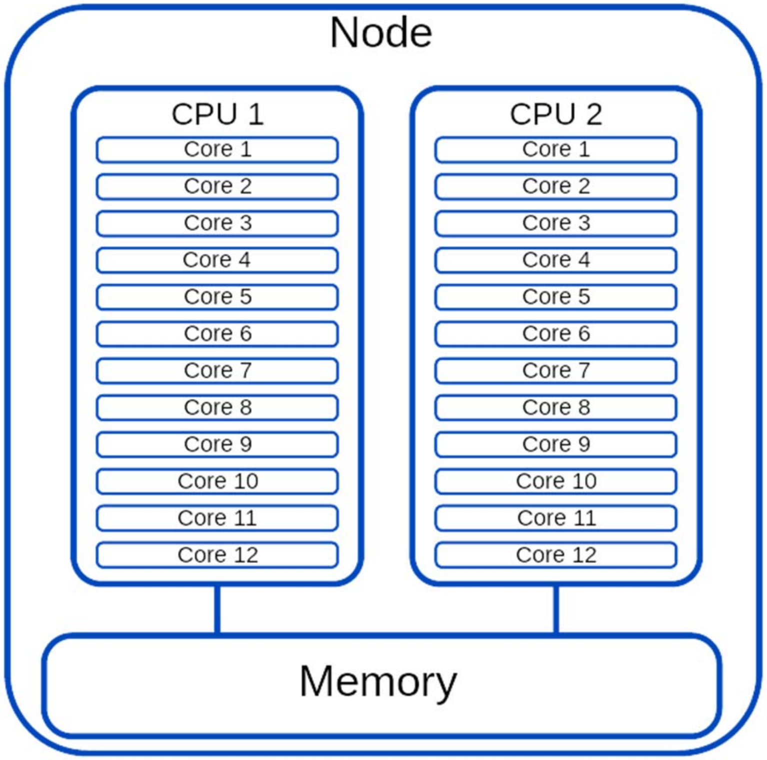
جامعــة خليفــة
Khalifa University

# Research Computing Introductory Session

Dr. Sergio Martinez

# What is the HPC cluster? - Some concepts

- **Node (Host):** Refers to the physical machine/server. In current systems, a node would typically include one or more processors, as well as memory and other hardware.
- **Processor:** Refers to the central processing unit (CPU), which contains one or more  cores.
- **Core:** Refers to the basic computation unit of the CPU. This is unit that carries out the actual computations.



# What is the HPC cluster? – Almesbar specifications

- **Number of nodes**
- Cores and memory
- Network and storage
- Performance

| Number of nodes | |
|---|---|
| 88 CPU nodes | 2x Intel Xeon Gold 6230R 26-Core 2.1GHz (Cascade lake) (Q1‘20) |
| 3 GPU nodes | 2x Intel Xeon Gold 6230R 26-Core 2.1GHz(Cascade lake) (Q1’20) |
| | 4x NVidia V100 |

# What is the HPC cluster? – Almesbar specifications

- Number of nodes
- **Cores and memory**
- Network and storage
- Performance

| Cores per node | |
|---|---|
| 52 | |
| **Memory per node** | |
| CPU nodes | 384 GB |
| GPU nodes | 640 GB + 128 GB (GPU) |

| Total CPU cores |
|---|
| 4,732 |
| **Total GPU cores** |
| 61,440 |
| **Total memory** |
| 35.25 TB |

# What is the HPC cluster? – Almesbar specifications

- Number of nodes
- Cores and memory
- **Network and storage**
- Performance

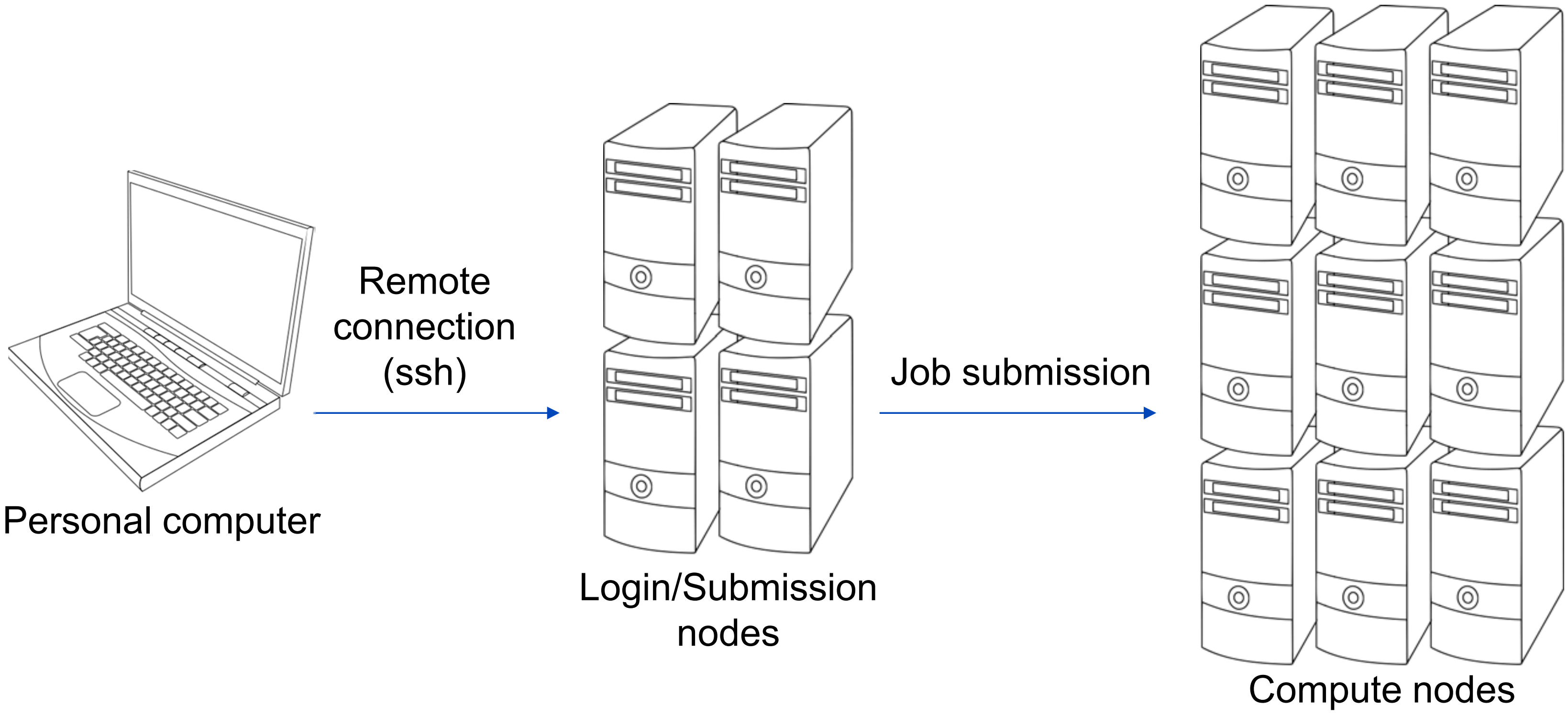
<table>
<tr><th colspan="2">Network interconnection</th></tr>
<tr><td colspan="2">Infiniband EDR (100Gb/s) and Infiniband HDR (200Gb/s)</td></tr>
<tr><th colspan="2">Storage</th></tr>
<tr><td>300 TB home</td><td>NFSv4 (10Gb/s)</td></tr>
<tr><td>947 TB scratch</td><td>Parallel file system (Lustre) (200Gb/s)</td></tr>
</table>

# What is the HPC cluster? – Almesbar specifications

- Number of nodes
- Cores and memory
- Network and storage
- **Performance**

| Theoretical peak of performance (CPU) |
|---|
| 307.5 TFOPS |
| **Achieved peak of performance (CPU)** |
| 205.0 TFOPS |

# What is the HPC cluster? - Layout



Personal computer

Remote connection (ssh)

Login/Submission nodes

Job submission

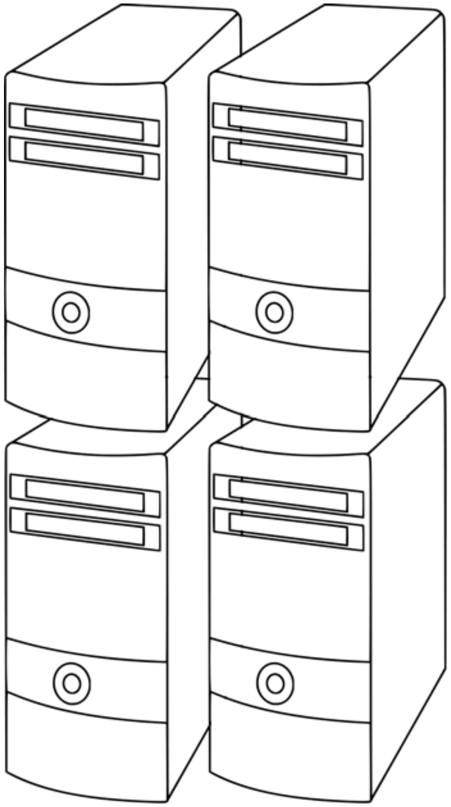
Compute nodes

# What is the HPC cluster? - Personal computer

- Develop testing code (IDE). ✓
- Prepare simulations. ✓
- Remote connection with the cluster. ✓
- File transfer with the cluster. ✓

# What is the HPC cluster? - Submission nodes

- Organize files (create, delete, move, copy).✓
- Create scripts. ✓
- Compile small programs.✓
- Run applications.✗
- Connect with compute nodes.✗
- Submit jobs.✓



# What is the HPC cluster? - Compute nodes

- Run applications directly. ✗
- Connect from other node. ✗
- Fully controlled by the scheduler ✓

# What is the HPC cluster? - Shared facility

- Although the number of available resources on the cluster is quite high, the number of active users is also high.
- There is no limitation on the number of resources that one user can request. We cannot guarantee that the requested resources will be available at the moment of submitting a job.
- Jobs will remain pending in a  queue until all the requested resources are available.
- Those resources are managed by the scheduler.
- Once the resources are available, jobs will start running on those resources.

# What is the HPC cluster? - Jobs

- Job is the unit of execution on the HPC cluster.
- Instead of running applications directly on a submission or on a compute node, in order to make an efficient use of the resources, we encapsulate our application in a job and we submit it through the scheduler.
- The syntax of a job allows us, among other things, to specify with accuracy what are the resources needed for our program to run and for how long we need those resources.

```
run python script
```

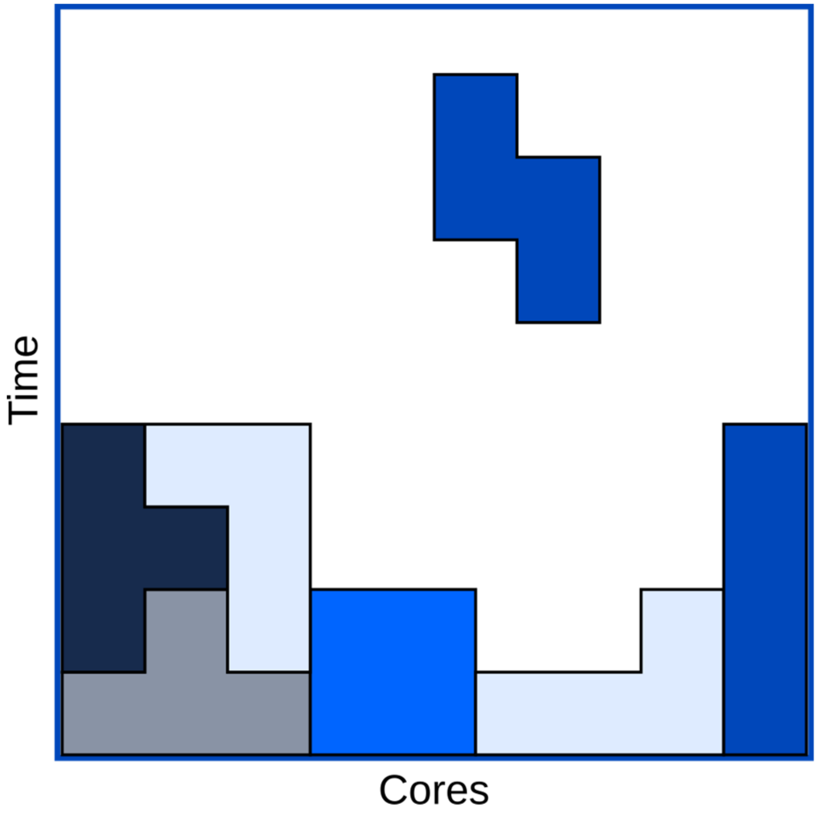
✗

```
Job start
       Cores = 2
       Nodes = 1
       Queue = gpu
       Walltime = 03:00
       run python script
Job end
```

✓

# What is the HPC cluster? - Scheduler

- The scheduler is a program that understands the job syntax and allocates the resources accordingly.
- The scheduler is configured in a way that enhances both the system and the users performance.
- It can manage multiple queues where jobs are submitted to.



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- **Batch and Interactive jobs.**
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- Backfill
- Fairshare (dynamic priority)

- **Batch jobs:** run in the background. The don’t require user interaction.
- **Interactive jobs:** allow input and output using command line interface or graphical user interface.

Interactive jobs are not recommended on a HPC cluster for several reasons:

- User might have to wait several hours before the application’s GUI opens.
- Graphics need to travel over the network, causing a bad user experience (clunky interface)

# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- **Array jobs**
- Exclusive mode
- Default wall time
- Maximum wall time
- Backfill
- Fairshare (dynamic priority)

- **Array syntax:** is an efficient way of submitting multiple jobs simultaneously
- Avoids “looping” and unnecessarily submitting a large amount of jobs with different JOBIDs.
- Ensemble solutions
- Parameter sweep

# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- **Exclusive mode**
- Default wall time
- Maximum wall time
- Backfill
- Fairshare (dynamic priority)

- **Exclusive mode:** guarantees that the nodes allocated for the jobs to run will not be shared by other jobs.
- In other words, the scheduler cannot place an exclusive job unless there is a host that has no jobs running on it.
- Useful for memory consuming applications, regardless the number of cores needed.

# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- **Default wall time**
- Maximum wall time
- Backfill
- Fairshare (dynamic priority)

In order to enhance the system performance, the scheduler needs to know the exact wall time of every single job in the system. This have a strong impact on dispatch planification, enabling features like backfill.

- Default wall time: if the user doesn’t specify the wall time of a job at submission time, the scheduler will assume a default one for that job.
- The more accurate an user is specifying the wall time of their jobs, the lesser their jobs will be pending in the queue.

# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- **Maximum wall time**
- Backfill
- Fairshare (dynamic priority)

In order to enhance the system performance, the scheduler needs to know the exact wall time of every single job in the system. This have a strong impact on dispatch planification, enabling features like backfill.

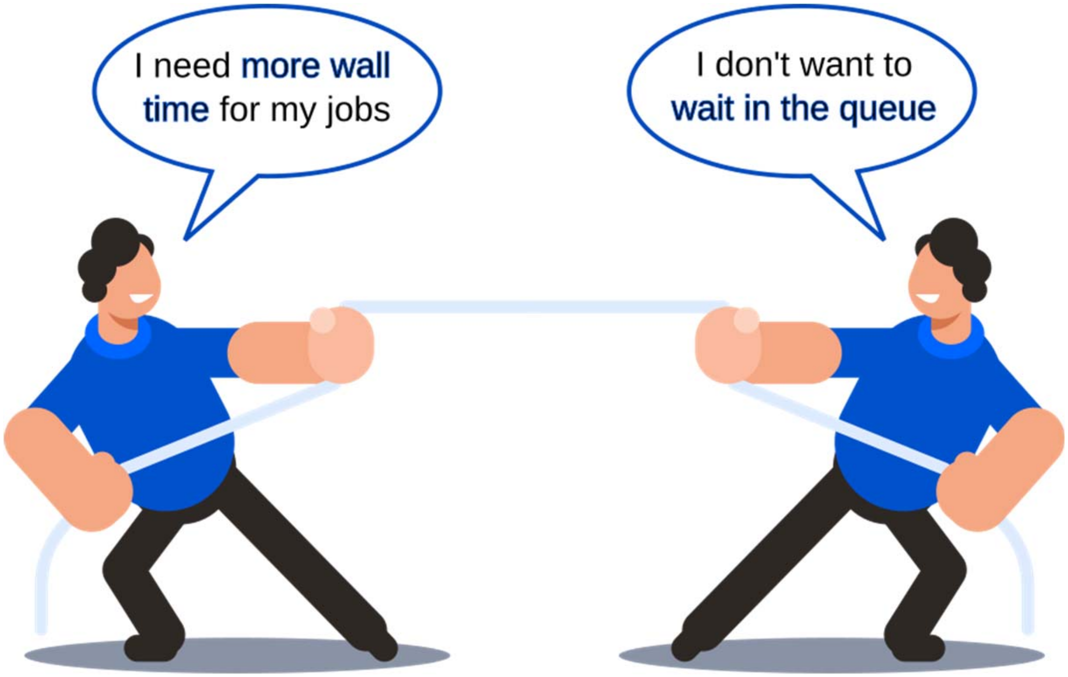
There is a limitation on the maximum wall time that a job can be running. If this limitation wasn't there:

- Overall pending time would increase notably.
- Non checkpointed jobs would be harmed if an error happens after running for long time.

# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- **Maximum wall time**
- Backfill
- Fairshare (dynamic priority)


I need **more wall time** for my jobs
I don't want to **wait in the queue**

# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)

The scheduler knows the wall time for all the jobs submitted to a queue.

Large jobs that cannot start running because lack of resources are scheduled for a specific time in the future and they wait in the queue.

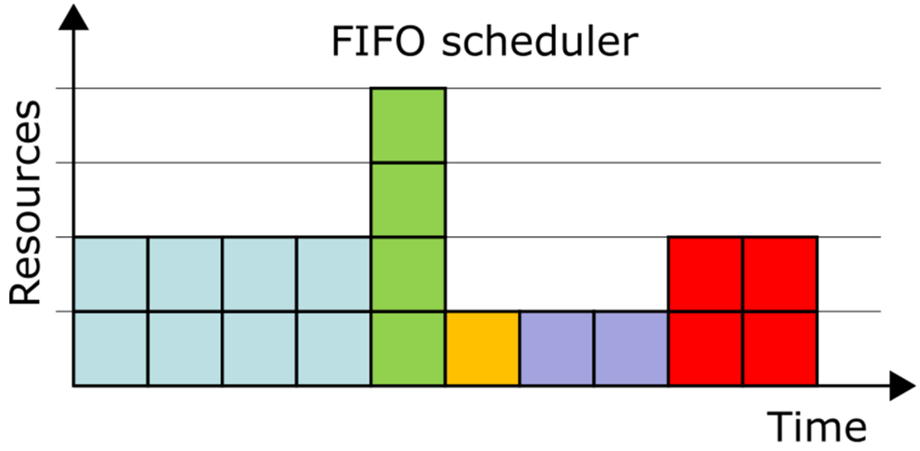
At the same time, as job finish their execution, resources become available. Those resources became an opportunity for small jobs to start running.

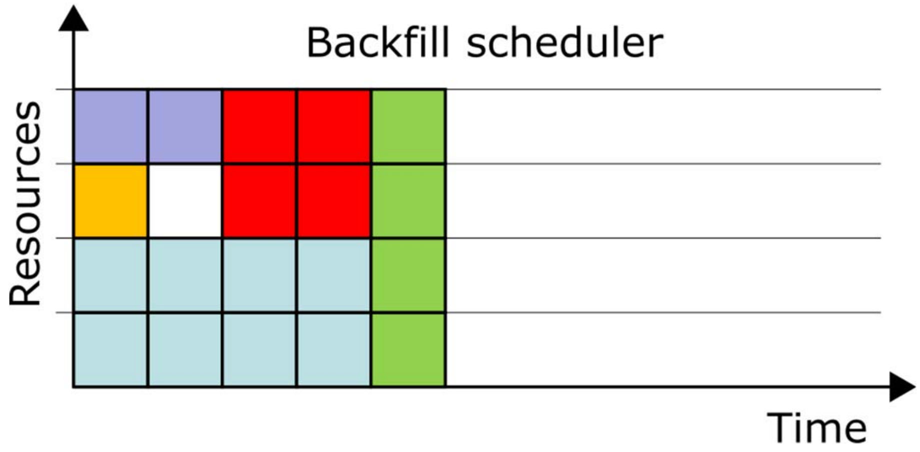
Backfill allows small jobs to jump in the queue and use those resources as long as they finish before large jobs start.

# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)

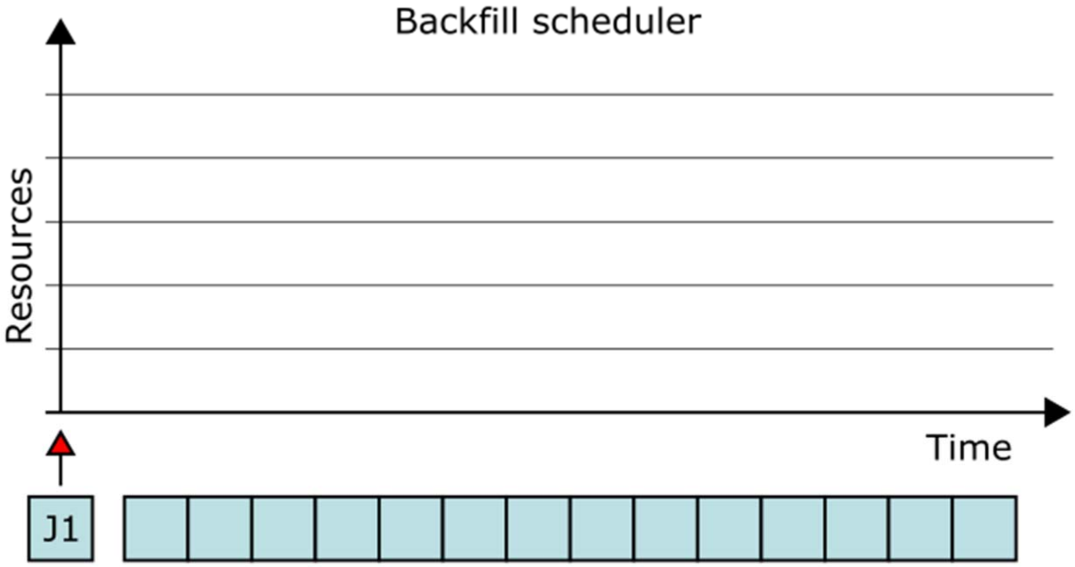




# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

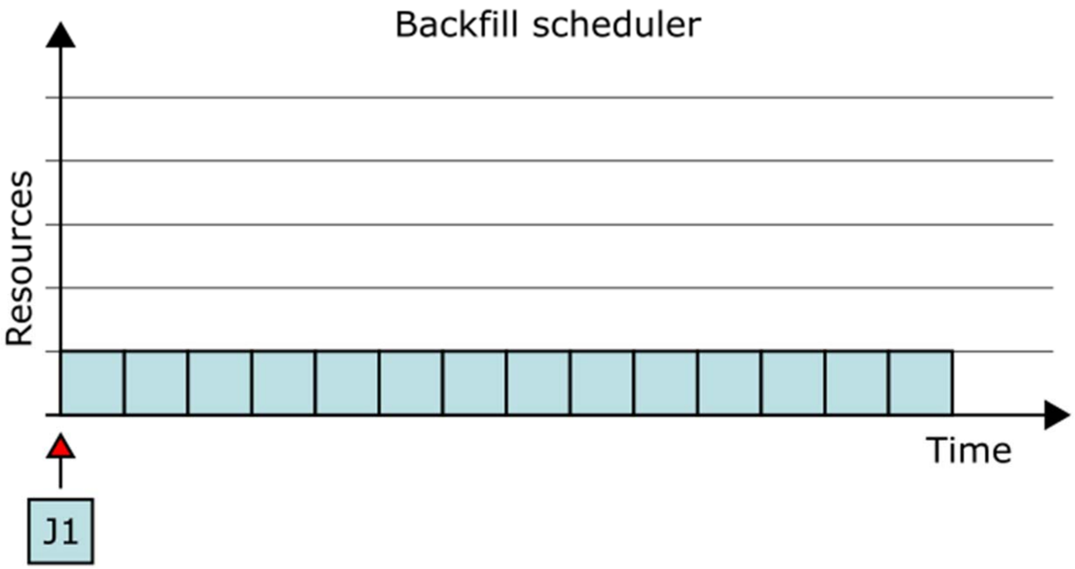
- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

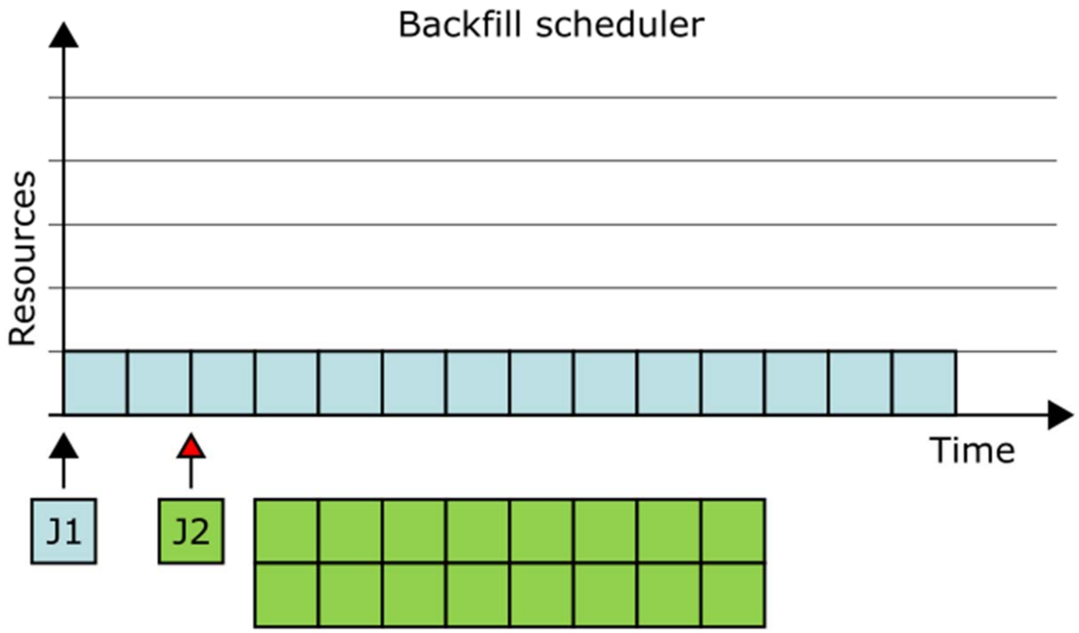
- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

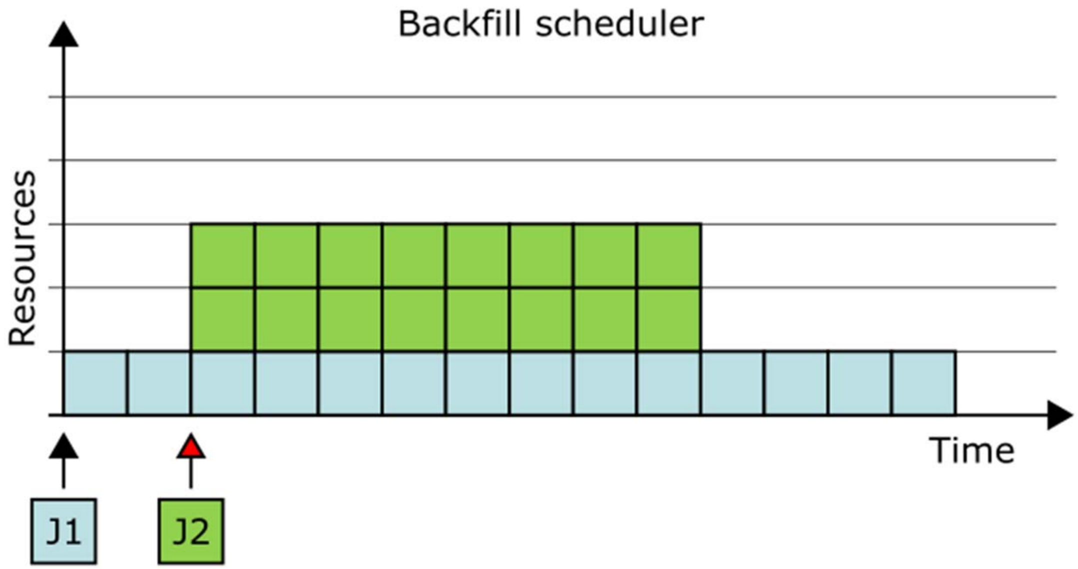
- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

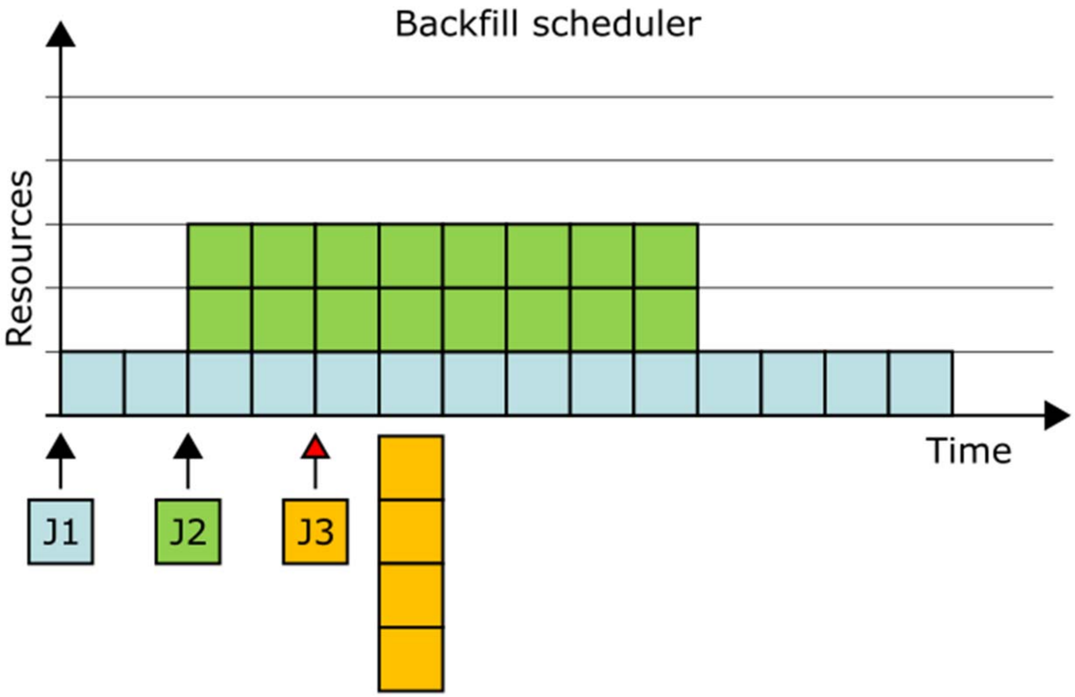
- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

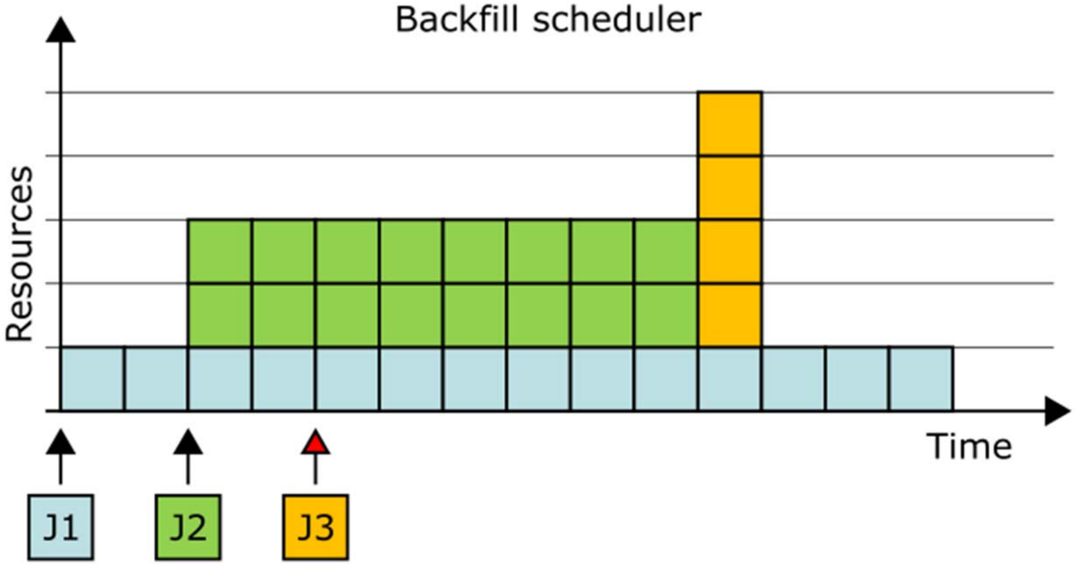
- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

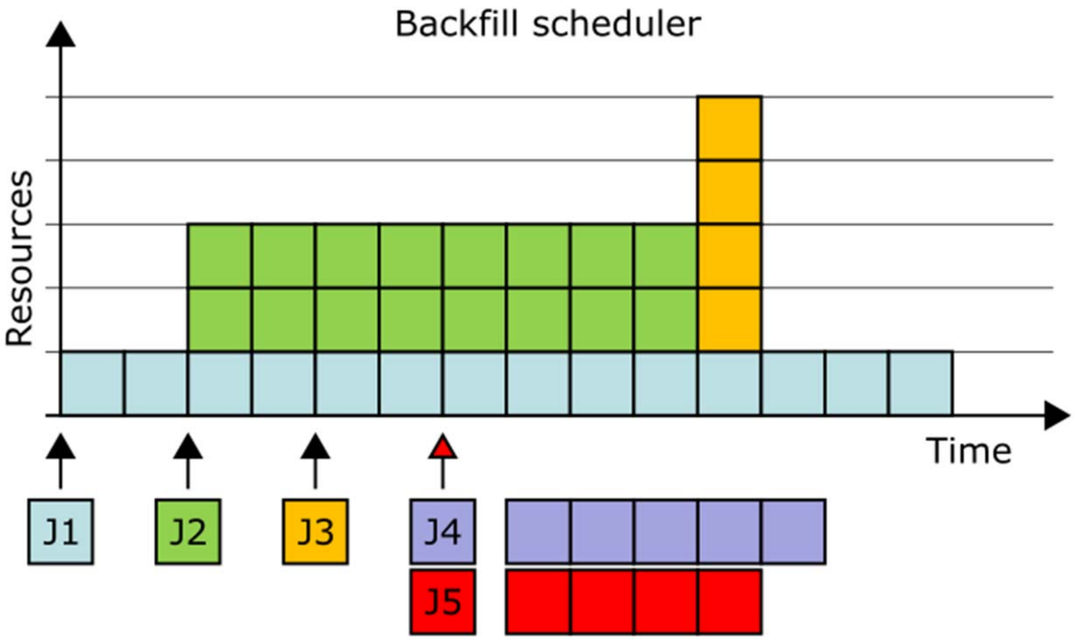
- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

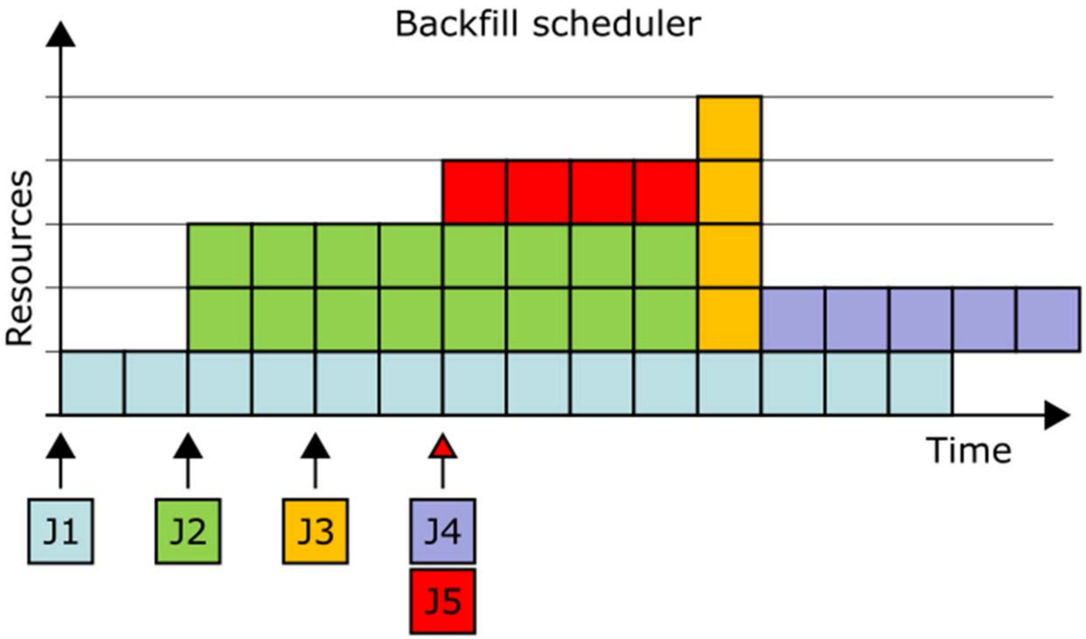
- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- **Backfill**
- Fairshare (dynamic priority)



# What is the HPC cluster? - Queues

Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- Backfill
- **Fairshare (dynamic priority)**

In order to enhance the user performance, fairshare feature calculates and assigns users' or projects' priority. This priority is inversely proportional to the utilization of the resources.

It is calculated first at project level and then at user level within the same project.

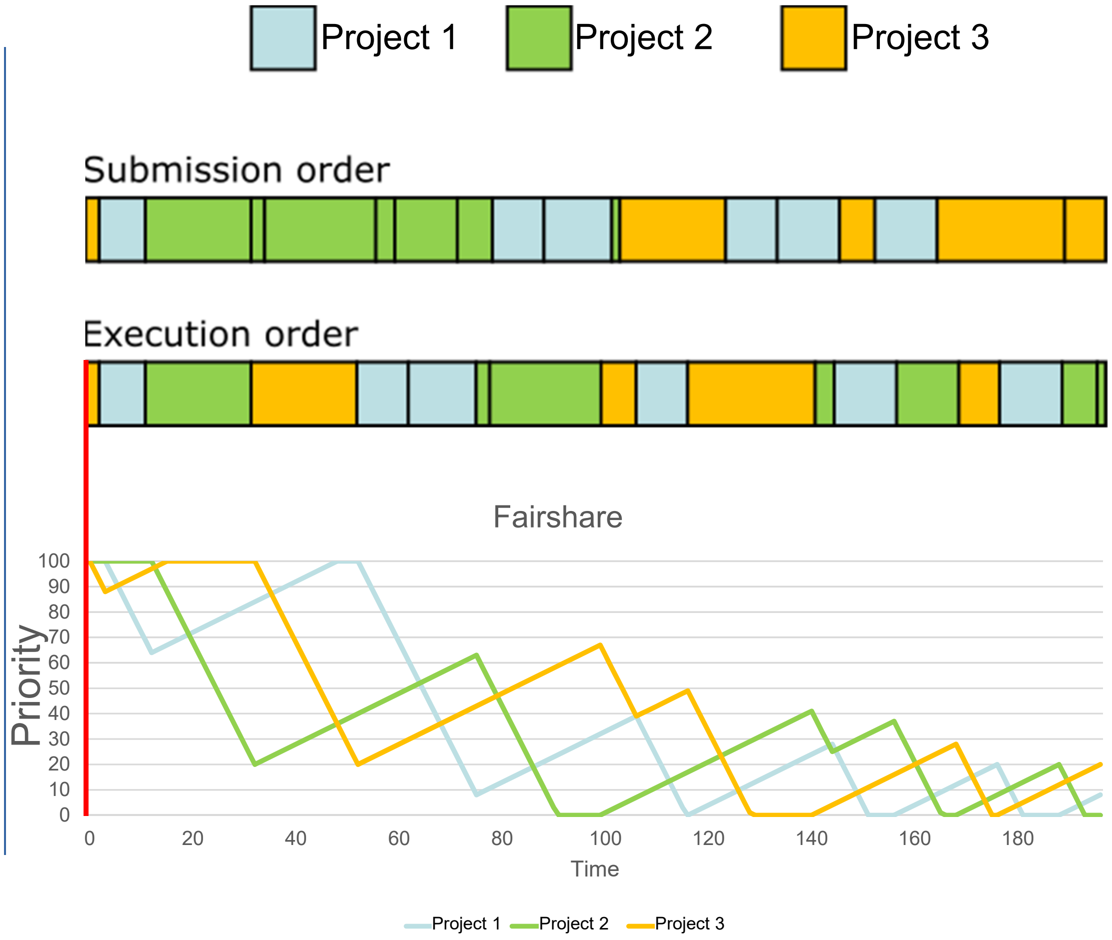
If the amount of resources used by all the jobs associated with Project 1 is higher than the amount of resources used by all the jobs associated with Project 2, then jobs submitted against Project 2 will have higher priority than jobs submitted against Project 1 and, hence, they will be dispatched sooner.

At the same time within the same project, if User 1 uses more resources then User 2, then User 2 will have higher priority than User 1 and, hence, their jobs will be dispatched sooner.

# What is the HPC cluster? - Queues

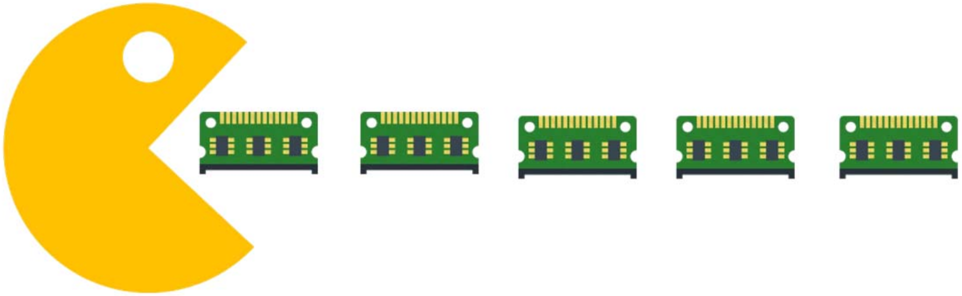
Queues are affected by different features and limitations:

- Batch and Interactive jobs.
- Array jobs
- Exclusive mode
- Default wall time
- Maximum wall time
- Backfill
- **Fairshare (dynamic priority)**



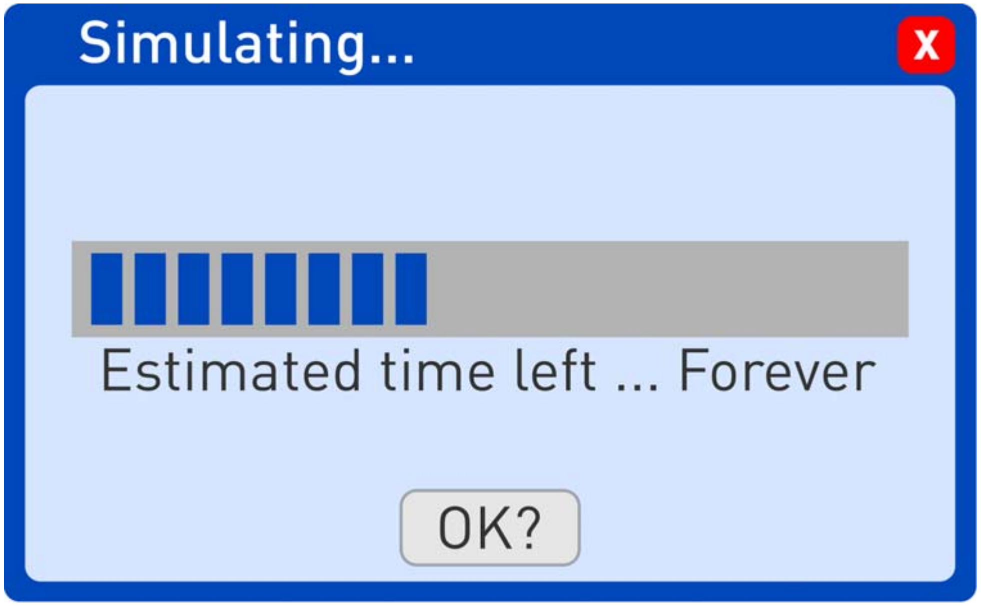
# What can you expect from the HPC cluster?

- **Solve problems that don’t fit in a single node memory space.**
- Solve problems that can’t be solved in a reasonable time.
- Solve problems that require large computational power using multiple nodes.
- Solve large amount of problems at the same time.
- Large amount of applications and libraries centrally installed and maintained.



# What can you expect from the HPC cluster?

- Solve problems that don’t fit in a single node memory space.
- **Solve problems that can’t be solved in a reasonable time.**
- Solve problems that require large computational power using multiple nodes.
- Solve large amount of problems at the same time.
- Large amount of applications and libraries centrally installed and maintained.



Simulating...

Estimated time left ... Forever

# What can you expect from the HPC cluster?

- Solve problems that don’t fit in a single node memory space.
- Solve problems that can’t be solved in a reasonable time.
- **Solve problems that require large computational power using multiple nodes.**
- Solve large amount of problems at the same time.
- Large amount of applications and libraries centrally installed and maintained.

# What can you expect from the HPC cluster?

- Solve problems that don’t fit in a single node memory space.
- Solve problems that can’t be solved in a reasonable time.
- Solve problems that require large computational power using multiple nodes.
- **Solve large amount of problems at the same time.**
- Large amount of applications and libraries centrally installed and maintained.

# What can you expect from the HPC cluster?

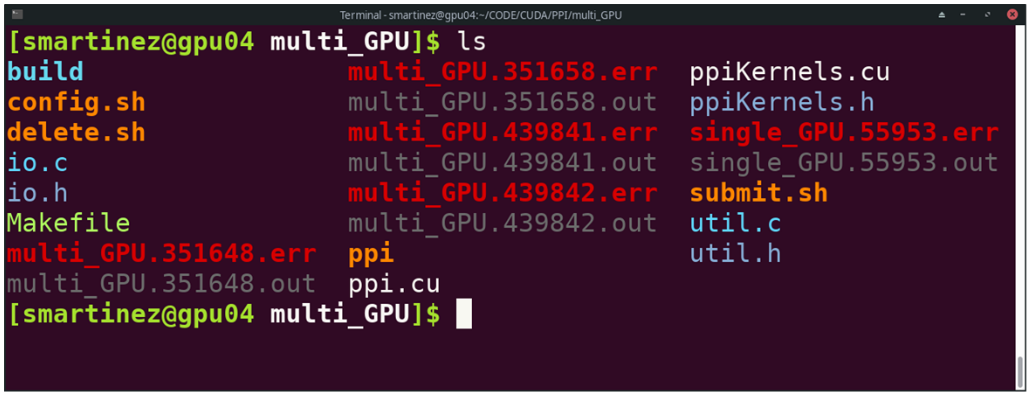
- Solve problems that don’t fit in a single node memory space.
- Solve problems that can’t be solved in a reasonable time.
- Solve problems that require large computational power using multiple nodes.
- Solve large amount of problems at the same time.
- **Large amount of applications and libraries centrally installed and maintained.**

SIMULIA
ABAQUS
ANSYS
COMSOL
intel
python™
1
4
2
3
QUANTUMESPRESSO
TensorFlow

# What do you need to know in order to use it?

- **Basic Linux commands.**
- Scheduler.
- Theoretical limits.
- Practical limits.
- Responsible usage.

- Command line interface.
- Remote connection (ssh).
- Navigating files and directories (cd, ls).
- Working with files and directories (mkdir, cp, mv rm).
- Shell scripts.



```
[smartinez@gpu04 multi_GPU]$ ls
build                  multi_GPU.351658.err  ppiKernels.cu
config.sh              multi_GPU.351658.out  ppiKernels.h
delete.sh              multi_GPU.439841.err  single_GPU.55953.err
io.c                   multi_GPU.439841.out  single_GPU.55953.out
io.h                   multi_GPU.439842.err  submit.sh
Makefile               multi_GPU.439842.out  util.c
multi_GPU.351648.err   ppi                   util.h
multi_GPU.351648.out   ppi.cu
[smartinez@gpu04 multi_GPU]$ 
```

# What do you need to know in order to use it?

- Basic Linux commands.
- **Scheduler.**
- Theoretical limits.
- Practical limits.
- Responsible usage.

Each **HPC cluster** has its own **scheduler** and hence its own **syntax**:

| | **SLURM (Almesbar HPC cluster)** |
|---|---|
| **Submit job** | sbatch |
| **Display resources** | sinfo |
| **Monitor jobs** | scontrol |
| **Kill jobs** | scancel |

# What do you need to know in order to use it?

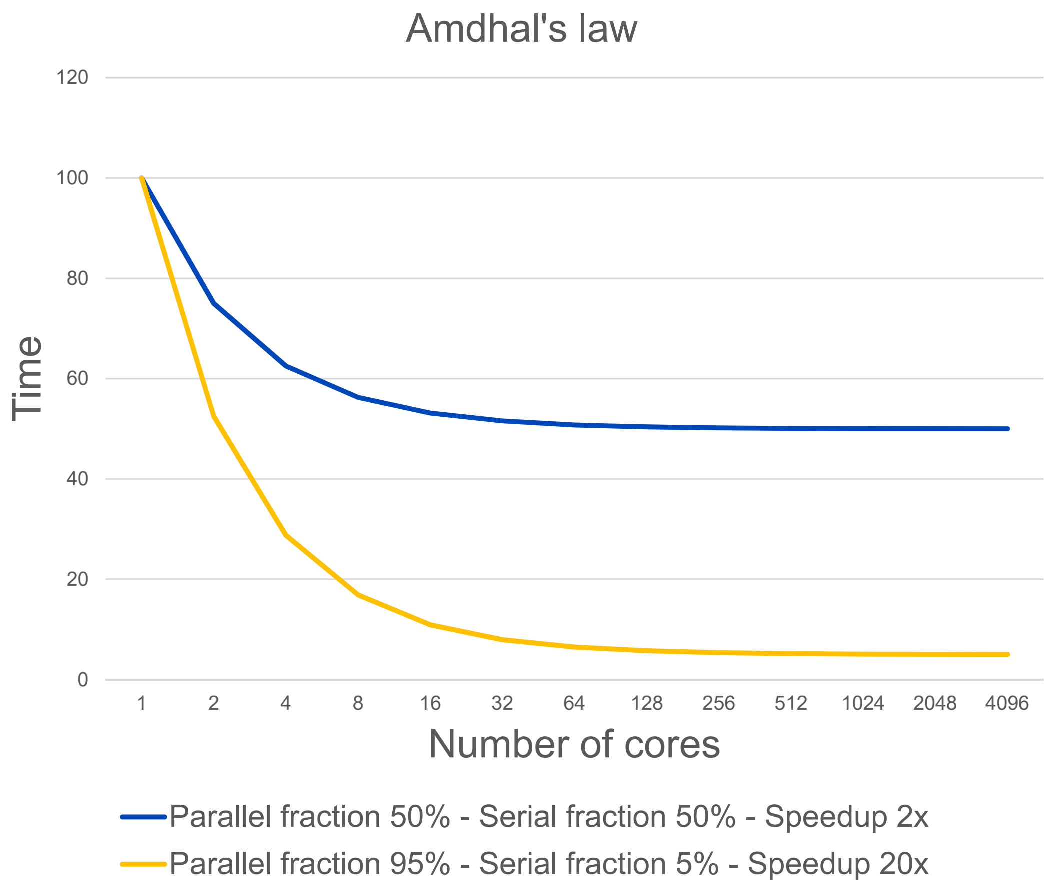
- Basic Linux commands.
- Scheduler.
- **Theoretical limits.**
- Practical limits.
- Responsible usage.

*“The more cores
I use in my jobs,
the faster I go”*

WRONG

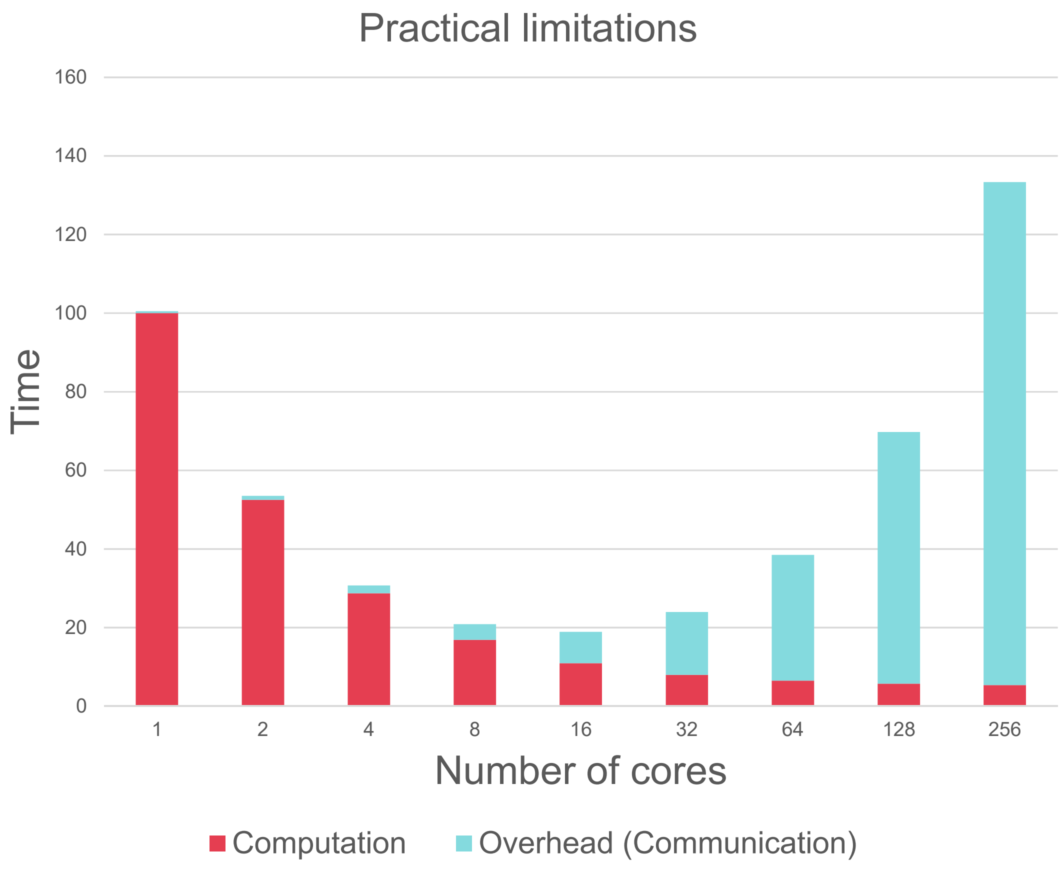
# What do you need to know in order to use it?

- Basic Linux commands.
- Scheduler.
- **Theoretical limits.**
- Practical limits.
- Responsible usage.



# What do you need to know in order to use it?

- Basic Linux commands.
- Scheduler.
- Theoretical limits.
- **Practical limits.**
- Responsible usage.



# What do you need to know in order to use it?

- Basic Linux commands.
- Scheduler.
- Theoretical limits.
- Practical limits.
- **Responsible usage.**

- Do not run applications on submission nodes.
- Do not submit unnecessary jobs.
- Do not request unnecessary resources.
- Do not connect directly to compute nodes.
- Try to optimize the use of the resources.

# Almesbar - Accounts, resource allocation and file sharing

- **User name and access**
- HPC Projects
- Jobs
- Queues
- Storage
- License applications

You will be using your KU Employee ID as your user name for accessing the HPC cluster.

You will be using SSH keys to identify yourself and access the HPC cluster.

SSH keys allow secure, password-less access to remote systems. They are very easy to set up and provide a greater level of security than using a password, while being convenient too.

We will discuss how to create and use your keys in the coming sections.

# Almesbar - Accounts, resource allocation and file sharing

- User name and access
- **HPC Projects**
- Jobs
- Queues
- Storage
- License applications

Resource allocation and file sharing on Almesbar HPC cluster is based on HPC projects.

Every single user account will be associated with at least one HPC project.

An HPC project will also be a shared space where project members can share files and submit their jobs.

Ultimately, every job has to be accounted against an HPC project.

Fairshare is applied to the HPC projects at first level.

# Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- **Jobs**
- Queues
- Storage
- License applications

Jobs number:

- Limit of 1000 jobs per user on **prod** and **gpu** queues
- Limit of 1 job per user on **devel** queue

Jobs size:

- Wall time limitations on each queue.
- There is no limit on the number of cores used by a job.

# Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- Jobs
- **Queues**
- Storage
- License applications

Access is restricted for train queue.

Research computing department uses train queue for compiling, installing, maintaining applications as well as for troubleshooting with the HPC users.

# Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- Jobs
- Queues
- **Storage**
- License applications

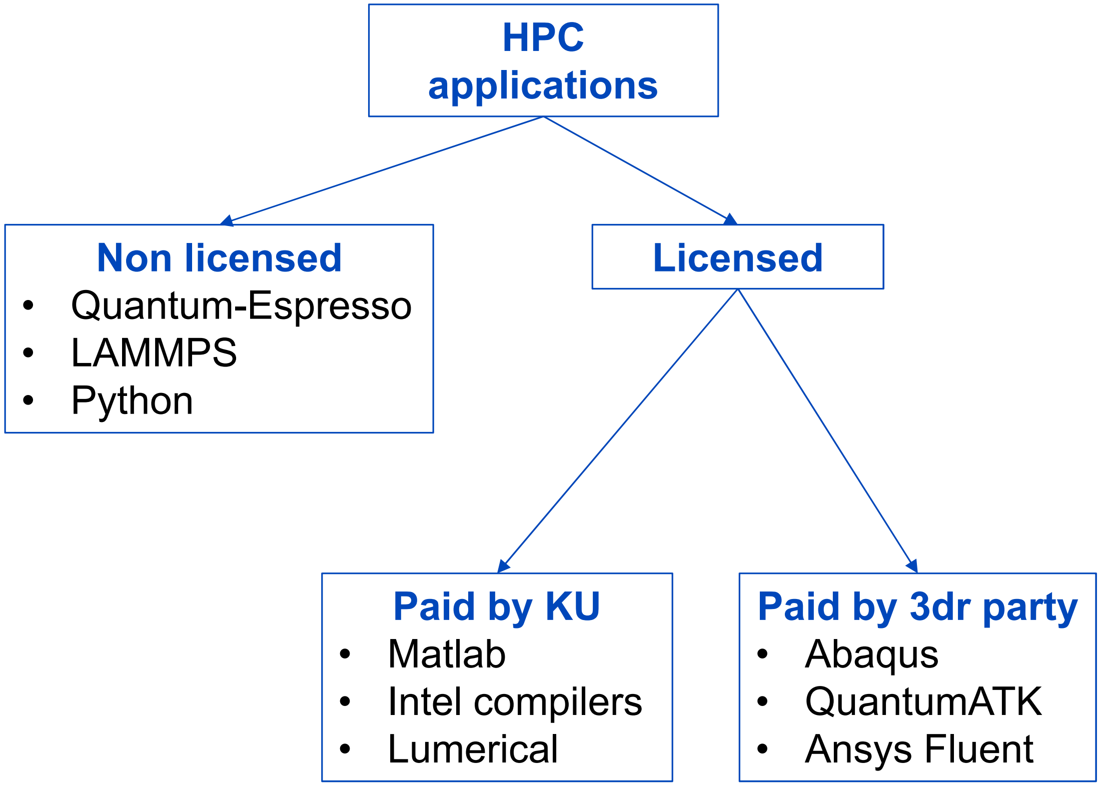
There will be quotas applied to the different storage places.

By default:

- Home directory (`/home/kunet.ae/username`): 50 GB
- Project directory (`/l/proj/projectname`): 2TB

# Almesbar - Accounts, resource allocation and file sharing

- User name and access
- HPC Projects
- Jobs
- Queues
- Storage
- **License applications**



HPC applications

Non licensed
- Quantum-Espresso
- LAMMPS
- Python

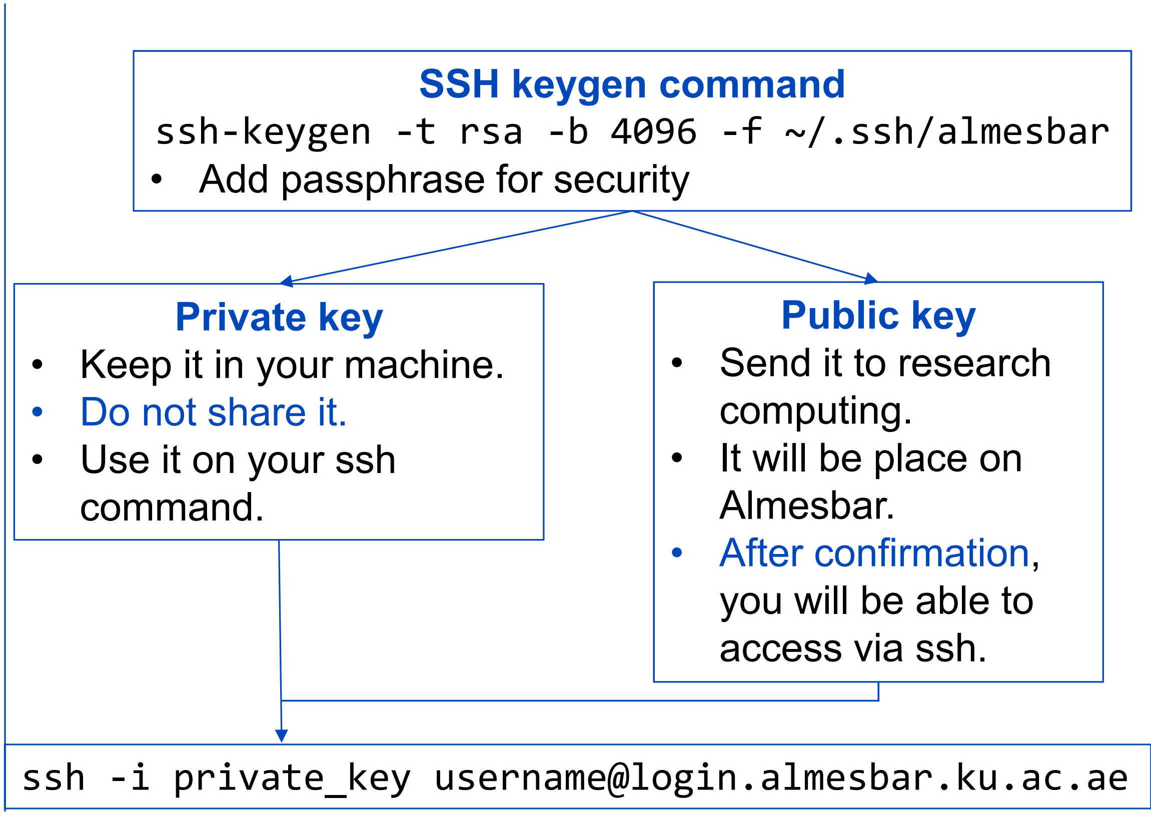
Licensed

Paid by KU
- Matlab
- Intel compilers
- Lumerical

Paid by 3dr party
- Abaqus
- QuantumATK
- Ansys Fluent

# Almesbar configuration

- **Access configuration**
- Queues (partitions)
- Modular environment
- Scheduler syntax



### SSH keygen command

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/almesbar
```

- Add passphrase for security

### Private key

- Keep it in your machine.
- Do not share it.
- Use it on your ssh command.

### Public key

- Send it to research computing.
- It will be place on Almesbar.
- After confirmation, you will be able to access via ssh.

```
ssh -i private_key username@login.almesbar.ku.ac.ae
```

# Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

| Production | |
|---|---|
| Name | prod |
| Purpose | Production jobs using CPU cores. |
| Number of nodes | 80 |
| Limits | Default wall time: 2h<br>Maximum wall time: 48h<br>Max number of jobs per user: 1000 |
| Access | Open |

# Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

| Development | |
|---|---|
| Name | devel |
| Purpose | For interactive and testing purposes, i.e, compiling, configuring and testing applications. Running GUI applications. |
| Number of nodes | 4 |
| Limits | Default wall time: 1h<br>Maximum wall time: 8h<br>Max number of jobs per user: 1 |
| Access | Open |

# Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

| GPU | |
|---|---|
| Name | gpu |
| Purpose | Production jobs using GPU cores. |
| Number of nodes | 3 |
| Number of GPUs per node | 4 |
| Limits | Default wall time: 2h<br>Maximum wall time: 48h<br>Max number of jobs per user: 1000 |
| Access | Open |

# Almesbar configuration

- Access configuration
- **Queues (partitions)**
- Modular environment
- Scheduler syntax

| Training | |
|---|---|
| Name | train |
| Purpose | Used by Research Computing department for compiling, installing, maintaining applications as well as for troubleshooting with the HPC users. |
| Number of nodes | 4 |
| Limits | No |
| Access | Restricted |

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

On a complex computer system, on which it is necessary to make available a wide choice of software packages in multiple versions, it can be quite difficult to set up the user environment so as to always find the required executables and libraries.

This is particularly true where different implementations or versions use the same names for files.

Environment modules provide a way to selectively activate and deactivate modifications to the user environment which allow particular packages and versions to be found.

Our modular environment is hierarchical and it has three levels:

Core, Compiler and MPI

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

## Useful commands:

| Command | Example | Description |
|---|---|---|
| module avail | module avail | Lists available modules in the system.<br>(D) means default.<br>(L) means it is already loaded.<br>Press Q to exit the list at any moment. |
| module list | module list | List your currently loaded modules. |
| module load module_name | module load matlab/R2021a | Loads the module for Matlab version R2019b |
| module show module_name | module show matlab/R2021a | Displays information about one or more module files. |
| module spider string | module spider wrf | List all module that contain the "string“ with details on how to load them. |
| module swap old_module new_module | module swap python/2.7.3 python/3.6.0 | Switches similar/conflicting modules |

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

**Core level:**

- Modules that are not dependent on Compiler or MPI implementation.
- Most of the installed applications that don't require to be compiled seats in this level. Also the compilers can be found in this level.
- This is the only level that is visible for `module avail` command after logging in the system.
- Only if user loads a module that is a compiler, for example `gcc/9.3` , the next level in the hierarchy (Compiler) will be available and became visible with module avail.

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

## Core level:

```
[username@login-3 ~]$ module avail
-------------------------- /apps/ku/modulefiles/Core --------------------------
   abaqus/2019             cuda/11.3                miniconda/3
   abaqus/2021    (D)      gcc/9.3                  molpro/2021.1.0
   ansys/2021R1            lumerical/2021R1         starccm/16.02.009
   ansysem/2021R1          matlab/R2020b
   comsol/56               matlab/R2021a    (D)
```

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

**Compiler level:**

- This level becomes available only if the user loads a compiler module, like `gcc/9.3` from the Core level.
- Then, all the applications compiled with that compiler (`gcc/9.3`) that don't require MPI to be compiled, will be listed here. This include MPI implementations like `openmpi` or `mvapich` for example.
- Only if user loads a module that is an MPI implementation, for example `openmpi/4.0` , the next level in the hierarchy (MPI) will be available and became visible with `module avail` command.

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

## Compiler level:

```
[username@login-3 ~]$ module load gcc/9.3
[username@login-3 ~]$ module avail

-------------------- /apps/ku/modulefiles/Compiler/gcc/9.3 --------------------
   hdf5/1.10.6         libpng/1.2.12     ucx/1.8.0
   jasper/1.701.0      openblas/0.3.7    wannier90/3.1.0
   libfabric/1.10.1    openmpi/4.0       zlib/1.2.11

-------------------------- /apps/ku/modulefiles/Core --------------------------
   abaqus/2019           cuda/11.3               miniconda/3
   abaqus/2021    (D)    gcc/9.3          (L)    molpro/2021.1.0
   ansys/2021R1          lumerical/2021R1        starccm/16.02.009
   ansysem/2021R1        matlab/R2020b
   comsol/56             matlab/R2021a    (D)
```

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

**MPI level:**

- This level becomes available only if the user loads an MPI implementation module, like `openmpi/4.0` from the Compiler level.
- Then, all the applications and libraries compiled with that compiler/MPI pair ( `gcc/9.3` and `openmpi/4.0` will be listed here.

# Almesbar configuration

- Access configuration
- Queues (partitions)
- **Modular environment**
- Scheduler syntax

## MPI level:

```
[username@login-3 ~]$ module load openmpi/4.0
[username@login-3 ~]$ module avail

---------------- /apps/ku/modulefiles/MPI/gcc/9.3/openmpi/4.0 -----------------
   elpa/2021.05.001        netcdf/4.7.3       vasp/6.2.0-wannier90-3.1.0
   fftw/3.3.8              phdf5/1.10.6       wps/4.0
   lammps/29Oct2020        pnetcdf/1.12.1     wrf/4.0
   netcdf-cxx/4.3.1        qe/6.7.0
   netcdf-fortran/4.5.2    scalapack/2.1.0

-------------------- /apps/ku/modulefiles/Compiler/gcc/9.3 --------------------
   hdf5/1.10.6         libpng/1.2.12     ucx/1.8.0
   jasper/1.701.0      openblas/0.3.7    wannier90/3.1.0
   libfabric/1.10.1    openmpi/4.0  (L)  zlib/1.2.11

-------------------------- /apps/ku/modulefiles/Core --------------------------
   abaqus/2019           cuda/11.3               miniconda/3
   abaqus/2021    (D)    gcc/9.3          (L)    molpro/2021.1.0
   ansys/2021R1          lumerical/2021R1        starccm/16.02.009
   ansysem/2021R1        matlab/R2020b
   comsol/56             matlab/R2021a    (D)
```

# Almesbar configuration

- Access configuration
- Queues (partitions)
- Modular environment
- **Scheduler syntax**

## Job submission control

| Command | Example | Description |
|---|---|---|
| salloc | salloc -n 1 | Obtain a job allocation. |
| sbatch | sbatch my_script.sh | Submit a batch script for later execution. |
| srun | srun -n 1 executable | If run from command line directly, srun will first create a resource allocation in which to run the parallel job.<br>srun can be run from within sbatch script or salloc shell |
| sinfo | sinfo -p prod | Display partition information |
|  | sinfo -N | Display nodes information |
| squeue | squeue -p prod | Display information about jobs |
| scontrol | scontrol show job 3420 | Show active job details |
| sacct | sacct -j 3420 | Show accounting information about active and completed jobs |
| scancel | scancel 3420 | Send a signal to kill the job |

# Almesbar configuration

- Access configuration
- Queues (partitions)
- Modular environment
- **Scheduler syntax**

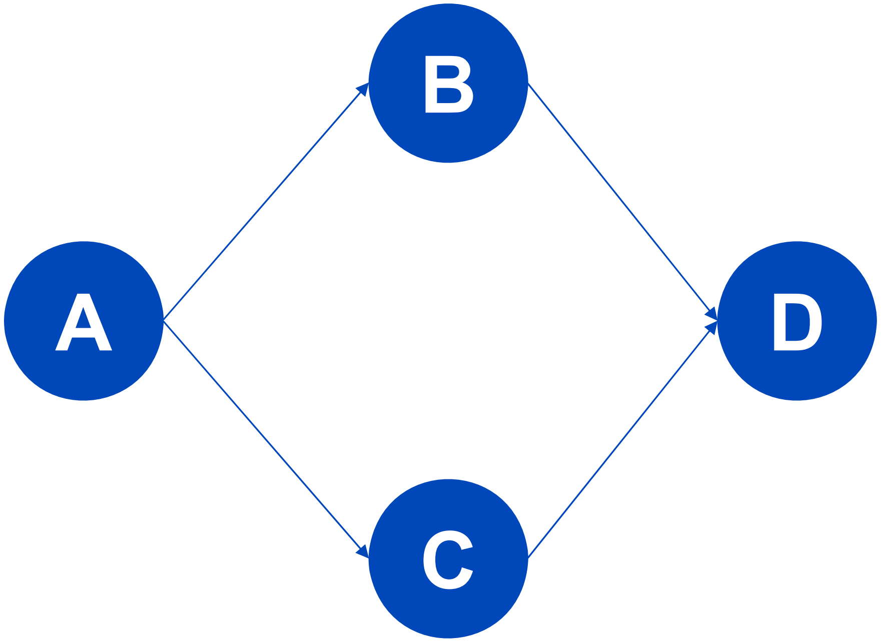
## Configuring a job

| Option | Example | Description |
|---|---|---|
| #SBATCH | | Scheduler directive to be used on job configuration scripts |
| --nodes | --nodes=2 | Will request tasks be run across 2 nodes |
| --ntasks | --ntasks=2 | Will start 2 MPI tasks |
| --ntasks-per-node | --ntasks-per-node=1 | Will start 1 task per requested node |
| --cpus-per-task | --cpus-per-task=10 | Will request 10 cores per task |
| --account | --account=project_name | The project your core hours will be 'charged' to. |
| --partition | --partition=prod | Submits job the specified partition. On Almesbar HPC cluster, the default queue is “prod" |
| --output | --output=file_name | Writes job output to “file_name”, %j can be used to include the JOBID, e.g. result.%j.out |
| --error | --output=file_name | Writes job errors to “file_name”, %j can be used to include the JOBID, e.g. result.%j.err |
| --job-name | --job-name=job_name | Assigns the specified name “job_name” to the job |
| --time | --time=DD-HH:MM:SS | Requests wall time limit |
| --mem | --mem=512MB | Requests a specific amount of memory for your job |
| --x11 | salloc -n 1 --x11<br>srun -n 1 --x11 xterm | Enables X11 graphical forwarding in interactive jobs |
| --pty | srun -n 1 --pty /bin/bash | Submits an interactive job that will create a terminal when the job starts. |
| --exclusive | --exclusive | Puts the host/s running your job into exclusive execution mode. |

# Our support

- **Pipeline optimization.**
- Programming consultancy.
- Code optimization.
- Build, install, maintain and test scientific applications.
- Training.
- Knowledge Base.
- Contact email.
- Location.

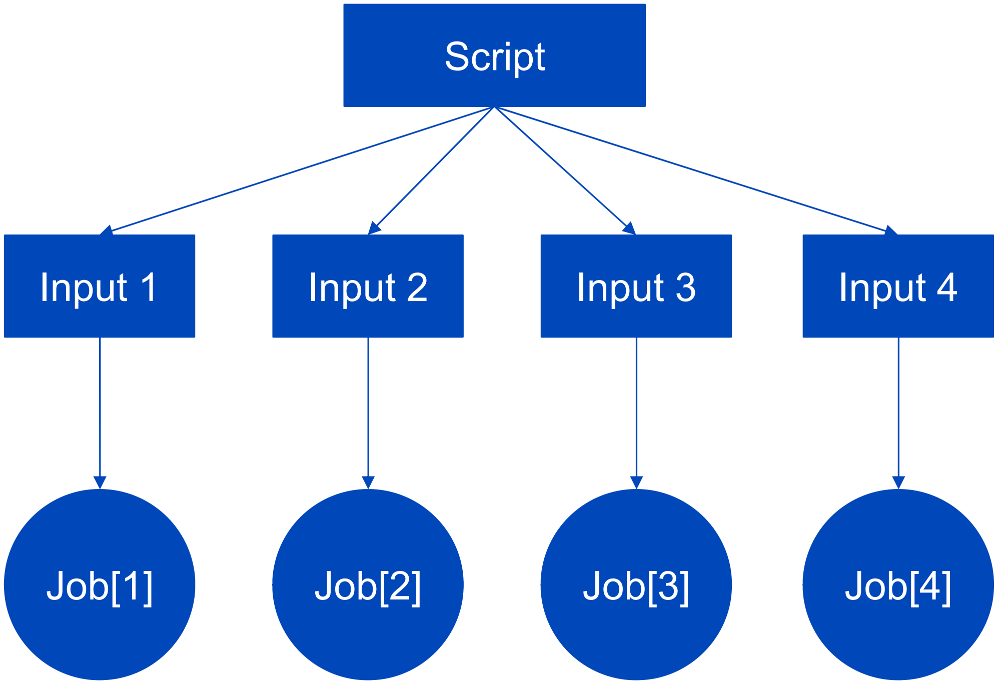
- Job dependencies



# Our support

- **Pipeline optimization.**
- Programming consultancy.
- Code optimization.
- Build, install, maintain and test scientific applications.
- Training.
- Knowledge Base.
- Contact email.
- Location.

- Array job: different parameters, but same executable.



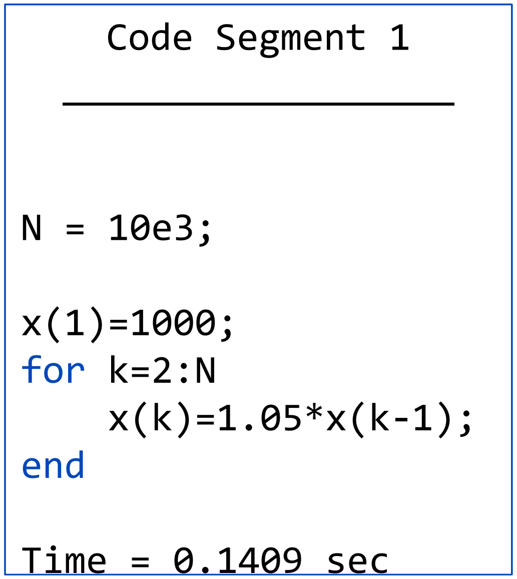
# Our support

- Pipeline optimization.
- **Programming consultancy.**
- Code optimization.
- Build, install, maintain and test scientific applications.
- Training.
- Knowledge Base.
- Contact email.
- Location.

NVIDIA.
CUDA
$_
C
C++
python™
MPI

# Our support

- Pipeline optimization.
- Programming consultancy.
- **Code optimization.**
- Build, install, maintain and test scientific applications.
- Training.
- Knowledge Base.
- Contact email.
- Location.

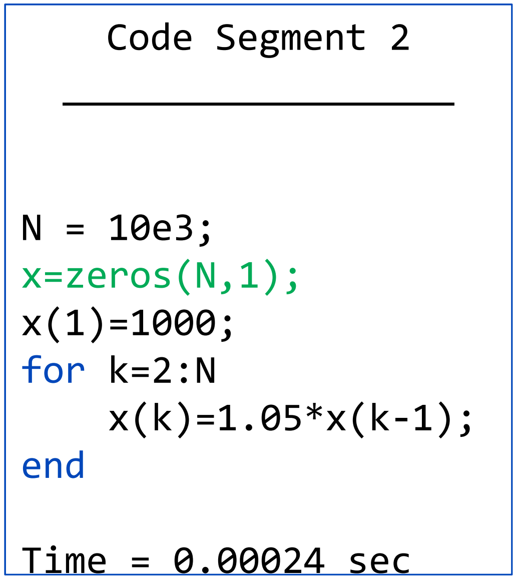


Code Segment 1

```
N = 10e3;

x(1)=1000;
for k=2:N
    x(k)=1.05*x(k-1);
end

Time = 0.1409 sec
```



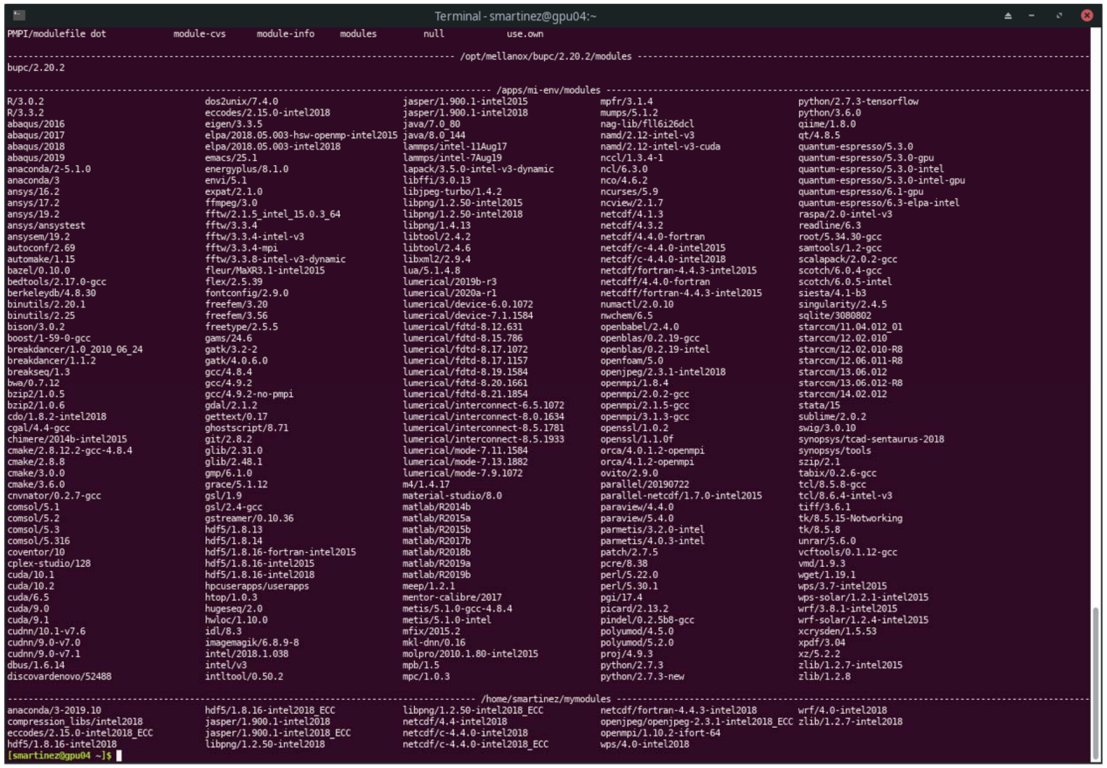
Code Segment 2

```
N = 10e3;
x=zeros(N,1);
x(1)=1000;
for k=2:N
    x(k)=1.05*x(k-1);
end

Time = 0.00024 sec
```

# Our support

- ➢ Pipeline optimization.
- ➢ Programming consultancy.
- ➢ Code optimization.
- ➢ **Build, install, maintain and test scientific applications.**
- ➢ Training.
- ➢ Knowledge Base.
- ➢ Contact email.
- ➢ Location.



Terminal - smartinez@gpu04:~

```
PMPI/modulefile dot          module-cvs     module-info     modules          null             use.own

---------------------------------------------------------------------------- /opt/mellanox/bupc/2.20.2/modules ------------------------------------------------------------------------------
bupc/2.20.2

------------------------------------------------------------------------------------- /apps/mi-env/modules --------------------------------------------------------------------------------
R/3.0.2                               dos2unix/7.4.0                        jasper/1.900.1-intel2015              mpfr/3.1.4                            python/2.7.3-tensorflow
R/3.3.2                               eccodes/2.15.0-intel2018              jasper/1.900.1-intel2018              mumps/5.1.2                           python/3.6.0
abaqus/2016                           eigen/3.3.5                           java/7.0_80                           nag-lib/fll6i26dcl                    qiime/1.8.0
abaqus/2017                           elpa/2018.05.003-hsw-openmp-intel2015 java/8.0_144                          namd/2.12-intel-v3                    qt/4.8.5
abaqus/2018                           elpa/2018.05.003-intel2018            lammps/intel-11Aug17                  namd/2.12-intel-v3-cuda               quantum-espresso/5.3.0
abaqus/2019                           emacs/25.1                            lammps/intel-7Aug19                   nccl/1.3.4-1                          quantum-espresso/5.3.0-gpu
anaconda/2-5.1.0                      energyplus/8.1.0                      lapack/3.5.0-intel-v3-dynamic         ncl/6.3.0                             quantum-espresso/5.3.0-intel
anaconda/3                            envi/5.1                              libffi/3.0.13                         nco/4.6.2                             quantum-espresso/5.3.0-intel-gpu
ansys/16.2                            expat/2.1.0                           libjpeg-turbo/1.4.2                   ncurses/5.9                           quantum-espresso/6.1-gpu
ansys/17.2                            ffmpeg/3.0                            libpng/1.2.50-intel2015               ncview/2.1.7                          quantum-espresso/6.3-elpa-intel
ansys/19.2                            fftw/2.1.5_intel_15.0.3_64            libpng/1.2.50-intel2018               netcdf/4.1.3                          raspa/2.0-intel-v3
ansys/ansystest                       fftw/3.3.4                            libpng/1.4.13                         netcdf/4.3.2                          readline/6.3
ansysem/19.2                          fftw/3.3.4-intel-v3                   libtool/2.4.2                         netcdf/4.4.0-fortran                  root/5.34.30-gcc
autoconf/2.69                         fftw/3.3.4-mpi                        libtool/2.4.6                         netcdf/c-4.4.0-intel2015              samtools/1.2-gcc
automake/1.15                         fftw/3.3.8-intel-v3-dynamic           libxml2/2.9.4                         netcdf/c-4.4.0-intel2018              scalapack/2.0.2-gcc
bazel/0.10.0                          fleur/MaXR3.1-intel2015               lua/5.1.4.8                           netcdf/fortran-4.4.3-intel2015        scotch/6.0.4-gcc
bedtools/2.17.0-gcc                   flex/2.5.39                           lumerical/2019b-r3                    netcdff/4.4.0-fortran                 scotch/6.0.5-intel
berkeleydb/4.8.30                     fontconfig/2.9.0                      lumerical/2020a-r1                    netcdff/fortran-4.4.3-intel2015       siesta/4.1-b3
binutils/2.20.1                       freefem/3.20                          lumerical/device-6.0.1072             numactl/2.0.10                        singularity/2.4.5
binutils/2.25                         freefem/3.56                          lumerical/device-7.1.1584             nwchem/6.5                            sqlite/3080802
bison/3.0.2                           freetype/2.5.5                        lumerical/fdtd-8.12.631               openbabel/2.4.0                       starccm/11.04.012_01
boost/1-59-0-gcc                      gams/24.6                             lumerical/fdtd-8.15.786               openblas/0.2.19-gcc                   starccm/12.02.010
breakdancer/1.0_2010_06_24            gatk/3.2-2                            lumerical/fdtd-8.17.1072              openblas/0.2.19-intel                 starccm/12.02.010-R8
breakdancer/1.1.2                     gatk/4.0.6.0                          lumerical/fdtd-8.17.1157              openfoam/5.0                          starccm/12.06.011-R8
breakseq/1.3                          gcc/4.8.4                             lumerical/fdtd-8.19.1584              openjpeg/2.3.1-intel2018              starccm/13.06.012
bwa/0.7.12                            gcc/4.9.2                             lumerical/fdtd-8.20.1661              openmpi/1.8.4                         starccm/13.06.012-R8
bzip2/1.0.5                           gcc/4.9.2-no-pmpi                     lumerical/fdtd-8.21.1854              openmpi/2.0.2-gcc                     starccm/14.02.012
bzip2/1.0.6                           gdal/2.1.2                            lumerical/interconnect-6.5.1072       openmpi/2.1.5-gcc                     stata/15
cdo/1.8.2-intel2018                   gettext/0.17                          lumerical/interconnect-8.0.1634       openmpi/3.1.3-gcc                     sublime/2.0.2
cgal/4.4-gcc                          ghostscript/8.71                      lumerical/interconnect-8.5.1781       openssl/1.0.2                         swig/3.0.10
chimere/2014b-intel2015               git/2.8.2                             lumerical/interconnect-8.5.1933       openssl/1.1.0f                        synopsys/tcad-sentaurus-2018
cmake/2.8.12.2-gcc-4.8.4              glib/2.31.0                           lumerical/mode-7.11.1584              orca/4.0.1.2-openmpi                  synopsys/tools
cmake/2.8.8                           glib/2.48.1                           lumerical/mode-7.13.1882              orca/4.1.2-openmpi                    szip/2.1
cmake/3.0.0                           gmp/6.1.0                             lumerical/mode-7.9.1072               ovito/2.9.0                           tabix/0.2.6-gcc
cmake/3.6.0                           grace/5.1.12                          m4/1.4.17                             parallel/20190722                     tcl/8.5.8-gcc
cnvnator/0.2.7-gcc                    gsl/1.9                               material-studio/8.0                   parallel-netcdf/1.7.0-intel2015       tcl/8.6.4-intel-v3
comsol/5.1                            gsl/2.4-gcc                           matlab/R2014b                         paraview/4.4.0                        tiff/3.6.1
comsol/5.2                            gstreamer/0.10.36                     matlab/R2015a                         paraview/5.4.0                        tk/8.5.15-Notworking
comsol/5.3                            hdf5/1.8.13                           matlab/R2015b                         parmetis/3.2.0-intel                  tk/8.5.8
comsol/5.316                          hdf5/1.8.14                           matlab/R2017b                         parmetis/4.0.3-intel                  unrar/5.6.0
coventor/10                           hdf5/1.8.16-fortran-intel2015         matlab/R2018b                         patch/2.7.5                           vcftools/0.1.12-gcc
cplex-studio/128                      hdf5/1.8.16-intel2015                 matlab/R2019a                         pcre/8.38                             vmd/1.9.3
cuda/10.1                             hdf5/1.8.16-intel2018                 matlab/R2019b                         perl/5.22.0                           wget/1.19.1
cuda/10.2                             hpcuserapps/userapps                  meep/1.2.1                            perl/5.30.1                           wps/3.7-intel2015
cuda/6.5                              htop/1.0.3                            mentor-calibre/2017                   pgi/17.4                              wps-solar/1.2.1-intel2015
cuda/9.0                              hugeseq/2.0                           metis/5.1.0-gcc-4.8.4                 picard/2.13.2                         wrf/3.8.1-intel2015
cuda/9.1                              hwloc/1.10.0                          metis/5.1.0-intel                     pindel/0.2.5b8-gcc                    wrf-solar/1.2.4-intel2015
cudnn/10.1-v7.6                       idl/8.3                               mfix/2015.2                           polyumod/4.5.0                        xcrysden/1.5.53
cudnn/9.0-v7.0                        imagemagik/6.8.9-8                    mkl-dnn/0.16                          polyumod/5.2.0                        xpdf/3.04
cudnn/9.0-v7.1                        intel/2018.1.038                      molpro/2010.1.80-intel2015            proj/4.9.3                            xz/5.2.2
dbus/1.6.14                           intel/v3                              mpb/1.5                               python/2.7.3                          zlib/1.2.7-intel2015
discovardenovo/52488                  intltool/0.50.2                       mpc/1.0.3                             python/2.7.3-new                      zlib/1.2.8

------------------------------------------------------------------------------------- /home/smartinez/mymodules ---------------------------------------------------------------------------
anaconda/3-2019.10                    hdf5/1.8.16-intel2018_ECC             libpng/1.2.50-intel2018_ECC           netcdf/fortran-4.4.3-intel2018        wrf/4.0-intel2018
compression_libs/intel2018            jasper/1.900.1-intel2018              netcdf/4.4-intel2018                  openjpeg/openjpeg-2.3.1-intel2018_ECC zlib/1.2.7-intel2018
eccodes/2.15.0-intel2018_ECC          jasper/1.900.1-intel2018_ECC          netcdf/c-4.4.0-intel2018              openmpi/1.10.2-ifort-64
hdf5/1.8.16-intel2018                 libpng/1.2.50-intel2018               netcdf/c-4.4.0-intel2018_ECC          wps/4.0-intel2018
[smartinez@gpu04 ~]$
```

# Our support

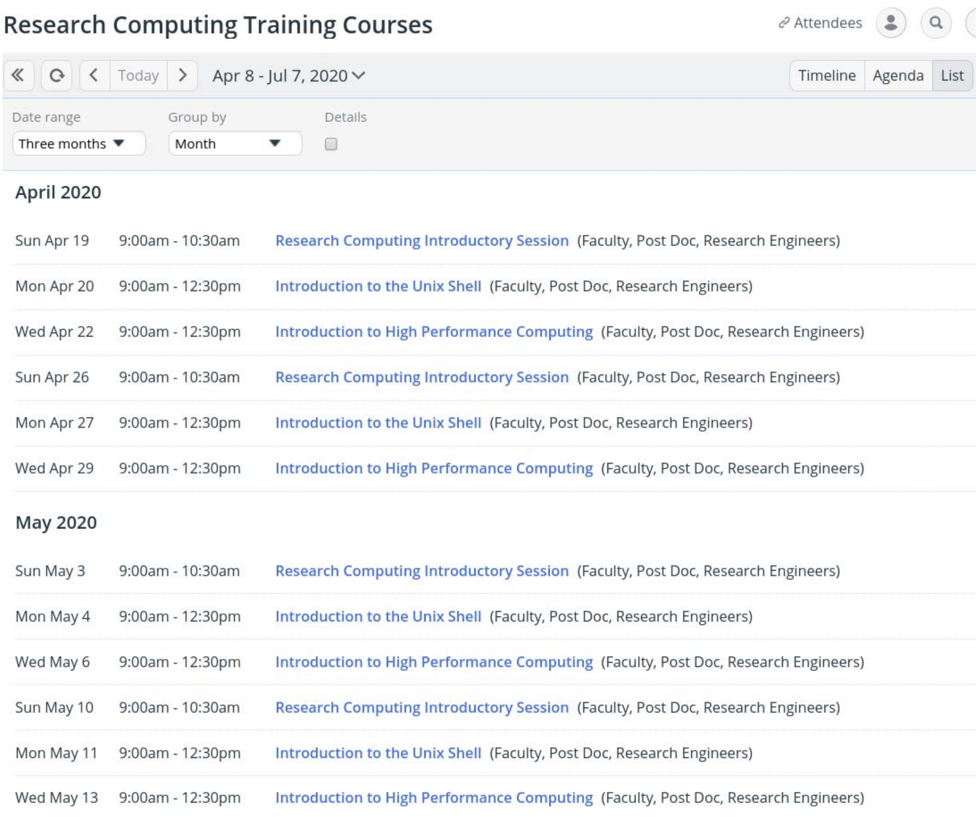
- Pipeline optimization.
- Programming consultancy.
- Code optimization.
- Build, install, maintain and test scientific applications.
- **Training.**
- Knowledge Base.
- Contact email.
- Location.

- Research Computing **Introductory Session**.
- Introduction to Linux command line.
- **High Performance Computing** on Almesbar.
- **Numpy** - Numerical Python.
- **Matplotlib** - Plotting with Python.

# Our support

- Pipeline optimization.
- Programming consultancy.
- Code optimization.
- Build, install, maintain and test scientific applications.
- **Training.**
- Knowledge Base.
- Contact email.
- Location.

https://teamup.com/ksquu2jki56nesnr6n



Research Computing Training Courses

April 2020

| | | |
|---|---|---|
| Sun Apr 19 | 9:00am - 10:30am | Research Computing Introductory Session (Faculty, Post Doc, Research Engineers) |
| Mon Apr 20 | 9:00am - 12:30pm | Introduction to the Unix Shell (Faculty, Post Doc, Research Engineers) |
| Wed Apr 22 | 9:00am - 12:30pm | Introduction to High Performance Computing (Faculty, Post Doc, Research Engineers) |
| Sun Apr 26 | 9:00am - 10:30am | Research Computing Introductory Session (Faculty, Post Doc, Research Engineers) |
| Mon Apr 27 | 9:00am - 12:30pm | Introduction to the Unix Shell (Faculty, Post Doc, Research Engineers) |
| Wed Apr 29 | 9:00am - 12:30pm | Introduction to High Performance Computing (Faculty, Post Doc, Research Engineers) |

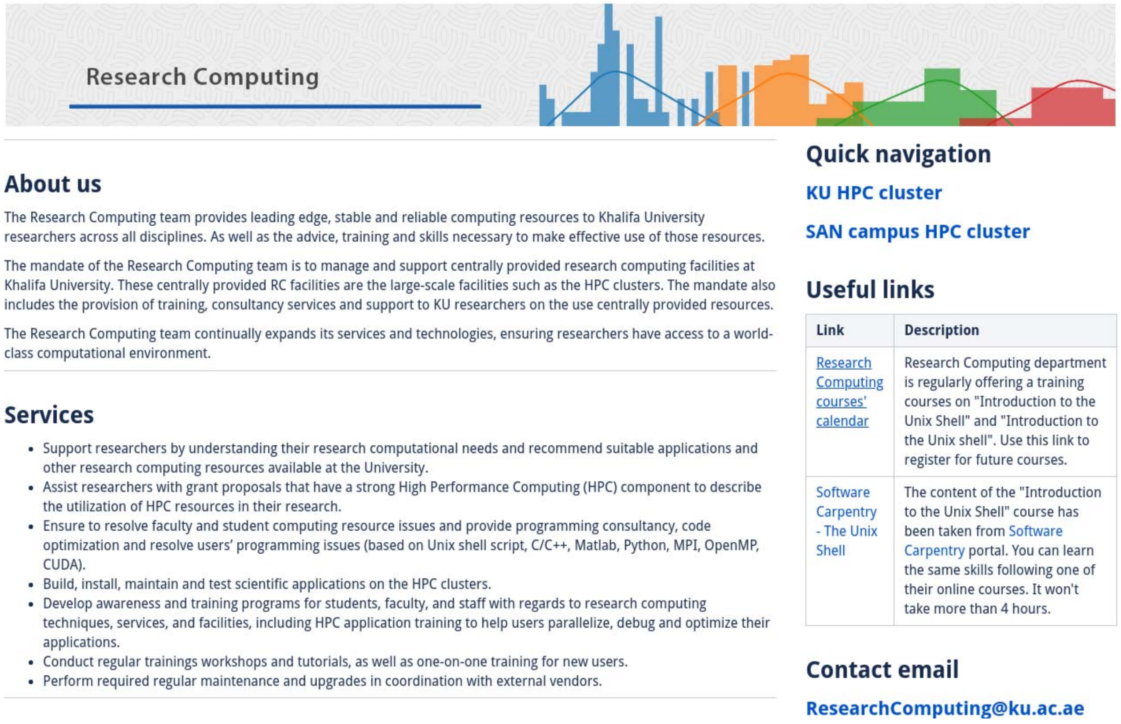
May 2020

| | | |
|---|---|---|
| Sun May 3 | 9:00am - 10:30am | Research Computing Introductory Session (Faculty, Post Doc, Research Engineers) |
| Mon May 4 | 9:00am - 12:30pm | Introduction to the Unix Shell (Faculty, Post Doc, Research Engineers) |
| Wed May 6 | 9:00am - 12:30pm | Introduction to High Performance Computing (Faculty, Post Doc, Research Engineers) |
| Sun May 10 | 9:00am - 10:30am | Research Computing Introductory Session (Faculty, Post Doc, Research Engineers) |
| Mon May 11 | 9:00am - 12:30pm | Introduction to the Unix Shell (Faculty, Post Doc, Research Engineers) |
| Wed May 13 | 9:00am - 12:30pm | Introduction to High Performance Computing (Faculty, Post Doc, Research Engineers) |

# Our support

- Pipeline optimization.
- Programming consultancy.
- Code optimization.
- Build, install, maintain and test scientific applications.
- Training.
- **Knowledge Base.**
- Contact email.
- Location.

https://wiki-researchcomputing.ku.ac.ae/



## About us

The Research Computing team provides leading edge, stable and reliable computing resources to Khalifa University researchers across all disciplines. As well as the advice, training and skills necessary to make effective use of those resources.

The mandate of the Research Computing team is to manage and support centrally provided research computing facilities at Khalifa University. These centrally provided RC facilities are the large-scale facilities such as the HPC clusters. The mandate also includes the provision of training, consultancy services and support to KU researchers on the use centrally provided resources.

The Research Computing team continually expands its services and technologies, ensuring researchers have access to a world-class computational environment.

## Services

- Support researchers by understanding their research computational needs and recommend suitable applications and other research computing resources available at the University.
- Assist researchers with grant proposals that have a strong High Performance Computing (HPC) component to describe the utilization of HPC resources in their research.
- Ensure to resolve faculty and student computing resource issues and provide programming consultancy, code optimization and resolve users' programming issues (based on Unix shell script, C/C++, Matlab, Python, MPI, OpenMP, CUDA).
- Build, install, maintain and test scientific applications on the HPC clusters.
- Develop awareness and training programs for students, faculty, and staff with regards to research computing techniques, services, and facilities, including HPC application training to help users parallelize, debug and optimize their applications.
- Conduct regular trainings workshops and tutorials, as well as one-on-one training for new users.
- Perform required regular maintenance and upgrades in coordination with external vendors.

## Quick navigation

KU HPC cluster

SAN campus HPC cluster

## Useful links

| Link | Description |
|---|---|
| Research Computing courses' calendar | Research Computing department is regularly offering a training courses on "Introduction to the Unix Shell" and "Introduction to the Unix shell". Use this link to register for future courses. |
| Software Carpentry - The Unix Shell | The content of the "Introduction to the Unix Shell" course has been taken from Software Carpentry portal. You can learn the same skills following one of their online courses. It won't take more than 4 hours. |

## Contact email

ResearchComputing@ku.ac.ae

# Our support

- Pipeline optimization.
- Programming consultancy.
- Code optimization.
- Build, install, maintain and test scientific applications.
- Training.
- Knowledge Base.
- **Contact email.**
- Location.

**ResearchComputing@ku.ac.ae**

# Our support

- Pipeline optimization.
- Programming consultancy.
- Code optimization.
- Build, install, maintain and test scientific applications.
- Training.
- Knowledge Base.
- Contact email.
- **Location.**

- **Sas Al Nakhl campus**
- **Habshan building**
- **Second floor**
- **Rooms 246 and 248**

جامعـــة خليفـــة
Khalifa University

# Thank You

ku.ac.ae