Appendix B
SLURM: A Resource Manager and Job Scheduler on Clusters of Machines

It can be tedious to write manually `hostfile` configuration files for executing MPI programs on a set of interconnected machines via the `mpirun --hostfile config` command, specially when we use a large cluster of machines. Fortunately, we can use a *Task scheduler* to allocate and plan the execution of programs. SLURM[2] that is the acronym for *Simple Linux Utility for Resource Management (SLURM)* is such a utility program to manage and share resources among users. SLURM schedules tasks (jobs) to be executed on a cluster of machines. *Pending jobs* are queued jobs waiting to be executed once resources get available for them. SLURM takes care of Input/Output (I/Os), signals, etc.

Jobs are submitted by users via shell commands, and they are scheduled according to the FIFO model (First-In First-Out). Script files to launch batched jobs can also be submitted to SLURM.

From the user standpoint, the four main commands prefixed with a 's' (for 'S'LURM) are:

- `sinfo`: display general information of the system,
- `srun`: submit or initialize a job,
- `scancel`: raise a signal or cancel a job,
- `squeue`: display information of system jobs with `R` indicating Running state and `PD` PenDing state,
- `scontrol`: administrator tool to set or modify the configuration.

We recommend the online SLURM tutorial[3] for further details.

A common scenario is to organize and configure a large set of machines into a few clusters of machines. For example, the author taught the contents of this textbook to about 280 students/year using 169 machines, and organized those computers into 4 clusters, 3 of 50 machines and one cluster of 19 machines. A student logged on a

[2] Available online at https://computing.llnl.gov/linux/slurm/.

[3] http://slurm.schedmd.com/tutorials.html.

F. Nielsen, *Introduction to HPC with MPI for Data Science*, Undergraduate Topics in Computer Science, DOI 10.1007/978-3-319-21903-5

machine can display the information related to the cluster the machine belongs to by typing the `sinfo` command:

```
[malte ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
Test*        up      15:00     19   idle
allemagne,angleterre,autriche,belgique,espagne,
    finlande,france,groenland,
hollande,hongrie,irlande,islande,lituanie,malte,monaco
    ,pologne,portugal,roumanie,suede
```

We can easily execute a program called `hostname` (just reporting on the console the names of the machines used) using 5 nodes (hosts) with a maximum of two processes per node as follows:

```
[malte ~]$ srun -n 5 --ntasks-per-node=2 hostname
angleterre.polytechnique.fr
autriche.polytechnique.fr
allemagne.polytechnique.fr
allemagne.polytechnique.fr
angleterre.polytechnique.fr
```

We can also use SLURM to execute a shell command `shell.sh` that launches a MPI program `myprog` as follows:

```
[malte ~]$ cat test.sh
#!/bin/bash
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/openmpi-1.8.3/lib/:
/usr/local/boost-1.56.0/lib/
/usr/local/openmpi-1.8.3/bin/mpirun -np 4 ./myprog

[malte ~]$ srun -p Test -n 25 --label test.sh
09: I am process 1 of 4.
09: I am process 0 of 4.
...
01: I am process 0 of 4.
01: I am process 2 of 4.
05: I am process 2 of 4.
05: I am process 3 of 4.
```

We summarize the main SLURM commands in the table below:

`salloc`	Resource allocation
`sbatch`	Give “batch” files
`sbcast`	Dispatch files on allocated nodes
`scancel`	Cancel the running “batch” file
`scontrol`	Control interface of SLURM
`sdiag`	To get the status report
`sinfo`	Display information related to the cluster of machines
`squeue`	Display the job queue
`srun`	Run a job
`sstat`	Report execution states
`strigger`	Manage and trigger signals
`sview`	Interface to view the cluster