Transition to SLURM

scitas.epfl.ch

July 9, 2014
What and why

PBSPro is not well suited to HPC and is also rather expensive!

SLURM is a modern HPC scheduler and is widely used.

Combined with political/structural changes (no more private nodes) the user experience will be much improved.

Goal: easier and better for you!
#!/bin/bash

#PBS -l select=2:ncpus=16:mem=32gb
#PBS -l walltime=02:00:00
#PBS -M your.email@epfl.ch
#PBS -o /scratch/gruyere/clara/moovit-results

module purge
module load intelmpi/4.1.3
mpirun /home/bob/code/milk < /home/bob/input/x23.dat
And now with SLURM

```
#!/bin/bash
#SBATCH --nodes 2
#SBATCH --ntasks 32
#SBATCH --cpus-per-task 1
#SBATCH --ntasks-per-node 16
#SBATCH --mem 32000
#SBATCH --time 02:00:00
#SBATCH --mail-user your.email@epfl.ch
#SBATCH --workdir /scratch/gruyere/clara/moovit-results

module purge
module load intelmpi/4.1.3
mpirun /home/bob/code/milk < /home/bob/input/x23.dat
```
SBATCH directives

--nodes 2
the number of nodes to use

--ntasks 2
the number of tasks (in an MPI sense) to run per job

--cpu-per-task 8
the number of cores per aforementioned task

--ntasks-per-node 1
the number of tasks per node

--mem 32000
the memory required in MB per node

--time 12:00:00  # 12 hours
--time 2-6      # two days and six hours
the time required
To submit jobs to the batch system the command is `sbatch`

$ sbatch myjob.sh
Submitted batch job 439

One can also pass arguments to `sbatch`

$ sbatch --partition fast myjob.sh
Submitted batch job 440

See `man sbatch` for all the options!
To cancel a specific job:

```
scancel <JOB_ID>
```

To cancel all your jobs:

```
scancel -u <username>
```

To cancel all your jobs in a particular state:

```
scancel -t PENDING -u <username>
```

qdel ⇒ scancel
Exercise

#!/bin/bash
#SBATCH --workdir /scratch/<username>
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 1
#SBATCH --mem 1024
sleep 10
echo "hello from $(hostname)"
sleep 10

Now adapt one of your PBS job scripts to SLURM and submit it using sbatch
What’s going on?

$ squeue

$ squeue -j <job id>

$ scontrol -d show job <job id>

$ sinfo -l

$ sshare -a

Try them and see what happens.
$ squeue

<table>
<thead>
<tr>
<th>JOBID</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3402</td>
<td>cpmd</td>
<td>kate</td>
<td>PD</td>
<td>0:00</td>
<td>4</td>
<td>(Resources)</td>
</tr>
<tr>
<td>3398</td>
<td>water</td>
<td>bob</td>
<td>PD</td>
<td>0:00</td>
<td>12</td>
<td>(Resources)</td>
</tr>
<tr>
<td>3406</td>
<td>tsk</td>
<td>sue</td>
<td>PD</td>
<td>0:00</td>
<td>8</td>
<td>(Priority)</td>
</tr>
<tr>
<td>3391</td>
<td>tsk</td>
<td>sue</td>
<td>R</td>
<td>5:49:44</td>
<td>12</td>
<td>b[401-412]</td>
</tr>
<tr>
<td>3401</td>
<td>ice</td>
<td>tim</td>
<td>R</td>
<td>17:10:01</td>
<td>2</td>
<td>b[413-414]</td>
</tr>
<tr>
<td>3393</td>
<td>QE</td>
<td>alex</td>
<td>R</td>
<td>23:49:13</td>
<td>1</td>
<td>b415</td>
</tr>
</tbody>
</table>
qstat -f ⇒ scontrol

eroche@bellatrix:jobs > scontrol -d show job 247
JobId=247 Name=j1
   UserId=eroche(141633) GroupId=scitas-ge(11902)
   Priority=6704 Nice=0 Account=scitas-ge QOS=normal
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 RequeueLimit=0 BatchFlag=1 ExitCode=0:0
   DerivedExitCode=0:0
   RunTime=00:00:13 TimeLimit=01:00:00 TimeMin=N/A
   PreemptTime=none SuspendTime=none SecsPreSuspend=0
   Partition=test AllocNode:Sid=bellatrix:18263
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=b[413-416]
   BatchHost=b413
   NumNodes=4 NumCPUs=64 CPUs/Task=16 ReqB:S:C:T=0:0:**:**
   Socks/Node=*
   NtasksPerN:B:S:C=0:0:**:** CoreSpec=0
   Nodes=b[413-416] CPU_IDS=0-15 Mem=32000
   MinCPUsNode=16 MinMemoryNode=32000M MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   Shared=0 Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/eroche/slurm/jobs/j1
   WorkDir=/scratch/eroche
   StdErr=/scratch/eroche/slurm-247.out
   StdIn=/dev/null
   StdOut=/scratch/eroche/slurm-247.out
Once a job has finished you need to use `sacct` to see what went on.

```bash
sacct -j <JOB_ID>
```

```bash
sacct -l -j <JOB_ID>
```
RTFM

man sbatch

man scancel

man sacct

man squeue

And don’t forget the official website at:

http://slurm.schedmd.com
What’s new?

salloc creates a reservation but doesn’t run any jobs

srun launches (parallel) jobs
salloc

s alloc accepts the same options as sbatch

salloc -N 2 -n 16 -c 32 -t 01:00:00

If the resources aren’t immediately available then the request will queue

$ s alloc -N 8 -n 8 -c 16
s alloc: Pending job allocation 248
s alloc: job 248 queued and waiting for resources

When queuing it has the same priority as any other job
**srun**

srun accepts the same options as `sbatch` and `salloc`

```
$ srun -N 4 -n 4 hostname
b413
b416
b415
b414
```

`srun` can be used instead of `mpirun` but it requires the MPI stack to be correctly configured.

On very large systems `srun` is much faster at launching processes.

If `srun` is not run inside a `salloc` session it will create an allocation so might have to queue.
Interactive jobs (qsub -l)

There are many ways to launch interactive jobs depending on the requirement.

(1) `salloc` and `srun/mpirun`

(2) `srun --pty bash -i`

(3) `salloc` and `ssh`

Try them out!
Bellatrix migration plan

▶ Now: test partition in place
▶ July: migration of the shared nodes
▶ August: migration of the private nodes

Aries will be migrated in October.