CSC

ICT Solutions for Brilliant Minds

How to Figure Out Batch Job Parameters

2024-11-27, Ari-Matti Sarén

What parameters need attention

- Number of cores (CPUs)
- Amount of Memory
- Time
- Special resources
  - Local disk
  - GPU

General notes on number of cores

- More is not always better
  - Using too many cores can actually make the job run slower
  - There is typically an optimum number
- If you reserve them, also make sure to use them
  - If software has a command line parameter for numer of cores/threads always set it
  - Defaults often do not work as expected

Check software documentation

- Can it use more than one core?
  - If there is no mention in the documentation and no command line parameter to set it: Probably not
- If it can use many cores, how is it implemented?
  - Threaded/shared memory applications must be run inside one node
    - This is the most common case
  - MPI applications can be spread on many nodes
    - Unless the documentation specially mentions this, it is safe to assume it is not supported

So how many cores to use?

- Check software documentation for recommendations
- Run tests and check
- You can use e.g. `seff` command
- Check CPU efficiency
  - Low efficiency can be a sign of too many cores used (but there are other possible reasons)
- Check job execution time (wall clock time)
  - As a rule of thumb: doubling the number of cores should make the job run 1,5X faster

Memory reservation

- Check software documentation
  - Some software parameters can have a big impact on memory usage
- Check available nodes
  - For example on Puhti most nodes have 192 GB of memory
  - Can affect job queuing time a lot
- Memory is a limited resource!
  - Start with lower memory reservation and increase reservation if job crashes
- After job finishes use `seff` to check

Time reservation

- When the time reservation ends, the job will be cancelled whether finished or not
- When testing or when unsure it is OK to reserve the partition maximum time
  - When the last job step is finishes job ends and resources are freed
  - BUs billed according to actual job duration
- When you get familiar with the program, try to use realistic reservations
  - Things like estimated job start times rely solely on job time reservations

Special resources: Local disk

- If the program generates a lot of small files or there is a lot of reading and writing files performance can be bad on */scratch*
  - Low CPU efficiency can be indicative of this
  - Using fast local disk can help
- Only available on some nodes
- Node specific
- Job specific
  - Only exist during the duration of the job, so remember to copy any results to */scratch* as part of the batch job

Local disk, cont.

- In addition to reserving it, remember to use it!
  - If reserved, variables $LOCAL_SCRATCH and $TMPDIR are set to use it
- Reservation:

```
#SBATCH --gres=nvme:50
```

- Usage:

```
cp input.file $LOCAL_SCRATCH
myprog --input $LOCAL_SCRATCH/input.file --output $LOCAL_SCRATCH/output.file
mv $LOCAL_SCRATH/output.file $SLURM_SUBMIT_DIR
```

Special resources: GPU

- For software that can greatly benefit from use of GPU
  - Rule of thumb: Use GPU if running on GPU is faster than a full-node CPU job on the same machine
- Remember to request GPU resources and select suitable partition

```
#SBATCH --gres=gpu:v100:1
#SBATCH --partition=gpu
```

Complex cases

- Jobs with many job steps with different requirements
  - Depending on job step time requirements consider breaking down to separate batch jobs
  - Batch jobs can be chained, i.e. job 2 starts after job 1 has finished

```
sbatch –d afterok:12345 myjob2.sh
```

- Job farming cases (*e.g.* array jobs) where sub jobs have different requirements (typically memory)
  - Depending on ratio of jobs with higher requirements, consider running first with lower reservation and re-run only the failed ones with higher reservation

So to sum it up

- Check the documentation
- Test, check & adjust
  - Also a good idea to keep notes, especially for applications you use less often
- From resource use point of view it is better to first try with lower reservation and re-run failed jobs than always reserve the maximum ”just in case”


facebook.com/CSCfi


twitter.com/CSCfi


youtube.com/CSCfi


linkedin.com/company/csc---it-center-for-science


github.com/CSCfi