Sorry for the inconvenience, documentation is still in preparation
Please feel free to send your questions to support.

Job Management

Overview / Jobs

Like many high-performance computing systems, ICI-SC's HPC systems are managed by a batch scheduling system. This means that in order to use the supercomputer, users must encapsulate the workload into a non-interactive job and submit that job to the scheduling system. SLURM is used as batch system on our BULLx DLC Cluster.

Job submission to SLURM is done using a job command file. The job command file is a shell script containing keywords embedded in comments beginning with #SBATCH followed by options. These keywords inform SLURM of the resources required for the job to run, the program to execute, where to write output files and the job environment. The job can specify a number of parameters to the scheduling system, including the following:

required attribute

- Resources Needed
  - Either number of nodes and processors per node, or total number of processors
  - Memory/RAM needed
  - Expected running time (or “walltime”)
  - Specific node features/attributes (GPUs, processor types, etc.)
  - Local disk space needed
- Notifications
  - Events to notify on (job abort, begin, end)
  - Email address to send notifications
- Job Name (used for output files)

SHORTEN your walltime to run more jobs and to let SLURM optimize better queue times scheduling!

Jobs are created by building a job script, usually written in bash, tcsh, perl, or python, that specifies the necessary parameters described above, and then contains appropriate scripting to launch the program that will do the job's work.

The batch job priority depends mainly on the job size: big jobs have a significant higher priority than smaller jobs.

Scheduling System

The scheduling system keeps track of the current state of all resources and jobs and decides, based on conditions and policies we have configured, where and when to start jobs. Jobs are started in priority order until no further jobs are eligible to start or it runs out of appropriate resources. The factors that are included in the priority calculation are:

- Historical Usage Patterns (eg. how much has been recently used)
  - Per user
  - Per research/project group
- Total time queued

When the scheduling system chooses to start a job, it assigns it one or more nodes/processors, as requested by the job, and launches the provided job script on the first node in the list of those assigned. The responsibility of taking advantage of all the requested resources is left up to the job script. Please do not request more resources unless you know you can use them.

Results

The output (both stdout and stderr) is sent to the file slurm-JOBNUMBER.out where **JOBNUMBER** is the unique job number assigned when the job is submitted. The output file is created in the directory where the job was submitted from. The output begins appearing as soon as the job starts running.

If you submit a job through XCS portal, a directory is created in `~/Jobs/JOBNUMBER` related to the job launched.

Options in SLURM

For reference, examples in the table below explain how to include the appropriate SLURM options for parallel jobs.

Options	**Meaning**
--nodes=[count]	Node count
--tasks-per-node=[count]	Processes per node
--ntasks=[count]	Total processes (across all nodes)
--cpus-per-task=[count]	CPU cores per process
--nodelist=[nodes]	Job host preference
--exclude=[nodes]	Job host to avoid
--time=[min] or --time=[dd-hh:mm:ss]	Wall clock limit, other formats: minutes, days-hours, days-hours:minutes, days-hours:minutes:seconds
--mem=[count]	RAM per node
--mem-per-cpu=[count][M or G]	RAM per CPU core
--output=[file_name]	Standard output file
--error=[file_name]	Standard error file

Options	Meaning
(default behavior)	Combine stdout and stderr
--array=[array_spec]	Launch job array
--mail-user=[email_address]	Email for job alerts
--mail-type=[BEGIN or END or FAIL or REQUEUE or ALL]	Email alert type
--account=[account]	Account to charge
--depend=[state:job_id]	Job dependency
--job-name=[name]	Job name
--constraint=[attribute]	Request node attribute (“sandy_bridge”, “haswell”, “eight”)
--partition=[name]	Submit job to specified partition

Job History and Accounting

How to retreive job history and accounting information.

Show SLURM Associations

Each user has SLURM associations that enable access to the SLURM partitions and clusters.

```
sacctmgr -s list user <username>
```

Show Job History and Accounting

The SLURM job accounting records show job details, including job steps and memory usage. The memory usage is accurate for jobs that use srun to launch tasks.

- use one of the *Liger User Commands (LUC)* to report your current usage (core hours) and the account usage and limit

  ```
  Mybalance
  ```

- show job account information for a specific job

  ```
  sacct -j jobid --
  format=User,JobID,Jobname,partition,state,time,start,end,elapsed,MaxRss
  ,MaxVMSize,nnodes,ncpus,nodelist
  ```

- show information for jobs currently in queue

  ```
  sacct --
  format=User,JobID,Jobname,partition,state,time,start,end,elapsed,MaxRss
  ,MaxVMSize,nnodes,ncpus,nodelist
  ```

- show all job information starting from a specific date

```
sacct --starttime 2014-07-01 --
format=User,JobID,Jobname,partition,state,time,start,end,elapsed,MaxRss
,MaxVMSize,nnodes,ncpus,nodelist
```