Introduction to HPC in Computational Modelling

PATC: HPC-based simulations, Engineering and Environment with applications in Bioengineering

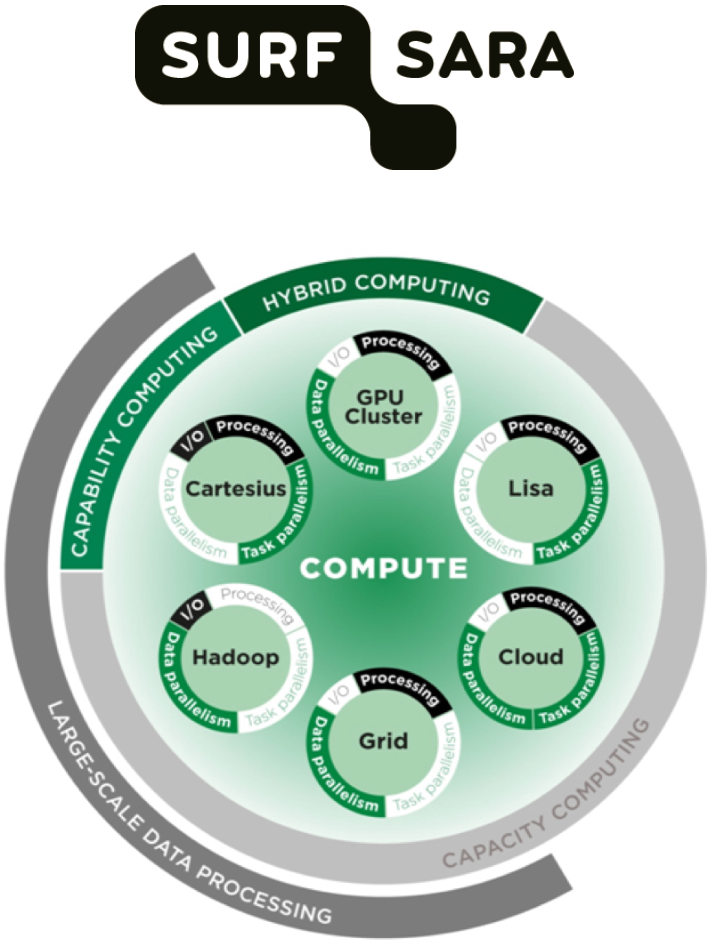
Marco Verdicchio
SURFsara

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675451.

HPC at SURFsara

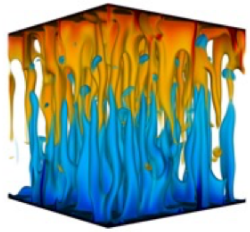
- HPC Consultant in the Supercomputing Team
- Science Park, Amsterdam
- About 70 op. people
  - 50% Consultants
  - 50% System programmers

HPC at SURFsara



HPC at SURFsara

- Maintain Cartesius and Lisa
- User support
- Training
- Development and Innovation
- Involvement in EU projects



Intel Parallel Computing Center

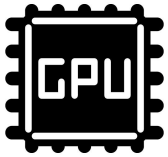
Introduction to HPC - Outline

- What is a Supercomputer?
  - Working on a Supercomputer
  - Supercomputer access and usage
- Getting started with HPC
  - Linux basic command
  - Bash scripting
- Running jobs
  - Use the batch system
  - Execute parallel programs

What is a Supercomputer?

User Experience

- Multiuser system
- Unix OS
- Optimized software



Compute power

- Many CPUs system
- Specialized Hardware
- Low-latency/High bandwidth Connections

Storage

- Efficient I/O
- Large Memories

What is a Supercomputer?

TOP 10 Sites for November 2017

For more information about the sites and systems in the list, click on the links or view the complete list.

1-100 | 101-200 | 201-300 | 301-400 | 401-500

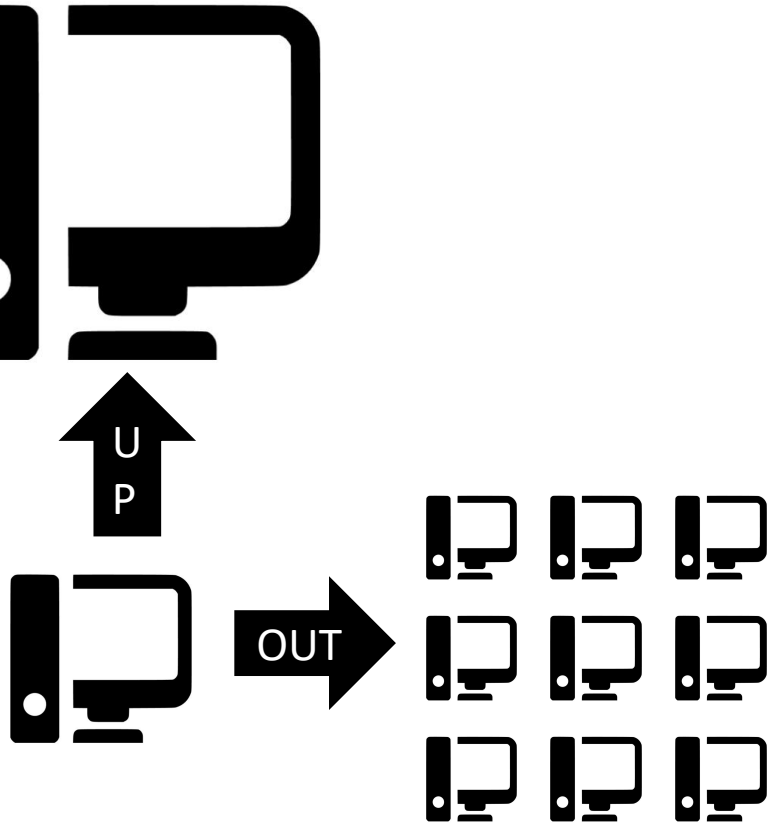
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
2	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P , NUDT National Super Computer Center in Guangzhou China	3,120,000	33,862.7	54,902.4	17,808
3	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	25,326.3	2,272
4	**Gyoukou** - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz , ExaScaler Japan Agency for Marine-Earth Science and Technology Japan	19,860,000	19,135.8	28,192.0	1,350
5	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209

www.top500.org

What is a Supercomputer?

Why, or more, when you need a Supercomputer?

- **Scale up**
  - Faster CPUs
  - Large memories
  - Specialized Hardware/Software
- **Scale out**
  - Large parallel applications
  - Many small- to medium- size jobs

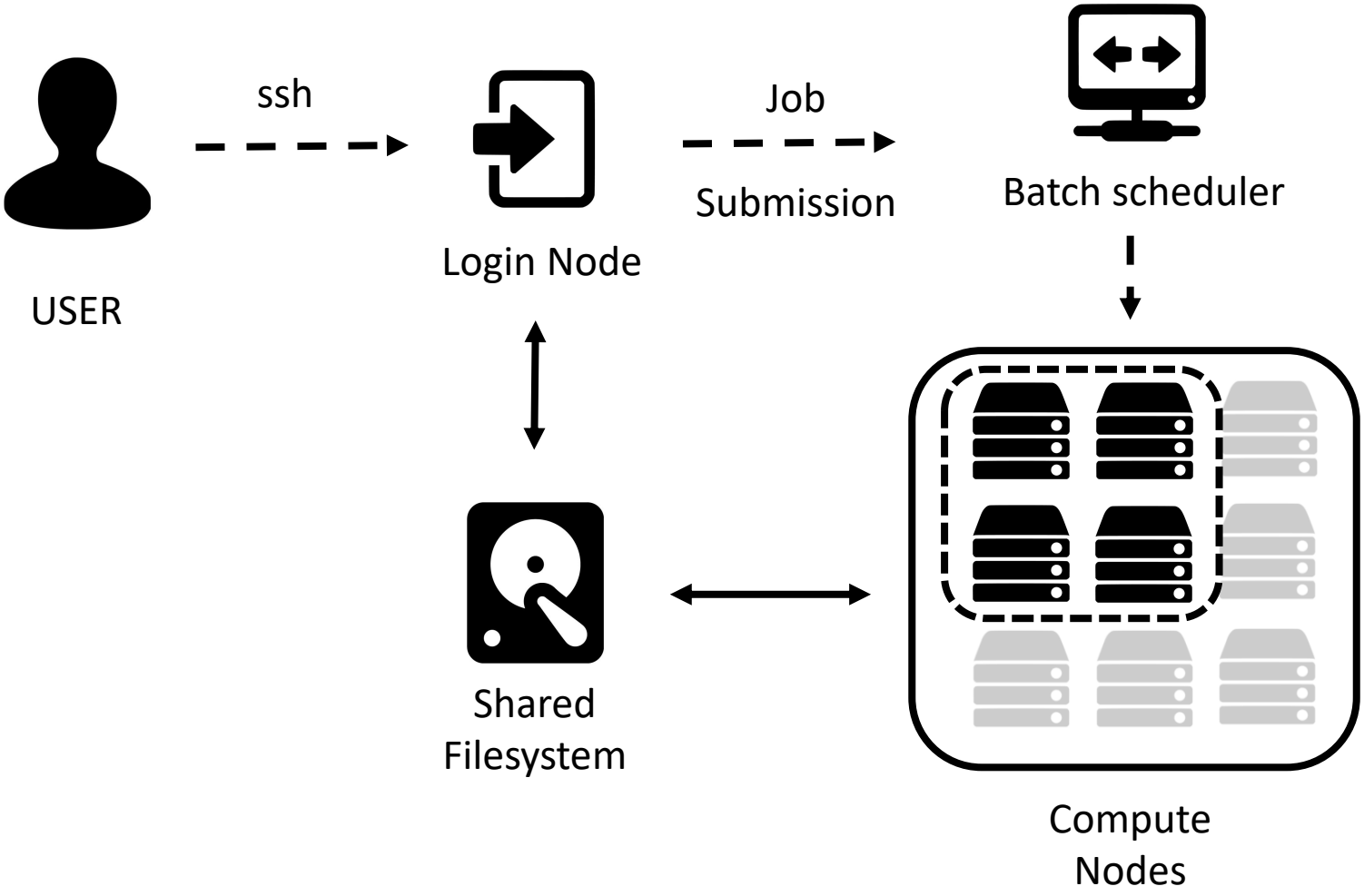


Working with a Supercomputer

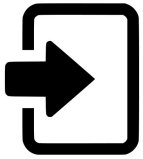
Is NOT like this…

Working with a Supercomputer

Working with a Supercomputer

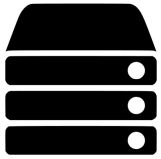


Working with a Supercomputer



Login node(s)

- Editing and transferring files
- Compile programs
- Prepare simulations



Compute nodes

- Multicore nodes
- Large memories
- High-speed interconnections

Batch scheduler

- Resource allocation
- Job queueing
- Accounting and

File system

- Parallel FS
- Efficient I/O
- Node local disks

Working with a Supercomputer

1. **Login and transfer files to the remote machine**
   - ssh, scp/ftp
   - Command line, GUI

2. **Prepare your job(s)**
   - Input preparation
   - Job submission script
   - Software preparation

3. **Submit your job and retrieve output**
   - Submit job to the batch system
   - Monitor job
   - Retrieve outputs / Remote visualization

Introduction to HPC in Computational Modelling

GETTING STARTED WITH HPC

Login to an HPC system

Getting an account on the system

- EU funded projects (PRACE, CompBioMed, etc.)
- National initiatives (NWO, RES)
- Special agreements with Universities or research centers
- Contact the HPC center
  - helpdesk@surfsara.nl
  - support@bsc.es

Login to an HPC system

Install UNIX tools on your local machine

Windows

Putty

MobaXterm (http://mobaxterm.mobatek.net)

Mac OSX

Terminal (pre-installed)

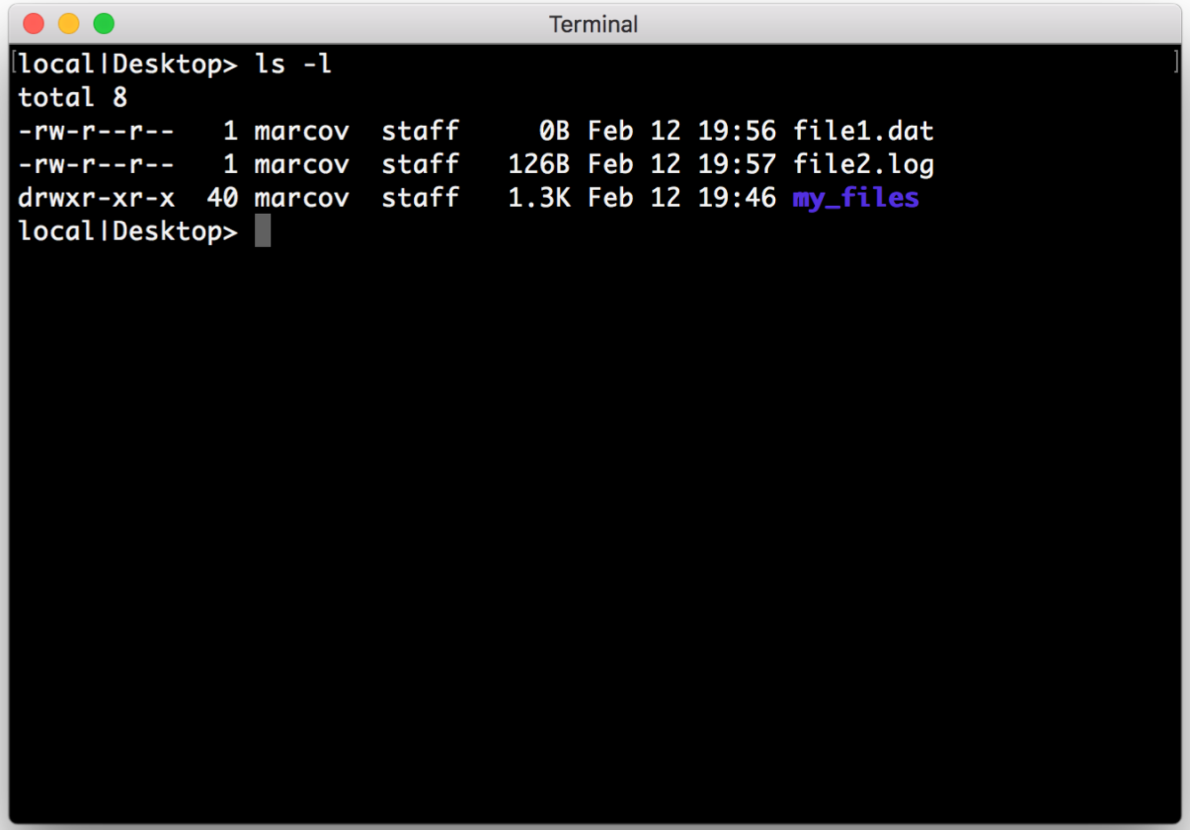
XQuartz (http://www.xquartz.org)

Linux

You are already well equipped!

Login to an HPC system

Terminal – Command line



Login to an HPC system

SSH, or Secure SHell

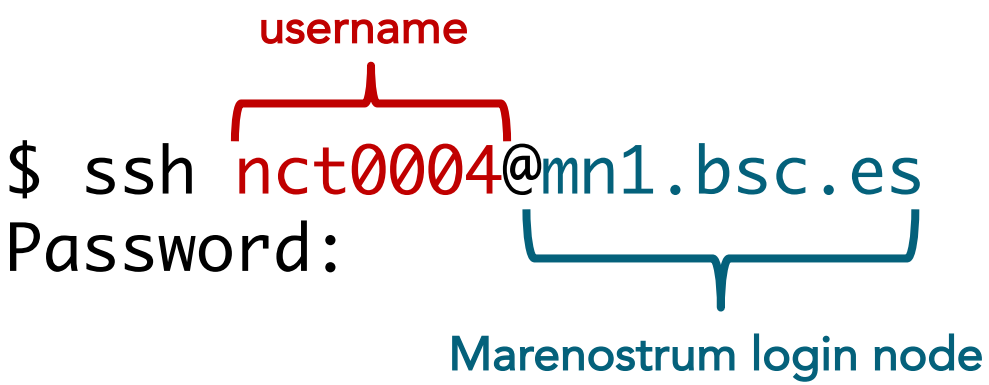
- establishing a cryptographically secured connection
- authenticating each side to the other
- passing commands and output back and forth

```
$ ssh nct0004@mn1.bsc.es
Password:
```

Login to an HPC system

SSH, or Secure SHell

- establishing a cryptographically secured connection
- authenticating each side to the other
- passing commands and output back and forth

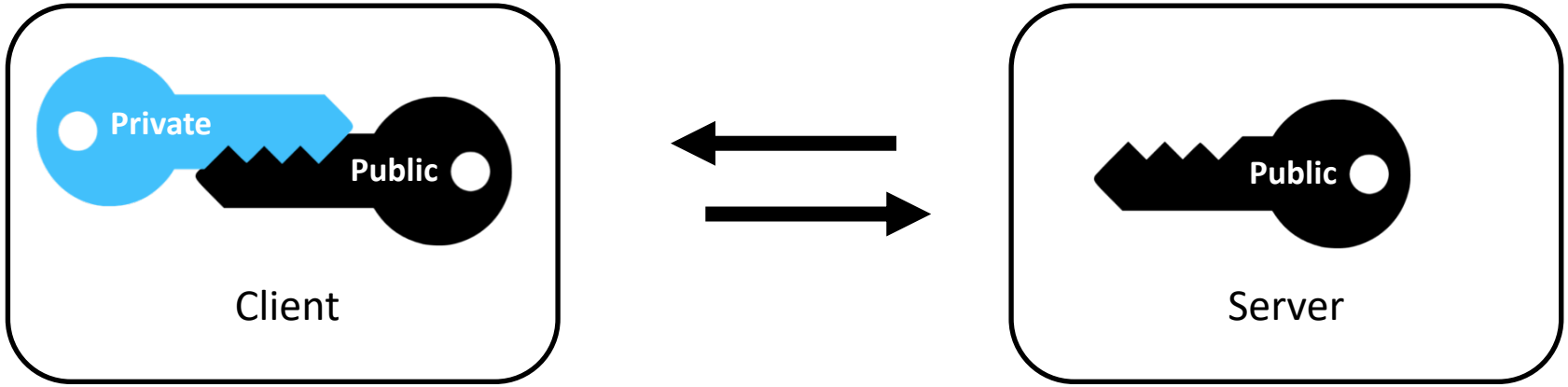


Login to an HPC system

SSH - Authentication and security

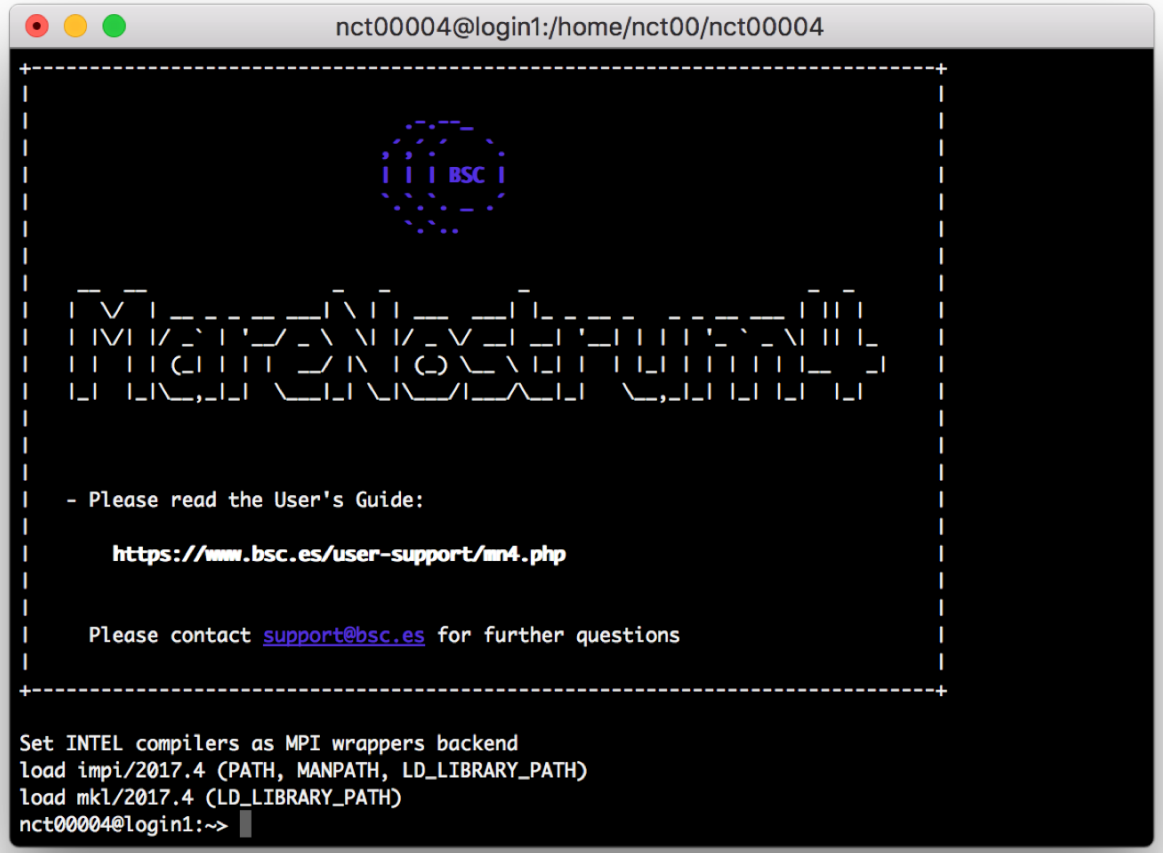
- Username/password (Symmetric encryption)

- Public and private keys (Asymmetric encryption)



Login to an HPC system

You are logged in!



Login to an HPC system

MareNostrum4

3456 nodes with a grand total of 165,888 processor cores and 390 Terabytes of main memory.

Compute nodes are equipped with:

- 2 sockets Intel Xeon Platinum 8160 CPU with 24 cores each @ 2.10GHz
- L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K
- 96 GB of main memory 1.880 GB/core
  - (216 nodes high memory, 10368 cores with 7.928 GB/core)
- 100 Gbit/s Intel Omni-Path HFI Silicon 100 Series PCI-E adapter
- 10 Gbit Ethernet
- 200 GB local SSD available as temporary storage during jobs

Introduction to HPC in Computational Modelling

GETTING STARTED WITH HPC

Linux basic commands

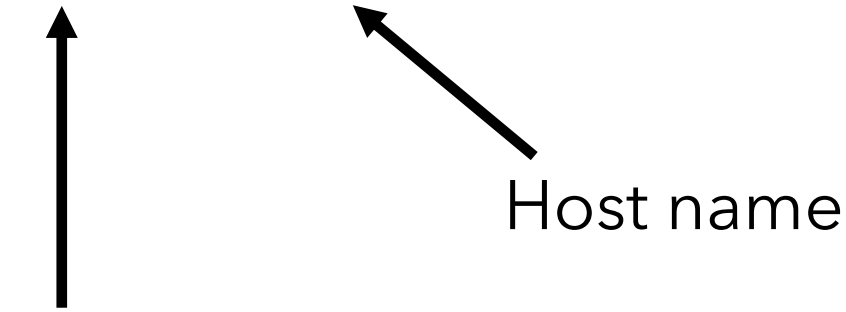
What is UNIX?

- Operating System
  - Program that controls all other parts of a computer system
  - Allocates computer's resources and schedules tasks
  - Allows the user to use the facilities provided by the system
  - Essential to all computer systems
- Multi-User and Multi-Tasking
  - Multiple users have multiple tasks running simultaneously
- Designed to be machine independent
- Setup as a software development environment
- Suitable for scientific applications

Linux basic commands

- After successful login

```
nct00004@login1:~>
```



Now the system is ready to accept commands

Linux basic commands

- First command

```
nct00004@login1:~> date
Wen Feb 14 11:00:00 CET 2018
nct00004@login1:~>
```

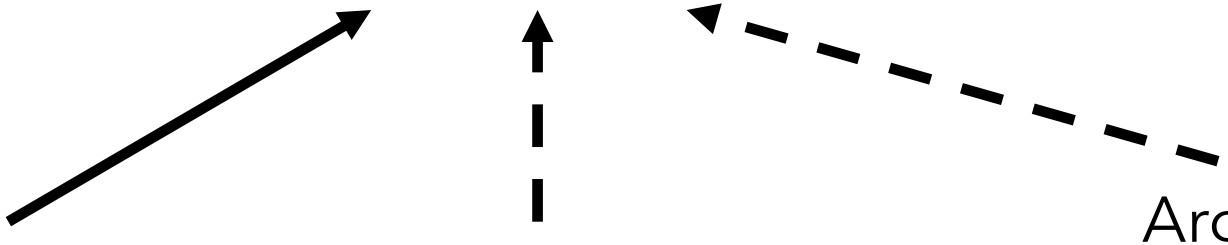
- A little bit more

```
nct00004@login1:~> uname -a
Linux login1 4.4.103-92.56-default #1 SMP Wed Dec 27 16:24:31 UTC
2017 (2fd2155) x86_64 x86_64 x86_64 GNU/Linux
nct00004@login1:~>
```

Linux basic commands

- Structure of a unix commands

```
nct00004@login1:~> mkdir –p dir1/subdir
```



Command

The UNIX shell (bash) tries to find a program called ‘mkdir’ and takes care that the system executes it.

Options

Passed to the command as a parameter(s) to change its default behavior.

Arguments

Taken as input by the program.

- Case-sensitive (everything!)
- Spaces used to separate command, options and arguments

Linux basic commands

- Where to find help?

--help flag

```
nct00004@login1:~> uname --help
```

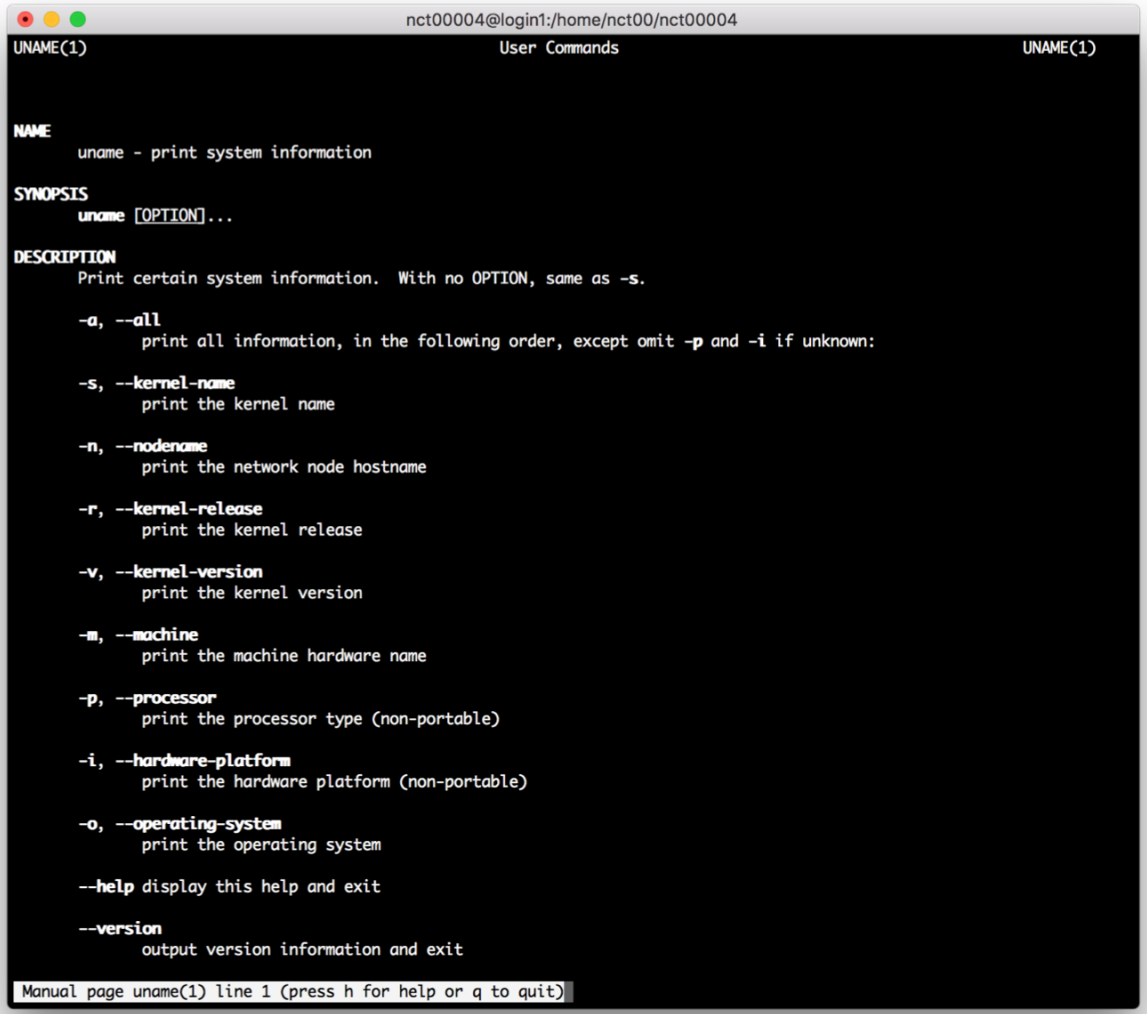
man built-in command

```
nct00004@login1:~> man uname
```

info built-in command

```
nct00004@login1:~> info uname
```

Linux basic commands



Linux basic commands – Looking around

- Where I am?

```
nct00004@login1:~> pwd
/home/nct00/nct00004
```

- What files are there?

```
nct00004@login1:~> ls
bin file1.txt file2.log
```

```
nct00004@login1:~> ls -l
total 0
drwxr-xr-x 2 nct00004 nct00 4096 May 5  2010  bin
-rw-r--r-- 1 nct00004 nct00    0 Feb 13 01:17 file1.txt
-rw-r--r-- 1 nct00004 nct00 1528 Feb 13 01:17 file2.log
```

Linux basic commands – Moving around

- Changing directory

```
nct00004@login1:~> cd dir/subdir
nct00004@login1:~/dir/subdir>
```

- Going “on level” up

```
nct00004@login1:~/dir/subdir> cd ..
nct00004@login1:~/dir>
```

- Going back to home folder (~)

```
nct00004@login1:~/dir/subdir> cd
nct00004@login1:~>
```

Linux basic commands – Moving around

Path to folders and files

- Relative path

```
nct00004@login1:~> ls dir/mydata.out
dir/mydata.out
```

- Absolute path

```
nct00004@login1:~> ls ~/dir/mydata.out
/home/nct00/nct00004/dir/mydata.out
```

Linux basic commands – Files management

- Directories

Create directories

```
nct00004@login1:~> mkdir dir
```

Remove directories (only if empty)

```
nct00004@login1:~> rmdir dir
```

Remove directory and its content (be careful with rm)

```
nct00004@login1:~> rm –r dir
```

Linux basic commands – Files management

- Files

Copy files

```
nct00004@login1:~> cp file1.txt file1.copy
```

Copy files to directories

```
nct00004@login1:~> cp file1.* dir/
```

Move files

```
nct00004@login1:~> mv file1.txt dir/file.txt.new
```

Linux basic commands – Files management

- Files (remote)

Copy files from your local machine to a remote host

```
~> scp file*.txt nct00004@mn1.bsc.es:/home/nct00/nct00004/
```

Copy files from remote host to local machine

```
~> scp nct00004@mn1.bsc.es:/home/nct00/nct00004/dir/mydata.out .
```

Copy folder from your local machine to a remote host

```
~> scp –r dir nct00004@mn1.bsc.es:
```

Linux basic commands

File editing/viewing

- touch: Creates a blank file with a specified name.
- less: View contents of specified file, page by page.
- cat: Display contents of a file.
- head/tail: Displays the first/last 10 lines of a file.

System tools

- history: Display a listing of the last commands you've run.
- find: Search files and directories.
- tar: Compress and extract files.
- top: Display processes running on the system.

...

Bash scripting

What if I want to run many bash commands?

…maybe in a workflow?

- **Bash scripts**

A Bash script is a plain text file which contains a series of commands.

Any command you can run on the command line can be put into a script (v.v.)

It will be executed like a normal program: `./script.sh`

Bash scripting

```
nct00004@login1:~> ls -l
total 0
drwxr-xr-x 2 nct00004 nct00 4096 May 5  2010  bin
-rw-r--r-- 1 nct00004 nct00    0 Feb 13 01:17 file1.txt
```

File permission

Every file/directory has 9 permission bits associated (+ initial to diff):

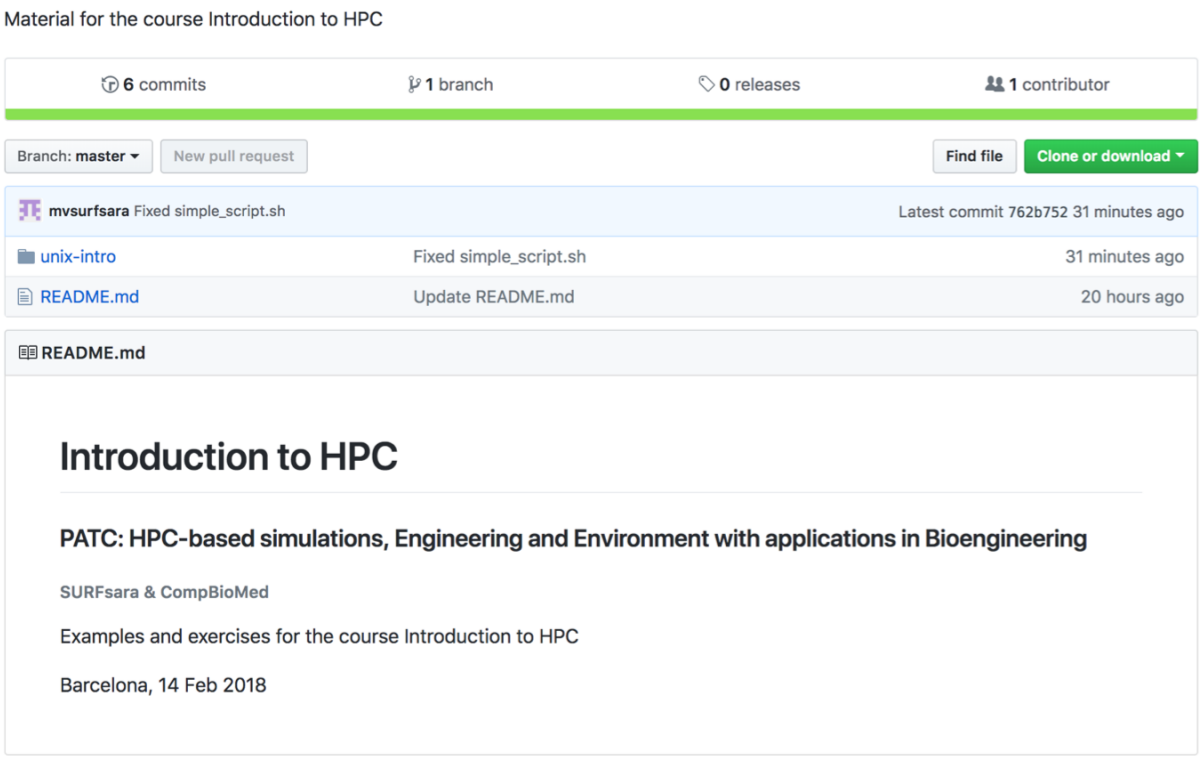
- 3 user based permission groups: owner(2-4), group(5-7), all users(8-10)
- x3 permission types: read(r), write(w), execute(x)

Permission can be changed with the command `chmod`

```
nct00004@login1:~> chmod u+x simple.sh
```

Training material

The material for this course is available at :
http://bit.ly/PATC_HPC



Training material

The material for this course is available at:
http://bit.ly/PATC_HPC

Transfer the zip file to MarenostrumIV

```
~> scp IntroHPC_PATC_CBM-master.zip nct00004@mn1.bsc.es:
```

(Windows users can use MobaXterm or Winscp)

Extract the zip in your home:

```
nct00004@login1:~> unzip IntroHPC_PATC_CBM-master.zip
```

Bash scripting

Simple bash script

unix-intro/simple.sh

```
#!/bin/bash

echo "Hi, I'm your first script."
echo


lscpu --help > cpu.log
lscpu >> cpu.log


echo "I've left something for you."
echo "Ciao"
```

Bash scripting

Standard input, output and error

Every program (bc, shell, ...) has three predefined input/output associated:

- Standard input (stdin): normally your keyboard
- Standard output (stdout): normally your screen
- Standard error (stderr): normally your screen

These can be redirected to a file or to another command.

```
nct00004@login1:~> echo "3/0" | bc > calc.log
```

Bash scripting - Ex

Simple bash script

unix-intro/simple.sh

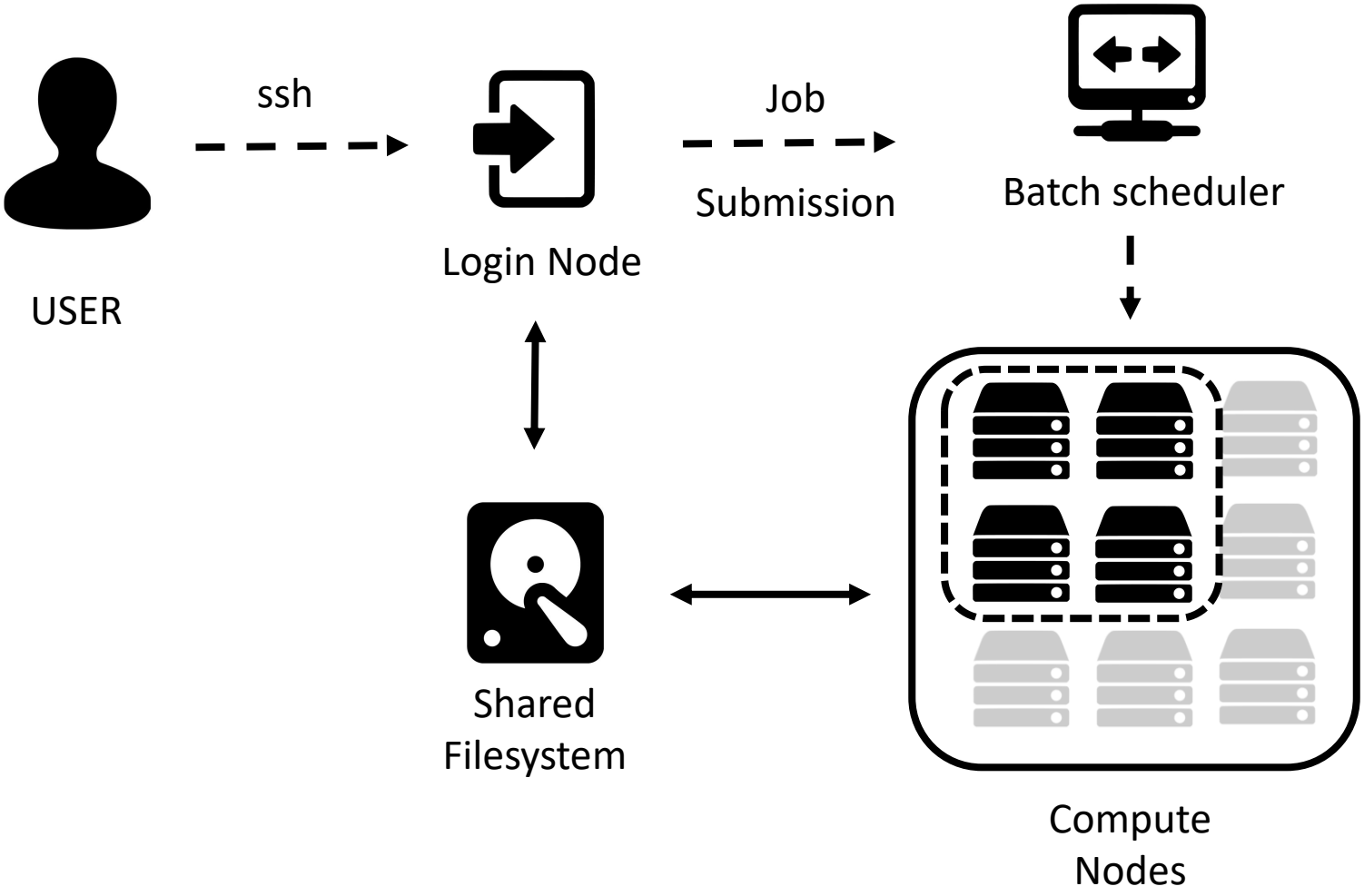
More complicated bash script

unix-intro/advanced.sh

Introduction to HPC

RUNNING JOBS

Batch system



Batch system

Supercomputers use batch systems to distribute computational tasks over the available nodes.

Instead of executing commands interactively, you prepare a ***job script***

- Script containing the commands to execute
- Resource characteristics (specific)

The batch system is responsible for allocating cores, processors or nodes to a job.

Batch system

Advantages of a batch system are:

- It allows to run MANY jobs at the same time.

The system takes care that they are run efficiently on the available resources.

- Multiusers, queue system.

A batch system allows users to always submit jobs, even if a lot of people are using the system at the same time. In addition take care of budgeting and fair resource usage.

- System load balance.

The system takes care of balancing the load across nodes and during time. In a batch system, most jobs may be submitted during office hours, but the scheduler will continue to start jobs at night as nodes become available.

Batch system

- **Queues**

There are several queues present in the machines and different users may access different queues.

```
nct00004@login1:~>bsc_queues

     QUEUE NAME   MAX TIME MAX  PROC
         debug   02:00:00       768
   interactive   02:00:00         4
      training 2-00:00:00       768
```

Batch system

- **Submitting jobs with SLURM**

The method for submitting jobs is to use the SLURM **sbatch** directives directly.

- submits a “job script” to the queue system:

```
nct00004@login1:~> sbatch <job script>
```

- shows all the submitted jobs ant their status:

```
nct00004@login1:~> squeue
```

- remove the job from the queue system:

```
nct00004@login1:~> scancel <jobid>
```

- run a command on the cluster

```
nct00004@login1:~> srun <command>
```

Batch system

- **SLURM Job directives**

A job script must contain directives to inform the batch system about the characteristics of the job. This directives appear as comments (#SBATCH) in the job script and have to conform with the sbatch syntax.

`#SBATCH --nodes=<num>`	request for <num> compute node
`#SBATCH --ntasks=<num>`	the number of processes to start
`#SBATCH --time=DD-HH:MM:SS`	total wall clock time of the job
`#SBATCH --qos=<queue>`	requesting a specific queue
`#SBATCH --task=<queue>`	requesting a specific queue
`#SBATCH --output=<file>`	name of the file where std out is printed
`#SBATCH --tasks-per-node=<num>`	
`#SBATCH --cpus-per-task=<num>`	

Batch system - Ex1

SLURM sequential

batch-sys/slurm1.sub

SLURM multinode

batch-sys/slurm2.sub

Software stack

Some software packages require certain settings in your user environment, like paths and environment variables.

Environment Modules

- Provide lots of useful software packges
- In many different versions
- Maintained by experts
- Optimized for the architecture

- Programming and scripting languages: Python, R, Matlab
- Compilers for C, C++, Fortran
- Specialized libraries (MKL, OpenMPI)
- Scientific codes

Software stack

Useful module commands

`module avail`	list modules
`module avail python`	list all installed versions of python
`module load python`	load the default python version
`module load python/2.7.13`	load a specific version of python
`module unload python`	unload python
`module list`	list currently loaded modules

MarenostrumIV uses **Lmod**, a Lua-based module system (TACC)

Software stack

Let's try it…

- List the default loaded modules

```
nct00004@login1:~> module list

Currently Loaded Modules:
 1) intel/2017.4 2) impi/2017.4 3) mkl/2017.4 4) bsc/1.0
```

- Which python I am using?

```
nct00004@login1:~> which python
/usr/bin/python
nct00004@login1:~> python --version
Python 2.7.13
```

Software stack

- Check which python is installed

```
nct00004@login1:~> module av python

---------------- /apps/modules/modulefiles/tools -----------------

  ANACONDA/5.0.1_python2   python/2.7.13       python/3.6.3_ML
  python/2.7.13_ML         python/3.6.1 (D)


 Where:
  D: Default Module


Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible
modules matching any of the
"keys".
```

Software stack

- List the default loaded modules

```
nct00004@login1:~> module load python
load python/3.6.1 (PATH, MANPATH, LD_LIBRARY_PATH, C_INCLUDE_PATH,
CPLUS_INCLUDE_PATH, PYTHONHOME)
```

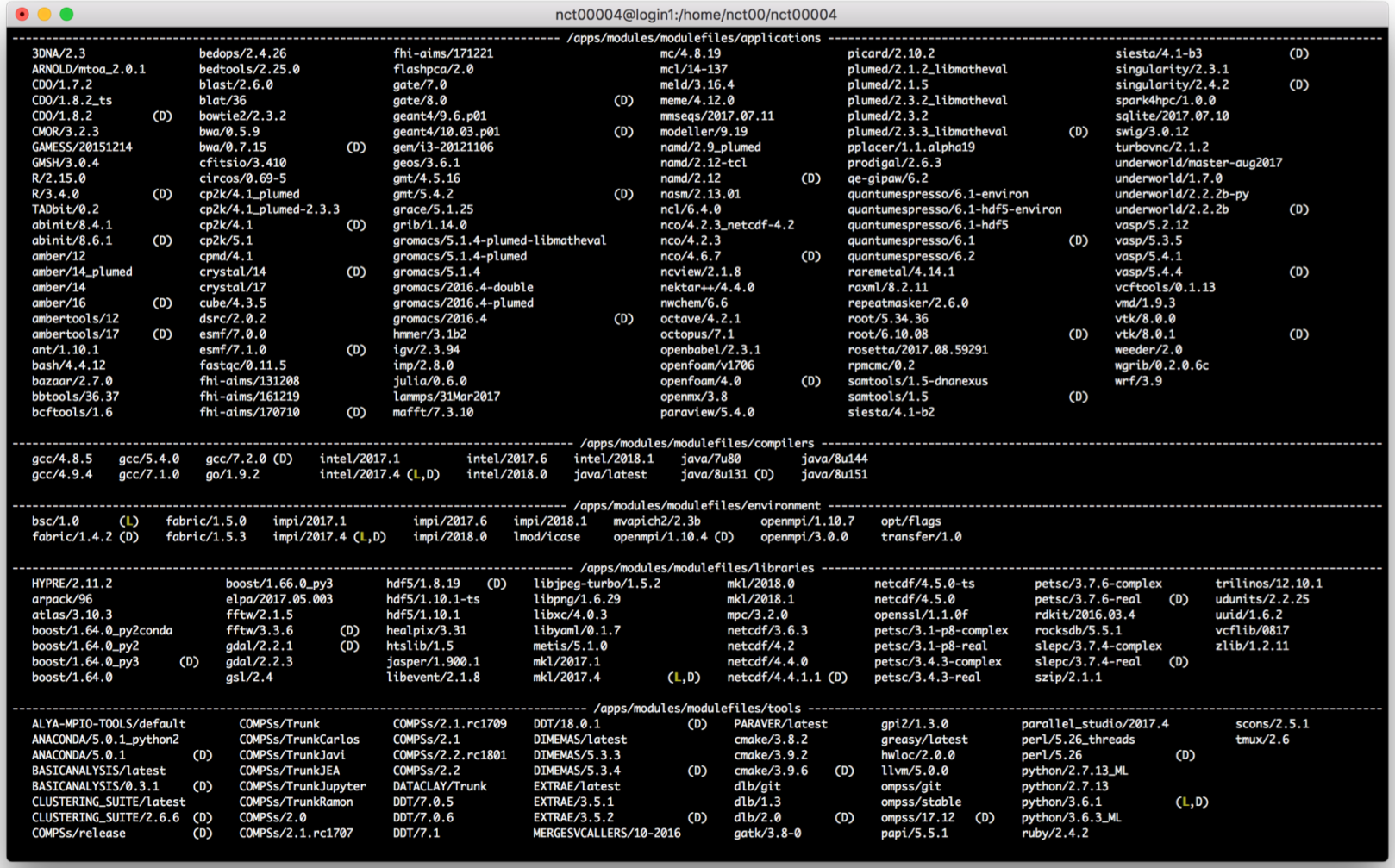
- Check is working

```
nct00004@login1:~> which python
/apps/PYTHON/3.6.1/INTEL/bin/python
nct00004@login1:~> python --version
Python 3.6.1
```

Have a look at all the software installed on MarenostrumIV with:

`module avail`

Software stack



File systems

Each user has several areas of disk space for storing files.

These areas may have size or time limits (`bsc_quota`).

Choose carefully where to store your data!

MarenostrumIV

- Root filesystem: Is the filesystem where the operating system resides.
- GPFS filesystems: GPFS is a distributed networked filesystem which can be accessed from all the nodes and Data Transfer Machine.
- Local hard drive: Every node has an internal hard drive ($TMPDIR).

File systems

The IBM General Parallel File System (GPFS) is a high-performance shared-disk file system providing fast, reliable data access from all nodes of the cluster to a global filesystem.

- **/apps**: Over this filesystem will reside the applications and libraries that have already been installed on the machine.

- **/gpfs/home**: This filesystem has the home directories of all the users, and when you log in you start in your home directory by default. It is highly discouraged to run jobs from this filesystem.

- **/gpfs/projects**: In addition to the home directory, there is a directory in /gpfs/projects for each group of users. This space is intended to store data that needs to be shared between the users of the same group or project.

- **/gpfs/scratch**: Each user will have a directory over /gpfs/scratch. Its intended use is to store temporary files of your jobs during their execution.

Batch system - Ex2

Running python on MarenostrumIV

run-jobs/python_seq.sub

run-jobs/pairwise.py

- Using a system installed software (modules)
- Make appropriate usage of disk space (scratch)

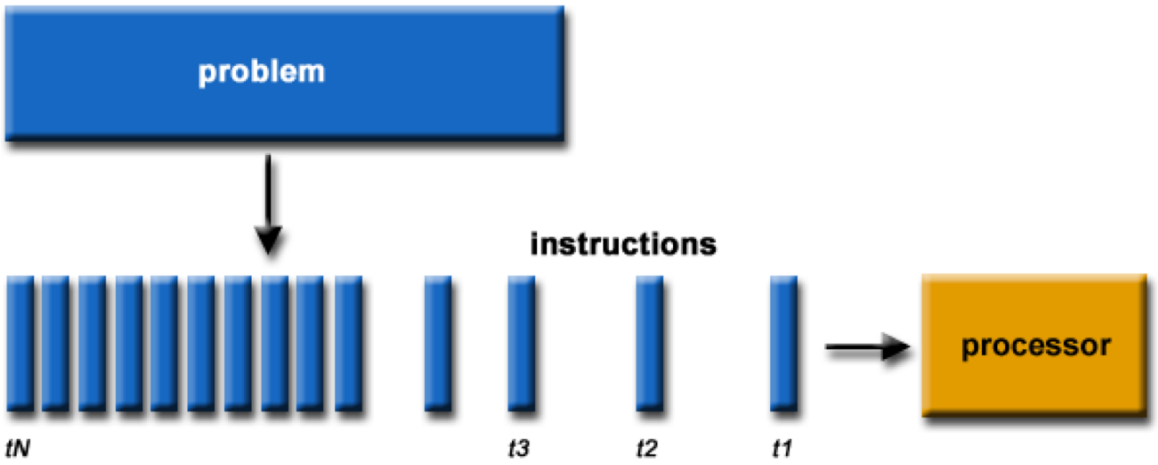
Introduction to HPC in Computational Modelling

RUNNING JOBS

Parallel jobs

Serial computing

A problem is broken into a discrete series of instructions, which are executed sequentially on a single processor

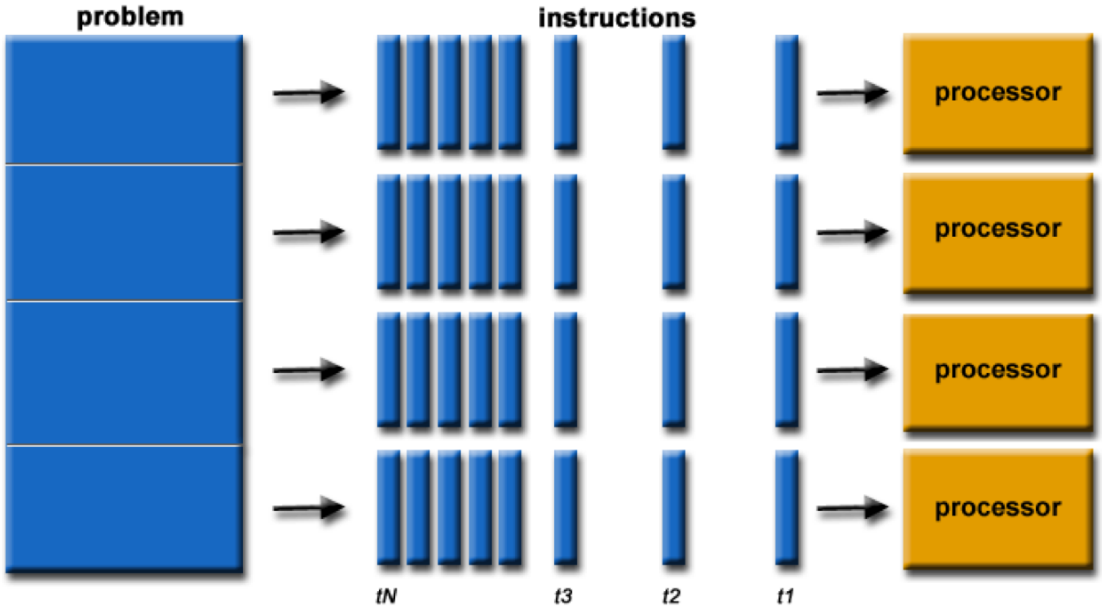


credits: https://computing.llnl.gov/tutorials/parallel_com

Parallel jobs

Parallel computing

A problem is broken into discrete parts that can be solved concurrently using simultaneously multiple resources



credits: https://computing.llnl.gov/tutorials/parallel_com

Parallel jobs - Ex

Running python in parallel on MarenostrumIV

batch-sys/python_par.sub

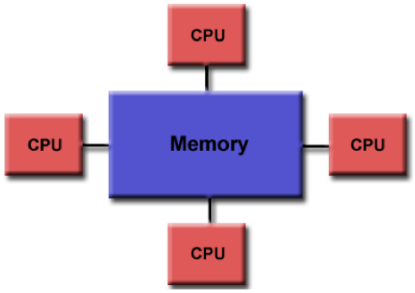
batch-sys/pairwise.py

Exploit background processes in UNIX

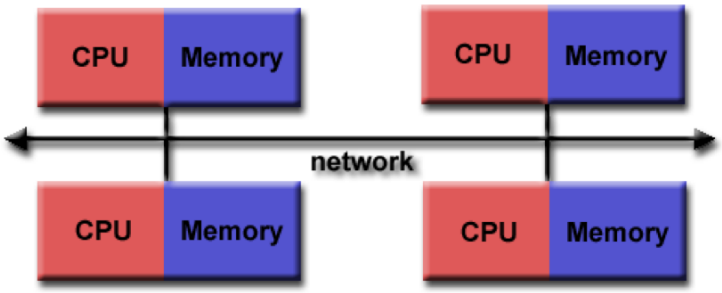
- `&` : at the end of a command put the process in bg and let you continue to work
- `wait` : wait that all the background processes have ended before continue

Parallel jobs

- Task parallel
  - many independent runs
  - needs orchestration
  - for monte-carlo, parameter sweeps
- Shared memory
  - always within one batch node
  - uses threads
  - often implicit
- Distributed memory
  - can use one or more batch nodes
  - uses separate processes
  - almost always using MPI
  - for PDE problems, time stepping



Shared memory topology

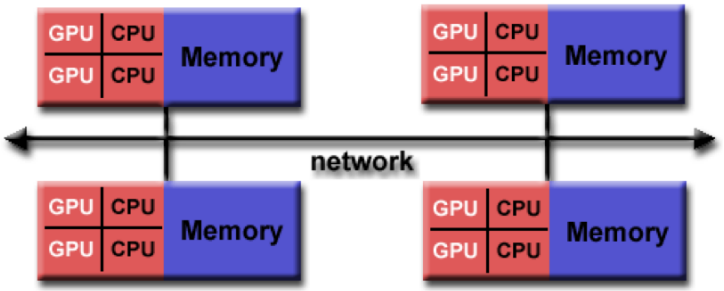


Distributed memory topology

credits: https://computing.llnl.gov/tutorials/parallel_com

Parallel jobs

- Task parallel
  - many independent runs
  - needs orchestration
  - for monte-carlo, parameter sweeps
- Shared memory
  - always within one batch node
  - uses threads
  - often implicit
- Distributed memory
  - can use one or more batch nodes
  - uses separate processes
  - almost always using MPI
  - for PDE problems, time stepping

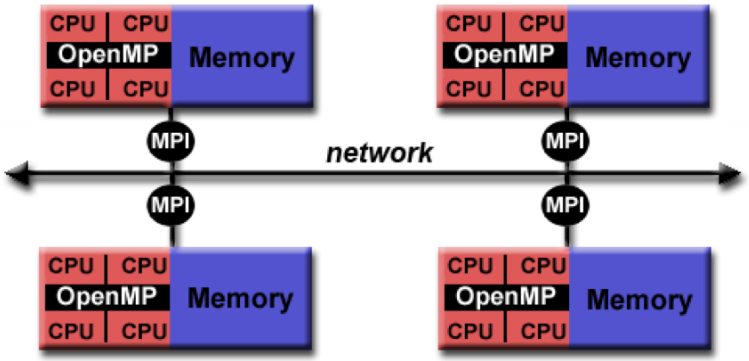


Hybrid topology

credits: https://computing.llnl.gov/tutorials/parallel_com

Parallel jobs

- Task parallel
  - many independent runs
  - needs orchestration
  - for monte-carlo, parameter sweeps
- Shared memory
  - always within one batch node
  - uses threads
  - often implicit
- Distributed memory
  - can use one or more batch nodes
  - uses separate processes
  - almost always using MPI
  - for PDE problems, time stepping



Hybrid topology

credits: https://computing.llnl.gov/tutorials/parallel_com

Parallel jobs - Ex

Running an MPI application

mpi-ex/run_wave

mpi-ex/Makefile

Uses MPI library to distribute work across multiple node.

The code needs to be compiled and right libraries loaded.

It generates a gif that we need to copy back at the end.

Thank you to my colleagues ...

Caspar van Leeuwen

Lykle Voort

Zheng Meyer-Zhao

Jeroen Engelberts

Ander Astudillo

Natalie Danezi

Damian Podareanu

Valeriu Codreanu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675451.