Bedienung des Jobverwaltungssystems (Batchsystem)

Der Slurm Workload Manager, kurz Slurm ist die Software, welche die Planung der Jobs übernimmt. Diese Anleitung befasst sich mit der Bedienung dieses Softwaresystems und soll Ihnen ermöglichen ein Script zu erstellen, mit dem Sie Ihren Job übermitteln können. Hierzu gibt es ein kleines Testscript zur Berechnung von PI, unter:

/home/public/public/software_demos_und_projekte/qsub_test_II/Beispiel_pi/pi_mpi

Wenn Sie das Script starten, ohne Änderungen vorzunehmen, wird der Job etwa 8 Minuten bis zu seiner Fertigstellung benötigen. Aber zuvor ein Blick auf das auszuführende Skript:

Wichtig ist hier die „submit.sh“-Datei, welche folgenden Inhalt hat:

submit.sh

```
1  #!/bin/bash
2  #SBATCH --output=output.dat
3  #SBATCH --partition=queue1
4  #SBATCH --job-name=pi_mpi
5  #SBATCH --nodes=2
6  #SBATCH --ntasks-per-node=4
7  #SBATCH --time=0:60:00
8
9  #ulimit -l
10 . /etc/profile.d/lmod.sh
11 #module add mpi/OpenMPI
12
13 scontrol show hostnames $SLURM_JOB_NODELIST > nodes.txt
14
15 export CORES=8
16
17 cd $SLURM_SUBMIT_DIR
18
19 mpirun -machinefile nodes.txt -np $CORES ./pi_mpi
```

Die ersten sieben Zeilen, die mit #SBATCH beginnen (abgesehen von der ersten), dies sind die Anweisungen für das Batchsystem:

#SBATCH --output= → Der Dateiname, in dem die Ergebnisse ausgegeben werden sollen, hier „output.dat“.

#SBATCH --partition= → Steht für die Queue, in die der Job eingereiht werden soll. „queue1“ ist hier die Standardqueue.

#SBATCH –-job-name= → Definiert den Namen des Jobs.

#SBATCH –-nodes= → Hier wird die Anzahl der zu reservierenden Nodes definiert. In diesem Fall sind es zwei.

#SBATCH –-ntasks-per-node= → Hier wird die Anzahl der zu reservierenden Prozessorkerne pro Node angegeben. In diesem Fall sind es vier.

#SBATCH –-time= → Hier werden für den Job 60 Minuten reserviert. Danach erfolgt, sollte der Job noch nicht fertig sein, der Abbruch.

Als nächstes werden wir diesen Job an das System übermitteln.
Befehl: *sbatch submit.sh*

```
nh256032@cc-main /home/public/public/software_demos_und_projekte/qsub_test_II/Beispiel_pi/pi_mpi $ ls
Makefile  nodes.txt  output.dat  pi_mpi  pi_mpi.cpp  submit.sh
nh256032@cc-main /home/public/public/software_demos_und_projekte/qsub_test_II/Beispiel_pi/pi_mpi $ sbatch submit.sh
Submitted batch job 316
```

Der Befehl: *squeue* Gibt eine Liste der aktuellen Jobs aus:
in unserem Fall ist der erste in der Liste unser Job. Er befindet sich noch im Pending-Zustand (“PD”), da er noch auf die Freigabe der benötigten Ressourcen wartet. Wenn der Job läuft, dann bekommt er den Status Running (“R”) zugewiesen.

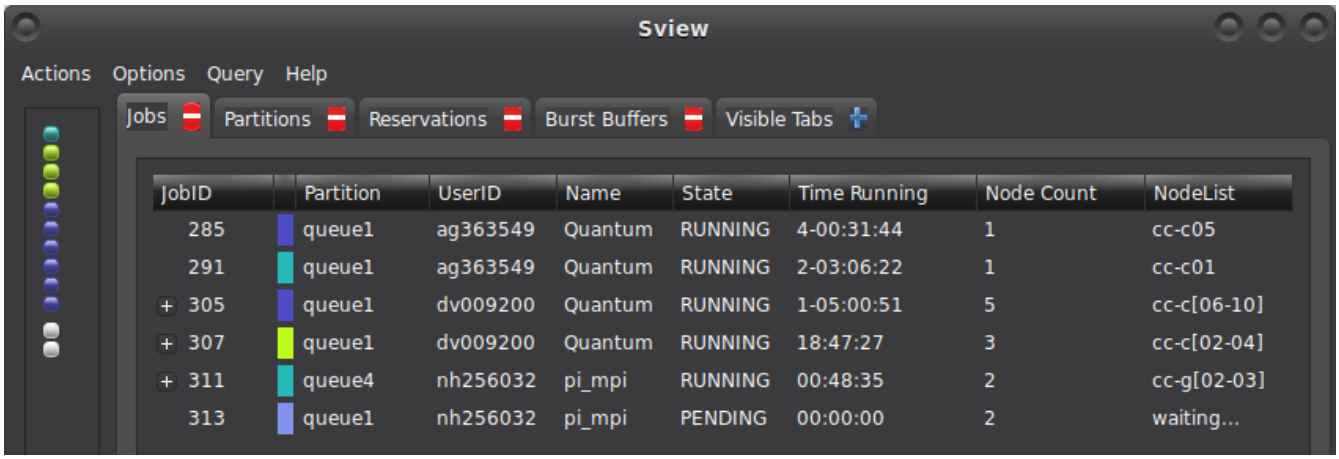
```
nh256032@cc-main /home/public/public/software_demos_und_projekte/qsub_test_II/Beispiel_pi/pi_mpi $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               313    queue1   pi_mpi nh256032 PD       0:00      2 (Resources)
               285    queue1  Quantum ag363549  R 4-00:34:42      1 cc-c05
               291    queue1  Quantum ag363549  R 2-03:09:20      1 cc-c01
               305    queue1  Quantum dv009200  R 1-05:03:49      5 cc-c[06-10]
               307    queue1  Quantum dv009200  R   18:50:25      3 cc-c[02-04]
```

Wichtig:

Befehl: *scancel [JobID]* Löscht den Job mit der entsprechenden ID.
z.B. *scancel 313* würde den Job mit der ID 313 abbrechen.

Zur Übersicht:

Befehl: *sview* Zeigt ein grafisches Fenster mit Informationen über die Systemauslastung. Die Reiter enthalten unter anderem Informationen zu den Jobs, Partitionen (Queues) und zu den einzelnen Nodes.



Auslastung der Nodes:

Ein grafisches Monitoring-System der realen momentanen Auslastung der Nodes finden Sie unter.
https://www.fh-muenster.de/pt/labore/campus-cluster/cluster_maschinenstatus.php