Requesting Compute Resources

When using Slurm to run your jobs, you have to be specific about requesting CPU cores and nodes. There are three options to consider when creating your job:

--nodes (or -N)
--ntasks (or -n)
--cpus-per-task (or -c)

Multithreaded Jobs

Applications that execute in parallel across multiple cores on a single node while using shared memory are multithreaded jobs. Slurm allocates one CPU core per task. To use multiple CPU cores in a multithreaded job, use the --cpus-per-task option. As our nodes include 128 cores each, the maximum number of cores that could be requested is 128.

Job Arrays

Job arrays are for managing a large number of similar jobs. For example, if you wanted to compare 100 similar files, you could do so using a job array. SLURM provides job array environment variables that allow multiple versions of input files to be easily referenced. In the next example, three input files called example_0.py, example_1.py, and example_2.py are used as input for three independent Python jobs:

```bash
#!/bin/bash
#SBATCH --mail-user=example@yale.edu
#SBATCH --mail-type=ALL
#SBATCH --ntasks=1
#SBATCH --time=2:00:00
#SBATCH --mem=2G
#SBATCH --array=0-2
#SBATCH --output=python_array_job_slurm_%A_%a.out

echo "SLURM_JOBID: " $SLURM_JOBID
echo "SLURM_ARRAY_TASK_ID: " $SLURM_ARRAY_TASK_ID
echo "SLURM_ARRAY_JOB_ID: " $SLURM_ARRAY_JOB_ID

# Load Anaconda distribution of Python
module load Anaconda2
python example_${SLURM_ARRAY_TASK_ID}.py
```