NLHPC
National Laboratory for High Performance Computing Chile

Introduction to Leftraru-Guacolda Cluster

Goals

- Accessing to cluster Guacolda-Leftraru
  - SSH
  - Infrastructure and resources
- Uing Slurm
  - Parameters
  - Interactive and non interactive tasks
  - Getting information about our tasks
  - Monitoring our tasks
- Using software through modules
  - Searching for software and their versions
  - Loading and using modules in our tasks
- Software scaling
- Practical exercises

NLHPC
National Laboratory for High Performance Computing Chile

Exercises and you

This course will encourage you to try some exercises and sharing your knowledge and ideas to solve each problem.

- Please, ask your questions during the whole class
- Exercises will be solved in groups
- Each exercise will require to
  - Look for a solution and share your knowledge with your classmates
  - Share your screen
  - Explain the outcome of each exercise
- Users accounts will be assigned by group. Check the chat and use the credentials to access to our cluster.

NLHPC
National Laboratory for High Performance Computing Chile

Infrastructure

Login/debug node (gn)

Debug partition

- 4 Nodes
  - Intel(R) Xeon(R) CPU E5-2660
  - 20 CPUs
  - 59 GB RAM

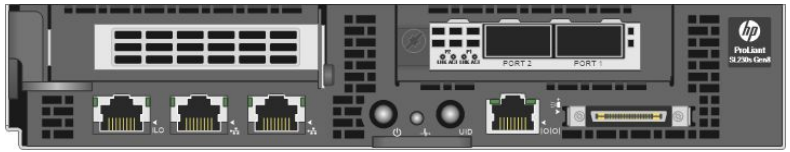
- Time limit: 30 minutes
- Using as main access and compilation tests

- **Total resources:**
- 80 CPUs
- 236 GB RAM

NLHPC
National Laboratory for High Performance Computing Chile

Infrastructure - Leftraru

Slims node(cn)



Slims partition

- 132 Nodes
  - Intel(R) Xeon E5-2660 v2
  - 20 CPUs
  - 46 GB RAM

- Time limit: 30 days
- **Default partition**

- **Total resources:**
- 2.640 CPUs
- 6.072 GB RAM

NLHPC
National Laboratory for High Performance Computing Chile

Infrastructure - Leftraru

General node (sn)

General partition

- 48 nodes
  - Intel(R) Xeon Gold 6152
  - 44 cores
  - 187 GB RAM DIMM DDR4
- Time limit: 30 days
- **Total resources:**
  - 2.112 CPUs
  - 8.976 GB RAM

NLHPC
National Laboratory for High Performance Computing Chile

Infrastructure - Leftraru

Largemem node (fn)

Largemem partition

- 9 nodes
  - Intel(R) Xeon Gold 6152
  - 44 cores
  - 765 GB RAM

- For tasks requiring 192G+ RAM
- Time limit: 30 days

- **Total resources:**

- 396 CPUs
- 6.885 GB RAM

NLHPC
National Laboratory for High Performance Computing Chile

Infrastructure - Guacolda

GPU node(gn)

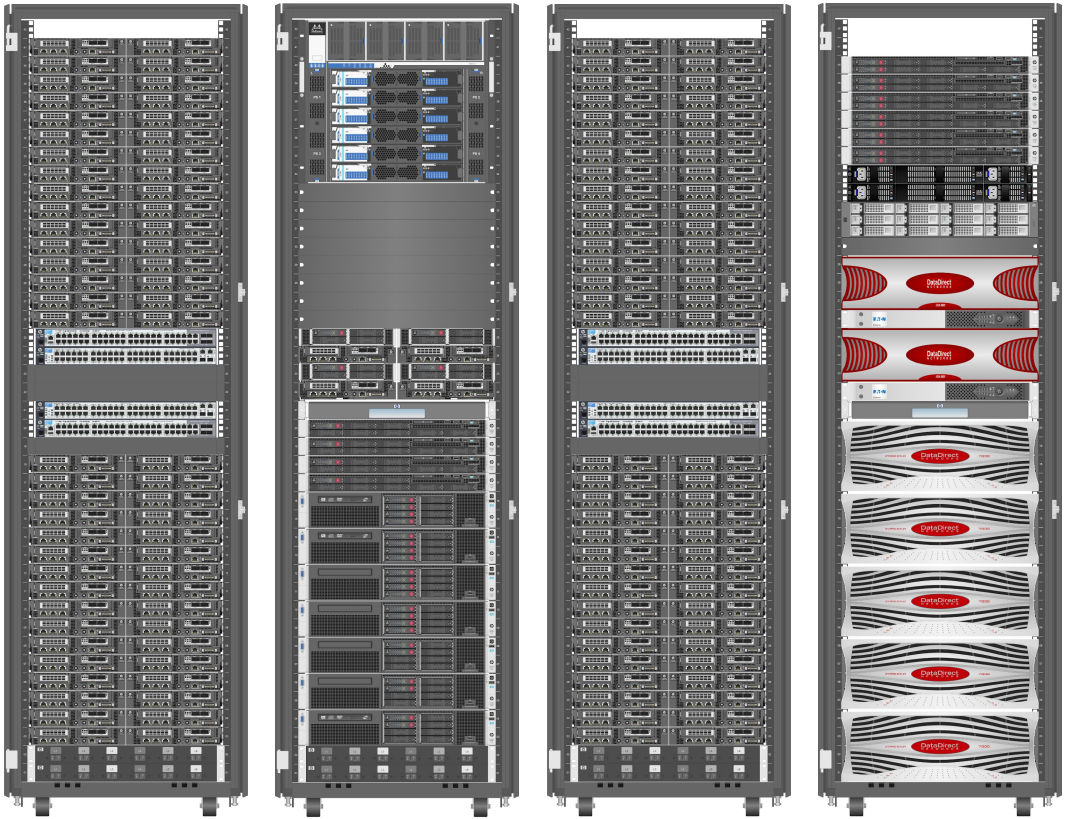
GPU partition

- 2 Nodes
  - Intel(R) Xeon Gold 6152
  - 44 cores
  - 187 GB RAM
  - 2 NVIDIA Volta V100 each node
    - 16GB
    - 5120 CUDA cores
- For tasks requiring GPU processing
- Time limit: 30 days
- **Total resources:**

- 88 CPUs
- 374 GB RAM
- 20.480 CUDA cores

NLHPC
National Laboratory for High Performance Computing Chile

NLHPC Infrastructure

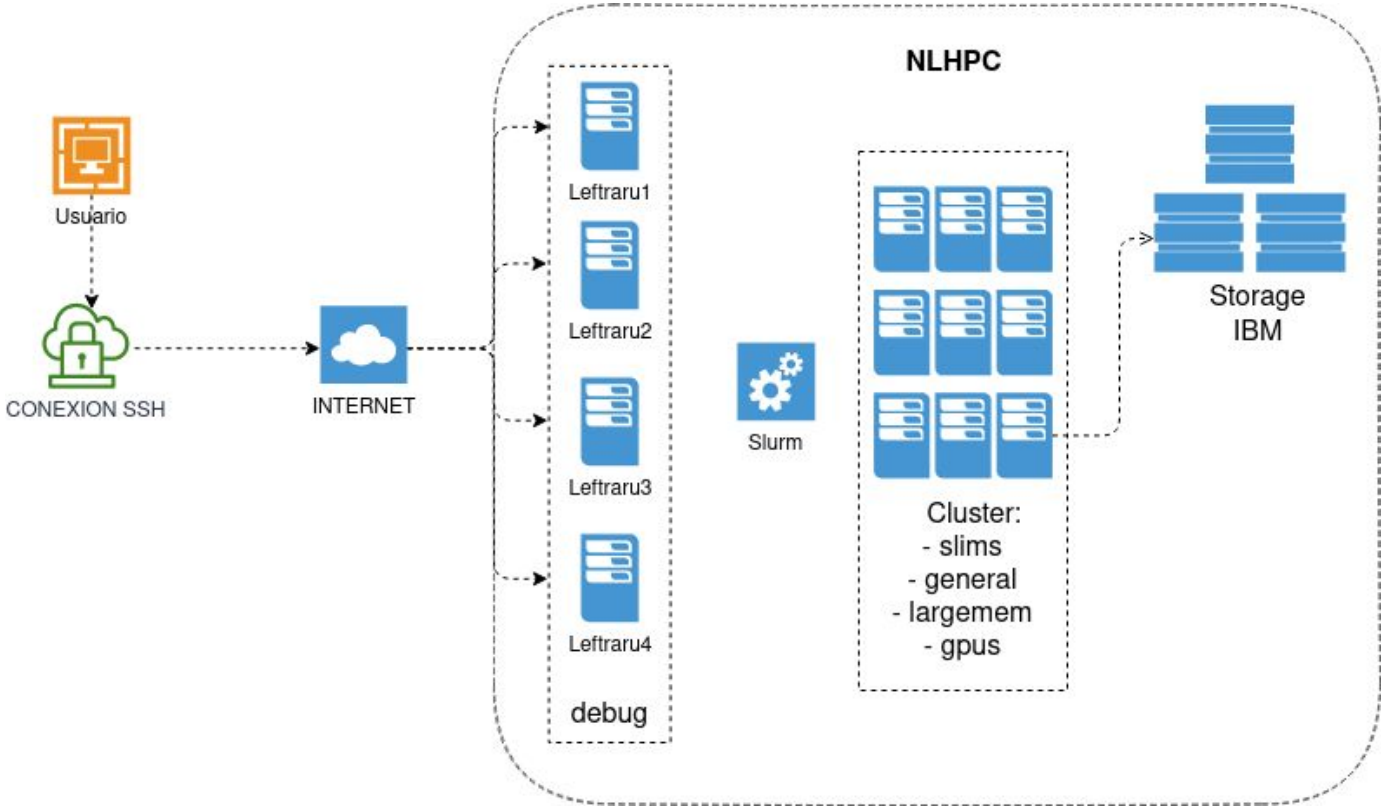


- 266 TFlops
- 5236 cores
- 191 nodes
- 4 PB of storage - IBM Spectrum Scale
- LAN Infiniband FDR 56Gbps

https://wiki.nlhpc.cl/Hardware_Disponible

NLHPC
National Laboratory for High Performance Computing Chile

Accessing the Cluster

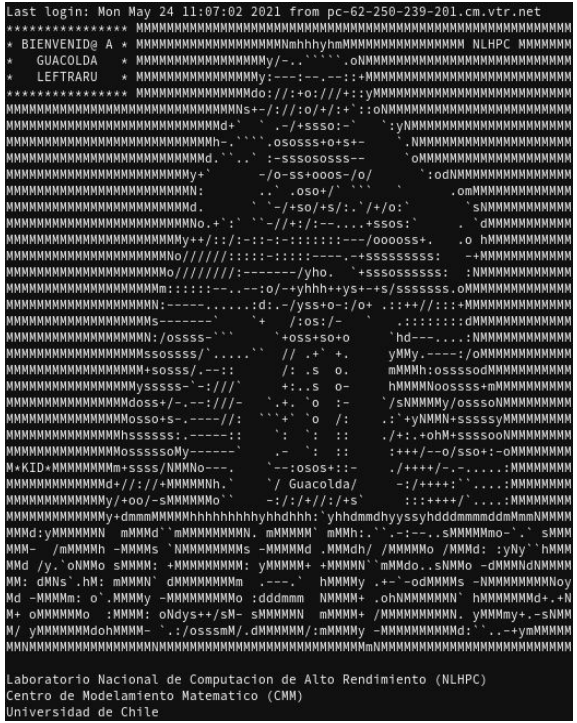


NLHPC
National Laboratory
for High Performance
Computing
Chile

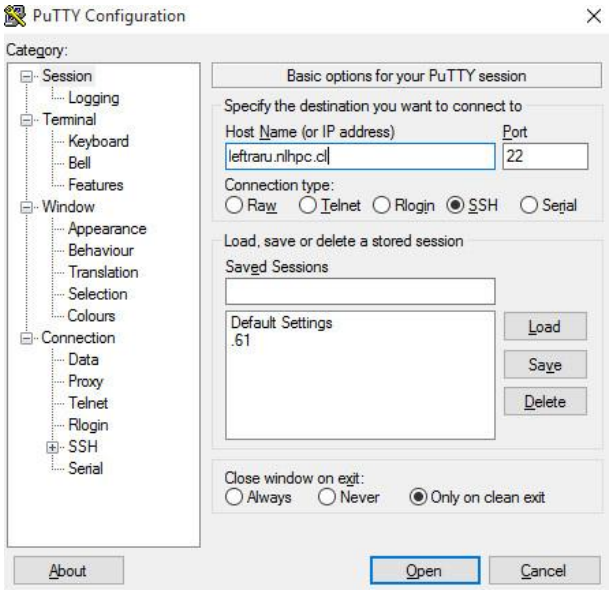
Accessing the Cluster

SSH Protocol
Host: leftraru.nlhpc.cl

Linux, macOS:
`ssh usuario@leftraru.nlhpc.cl`



Windows:



NLHPC
National Laboratory for High Performance Computing Chile

SSH Keys

Genearting SSH Keys

```
[dbowman@HAL ~] ssh-keygen -t ed25519
Enter file in which to save the key (/home/dbowman/.ssh/id_ed25519)
Enter passphrase (empty for no passphrase)
```

Copying SSH Keys

```
[dbowman@HAL ~] ssh-copy-id dbowman@leftraru.nlhpc.cl
```

Login into the cluster

```
[dbowman@HAL ~] ssh dbowman@leftraru.nlhpc.cl
```

SSH keys grants an extra layer of security.

NLHPC
National Laboratory for High Performance Computing Chile

What is SLURM?

- Resource manager
- Manages resources from partitions under Leftraru and Guacolda.
- Manages running and pending tasks in the cluster.
- Generate resources reservations
- Our configuration let the users run their tasks up to 30 days

Getting information from partitions

- ***sinfo***: show partitions and states

```
[dbowman@leftraru1 ~]# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
slims*       up   infinite      1  drain cn037
slims*       up   infinite     12    mix cn[023-024,026,045,072-073,079,087,096,107,129,131]
slims*       up   infinite     63  alloc cn[019-020,038-044,....108-128,130]
slims*       up   infinite     55   idle cn[001-018,021-022,...082-083,086,088-090,132]
general      up   infinite      9    mix sn[002,006,014-016,021,028,030-031]
general      up   infinite     39  alloc sn[001,003-005,007-013,...032-048]
largemem     up   infinite      2    mix fn[001,007]
largemem     up   infinite      2  alloc fn[002,004]
largemem     up   infinite      5   idle fn[003,005-006,008-009]
gpus         up   infinite      2    mix gn[001-002]
debug        up   infinite      4   idle leftraru[1-4]
```

squeue: list user's tasks

Run from the terminal:

```
[dbowman@leftraru1 ~]$ squeue
  JOBID     PARTITION  NAME       USER      ST    TIME  NODES NODELIST(REASON)
4400799     slims      example    usuario   R     0:00      1           cn042
```

sacct: get status from tasks


```
[dbowman@leftraru1 ~]$ sacct -X
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
24118136	14131-DIA+	slims	users	2	RUNNING	0:0
24118147	14132-DIA+	slims	users	2	RUNNING	0:0
24118148	14133-DIA+	slims	users	2	COMPLETED	0:0
24118154	14137-DIA+	slims	users	2	COMPLETED	0:0

Getting information from one task

```
[dbowman@leftraru1 ~]# scontrol -dd show job 9160565

JobId=9160565 JobName=w.fepc-f-cnt-oo.m2
   UserId=dbowman(wxyz) GroupId=users(wxyz) MCS_label=N/A
   Priority=109951 Nice=0 Account=users QOS=88-30-std
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=0 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=04:28:03 TimeLimit=3-00:00:00 TimeMin=N/A
   SubmitTime=2017-10-09T11:25:36 EligibleTime=2017-10-09T11:25:36
   StartTime=2017-10-09T15:04:40 EndTime=2017-10-12T15:04:40 Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=slims AllocNode:Sid=leftraru4:52235
   ReqNodeList=(null) ExcNodeList=(null)
```

NLHPC
National Laboratory for High Performance Computing Chile

Send tasks to SLURM

```
[dbowman@leftraru1 ~]$ srun hostname
cn028
[dbowman@leftraru1 ~]$ sbatch <job_script>
Submitted batch job 9142401
[dbowman@leftraru1 ~]$ squeue
             JOBID     PARTITION  NAME      USER    ST     TIME  NODES   NODELIST(REASON)
           9142401     slims      pi_levqu  root     R       0:08   1           cn105
[dbowman@leftraru1 ~]$ scancel 9142401
```

Command

command output

task cancelation

slurm workload manager

NLHPC National Laboratory for High Performance Computing Chile

SLURM parameters

Parameter	Usage	Description
-J	`-J my-task`	Task name
-p	`-p slims`	Partition to use
-n	`-n 1`	Number of process
-c	`-c 20`	CPUs by process
--ntasks-per-node	`--ntasks-per-node=20`	Process grouped by node
--mem-per-cpu	`--mem-per-cpu=2300`	RAM by CPU
-o	`-o output_%j.out`	Output log file
-e	`-e errors_%j.err`	Error log file
–mail-user	`–mail-user=user@abc.xyz`	Mail to send information
–mail-type	`–mail-user=ALL`	Type of information to send by mail

slurm
workload manager

NLHPC
National Laboratory
for High Performance
Computing
Chile

Exercise 1

- Run **hostname** command under *Slim* partition:
  - With one process
  - With two similar process
  - With two process in different nodes
  - With one process and two CPU
- Which are the results?
- About the *Slim* partition
  - How many cores can be reserved by process? Why?
  - Which is the difference with the *general* partition?
  - What will happen if the task is set up with a higher number of CPU available?
- If my task doesn't specify the partition to use, what will happen? why?

A simple SBATCH script

Use your preferred text editor like nvim, vi, nano, emacs to edit your first script as follows:

```
#!/bin/bash
#SBATCH -J my_script
#SBATCH -p slims
#SBATCH -n 1
#SBATCH -c 1
#SBATCH -o output_%j.out
#SBATCH -e errors_%j.err
#SBATCH --mail-user=foo@example.org
#SBATCH --mail-type=ALL

sleep 10
```

Run your script with

```
sbatch my_script.sh
```

NLHPC
National Laboratory for High Performance Computing Chile

Exercise 2

- Create a new script to run it with **`sbatch`**, following the next specs:
  - Use the *Slim* partition
  - Use only one CPU
  - Run the command **`stress -c 1`**
- **`stress`** command will test components from our system. In this case one CPU will be used up to 100% during an unlimited time.
  - For how long this tasks will be running?
  - Which commands can you use to get information from our task?
  - How can you cancel this task?
- To know if the task will be run or will be put on hold, you should know the Cluster state. Which command is available to know the current state of the cluster?

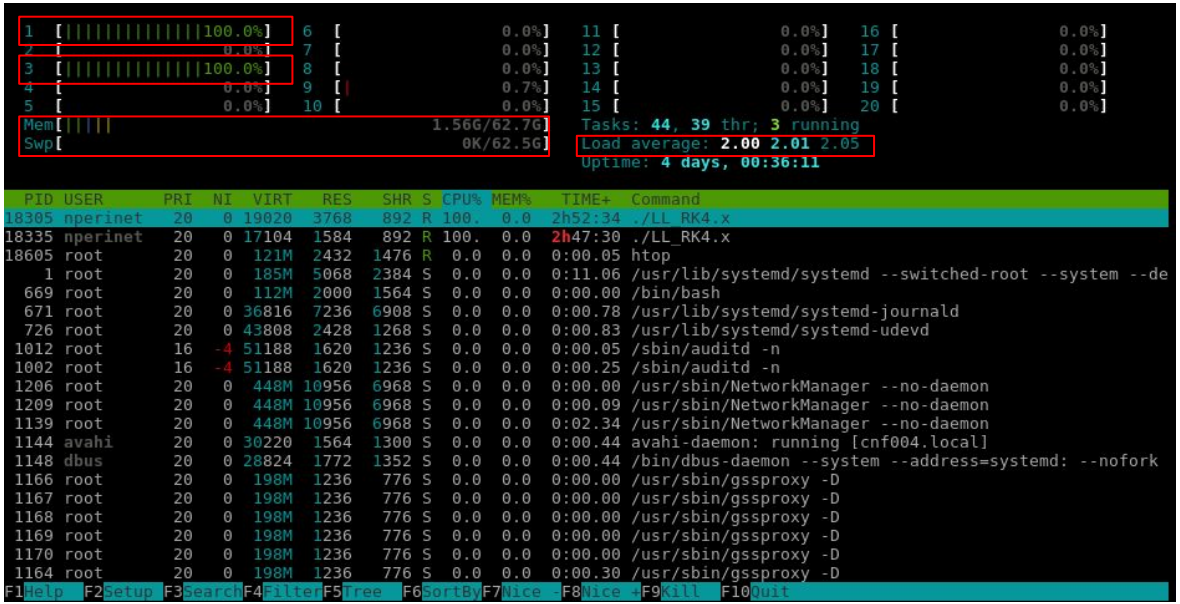
slurm
workload manager

NLHPC
National Laboratory for High Performance Computing Chile

Task monitoring with *htop*

To access to a specific node, your user must have a task running in it.

This will let you run **`htop`** and check your task.



NLHPC
National Laboratory for High Performance Computing Chile

SBATCH script generator

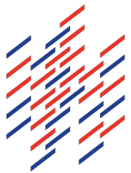
Follow the next link to fill a web form and get a copy&paste SBATCH script for our cluster.
https://wiki.nlhpc.cl/Generador_Scripts

Currently in Spanish.

NLHPC
National Laboratory for High Performance Computing
Chile

Exercice 3

- Edit a new script to run it using **`sbatch`**, follow the next requirements:
  - Use *slims* partition.
  - Request all the resources from one node.
  - Run the next command: **`stress -c 40 -t 10m`**.
- Verify in which node the task is running and then ssh into that node.
- Run the command **htop** inside the node.
- How many process are running?
- How many CPU% is using each process?
- Which changes should be applied to the script to get each process running at 100% of CPU?
- Compare the CPU usage from the initial exercise and the modified one.



RAM memory assignment

RAM limits:

- By default 1GB RAM will be assigned by CPU
- If more RAM than the available is assigned an error message will be displayed: “Exceeded job memory limit”
- Assigning RAM by core using the next parameter: **`#SBATCH --mem-per-cpu=2300`**

 # systemd-cgtop -m | grep job_id

```
Path                          Tasks   %CPU   Memory  Input/s Output/s

/                               485 4356.4    23.4G        -        -
/slurm                            -      -    14.3G        -        -
/slurm/uid_2398                   -      -    14.3G        -        -
/slurm/u...8/job_24117526         -      -    14.3G        -        -
/system.slice                     - 4354.2     8.4G        -        -
```

NLHPC
National Laboratory for High Performance Computing Chile

Exercice 4

- Edit a new script to run it using **`sbatch`**, follow the next requirements:
  - Use *slims* partition.
  - Do not assign RAM to your task.
  - Assign only one CPU.
  - Use output and error parameters.
  - Run **`stress -m 1 --vm-bytes 2048M -t 15m`**
- Check the output of your task, including STERR and STDOUT
- What is happening? Could you explain the situation?
- Modify your script and assign enough memory to run the command mentioned above

Modules System: LMOD

- Modules system lets to keep a catalog of software and several versions, letting to use them easyly
- Currently NLHPC has been using **Lmod** (https://github.com/TACC/Lmod).
- The current modul system is available for different architectures like AVX512, AVX, SSE4.2

NLHPC
National Laboratory
for High Performance
Computing
Chile

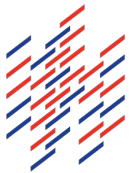
Lmod: Searching software

```
dbowman@leftraru1:/home/dbowman$ ml spider Python

----------------------------------------------------------------------------------------------------------------------
  Python:
----------------------------------------------------------------------------------------------------------------------
    Description:
      Python is a programming language that lets you work more quickly and integrate your systems more effectively.

     Versions:
        Python/2.7.15
        Python/3.7.2
        Python/3.7.3
     Other possible modules matches:
        Biopython  IPython  protobuf-python

----------------------------------------------------------------------------------------------------------------------
  To find other possible module matches execute:

      $ module -r spider '.*Python.*'

----------------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "Python" module (including how to load the modules) use the module's full
name.
  For example:

     $ module spider Python/3.7.3
```

NLHPC
National Laboratory for High Performance Computing Chile

Lmod: Loading software (and versions)

```
[dbowman@leftraru1 ~]$ ml Python/3.7.3
[dbowman@leftraru1 ~]$ ml

Currently Loaded Modules:
  1) GCCcore/8.2.0                      4) impi/2019.2.187   7) intel/2019b  10) libreadline/8.0  13) SQLite/3.27.1  16) libffi/3.2.1
  2) icc/2019.2.187-GCC-8.2.0-2.31.1    5) imkl/2019.2.187   8) bzip2/1.0.6  11) ncurses/6.1      14) XZ/5.2.4       17) Python/3.7.3
  3) ifort/2019.2.187-GCC-8.2.0-2.31.1  6) binutils/2.32     9) zlib/1.2.11  12) Tcl/8.6.9        15) GMP/6.1.2


[dbowman@leftraru1 ~]$ python -V
Python 3.7.3

[dbowman@leftraru1 ~]$ ml Python/2.7.15

The following have been reloaded with a version change:
  1) Python/3.7.3 => Python/2.7.15

[dbowman@leftraru1 ~]$ ml

Currently Loaded Modules:
  1) GCCcore/8.2.0                      4) impi/2019.2.187   7) intel/2019b  10) libreadline/8.0  13) SQLite/3.27.1  16) libffi/3.2.1
  2) icc/2019.2.187-GCC-8.2.0-2.31.1    5) imkl/2019.2.187   8) bzip2/1.0.6  11) ncurses/6.1      14) XZ/5.2.4       17) Python/2.7.15
  3) ifort/2019.2.187-GCC-8.2.0-2.31.1  6) binutils/2.32     9) zlib/1.2.11  12) Tcl/8.6.9        15) GMP/6.1.2


[dbowman@leftraru1 ~]$ python -V
Python 2.7.15
```



NLHPC
National Laboratory for High Performance Computing Chile

Exercice 5

- Download the following python script in your working directory: n-queens-problem-3.py (use **wget**)
- Edit a new **sbatch** *script* following the next criteria:
  - Configure your task to run under *slims* partition.
  - Assign only one CPU
  - Assign 2300Mb RAM to your task
  - Search and load Python 3.9.5 into your script
  - Run the python script

NLHPC
National Laboratory for High Performance Computing Chile

Disk quota

```
[dbowman@leftraru1 ~]# usoDisco

Uso de disco del usuario: dbowman

Cuota       = 200G

Utilizado   = 148.95G

% de utilización = 74.5%
```

NLHPC
National Laboratory for High Performance Computing Chile

Software Efficiency

- Related to software behaviour when it runs parallel (more than one CPU)
- Software can escalate using more than one processors [1..n]
- Efficiency is achieved when the scale is constant and above 0,5 factor
- This is important because a job will not run in half of the time if you use the double of resources
    - **Resource Efficiency usage** is the main goal
    - Optimal usage of resources will let us get or result in less time

Software Efficiency - Speedup and Efficiency

- **SpeedUp** is the metric value related to parallel processes and time execution
  - SpeedUp = Original Time / Improvement Time

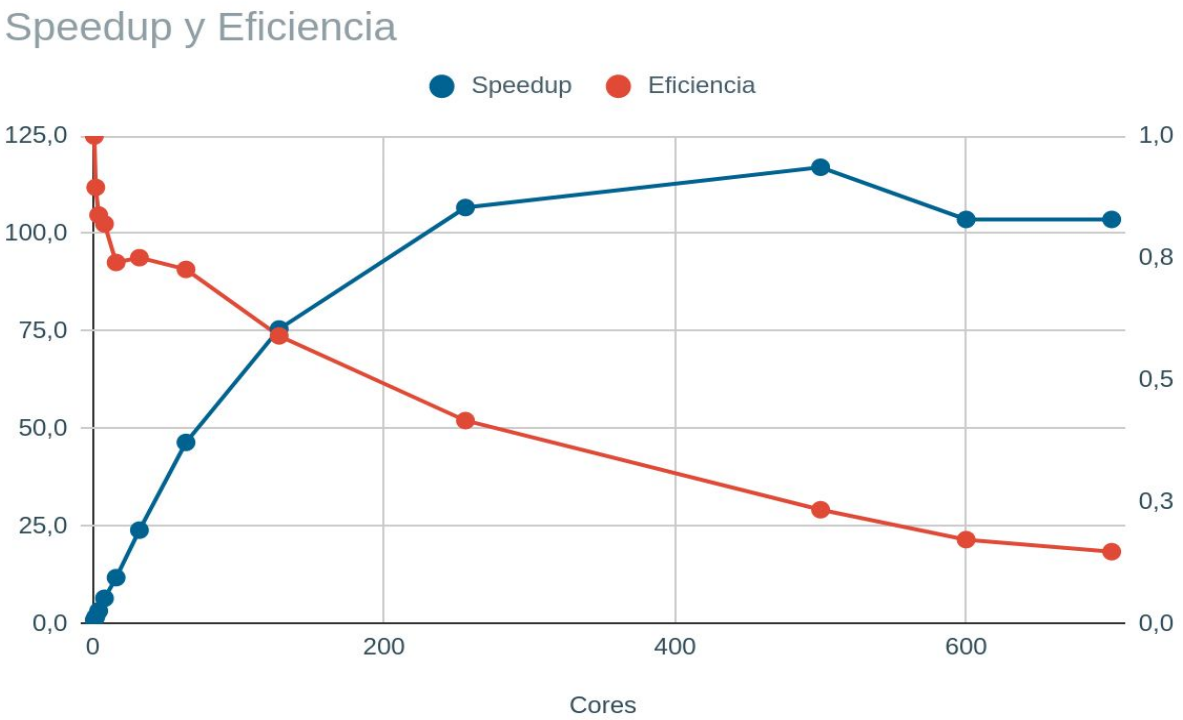
- **Efficiency** is the metric value related the **SpeedUp** divided by the amount of CPU used
  - Eficiencia = SpeedUp / CPU assigned

Procesadores	Tiempo de Ejecución	Speedup	Eficiencia
1	1:00:27	1,0	1,0
2	0:33:47	1,8	0,9
4	0:18:02	3,4	0,8
8	0:09:13	6,6	0,8
16	0:05:06	11,9	0,7
32	0:02:31	24,0	0,8
64	0:01:18	46,5	0,7
128	0:00:48	75,6	0,6
256	0:00:34	106,7	0,4
500	0:00:31	117,0	0,2
600	0:00:35	103,6	0,2
700	0:00:35	103,6	0,1

https://wiki.nlhpc.cl/Escalamiento

NLHPC
National Laboratory for High Performance Computing
Chile

Software Efficiency - Speedup and Efficiency - Graph



NLHPC
National Laboratory for High Performance Computing Chile

Using GPUS partition

- How to run a software using the GPU available in our Cluster?
  - Which options can NLHPC offer to the users?
  - Which modules are required to run software and use the GPU?

```
#!/bin/bash
#SBATCH -J gpu-example
#SBATCH -p gpus

#SBATCH -n 1
#SBATCH -c 1
#SBATCH --gres=gpu:1
#SBATCH --mem-per-cpu=4250

ml purge
ml fosscuda/2019b
ml NAMD/3.0alpha9
...
```

NLHPC
National Laboratory for High Performance Computing Chile

Links

Visit our webpage at

www.nlhpc.cl

Also we keep a public Wiki with useful information(only in Spanish):

https://wiki.nlhpc.cl/Bienvenida_NLHPC

To request user accounts, please visit:

https://solicitudes.nlhpc.cl/

Watch our current node status with our Dashboard

https://dashboard.nlhpc.cl/

If you have any inquiry, please send us an email to soporte@nlhpc.cl

NLHPC
National Laboratory
for High Performance
Computing
Chile

NLHPC
National Laboratory for High Performance Computing Chile

Thanks for your time!

www.nlhpc.cl

2023