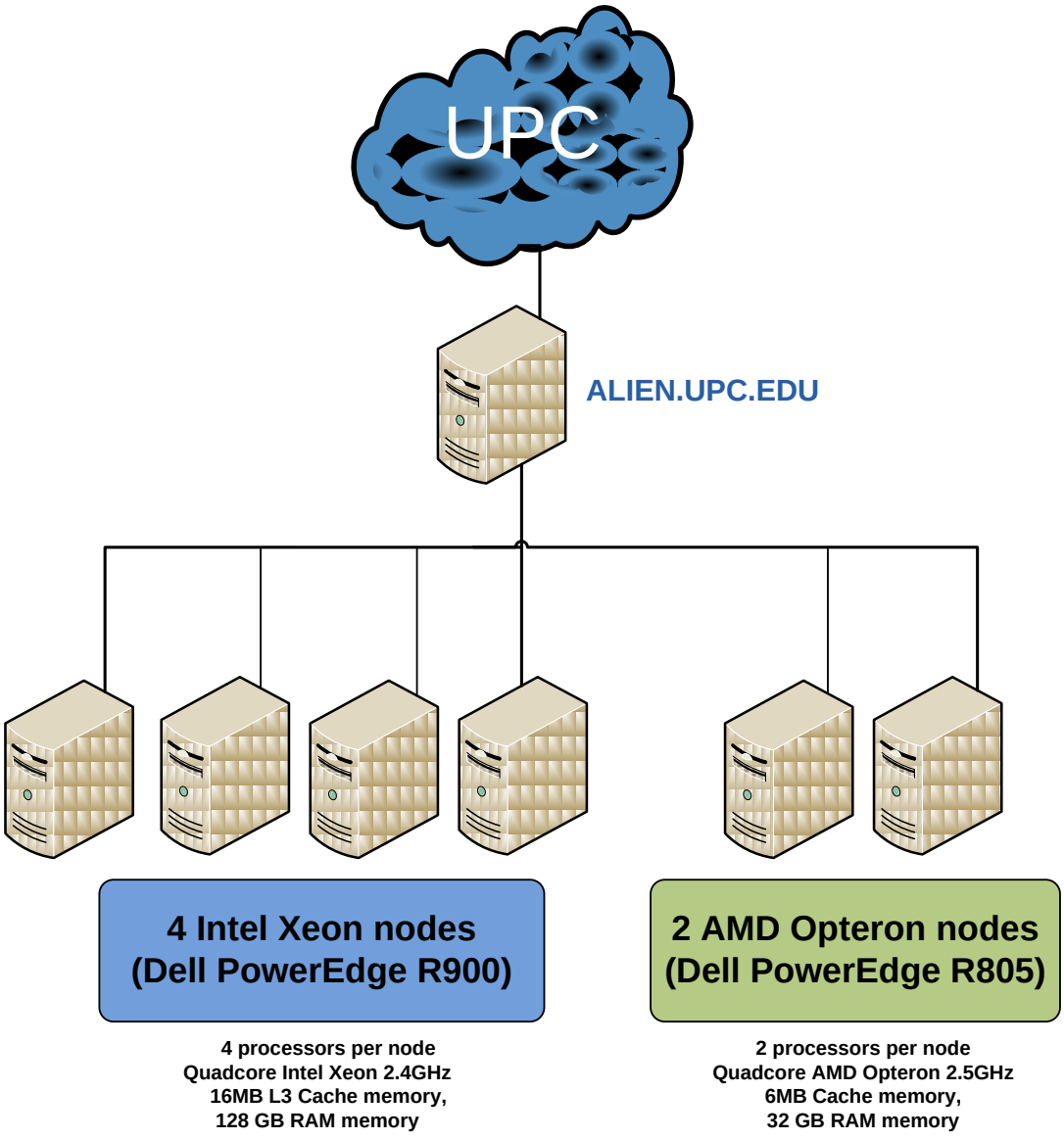
ALIEN: cluster computing

ALIEN is a high performance cluster formed by a set of Dell computers. It is a multitasking and multiuser environment with a Red Hat operative system using the set of tools known as cluster Oscar. This cluster is oriented for those users that execute programs with high performance requirements of memory as well as processor time.

One access node and six nodes for calculation form this cluster. The six calculation nodes are the ones that will execute the jobs the users send. Four of them have Intel processors meanwhile the other two have AMD processors. Adding all of them this cluster provides 64 Intel cores and 16 AMD.

An overview of the implementation for Alien and its characteristics is as follows:



Pictures of ALIEN cluster

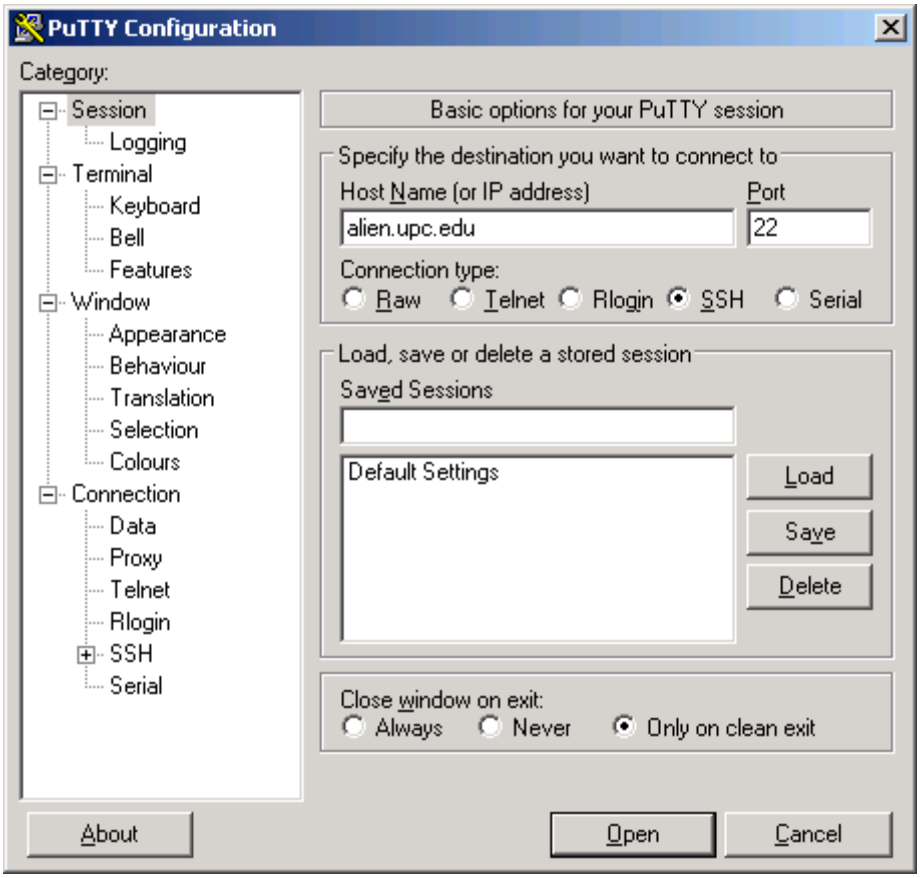
Accessing to ALIEN

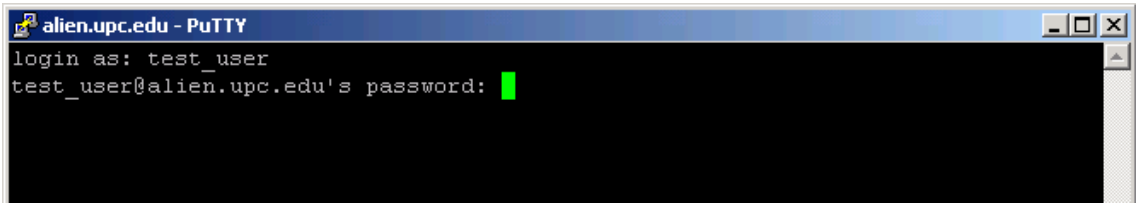
In order to gain access to the cluster it is required to ask for an account in the TIC Services Area of the ETSECCPB; a username and a password will be given and once you have this information, the connection will be using any tool that supports **`secure shell`** (**`ssh`**).

Login

Once the access account has been created, it is required to get connected every time you need to execute any job. It is suggested the use of puTTY, a tool that allows to do so configuring it as follows:

Host name	alien.upc.edu
Port number	22
User name	assigned_user
Password	assigned_password



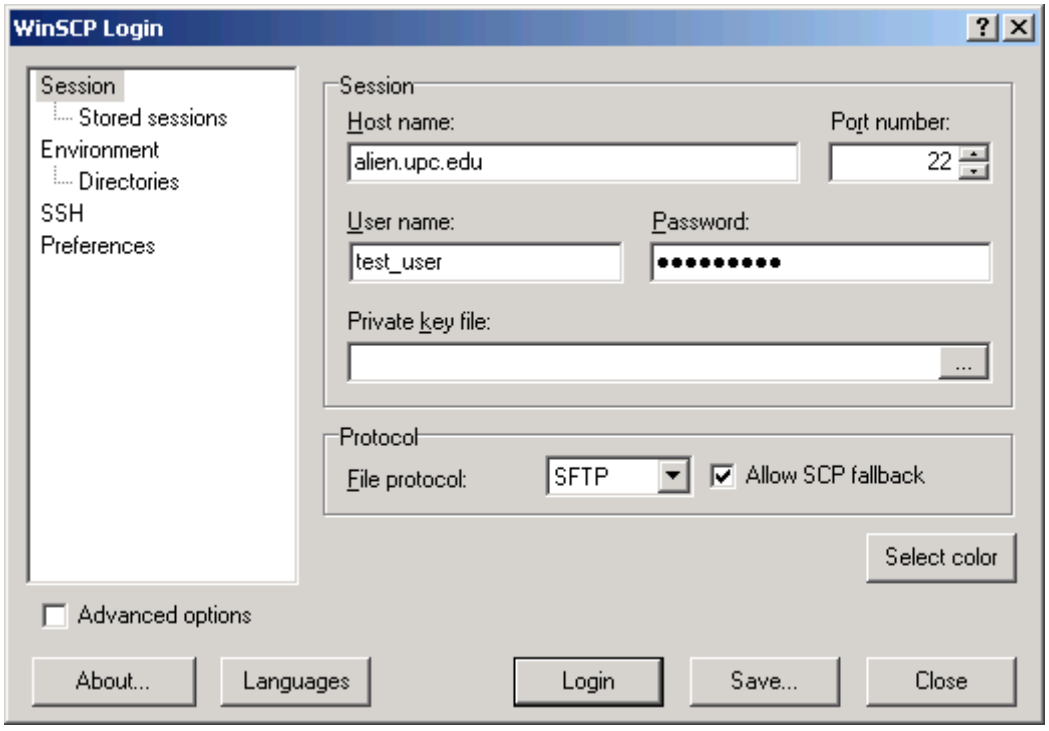


For more detailed information take a look to the following document:

http://www.camins.upc.edu/pdf/pdf-serveis/recursos-per-a-la-recerca/Manual%20de%20PuTTY.pdf

File Transfer

In order to get access to ALIEN and transfer files it is recommended to use a tool such as winSCP configuring it as follows.



For more detailed information take a look to the following document:

http://www.camins.upc.edu/pdf/pdf-serveis/recursos-per-a-la-recerca/manual-de-winscp

Disk space management

It is assigned to each user a personal space in the hard disk to store his/her files he/she is working on. This personal directory has the following *path*:

`/users/DEPARTAMENTO/username`

Initially, the hard disk quota is 1GB for storage; however in case it is required to increase it is possible by requesting it in the TIC Services Area of the ETSECCPB.

Additionally it has been created a temporary disk space useful for *scratch* as well as temporary data storage; it is managed as follows:

`/scratch/dia1` every **first day of the month**, all its content is **permanently deleted**

`/scratch/dia15` every **day 15 of the month**, all its content is **permanently deleted**

It is suggested to use the temporary */scratch* directory to store the results of the executed jobs sent to the batch queues since this directory does not have any quota restriction for users.

LINUX and basic commands

These are some of the basic commands in a Unix environment you will find useful by the time you work on it.

Command	Description	Example
pwd	shows the current working directory	`$ pwd`
mkdir	creates a new directory	`$ mkdir directory`
cd	changes from one directory to the one specified	`$ cd directory`
ls	lists the files and directories under the current directory	`$ ls`
mv	move files or subdirectories from one directory to any other specified or renames the file or directory	`$ mv test.txt /home/directory`
cp	copies files into the specified directory	`$ cp test.txt /home/user`
rm	deletes a file NOTE: once deleted it cannot be recovered	`$ rm test.txt`
rmdir	deletes the specified directory which must be empty	`$ rmdir directory`
more, cat	allows file visualization, not modifications allowed	`$ more test.txt` `$ cat test.txt`

For more detailed information take a look to the following document:

http://www.camins.upc.edu/pdf/pdf-serveis/pdf-recursos-per-la-docencia/manual-dunix

Software

Compilers

Fortran compilers installed are the following:

COMPILER	COMMAND
Intel Fortran Compiler (11.0.074)	ifort
GNU project Fortran 77 compiler	f77 g77
GNU Fortran 95 compiler	f95 gfortran g95

C/C++ compilers installed are the following:

COMPILER	COMMAND
Intel C++ Compiler (11.0.074)	icc
GNU project C and C++ compiler	cc gcc g++

Additionally it is installed the Intel Cluster Toolkit for Linux (version 3.2) which includes the following tools:

- Intel MPI Library for Linux
- Intel Math Kernel Library
- Intel Trace Analyzer and Collector
- Intel MPI Benchmarks (IMB)
- Intel Debugger for Linux

Parallel Computing

In ALIEN, the Intel and GNU the MPI libraries (Message Passing Interface) are installed through its OpenMPI implementation. These libraries are interfaces for parallel computing designed for those programs that take advantage of the existence of multiple processors.

The following link provides a user's guide for MPI programming in Fortran:

http://www.camins.upc.edu/pdf/pdf-serveis/recursos-per-a-la-recerca/manual-mpi-fortran

Choosing between Intel or GNU compiler

a) In order to use the **Intel**'s OpenMPI compiler:

   a.1) Execute the following command; from now on, all the sessions for this user will have the Intel's compiler as default:

   ```
   switcher mpi = openmpi-intel-1.2.4
   ```

   a.2) Login again to get the corresponding environment variables loaded.
   a.3) Now all is set to compile any MPI program in Fortran or C.

b) In order to use the **GNU**'s OpenMPI compiler:

   b.1) Execute the following command; from now on, all the sessions for this user will have the GNU's compiler as default:

   ```
   switcher mpi = openmpi-1.2.4
   ```

   b.2) Login again to get the corresponding environment variables loaded.
   b.3) Now all is set to compile any MPI program in Fortran or C.

Compilation Commands

Fortran uses the following command to compile a program ready for parallelization:

`mpif77` or **`mpif90`**

Meanwhile, C uses:

`mpicc`, **`mpiCC`**, **`mpicc++`** o **`mpicxx`**

Execute the following command to display the help file regarding this commands:

```
$ command -help
```

Queue Management

ALIEN provides a queue management system oriented to send jobs for their execution due to the interactive CPU time that this job may consume is limited by the system. Therefore, to execute any job it is required to send it to any predefined queue according the program specific needs. These queues are described in detail as well as its maximum CPU time, the maximum number of simultaneous jobs allowed to be executed, memory limitations and so on.

QUEUES' CHARACTERISTICS IN ALIEN

It has been defined a set of queues for the usage of the Intel processors and another set for the usage of the AMD processors. By the time a job is sent to the queues, consider the following recommendations:

- Each user is allowed to have five programs executing simultaneously across the whole set of queues.
- Additionally, each queue imposes restrictions for the limit of jobs executing per user (consult the limits).
- It is forbidden to link jobs execution, it is to say, that the same *shell script* executes one job after another executed before.
- Feel free to contact the TIC Services Area of the ETSECCPB to ask for access to restricted queues or in the particular case you need to increase the limits imposed for each queue.

Queues' characteristics could be modified regarding its maximum number of jobs and the memory limits according the users' needs, the computers' workload, among other circumstances, therefore consult the following link to have the most up to date characteristics: http://www.camins.upc.edu/serveis/recursos-recerca/cues-de-batch

The following table reflects the queues' characteristics for the **Intel nodes**. This information has been retrieved by the time this tutorial has been created, consult the most up to date information at the link mentioned before.

PUBLIC QUEUES FOR INTEL NODES

Queue's name	Maximum CPU time	Maximum number of jobs	Maximum number of simultaneous (per user)	Memory limits
iqShort	8 hours	16	3	8 GB
iqMedium	24 hours	16	3	4 GB
iqLarge	144 hours	16	3	4 GB
iqEterna	288 hours	8	2	8 GB

It has been defined two additional queues that offers more memory for those jobs that really require it, it has been named as **iqMemory** and the other one, called **iqParellel** useful for programs prepared to be parallelized using a maximum of 4 processors. The access to these queues is restricted, therefore, you have to request it.

RESTRICTED ACCESS QUEUES FOR INTEL NODES

Queue's name	Maximum CPU time	Maximum number of jobs	Maximum number of simultaneous (per user)	Memory limits
iqMemory	144 hours	4	1	32 GB
iqParallel	96 hours	4	1	4 GB

The following table describes the characteristics of the **AMD** nodes' queues where the users are allowed to send serial jobs, that is to say, not for parallel computing. This information has been retrieved by the time this tutorial has been created, consult the most up to date information at the link mentioned before.

PUBLIC QUEUES FOR AMD NODES

Queue's name	Maximum CPU time	Maximum number of jobs	Maximum number of simultaneous (per user)	Memory limits
aqShort	8 hours	4	2	4 GB
aqMedium	24 hours	4	2	2 GB
aqLarge	144 hours	2	1	2 GB
aqEterna	288 hours	2	1	4 GB

It has been defined two additional queues that offers more memory for those jobs that really require it, it has been named as **aqMemory** and the other one, called **aqParallel** useful for programs prepared to be parallelized using a maximum of 4 processors. The access to these queues is restricted, therefore, you have to request it.

RESTRICTED ACCESS QUEUES FOR AMD NODES

Queue's name	Maximum CPU time	Maximum number of jobs	Maximum number of simultaneous (per user)	Memory limits
aqMemory	144 hours	2	1	8 GB
aqParallel	96 hours	2	1	2 GB

SENDING JOBS TO THE QUEUE

The **qsub** command allows you to launch jobs to a defined queue using the following syntax:

```
qsub -q queue_name script
```

It is not possible to send a program by itself, for this purpose it is required to make a **shell script**. This script would look like the following:

```
#!/bin/sh
path/program_to_execute
```

Where **path** is the full name of the directory where the program to execute is.

Each job sent to the queue will generate two files: an error file and the one that contains the obtained results.

The name of these files would be, for instance:

script_name.e<job_identifier>

script_name.o<job_identifier>

That is to say, if the script sent is test**.sh** and the job identifier is **57**, the generated files are **test.sh.e57** and **test.sh.o57**; these two files are generated in the current directory where the **qsub** command was launched.

SENDING PARALLEL JOBS TO THE QUEUE

As mentioned previously, in order to send jobs for parallel execution it must be used the queues designed for this purpose have to be used:

- iqParallel : allows to execute a job for parallel computing using 4 Intel processors
- aqParallel : allows to execute a job for parallel computing using 4 AMD processors

The command used will be the following:

```
qsub -q iqParallel parallel_ script
```

The *parallel_ script* would be for instance:

```
#!/bin/sh
#PBS -j oe
cd path
mpirun –np 4 path/program_to_execute
```

Where **path** is the full name of the directory where the program to execute is.

CONSULTING JOBS IN THE QUEUE

The **qstat –a** command allows you to track the status of the jobs that are currently in the queue.

DELETING JOBS FROM THE QUEUE

The **qdel** command allows you to delete the tasks that are being executed currently or those that are waiting for execution. Its syntax is as follows:

```
qdel job_identifier
```

Scientific Software

Listed below are the scientific programs installed in ALIEN.

ABAQUS

Abaqus is a calculation program for finite elements analysis. It is oriented to solve problems such as solid mechanics, linear and non linear problems in static and dynamic ranges.

Abaqus/Standard and Abaqus/Explicit are available for the analysis and Abaqus/CAE for data preprocessing and post processing. Just as all the licenses, the ones we have are limited, for that reason we encourage you to make responsible and moderate use of them. It is recommended **not to send more than two jobs per user using Abaqus/Standard and Abaqus/Explicit**, and **just one job for Abaqus/CAE**. If the user exceeds these limits, the administrator is allowed to delete one of them without any previous warning.

LAUNCHING JOBS TO THE QUEUES USING ABAQUS

```
abaqus job=filename [.inp] queue=queue_name
```

EXECUTE THE POST PROCESS

To execute Abaqus/CAE and have a visualization or post processing of the obtained data, it is required an X server emulator such as Xwin32 or Cygwin, to get the visualization window. Once opened, execute:

```
abaqus cae
```

AVS

AVS/Express is a high performance tool for visualization recommended by the time a set of functionalities of data visualization or data analysis is required. It provides a last generation technology for the creation of advanced graphics applications, image processing, and data visualization among others. Its object oriented visual programming allows you to develop visual analysis applications.

It is required an X server emulator. Once opened, execute:

```
vxp
```

IMSL LIBRARIES (FORTRAN Y C)

FORTRAN, INTEL COMPILER

ALIEN by default loads the Intel Fortran Numerical Library version 6.0. Fortran libraries are divided into MATH/LIBRARY that includes Special Functions and STAT/LIBRARY.

The user's guide as well as the details about these libraries could be consulted in **/opt/vni/imsl/fnl600/help**.

The following command compiles and links a program:

```
$FC -o <executable_filename> $FFLAGS <compilable_filename> $LINK_FNL
```

Within the "**Using the IMSL Fortran Numerical Library**" section of the document located in **/opt/vni/imsl/fnl600/lxin100e64/notes/README** it is described in deep all the information you may need to compile Fortran applications.

There are also example programs within the **/opt/vni/imsl/fnl600/lnxin100e64/examples** directory.

C, INTEL COMPILER

ALIEN by default loads the Intel C Numerical Library.

Full documentation for these libraries is located in: **/opt/vni/imsl/cnl700/help**.

The following command compiles and links a program:

```
$CC -o <executable_filename > $CFLAGS <compilable_filename > $LINK_CNL
```

Within the ""**Using IMSL C Numerical Library**" section of the document located in **/opt/vni/imsl/cnl700/rdhin101e64/notes/README** it is described in deep all the information you may need to compile C applications.

There are also example programs within the **/opt/vni/imsl/cnl700/rdhin101e64/examples** directory.

C, GNU COMPILER

Since ALIEN loads by default Intel libraries for C, if you want to use the libraries for the GNU compiler the following script has to be executed:

```
./opt/vni/imsl/cnl700/rdhgc411e64/bin/cnlsetup.sh
```

MATLAB

MATLAB is a mathematic software that offers its own programming language, M language, integrated with its development environment.

Just as the programs mentioned before, a shell script is required to send jobs to the queue. This shell script has to have the following content, where *program_name* would be the filename of the Matlab file we want to get executed:

```
#!/bin/bash
#PBS -S /bin/bash

# PBS script for MATLAB batch job
# If this file is "matlab.pbs", submit with "qsub matlab.pbs"

# Check on some basics:

echo "Running on host: " `hostname`
echo "Changing to directory from which PBS script was submitted."

cd $PBS_O_WORKDIR

echo "Current working directory is now: " `pwd`
echo "Starting MATLAB at `date`"

/usr/local/bin/matlab -nojvm -nodisplay -nosplash < program_name.m > program_name.out


echo "MATLAB run completed at `date`"
```