Vienna Scientific Cluster: Introduction

Part 1

Thomas Jatschka

PhD Seminar - Dec 19, 2021

Vienna Scientific Cluster

- collaboration of several Austrian universities
- current flagship of the VSC family is VSC-4:
  - most powerful supercomputer in Austria, 2.7 PFlop/s
  - installed in 2019
  - 790 water cooled nodes
  - rank 199 in TOP500 list[1]
- predecessor VSC-3:
  - installed in 2014
  - 2020 oil cooled nodes (retired by 2021), 0.6 PFlop/s
  - special-purpose nodes: GPUs, bioinformatics
  - extension in 2018: $\approx$ 860 nodes

[1] `https://www.top500.org/system/179697/`

Vienna Scientific Cluster

Hardware

	bc4	VSC-3	VSC-4
nodes	16	2020	790
cores/node	20	16	48
cores	320	32320	37920
CPU	Intel Xeon E5-2640 2.40GHz	Intel Xeon E5-2650/2660 2.60GHz	Xeon Skylake Platinum 8174 3.10GHz
memory	160GB	131072GB	106368GB

Access

Prerequisites:

- user account (created by project leader)
- username
- password
- phone number (2 factor authentication)

Login:

- initially access restricted to IP addresses of partner universities (physical/VPN)
- SSH-keys over port 27:
  - ssh -p 27 <username>@vsc[3|4].vsc.ac.at

Restrictions:

- each project only has a limited number of core-hours

Simple Linux Utility for Resource Management (SLURM)

Comparison between SGE and SLURM[2]:

	SGE	SLURM
job submission	qsub [script_file]	sbatch [script_file]
job deletion	qdel [job_id]	scancel qdel [job_id]
job status	qstat	squeue
job ID	$JOB_ID	$SLURM_JOBID
job array index	$SGE_TASK_ID	$SLURM_ARRAY_TASK_ID

[2] `https://srcc.stanford.edu/sge-slurm-conversion`

Job Queuing

run.slrm:

```
#!/bin/bash

#SBATCH -J example
#SBATCH -N 1
#SBATCH --ntasks=48
#SBATCH --ntasks-per-node=48
#SBATCH --ntasks-per-core=1
#SBATCH --time=00:20:00
#SBATCH --mem-per-cpu=2G
#SBATCH --mem=8G

./script1.sh
./script2.sh
...
```

Job Queuing

run.slrm:

```
#!/bin/bash

#SBATCH -J example
#SBATCH -N 1
#SBATCH --ntasks=48
#SBATCH --ntasks-per-node=48
#SBATCH --ntasks-per-core=1
#SBATCH --time=00:20:00
#SBATCH --mem-per-cpu=2G
#SBATCH --mem=8G

./script1.sh
./script2.sh
...
```

- Node allocation policy (VSC-3): only complete compute nodes can be allocated to user jobs
- project's core-hours always calculated node-wise

Queuing Single Core Jobs

```
#!/bin/bash

#SBATCH -J example
#SBATCH -N 1
#SBATCH --ntasks=48
#SBATCH --ntasks-per-node=48
#SBATCH --ntasks-per-core=1
#SBATCH --time=00:20:00
#SBATCH --mem-per-cpu=2G
#SBATCH --mem=8G

for n in {1..48}; do
  ./script.sh $n &
done
wait
```

Queuing Single Core Jobs II

```
#!/bin/bash

#SBATCH -J example
#SBATCH -N 1
#SBATCH --ntasks=48
#SBATCH --time=00:20:00

max_num_tasks=48

for n in {1..500}; do
  ./script.sh $n &
  running_tasks=`ps -C script.sh --no-headers | wc -l`
  while (($running_tasks == $max_num_tasks)); do
    sleep 10
  done
done
wait
```

Modules

- use modules to set environment variables needed for a specific application:

```
module avail
module list
module unload <xyz>
module load <xyz>
module display <xyz>
module purge
```

Modules

```
#!/bin/bash

#SBATCH -J example
#SBATCH -N 1
#SBATCH --ntasks=48
#SBATCH --ntasks-per-node=48
#SBATCH --ntasks-per-core=1
#SBATCH --time=00:20:00
#SBATCH --mem-per-cpu=2G
#SBATCH --mem=8G

module purge
module load python/3.9.4-gcc-10.2.0-y355ixy

python script.py
```

GPU Partitions

- VSC-3+ provides several GPU partitions[3]:

```
#!/bin/bash

#SBATCH -J jobname
#SBATCH -N 1
#SBATCH --ntasks-per-node=4
#SBATCH --ntasks-per-core=1

#SBATCH --partition=gpu_rtx2080ti
#SBATCH --qos=gpu_rtx2080ti

./script
```

[3] https://wiki.vsc.ac.at/doku.php?id=doku:vsc3gpuqos

Jupytherhub

- VSC offers jupyterhub: `https://vsc.ac.at/jupyterhub`
- no VPN needed

Thank you!