Submitting and running jobs on PlaFRIM2

Redouane Bouchouirbat

Summary

1. Submitting Jobs: Batch mode - Interactive mode
2. Partition
3. Jobs: Serial, Parallel
4. Using generic resources Gres : GPUs, MICs.
5. Monitoring jobs.

Submitting Jobs :
Interactive mode-Batch mode

Batch mode & Interactive mode

What is a '**job**'?

Job = Your program + handling instructions

- Shell script/binary code which is run by the shell
- Allocation size:
  (Nodes or CPUs) + Aumount of Memory + Time
- Constraints (Features)
- **In batch mode**: Submitted in a single script file from the **devel** nodes**,** with sbatch command.
- **In interactive mode**: executing program with salloc/srun command on remote terminal/node

Batch & Interactive Jobs

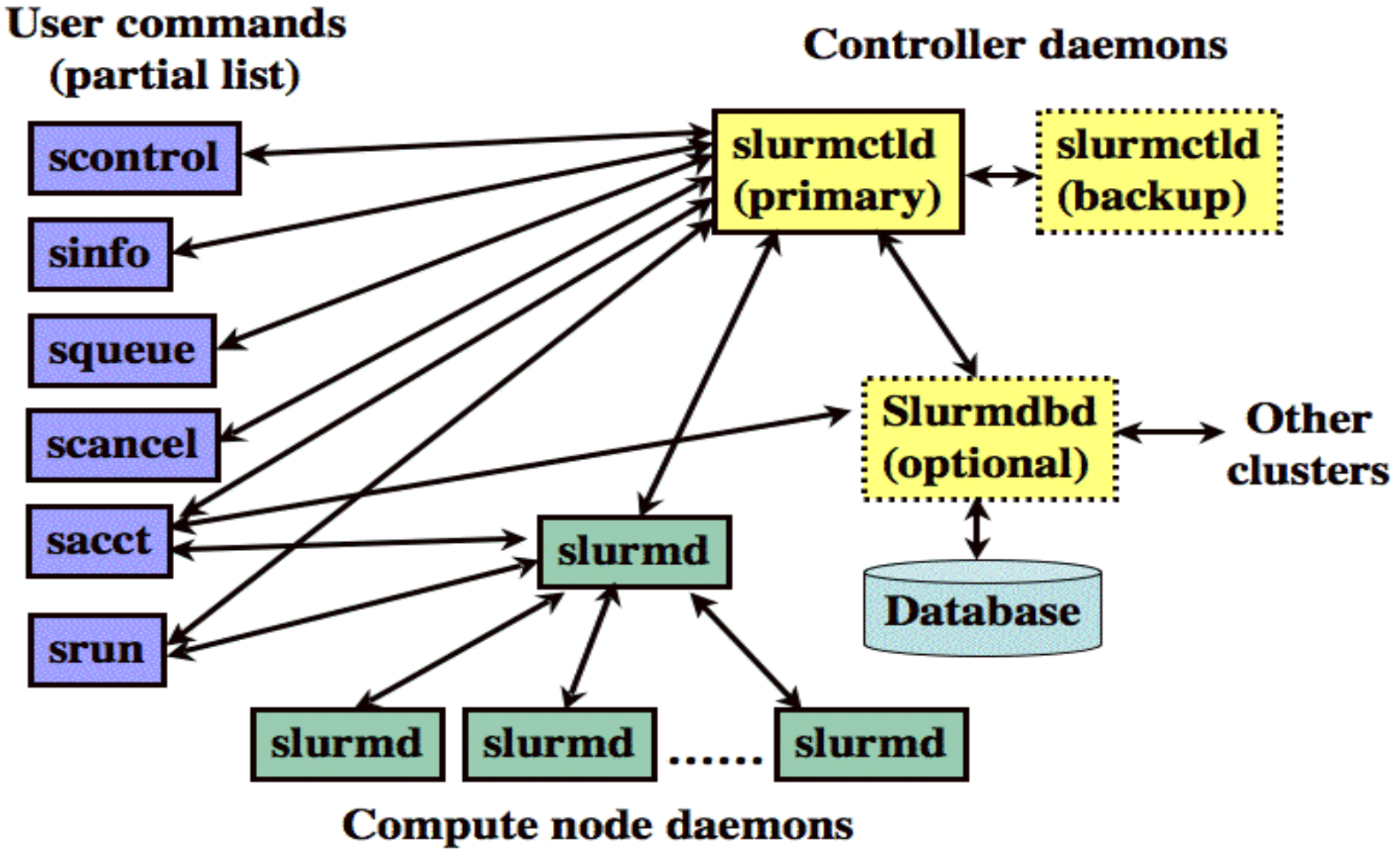
SLURM jobs's scheduling:

- Basic units of resources:
  - Nodes / CPU cores
  - Memory
  - Time
  - GPU cards
  - MICs Cards

- Compares user-specified requirements to resources available on compute nodes

- Starts jobs on available machine(s)

SLURM: Simple Linux Utility for Resource Management (open source llnl).



Batch & Interactive Jobs

- Load slurm module:

```
module load slurm/14.03.0
```

- Or automatically use initadd module's command (loaded at every login session):

```
module initadd  slurm/14.03.0
```

Interactive Jobs

- **srun** : launch command/program on **remote shell of all allocated nodes**

  usage : srun + option + <command/program>

  - srun - - ntasks=25 hostname:

    launch hostname command 25 times on available cores

  E.g: (use remote terminal as login terminal)

```
[bouchoui@devel13 ~]$ srun --pty bash -i
srun: job 35377 queued and waiting for resources
srun: job 35377 has been allocated resources
[bouchoui@miriel009 ~]$
```

  - By default one core is allocated for this job (you have to use nodes=1 one node when using --pty option)
  - The user environment is exported on the remote shell.
  - Most of the options valid for sbatch can be used with srun as well.

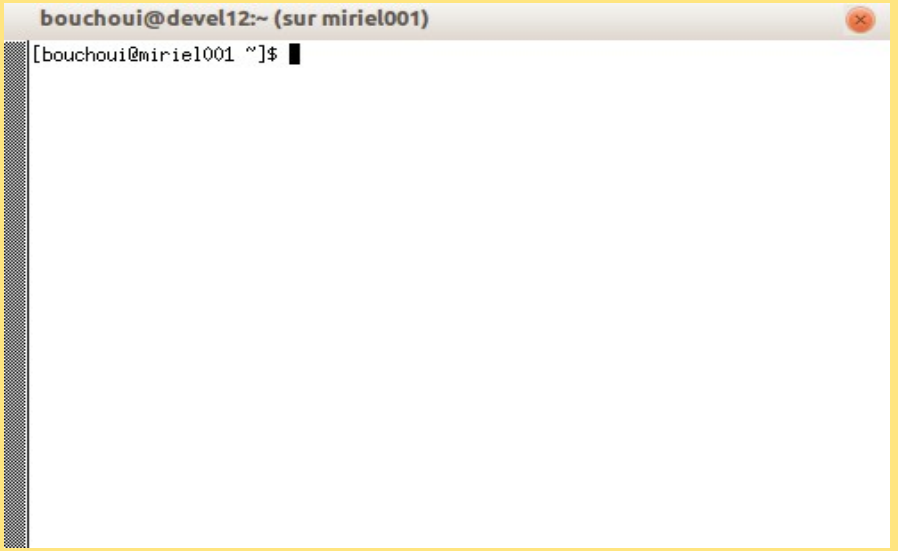
Interactive Jobs

- Using forwarding X11 (to have display)

  srun - -x11=first|all|last <command>

  E.g: (use remote xterm as login terminal)

- **[bouchoui@devel13 ~]$ srun –x11=first --pty xterm**



Interactive Jobs

- **salloc** : obtain a SLURM job allocation (a set of nodes), execute a command, and then release the allocation when the command is finished.

  - salloc --time=10:00 --mem-per-cpu=250 [<command/program>]

  - On other hand, one can just open a session and then use srun to start the program, every single srun considered by SLURM as another job step

```
bouchoui@devell13$  salloc --time=120:00 --mem-per-cpu=2500 --nodes 1 --ntasks 4
  salloc: Pending job allocation 1459
  salloc: job 1459 queued and waiting for resources
  salloc: job 1459 has been allocated resources
  salloc: Granted job allocation 1459
bouchoui@devel13$ srun hostname
   miriel021
```

Batch Jobs

```
#!/bin/sh
#PBS -N testPBS
#PBS -l  nodes=2:ppn=10
#PBS -l  walltime dd:hh:mm:ss
#PBS -o myout.out
#PBS -e myout.err

module purge
module load slurm/14.03.0
module load mpi/openmpi/gcc/1.8.4-tm


 mpirun -np  20 hostname
```

```
# !/bin/sh
#SBATCH --jobname=testSlurm
#SBATCH -N 2
#SBATCH - -ntasks-per-node=10
#SBATCH  - -time=d-hh:mm:ss
#SBATCH  -o myout.out
#SBATCH  -e myout.err

module purge
module load slurm/14.03.0
module load mpi/openmpi/gcc/1.8.4-tmi

mpirun hostname
```

Submission:

Torque: **qsub myJobpbs.sh**

SLURM: **sbatch myJobslurm.sh**

Batch & Interctive Jobs

What to do?	Torque	SLURM
Submit Batch job	qsub myjob.sh	sbatch myjob.sh
Intercative Job	qsub -I	salloc/srun
Cancel job	qdel jobid	scancel <jobid>

For more command http://slurm.schedmd.com/rosetta.pdf

Batch & Interctive Jobs

Accessing to compute nodes

- Job gets a unique jobID, from now on it is registered and whatever happens to the job it is already stored in DB of SLURM.
- SSH access to compute nodes is restricted.
- When job is running, user has access to the node(s) her/his job is running on.
- When user's job is terminated, all ssh sessions on the compute nodes will be killed, and all files/dirs of user on the local /tmp of the compute node will be removed.

Partitions

Partitions

Miriel nodes : miriel0[01-77]

Queue	Max User Running	Max User Queuable	Max Job Running	Max Job Queuable	MaxWalltime	Max Nodes	MaxCores
defq					2h	4	96
court	2	10	20	64	4h	42	1008
longq	2	10	16	64	72h	36	864
special	10	20	40	50	0.5h	77	1848

Note : VeryLong is removed from slurm's partitions

Partitions

Mistral nodes : mistral[001-018]

Queue	Max User Running	Max User Queuable	Max Job Queuable	Max Job Running	MaxWalltime	Max Nodes	Max Cores
court_mistral	2	10	20	10	4h	18	360 cores
long_mistral	2	5	10	10	3d=72h	16	320 cores

x2 MIC cards per node

MIC Card: Intel Xeon Phi 7120P

- Frequency : 1,238 GHz
- Cores count : 61 (244 threads)
- Memory : 16 GB GDDR5
- Memory rate : 2.75 GHz
- Bandwidth : 352 GB/s

Partitions

Sirocco nodes : sirocco[01-05]

Queue	Max User Running	Max User Queuable	Max Job Queuable	Max Job Running	MaxWalltime	Max Nodes	Max cores
court_sirocco	2	4	10	5	4h	5	120 cores
long_sirocco	1	2	4	2	72h=3d	2	48

X4 GPU cards per node

GPU Card: NVidia Tesla K40M

- Kepler GK110B
- Frequency: 745 to 875 MHz
- #Cores : 2880
- Memory: 12 GB GDDR5
- Peak Performance :
  - 4.2 TFLOPS single;
  - 1.4 TFLOPS double-precision
  - Memory BdW: 288 GB/s

Partitions

sinfo - view information about SLURM nodes and partitions.

```
]$ sinfo -p court
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
court        up    4:00:00      1 drain* miriel088
court        up    4:00:00      5    mix miriel[001,004-005,010,012]
court        up    4:00:00     22  alloc miriel[002,009,013-016,021-036]
court        up    4:00:00     50   idle miriel[003,006-008,011,017-020,037-077]
```

Jobs: Serial, Parallel

Jobs: Serial, Parallel

Transfer/copy remote files/directories to/from PlaFRIM2

- **git/svn**
  - Work with versonning tools
- **SSHFS**
  - Mount remote home/directories over SSH on local home/directories:
    $ mkdir <mountpoint>
    $ sshfs devel:/home/<login><directory> <mountpoint>
- **SCP/SFTP**
  - Copy individual files and directories:
    $ scp file1 devel:<where>/file1
- **Rsync** fast file transfer over SSH, it avoid copying existing files:
  $ rsync -auv --no-group source/ devel:target/

Jobs: Serial, Parallel

Serial jobs:(running on one core)

```
#!/bin/bash
# Name of your job
#SBATCH -J SerialTest
# output with the jobid
#SBATCH -o output%j.txt
# error message with the jobid
#SBATCH -e err%j.txt
# Ask for 1 nodes
#SBATCH -N 1 -n1
#SBATCH --mem=2000
# duration job 05 min
#SBATCH -t01:00:00

# load all modules that you need for your job (compiler, library ..., etc)
module purge
module load slurm/14.03.0
module load compiler/gcc/4.8.4
#example show my JObId

echo  $SLURM_JOBID

# here you lauch your application
./a.out
```

Jobs: Serial, Parallel

- **Multithreaded** (i.e. OpenMP)

```
#!/bin/sh
#SBATCH --time=2:00:00
#SBATCH --mem=1G
#SBATCH --cpus-per-task=12
export OMP_PROC_BIND=true
srun <openmp_program>
```

- **MPI**

```
#!/bin/sh
#SBATCH --time=30
#SBATCH --mem-per-cpu=1G
#SBATCH --nodes=4
#SBATCH --ntasks=48

module load <mpi/openmpi_Version>
Srun --mpi=pmi  <your_mpi_program>
```

Jobs: Serial, Parallel

Hybrid application (OpenMP+MPI)

$ cat slurmJob.sh

```
# !/bin/bash
#SBATCH -p court
#SBATCH --ntasks=24
#SBATCH --cpus-per-task=6
#SBATCH --mem-per-cpu=200
#SBATCH –mail-type=ALL
#SBATCH –mail=<your e-m@il adress>
#SBATCH --time=dd-hh:mm:ss
#
module purge
module load slurm/14.03.0
module load mpi/mvapich2/gcc/64/2.0b
module load compiler/gcc/4.8.4

export OMP_NUM_THREADS=6


srun -n 24 ./a.out
```

In hyper-threaded machine
Use :
 -B --extra-node-info=<sockets[:cores[:threads]]>

$sbatch slurmJob.sh

Jobs: Serial, Parallel

- Using infiniband with Intel-MPI application(on miriel servers)

```
srun hostname  -s| sort -u > mpd.hosts
export I_MPI_FABRICS=shm:tmi
mpiexec.hydra -f mpd.hosts -n $SLURM_NPROCS ./a.out
```

Or :

```
export I_MPI_PMI_LIBRARY=/cm/shared/apps/slurm/14.03.0/lib64/libpmi.so
  export I_MPI_FABRICS=shm:tmi
  srun -n $SLURM_NPROCS  ./a.out
```

- Using infiniband with OpenMPI application (miriel servers):

```
mpirun --mca btl openib,self ./a.out
```

Note:

- **Other available FABRICS for MPI communication network: ofa,tcp,dapl**
- **miriel servers have Intel True Scale Infiniband card, for inter-node communications, tmi is prefered**
- **Mistral and sirocco servers have Mellanox infiniband card, dapl or ofa is prefered.**

Jobs: Serial, Parallel

- Array of job

Usage: --array=[array_spec] $SLURM_ARRAY_TASK_ID

```
#!/bin/bash
#SBATCH -J jarray
# SBATCH -p court
#SBATCH -o jarray-%A-%a.out
#SBATCH --time=00:10:00
#SBATCH -N2
#SBATCH --ntasks-per-core=1
#SBATCH --mem-per-cpu=100
#SBATCH --array=1,4-6

module purge
module load slurm/14.03.0

echo "starting at `date` on `hostname`"

echo "SLURM_JOBID=$SLURM_JOBID"
echo "SLURM_ARRAY_JOB_ID=$SLURM_ARRAY_JOB_ID"
echo "SLURM_ARRAY_TASK_ID=$SLURM_ARRAY_TASK_ID"

echo "srun -l /bin/hostname"
srun -l /bin/hostname
srun sleep 100
srun echo "ended at `date` on `hostname`"
exit 0
```

```
$sbatch jobArray.slrm
Submitted batch job 35450
$squeue -r -u bouchoui
  JOBID  PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
  35450_1      defq   jarray bouchoui  R       0:09      2 miriel[006-007]
  35450_4      defq   jarray bouchoui  R       0:09      2 miriel[006-007]
  35450_5      defq   jarray bouchoui  R       0:09      2 miriel[008-009]
  35450_6      defq   jarray bouchoui  R       0:09      2 miriel[008-009]
```

Using generic resources Gres: GPUs, MICs,

Using generic resources Gres: GPUs, MICs,

--gres option: (generic resources)

#SBATCH --gres=name:N

- where 'name' is the generic resource name
- 'N' is a number of available resources (in addition to CPUs and memory).

E.g:

#SBATCH --gres=gpu:4 (with sirocco partition)
#SBATCH --gres=mic:2 (with mistral partition)

To see all available generic ressoures :
$ sinfo -o '%N %F %G'

```
@devel12 ~]$ sinfo -s -o "%.20N %.12F %.18G  "
NODELIST             NODES(A/I/O/      GRES
miriel[001-050,050-0  19/61/11/91      (null)
mirage08,sirocco06    0/1/1/2          gpu:2
sirocco01             0/1/0/1          gpu:4
mistral[01-18]        0/14/4/18        mic:2
sirocco[01-05]        0/5/0/5          gpu:4
miriel001             0/1/0/1          (null)
miriel001             0/1/0/1          (null)
mirage[01-07,09]      0/7/1/8          gpu:3
```

Monitoring jobs

Monitoring jobs

Job pending state

- (Priority): Your job has low priority.
- (Resources): Your job has enough priority to run, but there aren’t enough free resources.
- (ReqNodeNotAvail): You request something that is not available. (mem-per-cpu, cpu-per-node. Scheduled maintenance break).
- (QOSResourceLimit): Your job exceeds the QOS limits.
- (AssociationResourceLimit): The job exceeds some limit set on the association.

Monitoring jobs

- Using mail alert:

  --mail-type=[events] (E.g: BEGIN,END, FAIL, ...)
  --mail-user=[e-mail address]

Monitoring jobs

Sacct: display accounting data for all jobs and job steps (memory used, exit_code, nodes list … etc).

Usage: sacct + option

E.g:

- Display accounting data for jobs from the 4th to 15th December:

  sacct --format=jobid,elapsed,ncpus,ntasks,state,exitcode **-S** 12/04 **-E** 12/15

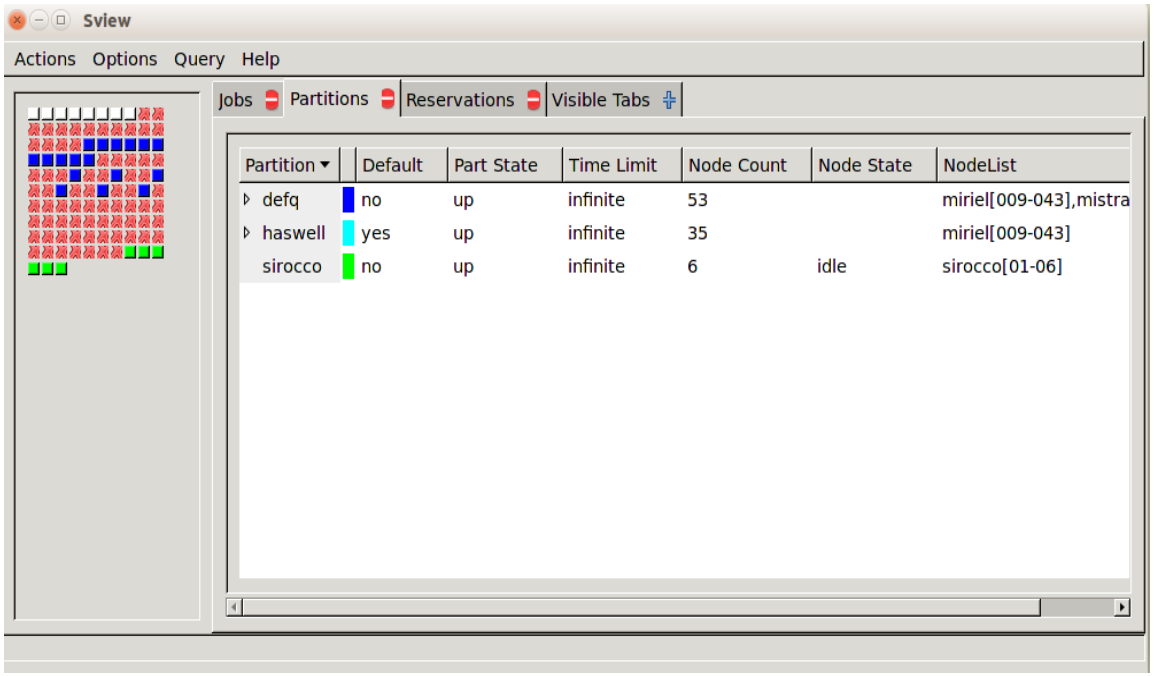
```
       JobID    Elapsed      NCPUS   NTasks      State ExitCode
------------ ---------- ---------- -------- ---------- --------
23907          00:00:04          1        1     FAILED      1:0
23909          00:00:01          4          COMPLETED      0:0
23909.batch    00:00:01          1        1  COMPLETED      0:0
23911          00:00:00          4          COMPLETED      0:0
23911.batch    00:00:00          1        1  COMPLETED      0:0
23912          00:00:00          4        COMPLETED      0:0
23912.batch    00:00:00          1        1  COMPLETED      0:0
```

- Display accounting data for all job of group sed-bdx:

  sacct - -format=jobid,start%-20,end%-20,elapsed,ncpus,ntasks,state,exitcode,NODELIST%-30,NNodes,user **-S** 12/10 **-E** 12/20 **-A sed-bdx**

Monitoring job

$ scancel <jobId_list>
$ squeue -u <user> or -j <jobId> ;
$ squeue –start
$ sview



Getting help:

- Send an e-mail (with subject and more details of your job issue, eventually join your bash script) to PlaFRIM Support: plafrim-support@inria.fr .
- Documentation on PlaFRIM Web-Site: https://plafrim.fr
- **Officiel documentation: http://slurm.schedmd.com/**

Thanks

Questions?