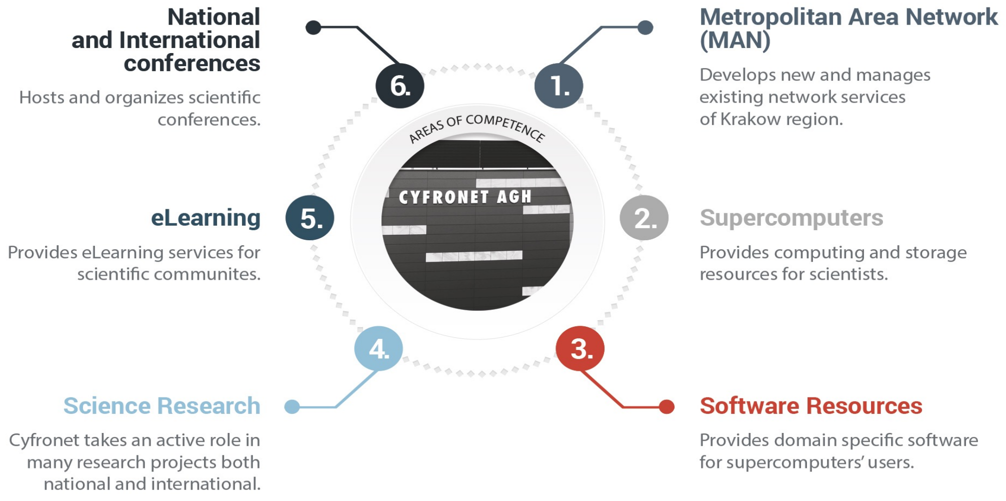
Agenda

- **Ares, Prometheus & Zeus clusters at ACC Cyfronet AGH**
    - available resources
    - access to clusters/data transfer
- **Performing calculations**
    - software environment management using Modules/Lmod
    - batch scripts
        - sequential and parallel runs
    - efficient usage of SLURM queuing system
- **Documentation and users' support**
- **Questions and exercises**
- **Zeus & Prometheus as a part of PLGrid Infrastructure**
- **PRACE and EuroHPC (LUMI) - computational opportunities**

Academic Computer Centre Cyfronet AGH

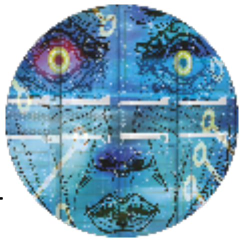
Academic Computer Centre Cyfronet AGH

- The biggest Polish Academic **Computer Centre**
  - **45+ years of experience** in IT provision
  - Centre of excellence in **HPC, Grid and Cloud Computing**
  - Home for **Ares**, **Prometheus** and **Zeus supercomputers**
  - **LUMI** consortium partner (EuroHPC pre-exascale supercomputer)
- Legal status: an **autonomous** within AGH University of Science and Technology
- Staff: >150 , ca. 60 in R&D
- Leader of **PLGrid**: Polish Grid and Cloud Infrastructure for Science
- NGI Coordination in **EGI e-Infrastructure**


AREAS OF COMPETENCE
CYFRONET AGH
1.
Metropolitan Area Network (MAN)
Develops new and manages existing network services of Krakow region.
2.
Supercomputers
Provides computing and storage resources for scientists.
3.
Software Resources
Provides domain specific software for supercomputers' users.
4.
Science Research
Cyfronet takes an active role in many research projects both national and international.
5.
eLearning
Provides eLearning services for scientific communites.
6.
National and International conferences
Hosts and organizes scientific conferences.

Academic Computer Centre Cyfronet AGH

Prometheus



- 2.4 PFLOPS
- 53 568 cores
- From 2015 to 2021 1st HPC system in Poland (440th on Top 500, 38th in 2015)

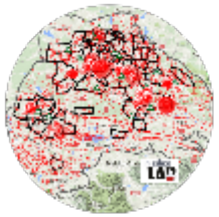
Zeus

- 374 TFLOPS
- 25 468 cores
- 1st HPC system in Poland (from 2009 to 2015, highest rank on Top500 – 81st in 2011)

Ares

- 4 PFLOPS
- 38 112 cores
- 267th on Top 500

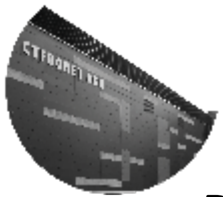
Computing portals and frameworks



- OneData
- PLG-Data
- Rimrock
- InSilicoLab

Storage

- 60+ PB
- hierarchical data management



Research & Development

- distributed computing environments
- computing acceleration
- machine learning
- software development & optimization

Data Centres

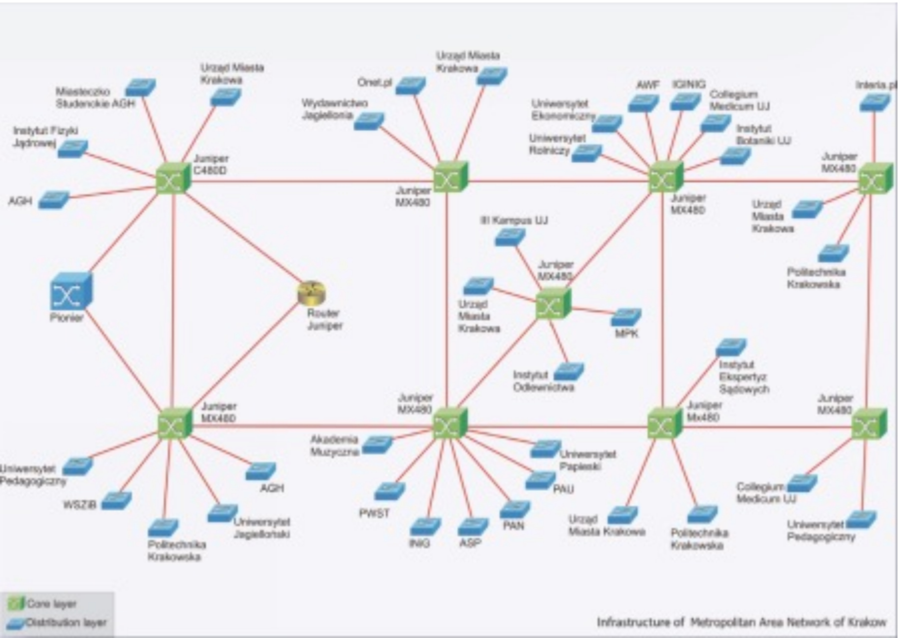
- 3 independent data centres
- dedicated backbone links

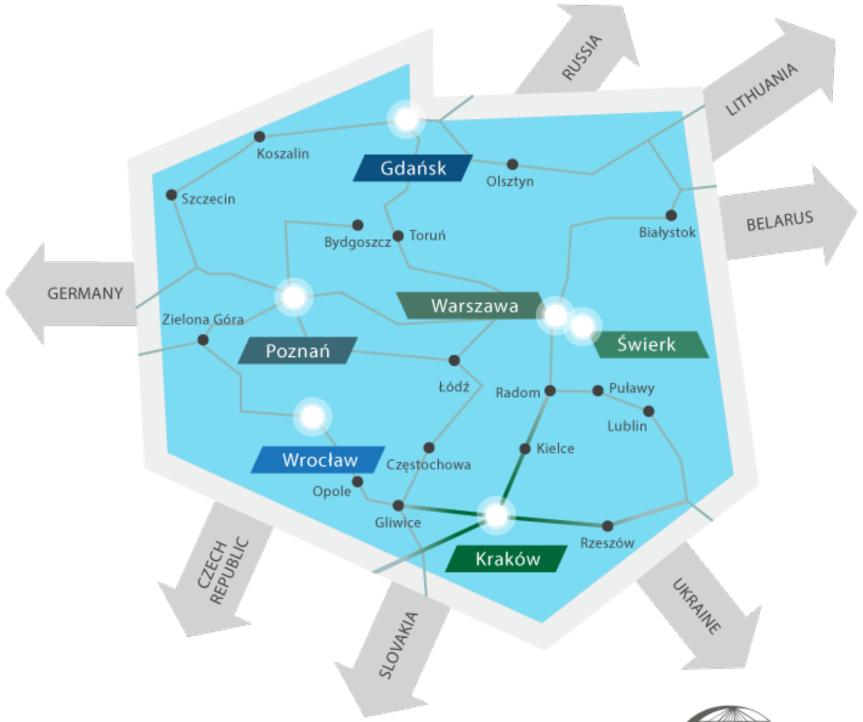
Computational Cloud

- based on OpenStack

Network backbone lines

- 4 main links to achieve maximum reliability
- Each link with 7x10Gbps capacity
- Additional 2x100Gbps dedicated links
- Direct connection with GEANT scientific network





Polish HPC @TOP500 – Nov 2021

- Supercomputers from Poland
  - 131 – Altair (PSNC) (PLGrid)
  - 267 – Ares (ACC Cyfronet AGH) (PLGrid)
  - 426 – Tryton Plus (TASK)
  - 440 – Prometheus (ACC Cyfronet AGH) – 2.4 Pflops (PLGrid)

What is HPC?

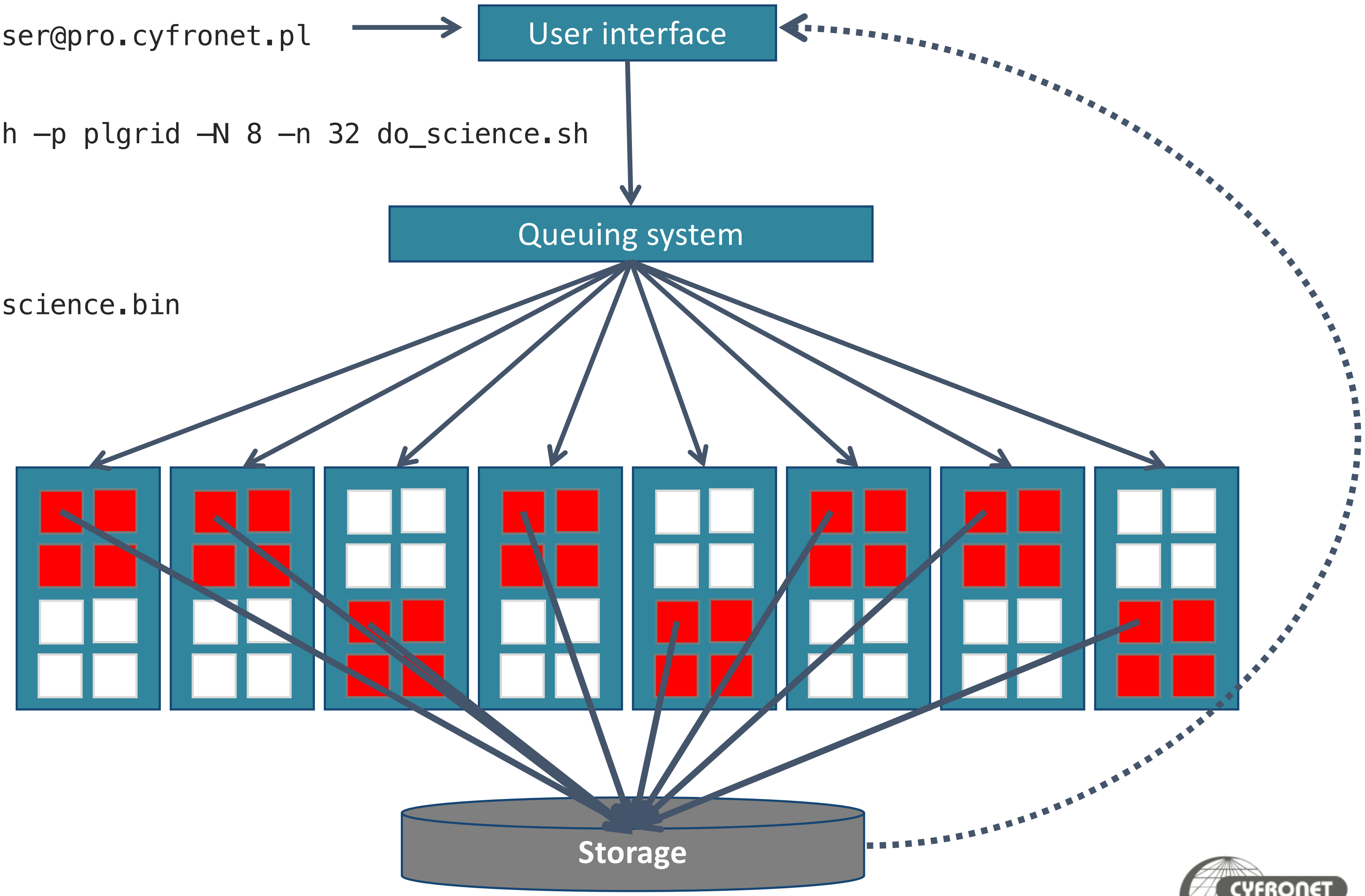
- High Performance Computing (HPC) – using supercomputers to solve problems that cannot be addressed by regular computes
  - use vast amount of processors/cores simultaneously
  - use huge memory allocations
  - use specialized computation accelerators (GPUs, FPGA, ASIC)
  - very fast access to data on dedicated high performance storage systems
- High Throughput Computing (HTC) – processing as much jobs (individual job could be quite simple) as possible using HPC infrastructure
- High Performance Data Analysis (HPDA) - using HPC infrastructure to analyse vast amount of data

High Performance Computing

```
ssh user@pro.cyfronet.pl

sbatch –p plgrid –N 8 –n 32 do_science.sh

srun science.bin
```



Prometheus HPC cluster

- Prometheus consist user interface nodes (UI), service nodes and worker nodes
  - 2 232 nodes worker nodes (2x Intel Xeon E5-2680v3 processors)
    - 72 nodes with additional GPPGU (2x nVidia Tesla K40XL)
  - 3 big memory nodes (2x Intel Xeon Gold 6128, 12 x 3.4 GHz, 768 or 1536 GB)
  - 4 ML/AI nodes (2 x Intel® Xeon® Gold 5220, 36 x 2.2 GHz, 386 GB, 8 x NVIDIA V100 SXM2 32GB HBM2)

Property	Prometheus
CPU frequency	2.50 GHz
RAM	128 GB
cores per node	24
InifiniBand interconnect	available, FDR 56 Gb/s

Ares HPC cluster

- Ares consist user interface nodes (UI), service nodes and worker nodes
  - 788 nodes CPU nodes (2x Intel Xeon Platinum 8268 processors, 48 x 2.9 GHz)
    - 532 nodes with 192 GB (4GB/core)
    - 256 nodes with 384 GB (8GB/core)
  - 9 ML/AI nodes (2 x Intel Xeon Gold 6242, 32 x 2.8 GHz, 384 GB, 8 x NVIDIA V100 SXM2 32GB HBM2)

Property	Ares	Ares GPU
CPU frequency	2.9 GHz	2.8 GHz
RAM	192/384 GB	384 GB
cores per node	48	32
InifiniBand interconnect	available, EDR 100 Gb/s	

Access to computing clusters

- All PLGrid HPC clusters use Linux as OS
  - CentOS 7 on Prometheus & Zeus
  - CentOS 8 on Ares
- HPC clusters contain
  - user interface (UI) node(s)
  - computing nodes (a.k.a worker nodes)
- User interface **must not be used** for computing
- Fair share between users tasks and computations provided by queuing system
  - SLURM on Ares, Prometheus & Zeus

Access to computing clusters

- User log on user interface (UI) node using SSH protocol
  - UI names:
    - login@zeus.cyfronet.pl
    - login@prometheus.cyfronet.pl (login@pro.cyfronet.pl)
      - two login nodes: login01 and login02
    - login@ares.cyfronet.pl
  - SSH clients
    - on Linux and MacOS included in OS
      - `ssh` command in terminal
    - on Windows
      - PuTTY - http://www.chiark.greenend.org.uk/~sgtatham/putty/
      - MobaXterm - http://mobaxterm.mobatek.net
  - copying files and directories
    - on Linux and MacOS included in OS
      - `scp` command in terminal
      - `rsync` command in terminal
    - on Windows
      - WinSCP - http://winscp.net/

Access to computing clusters

- User log on user interface (UI) node using SSH protocol
  - UI names:
    - login@zeus.cyfronet.pl
    - login@prometheus.cyfronet.pl (login@pro.cyfronet.pl)
      - two login nodes: login01 and login02
    - login@ares.cyfronet.pl
  - SSH clients
    - on Linux and MacOS included in OS
      - `ssh` command in terminal
    - on Windows
      - PuTTY - http://www.chiark.greenend.org.uk/~sgtatham/putty/
      - MobaXterm - http://mobaxterm.mobatek.net
  - copying files and directories
    - on Linux and MacOS included in OS
      - `scp` command in terminal
      - `rsync` command in terminal
    - on Windows
      - WinSCP - http://winscp.net/

Introduction to command line interface

- Computes can
    - run processes/programs
    - storage data
    - Interact with each other
    - Interact with users

- Methods of interaction with users
    - CLI (Command-Line Interface)
        - read-evaluate-print loop (REPL)
    - GUI (Graphical User Interface)

- Shell - computer program which exposes an operating system's services to a human user or other programs.
    - Bash, the Bourne Again Shell
- Shell advantages
    - job automation
    - pipelining/creating workflows from many programs
    - often the easiest way to interact with remote computers (i.e. HPC systems)

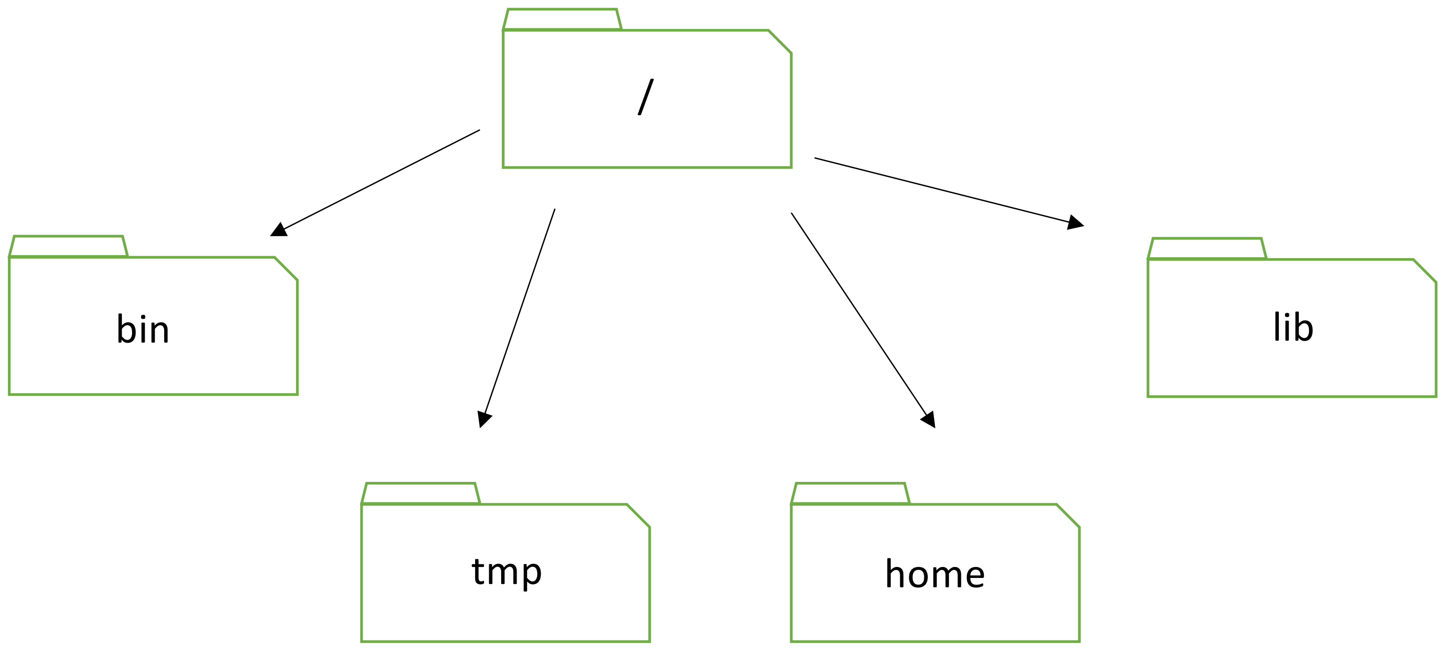
Linux shell

- When the shell is first opened, you are presented with a prompt, indicating that the shell is waiting for input:
  - `$`
- First command, which give information of our username
  - `whoami`
- What is happening after entering `whoami` to shell?
  - shell finds program `whoami`
  - execute it
  - writes program output to standard output (stdout)
  - shell presents a prompt, indicating that is ready for next task

Files and directories

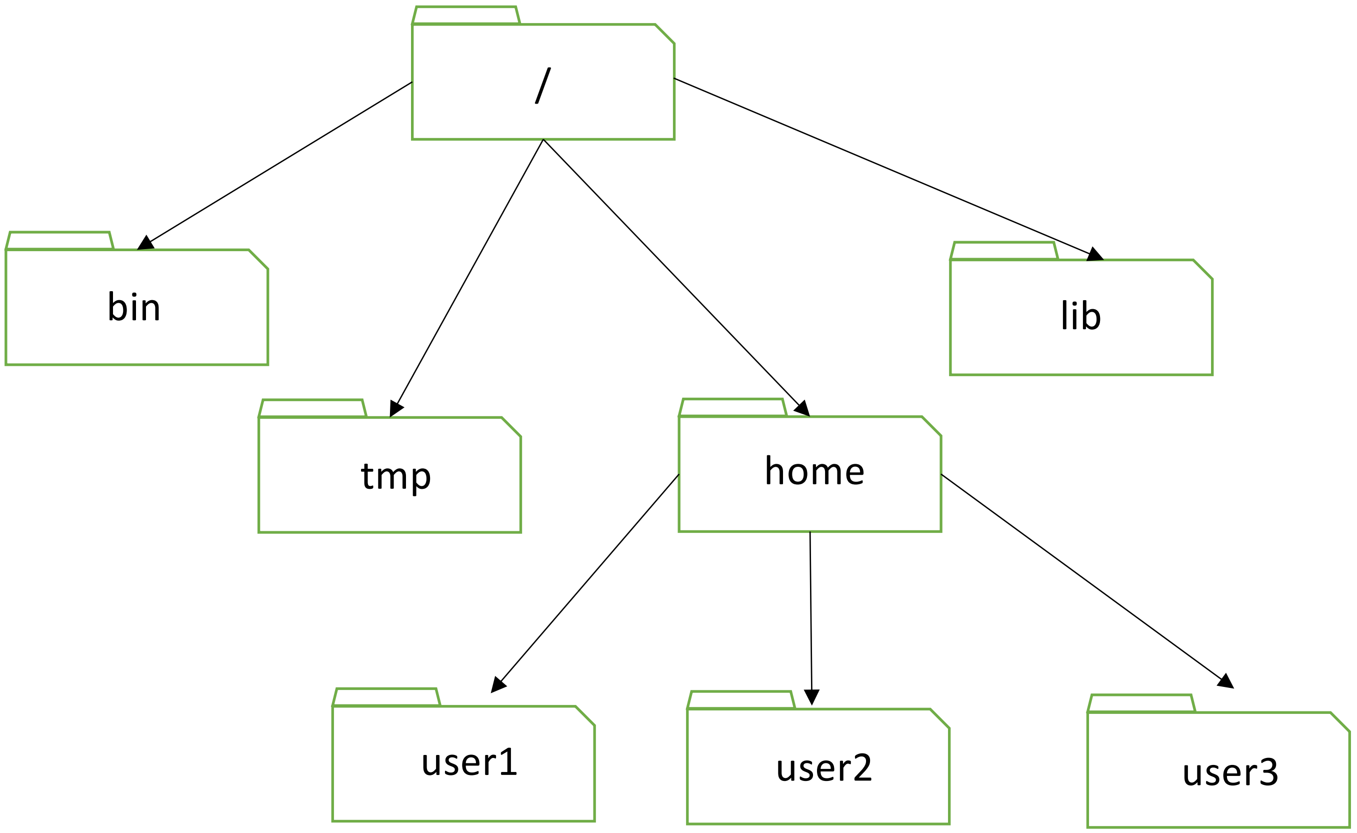
- File system is part of OS which handles data. Usually, it organize it into files and directories
- Tip: in Linux everything is a file (directory, devices, terminals)
- To check where terminal is located in file system type command
  - `pwd`
- This command prints out current working directory (`pwd = print working directory`)
- After start shell by default is pointing to user's home directory(`~/`, `$HOME`)
  - Linux/MacOS: /home/user, /people/user, /Users/user
  - Windows: C:\Documents and Settings\user, C:\Users\user
- At the top of file system is root directory (`/`)

Directories



- `/home` means that `home` is subdirectory of `/`

Directories



- `/home/user1` means that `user1` directory is subdirectory of `home`, which is subdirectory of root directory `/` `(root)`

Files and directories

- List contents of directory
  - `ls [opcje] [argumenty]`

- Używając flag/opcji możemy modyfikować zachowanie programów
  - `ls -F`
- Przydatne opcje polecenia `ls`
  - `-a` – wyświetla wszystkie pliki, w tym ukryte
  - `-t` – sortuje wyświetlanie według czasu modyfikacji
  - `-r` – odwraca kolejność sortowania
  - `-l` – wyświetla dodatkowe informacje w tzw. długim formacie

- Używając argumentów możemy wylistować zawartość innego katalogu
  - `ls -F pdb`

- Każda komenda w systemie linux ma stronę pomocy dostępną przez polecenie `man`
  - `man ls`

Pliki i katalogi

- Listowanie zawartości katalogów
    - używając relatywnych ścieżek
        - `ls -F pdb`
    - używając absolutnych ścieżek
        - `ls -F /home/szkolenie/pdb`

- Zmienianie katalogów – `cd` (change directory)
    - `cd pdb`

- Zmienianie katalogów
    - używając relatywnych ścieżek
        - `cd pdb`
        - `cd ..`
    - używając absolutnych ścieżek
        - `cd /home/szkolenie/pdb`

Pliki i katalogi

- Przydatne skróty do katalogów
  - `./` - bieżący katalog
  - `../` - katalog nadrzędny
  - `~/` - katalog domowy
  - `/` - katalog „root”, „korzeń” systemu plików

- Autouzupełnianianie
  - po wpisaniu fragmentu nazwy polecenia, pliku lub katalogu naciskając klawisz `Tab` powłoka
    - uzupełnia nazwę gdy jest unikalna
    - podaje możliwe uzupełnienia nazwy

- Przydatne skróty klawiszowe
  - `Ctrl + c` przerywa działanie komendy/programu
  - `Ctrl + r` przeszukuje historie wydanych komend
  - ↑ oraz ↓ przechodzenie po historii użytych komend
  - `Ctrl + l` czyści terminal

Tworzenie nowych plików i katalogów

- Do utworzenia nowego katalogu w systemie plików służy polecenie `mkdir`
  - `mkdir new_directory`
  - `mkdir /home/szkolenie/new_directory`
  - `mkdir -p /home/szkolenie/new_directory/another_directory`
- Do tworzenia nowych plików tekstowych można użyć edytora tekstu, np. `nano`
  - `nano new_file.txt`
- Rozszerzenia plików – czy są potrzebne?
  - `type file` – identyfikuje typ pliku
- Do kopiowania plików i katalogów służy polecenie `cp`
  - `cp first_file backup_file`
  - `cp -r first_directory backup_directory`
- Do przenoszenia plików i katalogów służy polecenie `mv`
  - `mv first_file second_file`
  - `mv first_directory second_directory`

Usuwanie plików i katalogów

- Do usuwania plików katalogu w służy polecenie `rm`
    - `rm file`
- Do usuwania pustych katalogów służy polecenie `rmdir`
    - `rmdir empty_directory`
- Do usuwania rekursywnego służy flaga `–r` polecenia `rm`
    - `rm -r first_directory`
    - `rm -ri first_directory`
    - `rm -rf first_directory`
- Usuwanie plików i katalogów w powłoce jest **nieodwracalne**!

Przeglądanie i edycja plików

- Do przeglądania plików można użyć
  - `cat file` – wyświetla zawartość pliku na ekranie (na standardowym wyjściu)
  - `more file` – wyświetla zawartość pliku z opcją przewijania
  - `less file` – wyświetla zawartość pliku z opcją przewijania, również wstecz
- Dla komend `more/less`
  - `/wzorzec` – wyszukanie wzorca w pliku
  - `q` – wyjście do terminala
- Do edycji plików służą edytory: nano, `vim`, `emacs`
  - `nano`, podstawowe komendy
    - `Ctrl + x` – wyjście do terminala
    - `Ctrl + o` – zapis zmian
- Linux/Unix (`LF`) oraz Windows (`CR+LF`) różnie kończą linie w plikach tekstowych
- Konwersja
  - `dos2unix plik`
  - `unix2dos plik`

Prawa dostępu do plików

- Ze względu na dostęp do plików/katalogów użytkownicy podzieleni są na
  - właściciel `(user)`
  - grupa, do której należy właściciel `(group)`
  - pozostali użytkownicy `(others)`

- Prawa dostępu do pliku/katalogu
  - odczyt `(read,r,4)`
  - zapis `(write,w,2)`
  - prawo do wykonania `(execute,x,1)`

- Komenda `chmod` zmienia prawa dostępu
  - składnia
    - komu: `u (user), g (group), o (others), a (all)`
    - operator: + (dodanie praw), −(odjęcie praw), = (ustawienie na podane prawa)
    - prawa: `r (read), w (write), x (excecute)`
- Uruchomienie programu
  - `./program.exe`

Potoki i filtrowanie – komendy pomocnicze

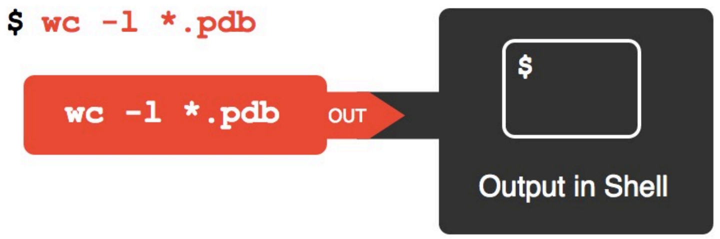
- Do odczytania ilości znaków/linii w pliku służy polecenie `wc`
  - `$ wc -l plik.txt`
- Do sortowania w pliku służy polecenie `sort`
  - `$ sort new_file.txt`
- Do wyświetlania wyłącznie początku/końca pliku służą polecenia `head/tail`
  - `$ head new_file.txt`
  - `$ tail new_file.txt`
- Do wypisywania tekstu służy polecenie `echo`
  - `$ echo I am who am I`
- Do wypisywania plików służy polecenie `cat`
  - `$ cat new_file.txt other_new_file.txt`

Strumienie i przekierowania

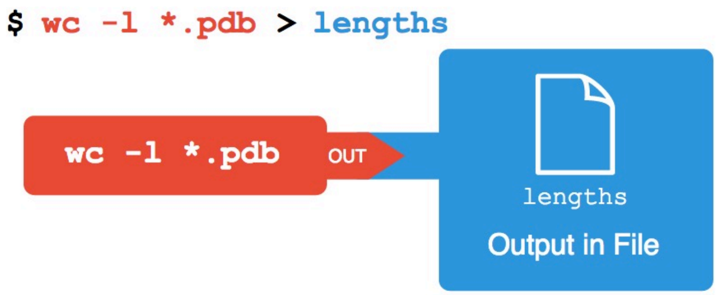
- Znak `*` oznacza dowolny symbol(e) w nazwie pliku (tzw. wildcard)
- Każdy proces w systemie posiada swoją tablicę deskryptorów plików (posiada trzy standardowe strumienie) do komunikacji:
  - standardowy strumień wejścia (0, stdin)
  - standardowy strumień wyjścia (1, stdout)
  - standardowy strumień błędów (2, stderr)
- Do przekierowania strumienia stdout z programu służy symbol `>` lub `>>`
  - `wc -l *.pdb > lines.txt`
  - `echo hello >> hello.txt`
- Do przekierowania strumienia stderr z programu służy symbol `2>` lub `2>>`
  - `wc -l *.pdb 2> lines.txt`
  - `echo hello 2>> hello.txt`
- Do przekierowania strumienia stdin do programu służy symbol `<`
  - `wc -l < test.pdb`

Potoki i filtrowanie

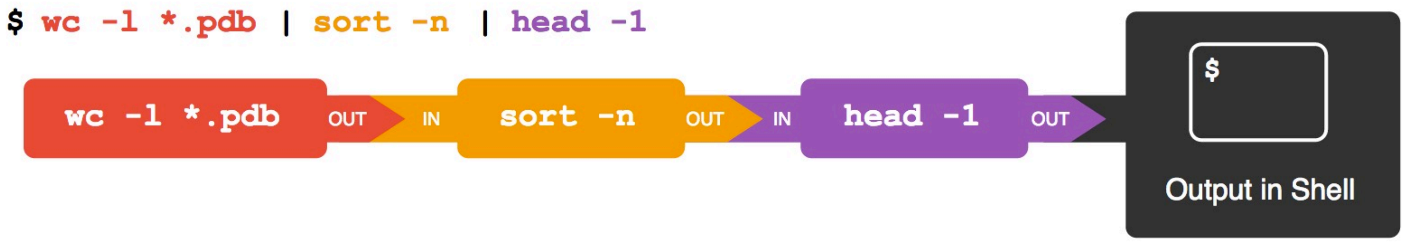
- Potoki (pipes, |) pozwalają łatwo połączyć pracę kilku programów w jeden strumień



```
$ wc -l *.pdb
```



```
$ wc -l *.pdb > lengths
```



```
$ wc -l *.pdb | sort -n | head -1
```

Wyszukiwanie plików oraz wzorców w pliku

- Do przeszukiwania w plikach wzorca służy polecenie `grep`
  - `$ grep wzorzec plik`

- Do wyszukiwania plików/katalogów służy polecenie `find`
  - `$ find directory -name filename`

- Oba polecenia można łączyć
  - ``$ grep wzorzec `find directory -name filename` ``
  - `$ grep wzorzec $(find directory -name filename)`

Skrypty

- Skrypty pozwalają na zapisanie kilku komend wykonywanych sekwencyjnie przez powłokę po uruchomieniu danego skryptu
- Łącząc skrypty, pętle i potoki można wykonywać automatycznie nieskończoną ilość operacji automatycznie na każdym obiekcie z danej grupy
- Czasami lepiej poświęcić chwilę na napisanie dobrego skryptu, który zautomatyzuje pracę, niż wykonywać całość ręcznie
- Do skryptów można przekazać dodatkowe parametry i odczytać je poprzez zmienne `$1`, `$2`, …, `$@`
- Skrypty uruchamia się poprzez podanie ich jako argument powłoki `bash`
    - `bash scriptname`
    - nadanie praw do wykonywania plikowi skryptu

- Dzięki skryptom można całkowicie zautomatyzować swoją pracę!

Prometheus – file systems

- Storage of data – NFS (quite slow, should not be used for heavy I/O calculations)
    - `$HOME` – user's home directory
        - quota 40 GB
    - `$PLG_GROUPS_STORAGE` – additional storage gained through PLGrid grants system
- Temporary scratch file systems
    - `$SCRATCH` – distributed scratch Lustre file system
        - accessible from all nodes of cluster (including UI)
        - `$TMPDIR` and `$SCRATCHDIR` – unique subdirectories on `$SCRATCH` created for the job at it's start
- To check quota use `pro-fs`

Software

- Scientific software usually needs specific runtime environment (i.e. additional libraries) and sometimes technical knowledge is needed to install them efficiently
- Modules and Lmod packages are solutions for loading runtime environments on every cluster in PLGrid infrastructure
- Advantages
  - simplicity of preparing software to run efficiently
  - computation scripts could be transferable between HPC clusters
  - possibility of concurrent runs of different versions of software
  - on hybrid HPC systems transparent switching to most efficient version of software
- Drawbacks
  - additional command to remember .-)

Software – modules

- Load environment for scientific package
    - `module add <module-name>` (i.e. `module add plgrid/apps/r`)
    - `module load <module-name>` (i.e. `module load plgrid/apps/matlab`)
- Remove module
    - `module rm <module-name>` (i.e. `module rm plgrid/apps/r`)
    - `module unload <module-name>` (i.e. `module unload plgrid/apps/matlab`)
- Listing of all available modules
    - `module avail`
    - `module avail plgrid/tools` (only from `tools` branch)
    - `module avail plgrid/apps/r` (all available R versions in `plgrid/apps`)
    - `module spider python` (all available Python versions)
    - `module spider “/r/”` (all available R versions, regexp search)
- Listing of loaded modules
    - `module list`

Software – modules

- Clearing all loaded modules
    - `module purge`
- Saving collection of modules for later use, restoring it and listing saved collections
    - `module save [collection]`
    - `module restore [collection]`
    - `module savelist`
    - `module describe [collection]`
- `ml` is shorthand for `module` command
    - `ml = module list`
    - `ml <module-name> = module load <module-name>`
    - `ml -<module-name> = module unload <module-name>`
    - `ml av <string> = module avail <string>`
- Getting help
    - `module help`
    - `ml -h`

Software – modules

- Each software package installed in PLGrid infrastructure has it's own module
  - `plgrid/<branch>/<software-name>/<version>`
- Branch kinds
  - `apps` – for most of scientific packages
  - `libs` – for software libraries
  - `tools` – for toolkits and helper packages
- User's own modules
  - `module use path` – adds `path` with additional modules
- Examples:
  - `plgrid/tools/intel/19.0.5`
  - `plgrid/apps/r/3.6.0`
  - `plgrid/tools/python/3.6.5`
  - `plgrid/apps/relion`

https://apps.plgrid.pl/

SLURM queuing system – commands

- User interact with SLURM queuing system using commands
    - `sbatch` – to submit new job to queue
    - `squeue` – gives information about jobs running in queuing system
    - `scancel` – deletes jobs from queue
    - `sinfo/scontrol` – gives detailed information about queue, job or node
    - `smap` – gives graphical information about state of HPC cluster
    - `srun` – runs interactive job or step in batch job

- Each job has got **unique job identifier** (jobID)

Queuing system

- Queuing system
  - manage all computational task on cluster
  - monitor available resources
  - acts as matchmaker between needs of jobs and resources
  - empowers fair share between different users
- All computational tasks are run as **jobs** queued in **queues** and run according to their priority and available resources.
- Priority of job depends on
  - amount of resources obtained by user in computational grant
  - amount of resources requested by job
    - **maximum wall time of computation** is most essential resource
  - amount of other resources concurrently used by job's owner

Queuing systems in PLGrid

- HPC clusters available in PLGrid use several kinds of queuing systems
    - SLURM (http://slurm.schedmd.com)
    - PBS Pro (http://pbspro.org)

HPC Centre	Cluster	Queuing system
ACC Cyfronet AGH	Prometheus	SLURM
	Zeus	SLURM
ICM	Topola	SLURM
PSNC	Eagle/Altair	SLURM
TASK	Tryton	SLURM
WCSS	Bem	PBS Pro

SLURM queuing system – job submission

- Command `sbatch` submits new job in queue
- All parameters describing job's requirements could be included in batch script and given to queuing system using command
  - `sbatch [options] script.slurm`
- Example script

```
#!/bin/env bash

# Commands that will be run after start of the job
echo "Computation started on work node: ";
hostname

module add plgrid/apps/matlab

matlab –nodisplay <matlab.in >matlab.out
```

SLURM queuing system – job monitoring

- Commands `squeue` and `pro-jobs` give view of jobs scheduled in queuing system
- Jobs States
  - `PD` – queued
  - `R` – running
  - `CF` – configuring (resources for job are being prepared)
- Additional helpful flags
  - `squeue --user $USER` – information about `$USER`'s jobs
  - `pro-jobs -j <jobID>` – information about specified jobs
  - `pro-jobs -N` – additional information about information about exec nodes
  - `pro-jobs -q/-r` – information about queued (pending)/running jobs only
  - `pro-jobs -h` – help screen
- In addition `scontrol`, `sinfo` and `smap` give information about status of cluster
  - `scontrol show job <jobID>` – information about `<jobID>` job
  - `scontrol show node <nodes_list>` – information about nodes

Available partitions

Partitions	max time	Information
plgrid-testing	1:00:00	for test runs (small number of jobs)
plgrid-short	1:00:00	
plgrid	3-00:00:00	
plgrid-now	12:00:00	interactive runs, max one job on one node
plgrid-long	7-00:00:00	*
plgrid-gpu	3-00:00:00	nodes with GPGPU*
plgrid-gpu-v100	3-00:00:00	nodes with V100 GPGPU*
plgrid-bigmem	3-00:00:00	big mem nodes*

- In SLURM queues are called partitions
- `scontrol show partitions <patition_name>` – detailed information about partition
- `sinfo` – lists all available nodes in all partitions
    - `sinfo –p <partition_name>` – lists information only about partition
- default time in all `plgrid*` partitions is set to 15 minutes

* - partitions available after request

SLURM – example Python job

```
#!/bin/env bash

# Commands that will be run after start of the job
echo "Computation started on work node: "; hostname

module add plgrid/tools/python

./python-script.py > python.log
```

- SLURM options provide information about job requirements to queuing system. They could be
  - given in command line `sbatch [SLURM options]`
  - included in first lines of batch script with `#SBATCH` at start of line

SLURM – `sbatch` command options

- `sbatch` command uses various options to provide queuing system with additional info about the job
  - `-p <partition>, --partition=<partition>` defines partition
  - `-J <jobname>, --job-name=<jobname>` give name to job
  - `-a, --array=<indexes>` submit a job array
  - `--mail-user=<user’s e-mail>` setting email for notifications
  - `--mail-type=<type>` information when notifications should be send: at beginning (`BEGIN`), end (`END`) or execution error (`FAIL`)
  - `-A <grantID>, --account=<grantID>` information about computational grant (if omitted job use default)

- When option `-p` is omitted job is queued into default partition (on Prometheus `plgrid`)

SLURM – job requirements specification

- There are several recourses available for job
  - `-t, --time=<time>` total maximal execution wall time of job
  - `-N, --nodes=<nodes>` amount of nodes allocated to job
  - `-n, --ntasks=<ntasks>` amount of tasks invoked in whole job
  - `--ntasks-per-node=<ntasks>` amount of tasks invoked on each node
  - `--cpus-per-task=<cores>` amount of cores per each task (i.e. when using threads in OpenMP)
  - `--mem=<MB>`amount of memory per node requested by job
  - `--mem-per-cpu=<MB>` amount of memory per core requested by job

- Parameter formats
  - time format: `"min", "min:sec", "hours:min:sec", "days-hours", "days-hours:min" and "days-hours:min:sec"`
  - memory: `MB (=1024kB), GB (=1,024MB)`

SLURM – example serial job

```
#SBATCH --job-name=serial.job
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=10:00
#SBATCH --mem=24000
#SBATCH --partition=plgrid
#SBATCH --account=plgtraining2021

module add plgrid/tools/intel

icc -xHost hello.c -o hello.x

./hello.x
```

- In SLURM job is sent to partition not to queue
  - flag `-p <partition_name>` or `--partition <partition_name>`
  - partition for PLGrid users: `plgrid*`

SLURM – example parallel job

```
#SBATCH --job-name=parallel-srun
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=2
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=1GB
#SBATCH --partition=plgrid
#SBATCH --account=plgtraining2021

module add plgrid/tools/intel

icc -xHost hello.c -o hello.x

srun ./hello.x
```

- `srun` inside batch job executes command `./hello.x` on allocated resources according to `requested --ntask` or `--nodes*--ntasks-per-node` flags
  - variable SLURM_NTASKS holds information about number of tasks to be run
- each `srun` could request more than one core
  - `srun -nodes=x --ntasks=y --cpus-per-task=z …`

SLURM – example parallel OpenMP job

```
#SBATCH --job-name=parallel-openmp
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=24
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=2GB
#SBATCH --partition=plgrid
#SBATCH --account=plgtraining2021

module add plgrid/tools/intel

icc -xHost –qopenmp hello.c -o hello.x

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

./hello.x
```

- When use OpenMP
  - use `--cpus-per-task=<cores_per_job>` and `--nodes=1` for request of resources
  - variable `SLURM_CPUS_PER_TASK` holds information about number CPUs allocated to each task

SLURM – example parallel MPI job

```
#SBATCH --job-name=distributed-mpi
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=24
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=1GB
#SBATCH --partition=plgrid
#SBATCH --account=plgtraining2021

module add plgrid/tools/impi

mpiicc -xHost hello.c -o hello.x

mpiexec -np $SLURM_NTASKS ./hello.x
```

- When software is parallelized using MPI
  - use `--ntasks-per-node=<cores_per_node` and `--nodes=<no_of_nodes>` for request of resources
  - variable `SLURM_NTASKS` holds information about number of tasks to be run

SLURM – example parallel hybrid MPI/OpenMP job

```
#SBATCH --job-name=mpi-openmp
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=6
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=2GB
#SBATCH --partition=plgrid
#SBATCH --account=plgtraining2021

module add plgrid/tools/impi

mpiicc -xHost -qopenmp hello.c -o hello.x

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

mpiexec -np $SLURM_NTASKS ./hello.x
```

- When hybrid MPI/OpenMP
  - use `--cpus-per-task=<cores_per_job>` and `$SLURM_CPUS_PER_TASK` for distribution of threads
  - use `--ntasks-per-node=<cores_per_node>` for request of MPI processes
  -

SLURM – environmental variables

- SLURM adds environmental variables which could ease performing computation

Variable	**Description**
`SLURM_JOB_ID`	job identifier (jobID)
`SLURM_SUBMIT_DIR`	dir, from which batch script was submitted to queuing system
`SLURM_NTASKS`	total number of tasks (i.e. MPI processes) in the current job
`SLURM_NTASKS_PER_NODE`	number of tasks to be run on one node
`SLURM_NODELIST`	list of nodes allocated to the job
`SLURM_CPUS_PER_TASK`	number of cores requested per task
`TMPDIR`, `SCRATCHDIR`	scratch file temporary directories for job
`SCRATCH`	$USER's root scratch directory on distributed Lustre file system
`SCRATCHDIR`	unique directory for the job on `$SCRATCH`

- Environment variables can be used to control distribution of job
  - MPI jobs: `SLURM_NTASKS` to run MPI processes (using `srun`) variable
  - OpenMP jobs: `SLURM_CPUS_PER_TASK` to run proper number of threads
  - hybrid MPI/OpenMP jobs: combine `SLURM_NTASKS` to run MPI processes and `SLURM_CPUS_PER_TASK` to expand threads

SLURM – interactive jobs

- Interactive work on cluster should be done using interactive jobs trough `srun` command
    - `srun -p plgrid -A <grant_id> -n 1 --pty /bin/bash`

- User interface **must not be used** for computing

- High priority queue `plgrid-now` for interactive work
    - one job on one node up to 12:00:00

- To attach terminal to running batch job
- `srun -N1 -n1 --jobid=<jobID> --pty /bin/bash`
- `srun -N1 -n1 --jobid=<jobID> -w <nodeID> --pty /bin/bash`
- `sattach <jobid.stepid>`

- Prometheus helper script `ssh_slurm`
    - `ssh_slurm <jobid> <dest_host> [command]`

SLURM `scancel` – deleting jobs

- `scancel` command is used to delete unwanted jobs from queuing system
  - `scancel <JobID>`

- Information about jobs which cannot be deleted using `scancel` should be sent to system administrators through
  - Helpdesk PLGrid PL
    - https://helpdesk.plgrid.pl
    - helpdesk@plgrid.pl
  - directly to system administrators prometheus@cyfronet.pl

Job monitoring with `pro–jobs*`

- `pro–jobs/hpc–jobs` and `pro–jobs–history/hpc–jobs–history` could be used to monitor efficiency of jobs
  - memory usage
  - CPU usage

- `pro–jobs/hpc–jobs` – running and queued jobs
- `pro–jobs–history/hpc–jobs–history` – historical data of completed jobs

- `pro–jobs*/hpc–jobs*` usage
  - `pro–jobs –N` – additional information about nodes of job(s)
  - `pro–jobs –v` – more detailed information about job(s)
  - `pro–jobs –j (<jobID>)` – information only about job(s)
  - `pro–jobs –h` – help screen
  - `pro–jobs–history –d <period>` jobs completed in last <period> days

Best practices

- SLURM job batch script is always started in directory from which it was submitted to queuing system. Access to that directory is also possible with `SLURM_SUBMIT_DIR`

- All batch jobs have got file in which data from standard outputs (both standard output stream `stdout` and standard error stream `stderr`) is stored named `slurm-<JobID>.out`
  - those file should not be big (less than several MBs) and are stored in `SLURM_SUBMIT_DIR`
  - `-o, --output=<file>` and `-e, --error=<file>` - options to redirect `stdout` and `stderr`

- When commands in SLURM script print big amount of data into output streams user should redirect that data to file(s)
  - for standard output stream (*stdout*): `command > file.out`
  - for standard error stream (*stderr*): `command 2> file.err`
  - for both streams to one file: `command &> file.log`

- `$HOME` and `$PLG_GROUPS_STORAGE` **must not be used** for heavy I/O computations

Best practices

- During batch job submission user should always
  - specify maximal time of job execution (parameter `t/time`)
  - specify maximal RAM amount needed by job through `mem` (or `mem−per−cpu`)
  - enable checkpoints
  - for parallel computations use all cores on nodes when possible
  - when big amount of data is used in computation always use `$SCRATCH` for files
  - when big amount of data is going to be passed to standard output streams redirect it to files and use `$SCRATCH`
  - load runtime environment of software via `module` command in batch script
  - do not load software modules in scripts loaded at user's login (i.e.`.bashrc`)

Computational grants

- Obtained through PLGrid Portal - https://bazaar.plgrid.pl/
  - distinct grants for GPGPU
- Commands
  - `plg-show-grants (pro-show-grants)`
  - `plg-show-grant-details <account> (pro-show-grant-details <account>)`
  - `plg-show-default-grant (pro-show-default-grant)`
- Accounting portal - https://accounting.plgrid.pl/

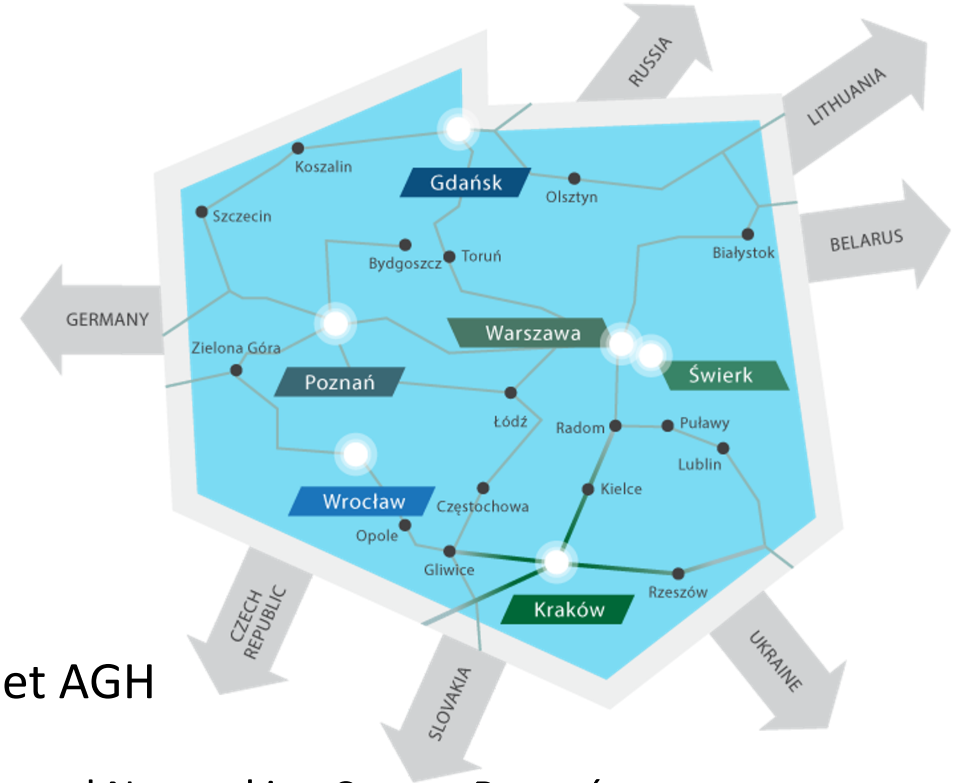
Additional „filesystems”

- MEMFS
  - `-C memfs`
  - `$MEMFS`
  - use memory as filesystem (120GB max)
    - Accessible only within node
  - available during JOB and lost after it finishes

- LOCALFS
  - `-C localfs`
  - `$SCRATCH_LOCAL`
  - use file as filesystem (512GB per node)
  - Each node has its own file! (not a shared filesystem)
    - Accessible only within node
  - Available during JOB and lost after it is finished

PLGrid Infrastructure

PLGrid Infrastructure



- Projects:
  - PL-Grid
  - PLGrid Plus
  - PLGrid NG
  - PLGrid Core
- PLGrid Consortium
  - Coordinator: ACC Cyfronet AGH
  - Partners:
    - Poznan Supercomputing and Networking Center, Poznań
    - Interdisciplinary Centre for Mathematical and Computational Modelling, Warszawa
    - Wroclaw Centre for Networking and Supercomputing, Wrocław
    - Tricity Academic Computer Centre, Gdańsk
    - National Centre for Nuclear Research, Świerk

http://www.plgrid.pl/en/

PLGrid - computational infrastructure for science

- The PLGrid Infrastructure is available free of charge for Polish researchers and all those engaged in scientific activities in Poland
- On-line registration through PLGrid Users' Portal – https://portal.plgrid.pl
- User verification based on Polish Science Database – https://www.nauka-polska.pl

On PLGrid Users Portal user can

- apply for access to tools and services
- monitor utilization of resources
- manage their computational grants and grid certificates

Access to all PLGrid resources through **one account** and **one passphrase** (or grid certificate)

PLGrid - accessing resources

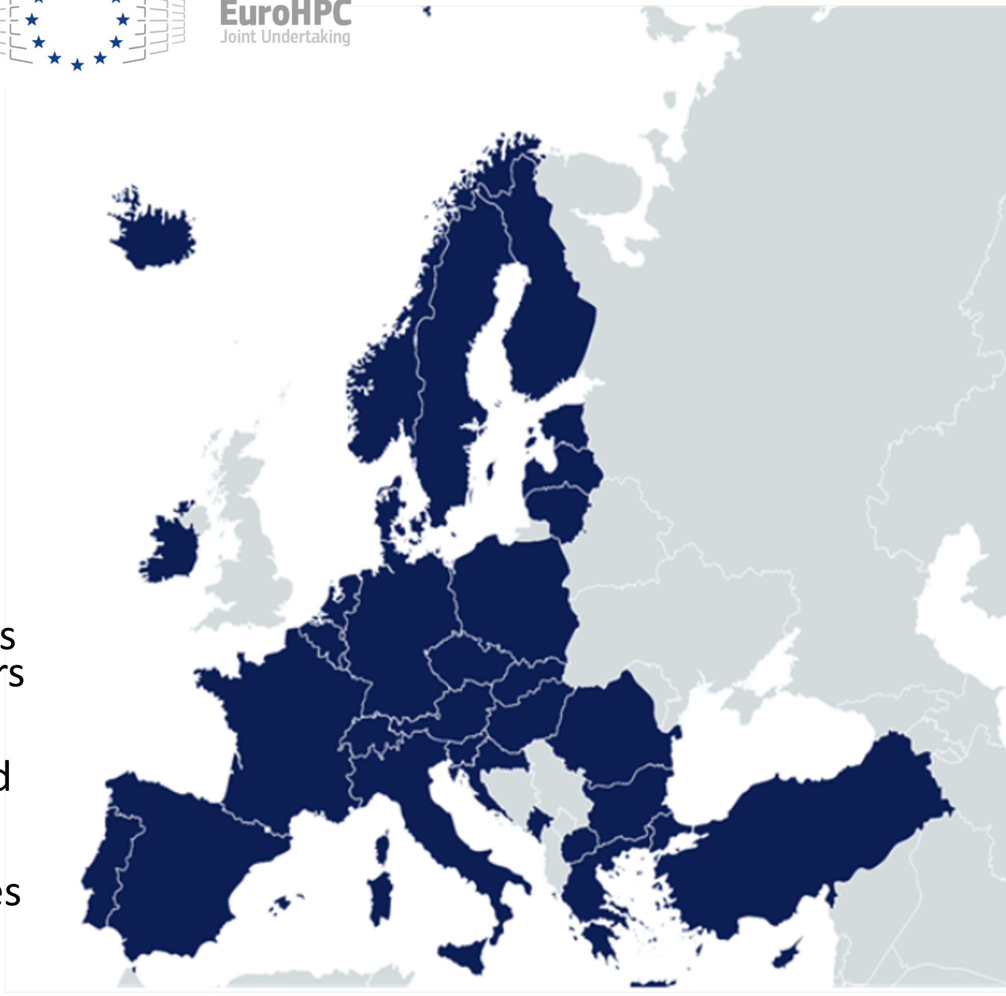
Steps necessary to grant access to PLGrid resources

- Create account at PLGrid Users' Portal – https://portal.plgrid.pl
- Create (Scientific) Affiliation
- Create Team
- Create Computational Grant for the team
- Apply for necessary services/entry points at Services and Applications Catalogue - https://apps.plgrid.pl

EuroHPC

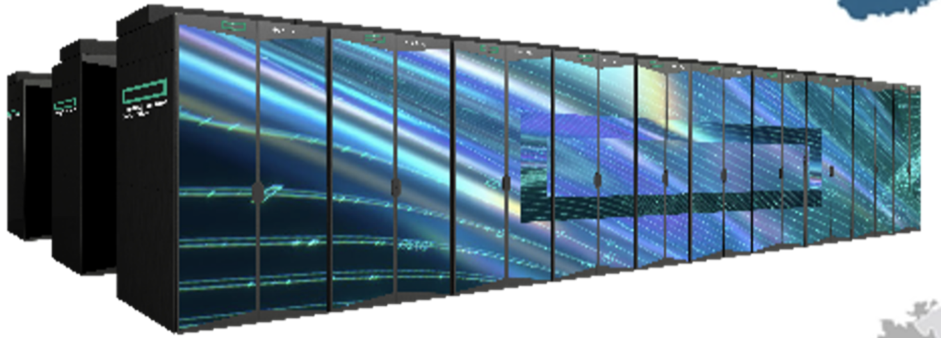
- The European High Performance Computing Joint Undertaking
  - 32 participating countries
  - the European Union (represented by the European Commission)
  - private partners

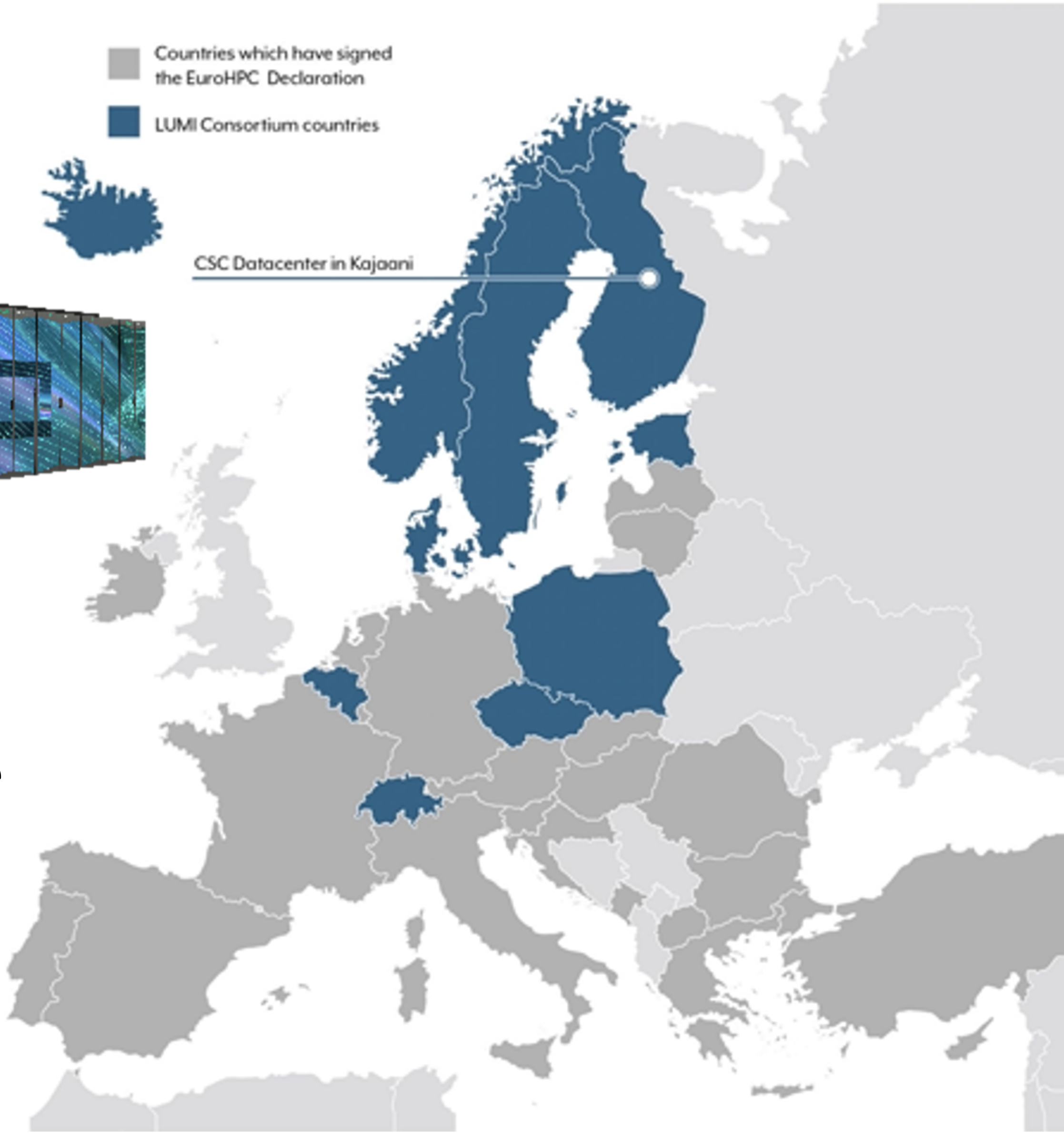
- Goals
  - deploy top-of-the-range supercomputing infrastructures across Europe to support European HPC users wherever they are in Europe
  - implement an ambitious research and innovation agenda to develop a competitive HPC ecosystem and supply chain in Europe, which includes hardware, software, applications but also training and skills



https://eurohpc-ju.europa.eu/

LUMI Consortium




Countries which have signed the EuroHPC Declaration
LUMI Consortium countries
CSC Datacenter in Kajaani

- LUMI will be an **HPE Cray EX** supercomputer manufactured by Hewlett Packard Enterprise
- Peak performance over 550 petaflop/s makes the system one of the world's fastest
- Available for users in
  - LUMI-C Q4 2021
  - LUMI-G Q1 2022

https://www.lumi-supercomputer.eu/

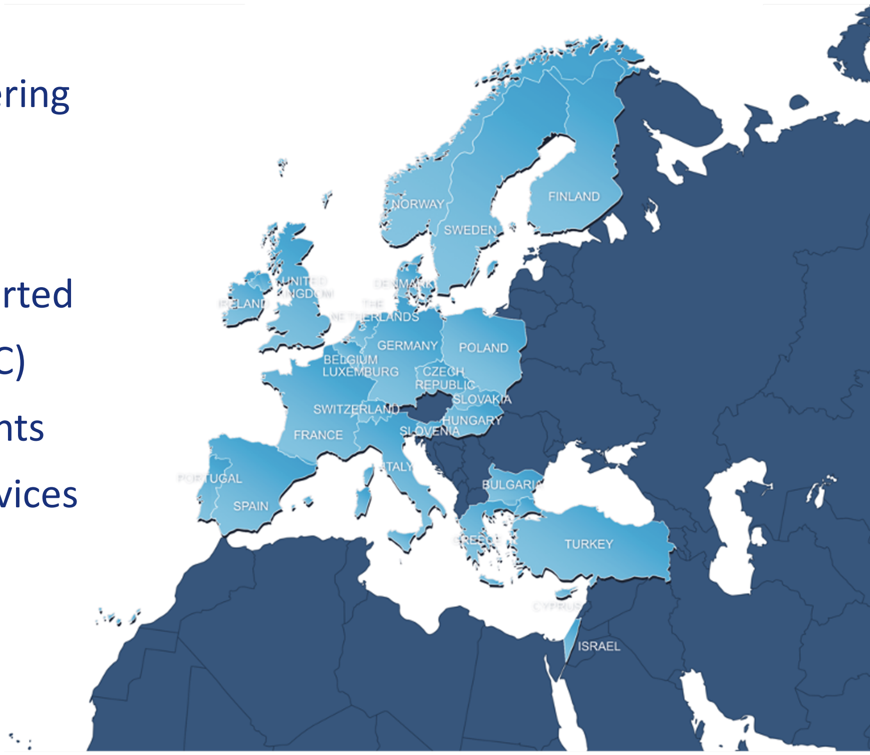
- National Competence Centres for EuroHPC
- Goals
  - Establishing network of national HPC competence centers in all EuroHPC member states
  - Focus on cooperation between all stakeholders in european HPC
  - Training of scientific staff and development of HPC software in both academia and industrial environments

www.eurocc-project.eu

cc.eurohpc.pl

Partnership for Advanced Computing in Europe

- Open access to world-class HPC systems to EU scientists and researchers
- Variety of architectures to support the different scientific communities
- High standards in computational science and engineering
- Peer Review at European level to foster scientific excellence
- Robust and persistent funding scheme for HPC supported by national governments and European Commission (EC)
- Support the development of intellectual property rights (IPR) in Europe by working with industry and public services
- Collaborate with European HPC industrial users and suppliers
- Training and Outreach for HPC scientist and students



https://prace-ri.eu/

PRACE | members

Hosting Members

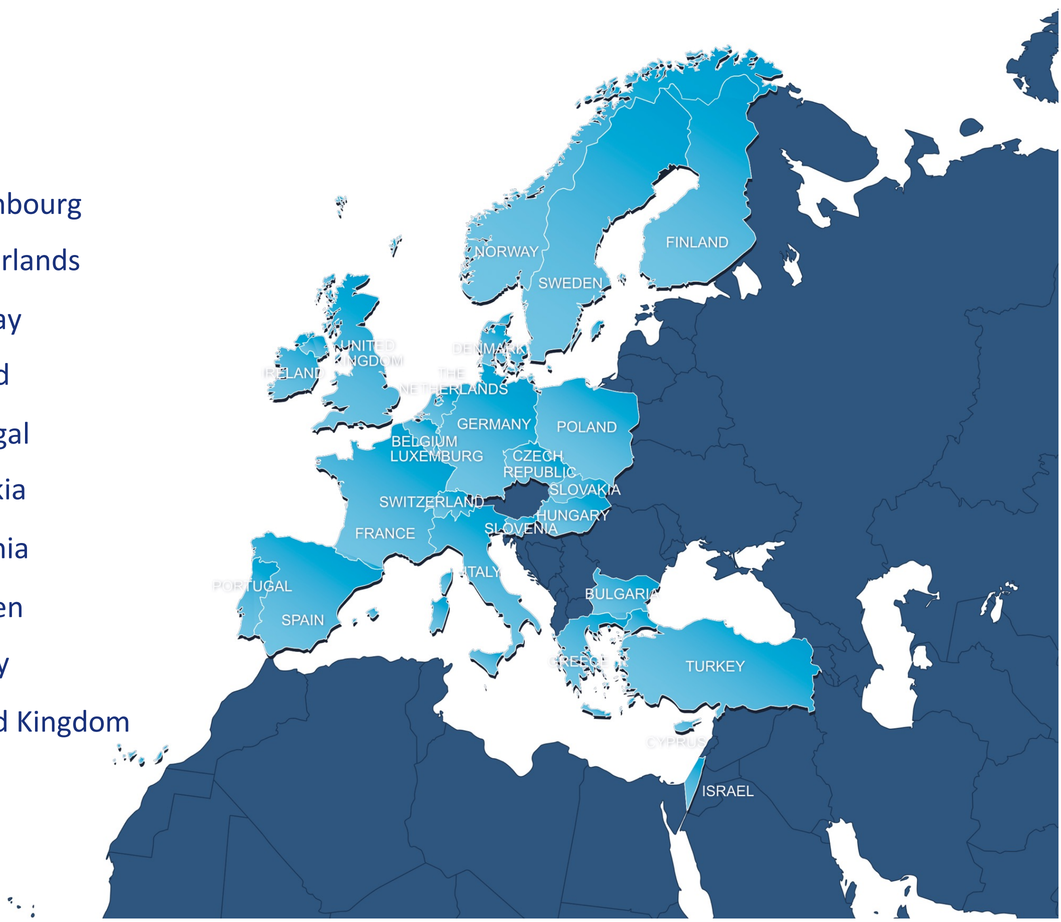
- France
- Germany
- Italy
- Spain
- Switzerland

General Partners (PRACE 2)

- Belgium
- Bulgaria
- Cyprus
- Czech Republic
- Denmark
- Finland
- Greece
- Hungary
- Ireland
- Israel
- Luxembourg
- Netherlands
- Norway
- Poland
- Portugal
- Slovakia
- Slovenia
- Sweden
- Turkey
- United Kingdom

Observers

- Croatia
- Romania



PRACE | Tier-0 Systems

MareNostrum: IBM
BSC, Barcelona, Spain

Piz Daint: Cray XC50
CSCS, Lugano, Switzerland

SuperMUC-NG: Lenovo ThinkSystem
GAUSS @ LRZ, Garching, Germany

Joliot Curie: BULL Sequana X1000
GENCI/CEA, Bruyères-le-Châtel, France

MARCONI: Lenovo
CINECA, Bologna, Italy

JUWELS: BULL Sequana X1000
GAUSS @ FZJ, Jülich, Germany

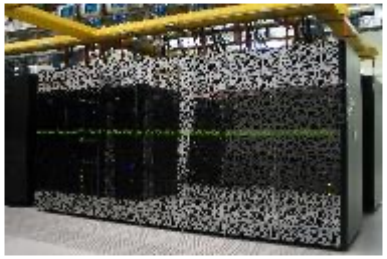
PRACE | Tier-1 Systems

ARCHER: Cray XC30
EPCC, Edinburgh, UK
#252 Top 500

Prometheus: HPE Apollo 8000
ACC Cyfronet AGH-UST, Krakow, Poland
#174 Top 500

Beskow: Cray XC40
KTH, Stockholm, Sweden
#151 Top 500

Salomon: SGI ICE X
IT4I, Ostrava, Czech Republic
#282 Top 500



Cartesius: Bull Bullx B720/B710
SURFSara, Amsterdam, The Netherlands
#455 Top 500

Puhti: BullSequana X400
CSC, Espoo, Finland

PRACE | project access

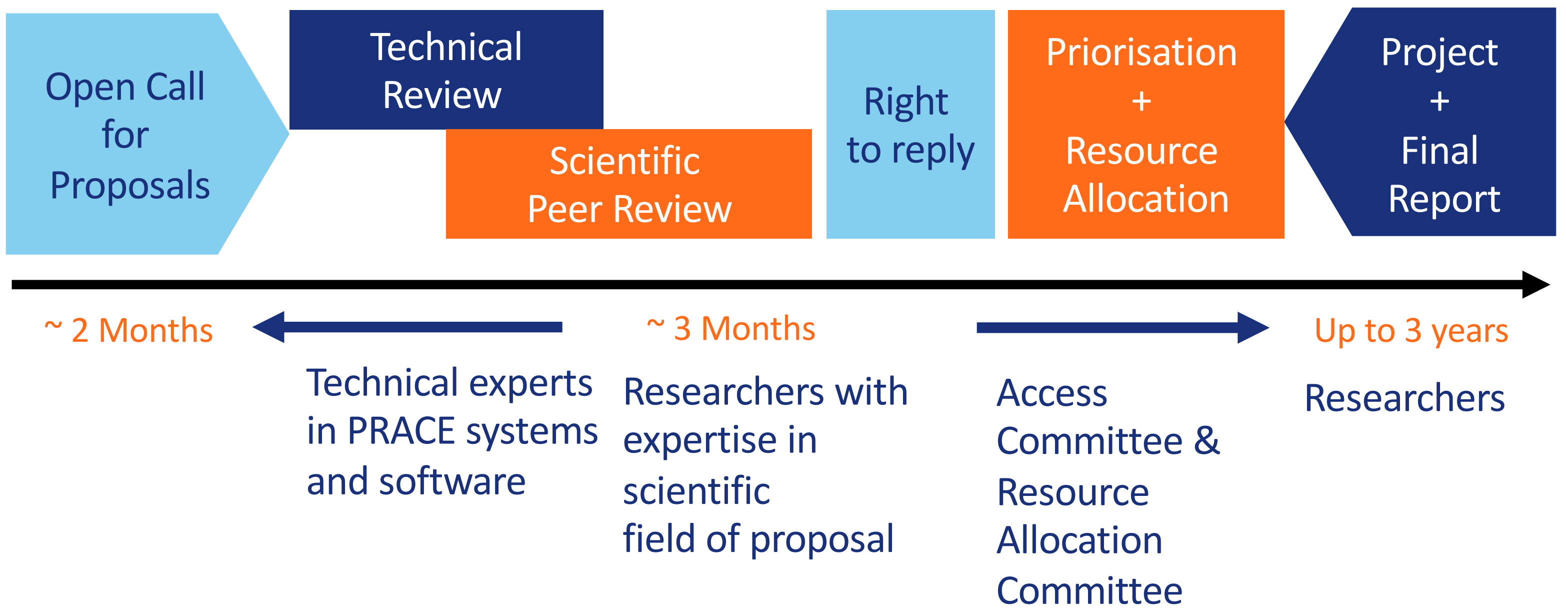
- Free-of-charge required to publish results at the end of the award period
- Preparatory Access (2 to 6 months)
- Project Access (12, 24 or 36 months)
- SHAPE Programme (2 to 6 months)
- Distributed European Computing Initiative (Tier-1 12 months)

Criterion: Scientific Excellence Assessed by an improved review process

www.prace-ri.eu/call-announcements/

CYFRONET

PRACE | project access

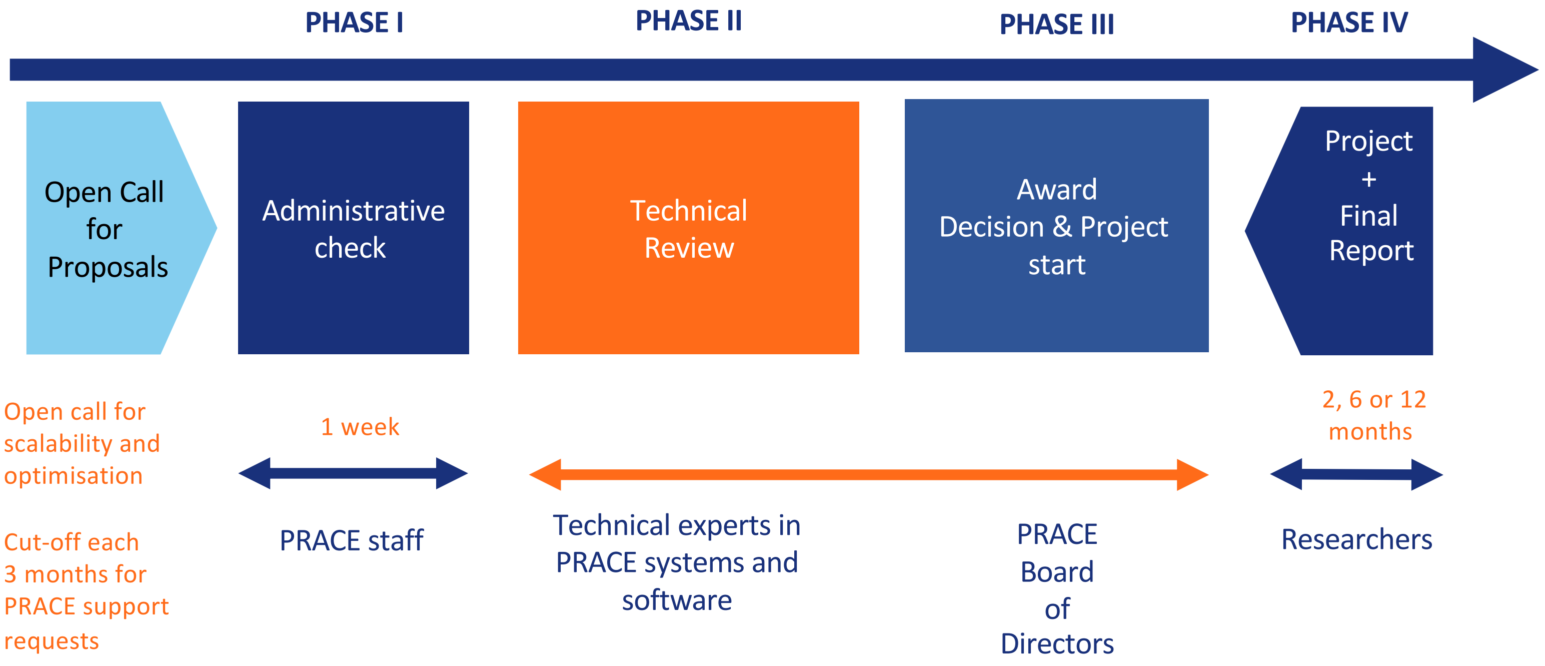

Open Call for Proposals
Technical Review
Scientific Peer Review
Right to reply
Priorisation + Resource Allocation
Project + Final Report
~ 2 Months
~ 3 Months
Up to 3 years
Technical experts in PRACE systems and software
Researchers with expertise in scientific field of proposal
Access Committee & Resource Allocation Committee
Researchers

http://www.prace-ri.eu/prace-project-access/

PRACE | project access

- 24th Call for Proposals for Project Access
  - Opening of the call: 9 September 2021
  - Closing of the call: 2 November 2021, 10:00 CET
  - Allocation period for awarded proposals: April 2022 – March 2023
  - Type of Access: Project Access and Multi-Year Project Access
- Applications for Project Access must use codes that have been previously tested and
  - demonstrate high scalability and optimization to multi-core architectures
  - demonstrate a requirement for ensemble simulations that need a very large amount of CPU/GPU

PRACE | preparatory access



http://www.prace-ri.eu/prace-preparatory-access/

PRACE | Distributed European Computing Initiative

- 17th Call for Proposals for DECI (Tier-1)
  - Opening of the call: 16 December 2020
  - Closing of the call: 31 January 2019, 18:00 UTC
  - Allocation period for awarded proposals: June 2021 – May 2022
  - Type of Access: DECI (Tier-1)
- Applications for DECI:
  - projects requiring access to Tier-1 resources that are not currently available in PI's own country or for international collaborations
  - individual projects limited to around 5 million machine hours (2.5 million machine hours in average)

PRACE | Training and Outreach activities

provide a sustained, high-quality training and education service for the European HPC community

- 6 PRACE Advanced Training Centres (PATCs) and 4 Training Centres (PTCs)
- PRACE training events: Seasonal Schools, International HPC Summer School, On-demand training events
- Summer of HPC (programme for undergraduate and postgraduate students)
- PRACE Training and Events portal
- CodeVault, Massive Open Online Courses (MOOCs)

Training topics

Different levels of training

- Basic, intermediate, advance

High performance computing

- Parallel programming
- Accelerators
- Performance optimization

Domain-specific topics

- Simulation software
- Visualization
- Data intensive computing

PRACE | Training and Events Portal

- www.training.prace-ri.eu
- Single hub for the PRACE training events, training material and tutorials
- PATC Programme 2020-2021
  - Online training events due to COVID19
  - New courses on forward-looking topics
  - New hardware and programming paradigms
  - Data science
  - Collaboration with CoEs on several courses

Ministry of Science and Higher Education
Republic of Poland

"Prace realizowane przy wsparciu Ministerstwa Nauki i Szkolnictwa Wyższego, decyzja nr DIR/WK/2016/18"

CYFRONET

Umiesz liczyć?
LICZ U NAS!