HPC Tuning and debugging

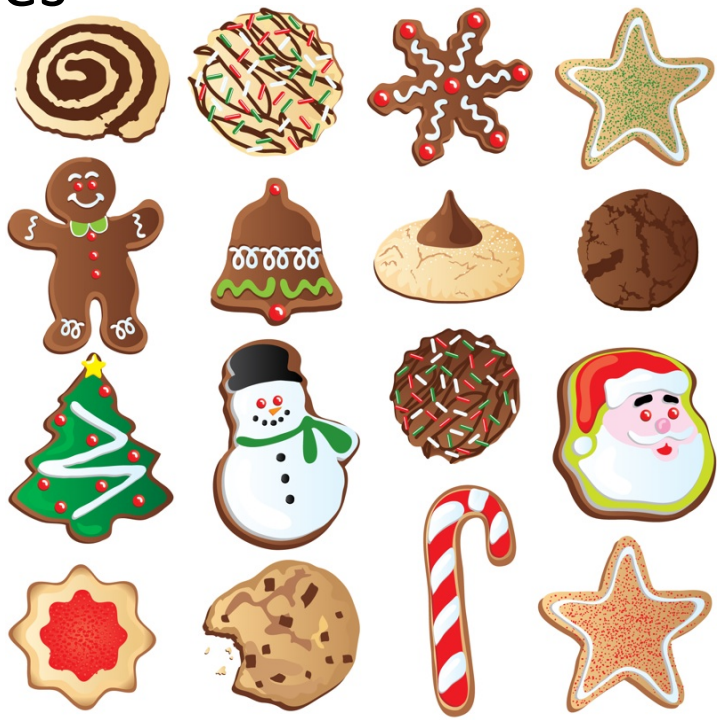
Getting started

Batch Jobs

Batches of Cookies

A friend offers to bake some of your famous cookies

What do you need to do?



For each cookie type you need to provide a recipe, which includes

- List of ingredients
- How hot the oven needs to be and how long to bake
- Actual instructions on what to do.

Depending on the size of your friends oven some cookies are baked at the same time and others when there is room in the oven.

Batch Jobs

Need a batch script describing the work to be done. (recipe)

- Describe the set of requested resources needed.
- Have a requested walltime, a time for which the job will run.

  (If it exceeds this time it will be killed.)
- Actual instructions on what to do.

Batch Jobs can be started by the scheduler when enough resources are available.

Simple Slurm job script (recipe)

```
#!/bin/bash
#SBATCH --ntasks=1
#SBATCH --nodes=1
#SBATCH --time=0-00:02
#SBATCH --mail-type=ALL
#SBATCH --mail-user=no.email@ubc.ca
#SBATCH –o my-output-file-%j.out
#SBATCH --job-name=my-named-job
sleep 1000;  # Replace with a line running code
```

How to develop a new recipe.

Would you use the same process and have your friend bake cookies?

or would you ask to use your friends bakery interactively?

If you were trying to make cookie production more efficient would you ask your friend to fill out a some questions and report back to you or go into the bakery yourself and observe the process interactively yourself?

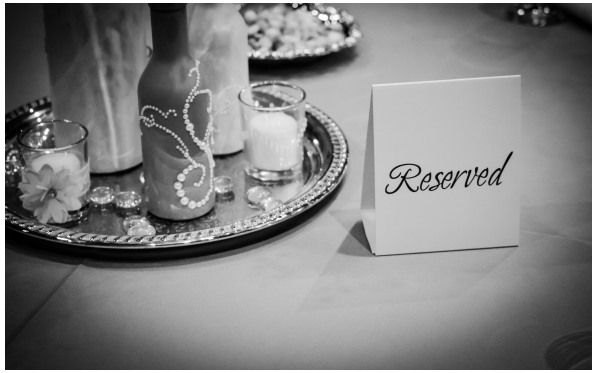
Interactive Use

In a Slurm batch cluster

Interactive Jobs

Interactive jobs are used to run a program on the cluster and interact with it while it is running often for graphical or interactive work, debugging, and profiling.

There is a set of reserved nodes for interactive usage that are set aside for interactive jobs, but the jobs are not limited to run on these nodes.



Interactive Jobs have limits.

To ensure more people get to take advantage of these resources

- Limit of 1 interactive job per user.
- Limit of max walltime allowed per job.
- Limits on what type of jobs can run:
  - No interactive job arrays.

If the resources to run your interactive job cannot be found within a reasonable time your request will be canceled.

2 ways of Interacting with Jobs

1.) Requests an interactive job with the salloc command

salloc --ntasks=1 --nodes=1 --time=0-01:20

You would run your work interactively at this point.

2.) Run a regular non-interactive job and then interact with it after it starts

- The interactive limits don't apply for this job as it is a normal job.
- These jobs can't use the resources reserved for interactive use.
- It may take a long time for these jobs to start

After the job starts interact with it by attaching to the already running job

srun --jobid=<jobid> --pty –xll

You would run your work interactively at this point.

Any resource you use will be attributed to the job.

Profiling by attaching to an already running job

Run script to keep a record of any outputs

```
script profile-jobid-todaysdate.txt
```

Attach to an already running job

```
srun --jobid=<jobid> --pty bash
```

Remember that you are logged in only on one machine and your job may span more than one

List environment variables with printenv

```
printenv
```

To check memory usage use

```
top -u $USER
```

Run any other UNIX commands to find the state of your program

Type “exit” to exit the the job

Type “exit” to exit the script recording environment.

Interactive Jobs for debugging

Run script to keep a record of any outputs

```
script profile-jobid-todaysdate.txt
```

Use salloc instead of sbatch to launch interactive jobs.

```
salloc  --ntasks=4 --mem-per-cpu 4000 -t 0-00:20
```

List environment variables with printenv

```
printenv
```

To check memory usage use

```
top –u $USER
```

Remember that you are logged in only on one machine and your job may span more than one

Unix processes

- Running program have
  - **PID** (process id) which is a uniq number
  - Owner with a user id
  - Has some process that started it a parent which has its own PID the PID of the parent process is (**PPID** parent process pid)
- “pstree” command show the family tree of processes
- Can be killed with the “kill“ command
- Demo foreground and background commands (fg,bg,&,jobs) commands

Some Unix commands for debugging and profiling

List environment variables with printenv: `printenv`

To check memory usage: `top -u $USER`

Processes	Memory	Fileystem/ IO	Network	Other	Advanced
ps pstree top htop	top htop free	df lsof -p quota pidstat -d -p pid htop dstat iotop iostat	netstat	uptime hostname whoami who	sar perf

You can also look at the filesystem of the node directly

Cgroup: ls /sys/fs/cgroup/cpuset/slurm/uid_$SLURM_JOB_UID/job_$SLURM_JOB_ID/step_$SLURM_STEPID

/proc/meminfo

Profiling Your Jobs

Slurm Jobs and memory

- Always ask for slightly less than total memory on node as some memory is used for OS, and your job will not start until enough memory is available.
- You may specify the maximum memory available to your job in one of 2 ways.
  - Ask for a total memory used by your jobs (MB)

    #SBATCH --mem=4000
  - Ask for memory used per process/core in your job (MB)

    #SBATCH --mem-per-cpu=2000

Slurm Jobs and memory

It is very important to specify memory correctly

- If you don’t ask for enough and your job uses more ,your job will be killed.
- If you ask for too much, it will take a much longer time to schedule a job, and you will be wasting resources.
- If you ask for more memory than is available on the cluster your job will never run. The scheduling system will not stop you from submitting such a job or even warn you.
- If you don’t specify any memory then your job will get a very small default maximum memory.

Find out how much memory a job used.

Command	Flags	What its used for
sstat		Display various status information of a running job
	–j <jobid>	Displays information about the specified job
	--format= AveCPU,MaxRSS,MaxVMSize,JobID	limits the information to that about memory (MaxVMSize is requested memory) (MaxRSS is memory used)
sacct		Displays slurm accounting data
	–j <jobid>	Displays information about the specified job
	-u $USER	Displays information about jobs belong to a specific user
	--format= JobID,AveCPU,MaxRSS,MaxVMSize	limits the information to that about memory
salloc		Submit to run Job Interactively, use unix system utilities such as top
	Same flags as sbatch	Note not all sbatch flags work

Virtual and Physical memory

A program can ask an OS (operating system) to use a chunk of memory. This is virtual or requested memory: MaxVMSize

MaxRSS is the amount of memory used by your code.

These can and often are different because the OS grants memory request without actually giving the memory until it us used.