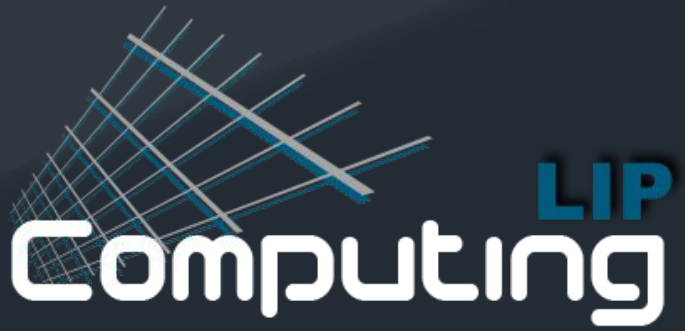


HPC: advanced level

European Commission
Cofinanciado por:
COMPETE 2020
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional
Lisb@20 20
FCT Fundação para a Ciência e a Tecnologia

Outline:

- Using slurm
  - How to retrieve job information from slurm
    - Jobs information
    - Show job accounting data
    - Stop or Cancel jobs
- How to run custom software
  - conda example

Slurm

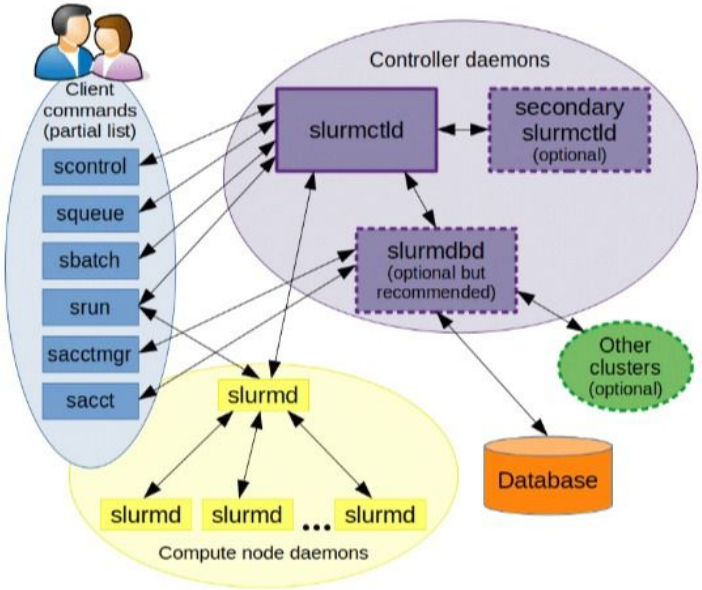
LIP Computing

Slurm

- Slurm:
  - Open Source workload manager
  - Manages resources and provide access to resources
  - arbitrates contention from users requests to available resources according to “predefined” policies”

Used at INCD since 2020

Full documentation:
https://slurm.schedmd.com/overview.html



Using slurm commands

squeue: job list information

```
[user@cirrus09 grom-cpu-1]$ squeue
JOBID    PARTITION NAME      USER      ST TIME      NODES CPUS TRES_PER_NODE NODELIST
13082168 short     grom-cpu.s user      R  0:05      1     1    N/A          hpc078
```

- We configure **squeue** in order to show only the user tasks because the number of running jobs is usually too big and it may be disturbing to read; the user can change this behavior unsetting the environment variable **SQUEUE_USERS**, to make it permanente insert the instruction on user configuration file **~/.bash_profile**:

```
[user@cirrus09 grom-cpu-1]$ unset SQUEUE_USERS
[user@cirrus09 grom-cpu-1]$ echo 'unset SQUEUE_USERS' >> ~/.bash_profile
[user@cirrus09 grom-cpu-1]$ squeue
JOBID PARTITION   NAME   USER    ST TIME  NODES CPUS TRES_PER_NODE NODELIST
11302551      fct   cpu.sh   user1  PD  0:00      1     16  N/A
12605493      fct   OF13     user2  PD  0:00      1     64  N/A
...
```

scontrol: running job details

```
[jpina@cirrus09 grom-cpu-1]$ scontrol show jobid -dd  13082168
JobId=13082168 JobName=grom-cpu.sh
   UserId=jpina(5800003) GroupId=csys(5800000) MCS_label=N/A
   Priority=49758 Nice=0 Account=csys QOS=normal
   JobState=FAILED Reason=NonZeroExitCode Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=127:0
   DerivedExitCode=0:0
   RunTime=00:00:06 TimeLimit=1-06:00:00 TimeMin=N/A
   SubmitTime=2023-10-16T11:27:39 EligibleTime=2023-10-16T11:27:39
   AccrueTime=2023-10-16T11:27:39
   StartTime=2023-10-16T11:27:40 EndTime=2023-10-16T11:27:46 Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-10-16T11:27:40 Scheduler=Main
   Partition=short AllocNode:Sid=cirrus09:199081
```

scontrol: running job details(cont.)

```
ReqNodeList=(null) ExcNodeList=(null)
 NodeList=hpc078
 BatchHost=hpc078
 NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
 TRES=cpu=1,mem=5000M,node=1,billing=1
 Socks/Node=* NtasksPerN:B:S:C=1:0:*:1 CoreSpec=*
 JOB_GRES=(null)
   Nodes=hpc078 CPU_IDs=0 Mem=5000 GRES=
 MinCPUsNode=1 MinMemoryCPU=5000M MinTmpDiskNode=0
 Features=el8 DelayBoot=00:00:00
 Reservation=tut
 OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
 Command=/users3/csys/jpina/grom-cpu-1/grom-cpu.sh
 WorkDir=/users3/csys/jpina/grom-cpu-1
 StdErr=/users3/csys/jpina/grom-cpu-1/slurm-13082168.out
 StdIn=/dev/null
 StdOut=/users3/csys/jpina/grom-cpu-1/slurm-13082168.out
 Power=
```

sstat: status information of a running job

```
[user@cirrus09 grom-gpu]$ sstat --format=AveCPU,AvePages,AveRSS,AveVMSize,JobID -j 13082190 --allsteps
   AveCPU   AvePages     AveRSS  AveVMSize JobID
---------- ---------- ---------- ---------- ------------
 00:00:00          0         76K    217084K 13082190.ex+
 00:00:00          0        436K    222664K 13082190.ba+
```

sacct: account info. of completed job

```
[user@cirrus09 grom-gpu]$ sacct -j 13082190 --format=JobID,JobName,MaxRSS,Elapsed
JobID             JobName   MaxRSS Elapsed
------------------ -------------- ---------- ----------
13082190          prod01              00:00:11
13082190.ba+  batch            436K 00:00:11
13082190.ex+  extern            76K 00:00:11
13082190.0    clean tmp          0 00:00:00
```

seff: simple accounting of completed jobs

```
[user@cirrus09 grom-gpu]$ seff 13082190
Job ID: 13080911
Cluster: production
User/Group: afernandes/cosmo
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 16:14:02
CPU Efficiency: 99.98% of 16:14:15 core-walltime
Job Wall-clock time: 16:14:15
Memory Utilized: 566.80 MB
Memory Efficiency: 3.54% of 15.62 GB
```

scancel: cancel jobs

```
[user@cirrus09 grom-gpu]$ scancel 13082190
```

sinfo: nodes and partitions information

```
[user@cirrus09 ~]$ sinfo
PARTITION   AVAIL  TIMELIMIT  NODES  STATE NODELIST
medusa         up   infinite      3   alloc  hpc[049,059,066]
fast_medusa    up   infinite      2    idle  gorgon[201,301]
bob            up 4-00:00:00     35    idle  hpc[084-118]
hpc*           up 4-00:00:00      2    resv  hpc[078-079]
hpc*           up 4-00:00:00      5     mix  hpc[046-047,070,074-075]
hpc*           up 4-00:00:00     10   alloc  hpc[048,066-069,071-073,076-077]
teste          up   infinite      1    drng  hpc062
teste          up   infinite      1     mix  hpc063
gpu            up 4-00:00:00      1    drng  hpc062
```

Careful show all queues
(even those were not allowed)

sinfo: nodes and partitions information (cont.)

[user@cirrus09 ~]$ **sinfo -o "%10P %.6D %.11T %11G %N"** **(show GPU in a readable manner)**

```
PARTITION   NODES       STATE GRES        NODELIST
medusa          1       mixed (null)        hpc066
medusa          2  allocated (null)        hpc[049,059]
fast_medus      2        idle (null)        gorgon[201,301]
bob            35        idle (null)        hpc[084-118]
hpc*            2    reserved (null)        hpc[078-079]
hpc*           13       mixed (null)        hpc[047,066-077]
hpc*            1   allocated (null)        hpc048
hpc*            1        idle (null)        hpc046
gpu             1     drained gpu:v100s:1 hpc062
gpu             1    reserved gpu:a100:1( hpc064
gpu             1       mixed gpu:v100s:1 hpc060
gpu             1       mixed gpu:t4:2      hpc061
gpu             1       mixed gpu:t4:2(S:   hpc063
gpu             1       mixed gpu:a100:1( hpc065
```

pestat: detailed resources and job information

```
[user@cirrus09 ~]$ pestat -p gpu
Print only nodes in partition gpu
Hostname Partition Node  Num_CPU  CPUload  Memsize  Freemem  GRES/                    Joblist
                   State Use/Tot           (MB)     (MB)  node                     JobId User GRES/job ...
 hpc060        fct+   mix   81   96    81.21   512000   334096  gpu:v1 00s:1,gpu:t4:1 12605492 jgadelho  13082087 fmartins  13084780 fmartins
 hpc061         fct   mix   80   96    66.40*  512000   387650  gpu:t4:2              12605491 jgadelho  13025139 apontes
 hpc062         gpu drain*   0   96     0.00   512000   222459  gpu:v100s:1(S:0),gpu:t4:1(S:1)
 hpc063         gpu   mix    4   96     7.73*  512000   246004  gpu:t4:2(S:0)         13080345 gabriel  13080343 gabriel
 hpc064         gpu  resv*   0   96     0.03   515316   368511  gpu:a100:1(S:0)
 hpc065         gpu   mix    1   96     2.05*  515316   360230  gpu:a100:1(S:0)       13073532 balmeida
```

sprio: pending jobs scheduled priority

```
[user@cirrus09 ~]$ sprio
```

JOBID	PARTITION	PRIORITY	SITE	AGE	FAIRSHARE	PARTITION	QOS	TRES
11302551	fct	90384	0	10000	68731	1	10000	cpu=833,mem=820
11302725	fct	89558	0	10000	68731	1	10000	cpu=417,mem=410
11302726	fct	89558	0	10000	68731	1	10000	cpu=417,mem=410
12605431	hpc	101369	0	10000	24159	1	5000	cpu=25000,mem=37209
12605493	fct	17486	0	10000	841	1	0	cpu=3333,mem=3311
12605500	fct	17257	0	10000	612	1	0	cpu=3333,mem=3311
13073540	gpu	608437	0	1520	5581	1000	0	cpu=174,mem=162,gres
13073542	gpu	608437	0	1520	5581	1000	0	cpu=174,mem=162,gres

…

Using slurm command alias

Useful command alias

- We defined some alias for users convenience:
  - **ninfo**: sinfo format presenting some useful node resources

```
[user@cirrus08 ~]$ ninfo -p gpu
NODELIST   PARTITION CPUS MEMORY FREE_MEM CPU_L STATE GRES              TIMELIMIT  ACTIVE_FEATURE  REASON
   hpc060       gpu   96 512000   334108 81.13   mix gpu:v100s:1,gpu 4-00:00:00 el7,epyc2,gpu    none
   hpc061       gpu   96 512000   388431 65.94   mix gpu:t4:2        4-00:00:00 el7,epyc2,gpu    none
   hpc062       gpu   96 512000   222422  0.09 drain gpu:v100s:1(S:0 4-00:00:00 el8,epyc2,gpu    tutorial
   hpc063       gpu   96 512000   245915  8.85   mix gpu:t4:2(S:0)   4-00:00:00 el8,epyc2,gpu    none
   hpc064       gpu   96 515316   368476  0.19  resv gpu:a100:1(S:0) 4-00:00:00 el8,epyc3,gpu    none
   hpc065       gpu   96 515316   360200  2.18   mix gpu:a100:1(S:0) 4-00:00:00 el8,epyc3,gpu    none
```

Useful command alias(cont.)

- **ginfo**: sinfo format with emphasis on resources presentation, e.g. GPU

```
[user@cirrus08 ~]$ ginfo -p gpu
 NODELIST PARTITION CPUS MEMORY FREE_MEM CPU_L STATE  TIMELIMIT  ACTIVE_FEATURE  GRES
    hpc060       gpu   96 512000   318915 83.74   mix  4-00:00:00 el7,epyc2,gpu      gpu:v100s:1,gpu:t4:1
    hpc061       gpu   96 512000   387851 66.25   mix  4-00:00:00 el7,epyc2,gpu      gpu:t4:2
    hpc062       gpu   96 512000   222444  0.00 drain  4-00:00:00 el8,epyc2,gpu      gpu:v100s:1(S:0),gpu:t4:1(S:1)
    hpc063       gpu   96 512000   246133  8.76   mix  4-00:00:00 el8,epyc2,gpu      gpu:t4:2(S:0)
    hpc064       gpu   96 515316   368566  0.12  resv  4-00:00:00 el8,epyc3,gpu      gpu:a100:1(S:0)
    hpc065       gpu   96 515316   360281  2.03   mix  4-00:00:00 el8,epyc3,gpu      gpu:a100:1(S:0)
```

Useful command alias(cont.)

- **mprio**: sprio command format with a friendly view, particularly present the list on a reverse priority order

```
[user@cirrus08 ~]$ mprio
     JOBID PARTITION  PRIORITY       USER    AGE  FAIRSHARE  PARTITION   QOS                                 TRES
  13073540       gpu    608444   balmeida   1528       5581       1000     0 cpu=174,mem=162,gres/gpu=100000,gres/gpu
  13073542       gpu    608444   balmeida   1528       5581       1000     0 cpu=174,mem=162,gres/gpu=100000,gres/gpu
  13085480       gpu    111159  hmonteiro    163       1988       1000  5000 cpu=347,mem=2661,gres/gpu=100000
  13085625       gpu    110152  hmonteiro    154       1988       1000  5000 cpu=347,mem=1663,gres/gpu=100000
  13080356       gpu    105661    gabriel    778       3211       1000     0 cpu=347,mem=325,gres/gpu=100000
  12605431       hpc    101369   eribeiro  10000      24159          1  5000 cpu=25000,mem=37209
  11302551       fct     90384    bvictor  10000      68731          1 10000 cpu=833,mem=820
…
```

Useful command alias(cont.)

- **macct**: sacct command format with extended job details

```
[user@cirrus08 tensorflow-cpu]$ macct -j 13083408
JobID           JobName Partition User Account    QOS Res  AllocCPUS    Elapsed   CPUTime MaxRSS MaxVMSize      State NodeList
------------ ---------- --------- ---- --------- ------ ---- ---------- ---------- ---------- ------ --------- ---------- --------
13083408        keras.sh       hpc user    group normal tut          1 00:00:06   00:00:06                 OUT_OF_ME+ hpc078
13083408.batch  batch                      group                     1 00:00:06   00:00:06      0  190912K OUT_OF_ME+ hpc078
13083408.extern extern                     group                     1 00:00:06   00:00:06      0  147896K OUT_OF_ME+ hpc078
13083408.0      clean tmp                  group                     1 00:00:00   00:00:00      0  195908K  COMPLETED hpc078
```

Submit a job with memory request

Submit a job with memory request

We will train a machine learning model on a CPU and on a GPU:

- For the CPU run go to **tensorflow-cpu** directory of support material and examine the submission script **keras.sh**:

```
[user@cirrus08 ~] git clone https://gitlab.a.incd.pt/hpc/tutorial/module0.git
[user@cirrus08 ~]$ cd ~/modulo0/tensorflow-cpu
[user@cirrus08 tensorflow-cpu]$ cat keras.sh
#!/bin/bash
#SBATCH --partition=hpc
#SBATCH --ntasks=1

module load tensorflow

python keras_example_small.py
```

Submit a job with memory request(cont.)

- Submit and wait for completion:

```
[user@cirrus08 tensorflow-cpu]$ sbatch keras.sh
Submitted batch job 13083408

[user@cirrus08 openmpi]$ squeue
JOBID    PARTITION NAME     USER ST TIME NODES CPUS TRES_PER_NODE NODELIST
13083408 hpc       keras.sh user R  0:05 1     16   N/A           hpc078
```

- Check the slurm output file:

```
[user@cirrus08 tensorflow-cpu]$ cat slurm-13083408.out
…
/dev/shm/job13083408/slurm_script: line 9: 4176757 Killed                python
keras_example_small.py
slurmstepd: error: Detected 1 oom-kill event(s) in StepId=13083408.batch. Some of your
processes may have been killed by the cgroup out-of-memory handler.
```

Submit a job with memory request(cont.)

- Examining the output file we see that the job was aborted due to insufficient memory, by default the job get **8GB** per task (*we introduce a dummy array allocation on the **python** script to illustrate this issue*)

- We can check the job account to get the same information with command **sacct**:

```
[user@cirrus08 tensorflow-cpu]$ sacct -j 13083408
JobID           JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
13083408        keras.sh        hpc      group          1 OUT_OF_ME+    0:125
13083408.ba+ batch                       group          1 OUT_OF_ME+    0:125
13083408.ex+ extern                      group          1 OUT_OF_ME+    0:125
13083408.0      clean tmp                group          1  COMPLETED      0:0
```

Submit a job with memory request(cont.)

- The command **seff** can also be used and presents the same information with a different view:

```
[user@cirrus08 tensorflow-cpu]$ seff 13083408
Job ID: 13083408
Cluster: production
User/Group: martinsj/csys
State: OUT_OF_MEMORY (exit code 0)
Cores: 1
CPU Utilized: 00:00:05
CPU Efficiency: 83.33% of 00:00:06 core-walltime
Job Wall-clock time: 00:00:06
Memory Utilized: 0.00 MB (estimated maximum)
Memory Efficiency: 0.00% of 7.81 GB (7.81 GB/core)
```

Submit a job with memory request(cont.)

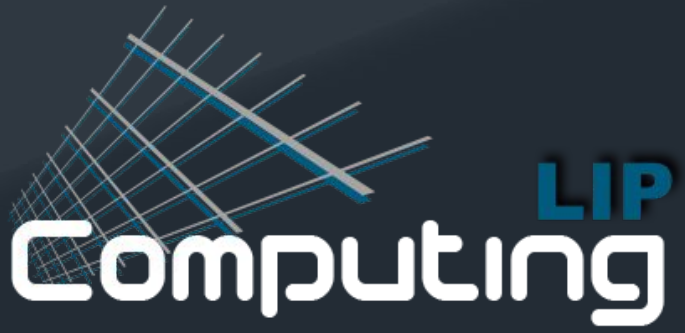
- We need to find out how much memory is need, the account commands **sacct** and **seff** don´t help, so we can apply one of the following strategies:
  - Analyze the code and predict the memory needed observing the number and type of allocated variables and functions calls and include the head and stack, for example; this is not an easy task and the best option is to adopt the next strategy.
  - Launch jobs with increasing memory requests until we get a successful run; we can start with the default **8GB** and double the request on each iteration.
- Include the option "**--mem=16G**" on the submission script **keras.sh**:

```
[user@cirrus08 tensorflow-cpu]$ cat keras.sh
#!/bin/bash
#SBATCH --partition=hpc
#SBATCH --mem=16G
#SBATCH --ntasks=1

module load tensorflow
python keras_example_small.py
```

Submit a job with memory request(cont.)

- If the job fail again then double the memory request, "**--mem=32G**", and resubmit **keras.sh** script.
- Keep doubling the requested memory until you obtain a successful run, our example run successfully with **16GB**.



LIP Computing Activities

LIP workshop 2018

European Commission
Cofinanciado por:
COMPETE 2020
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional
Lisb@20
FCT Fundação para a Ciência e a Tecnologia

How to start an interactive session

- It may be convenient (and faster) to test and troubleshoot applications on the actual execution nodes, the direct **ssh** session is not allowed so we start an interactive session on the workernodes with **srun** command:

```
[user@cirrus08 ~]$ srun -p hpc --job-name "my_interactive" --pty bash -i
srun: job 6959936 queued and waiting for resources
srun: job 6959936 has been allocated resources
[user@hpc079 ~]$ _
```

- You'll get one CPU "*core*" and only one, for example the GPU will be off limits:

How to start an interactive session(cont.)

- To request a GPU we must request the "**gpu**" partition and the appropriate resource:

```
[user@cirrus08 ~]$ srun -p gpu –gres=gpu --job-name "my_interactive" --pty bash -i
srun: job 6959936 queued and waiting for resources
srun: job 6959936 has been allocated resources
[user@hpc064 ~]$ nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
…
```

How to run custom software (conda example)

LIP Computing

Install conda environment

[jpina@cirrus09 ~]$ mkdir conda-test
[jpina@cirrus09 ~]$ cd conda-test/

```
[jpina@cirrus09 conda-test]$ conda create -n myconda python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /users3/csys/jpina/.conda/envs/myconda

  added / updated specs:
    - python=3.7
```

test conda

```
[jpina@cirrus09 conda-test]$ conda activate myconda
(myconda) [jpina@cirrus09 conda-test]$ pwd
/users3/csys/jpina/conda-test
(myconda) [jpina@cirrus09 conda-test]$ python3
Python 3.7.12 | packaged by conda-forge | (default, Oct 26 2021, 06:08:21)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

Submit conda job

```
cp ../hello/hello.sh .

 vi hello.sh

Cat hello.sh

#!/bin/bash
#SBATCH -p short
#SBATCH --tasks-per-node=1
#SBATCH --nodes=1
echo "Hello, ready to business"

python3 -V
source /etc/profile.d/conda.sh
conda activate /users3/csys/jpina/.conda/envs/myconda
python3 -V

echo "Done"
```

Submit conda job

```
[jpina@cirrus09 conda-test]$ cat slurm-13088770.out
* ------------------------------------------------------------
* Running PROLOG for hello.sh on Mon Oct 16 21:08:55 WEST 2023
*   JOB_NAME             : hello.sh
*   JOB_ID               : 13088770
*   JOB_PARTITION        : short
*   JOB_USER             : jpina
*   JOB_ACCOUNT          : csys
*   JOB_QOS              : normal
*   NODE_LIST            : hpc078
*   SLURM_NNODES         : 1
*   SLURM_NPROCS         : 1
*   SLURM_NTASKS         : 1
*   SLURM_CPUS_ON_NODE   : 1
*   SLURM_TASKS_PER_NODE : 1
*   SLURM_JOB_CPUS_PER_NODE : 1
*   SLURM_MEM_PER_CPU    : 5000
*   SUBMIT_HOST          : cirrus09.a.incd.pt
*   WORK_DIR             : /users3/csys/jpina/conda-test
*   JOB_SCRIPT           : /users3/csys/jpina/conda-test/hello.sh
* ------------------------------------------------------------
Hello, ready to business
Python 3.6.8
Python 3.7.12
Done

echo "Done"
```

Next Courses

December 2023

05 Dec Jorge Gomes, Mario David, "Advanced use of Containers in HPC (hand-on tutorial)"

November 2023

10 Nov Jorge Gomes, Mario David, "Using Containers in HPC (hand-on tutorial)"

Registration: https://indico.eurocc.fccn.pt/