Introduction to ZECM (a.k.a. ZECM) HPC Cluster

Sebastian Kraus

Team IT am Institut für Chemie

October 31, 2018

rev. 1.2, 2019/02/05

1 Overview HPC computing facility at ZECM

Cluster infrastructure

Hardware equipment - Cluster partitions

- 3 partitions: *standard*, *smp* and *gpu*
  - **smp partition**: *3 big(!) nodes*, each with 3 TB memory and 64 cores (2.10GHz, Intel Xeon E7-4850 v4, AVX2)
  - **standard partition**: *132 nodes*, each with 250 GiB memory and 20 cores (2.20GHz, Intel Xeon E5-2630 v4, AVX2)
  - **gpu partition**: *21 nodes*, each with 500 GiB memory, 2x 16GB Nvidia Tesla P100 and 20 cores (2.20GHz, Intel Xeon E5-2630 v4, AVX2)
- *fast node interconnect*: Intel's proprietary Omni-Path architecture
- *Distributed Parallel File System for high-bandwidth network disk storage*: Fraunhofer's BeeGFS

Software equipment

- Workload/scheduler facility **SLURM** (**S**imple **L**inux **U**tility for **R**esource **M**anagement): managing *workload* on *cluster* of *computing nodes* grouped by *partitions*; accessed via *queues* by *jobs* demanding for *resource allocations*
- **STUBL** (**S**LURM **T**ools and **UB**i**L**ities): Collection of user-friendly wrappers for bare **SLURM** commands

Software equipment

- Available Quantum Chemistry packages:
  TURBOMOLE 7.3 (Kaupp‘s group)
  GAUSSIAN 16 (**OpenMP**, but no **LINDA**)
  MRCC (in planning)

- Available Developer modules: Compiler suites: **GNU** 7.2.x and **Nvidia CUDA** 9.0.x/9.2.x compilers; MPI: **Open MPI** 3.0.x/3.1.x

- *Cluster logon* via:

```
ssh username@gateway.hpc.tu-berlin.de
```

Concepts of SLURM workload manager

SLURM: Design aspects

- Purpose *Fair sharing of compute resources* by *quoting* wrt. to jobs and users on *clusters* (and federations)
- Jobs and Steps *consuming computing resources* via *allocation*: *cores*, *runtime*, *main memory*, *accelerator cards*, further *general resources* . . .
- Queueing job's priority: Weighted by aspects of *runtime*; available and used *resources* accounted per *user/group/project*

SLURM: Functioning and lexical terms

- General terms: *job* (general resource allocation), *job step* (separate tasks/processes within job), *task* (an OS-process), *core* (no hyper-threading per default)
- A *job* creates a context for one or several *tasks*, each running on a single *core*. A *job* can be divided into several *steps* each launched by *srun*.
- Compared to *Sun's Gridendgine*: **SLURM** without concept of *parallel environments*, direct implementation of distributed jobs (*tasks per node*, see below)

Using SLURM with interactive and batch jobs

SLURM commands - Essentials

- General command line interfaces (see **SLURM** manpages):

  1. *Resource allocation*: salloc
  2. *Running jobs*: sbatch (batch jobs via scripts)
     srun (interactive jobs and job tasks)
  3. *Cluster/Queue status*: sinfo, squeue
     sqstat and snode (**STUBL**)
  4. *Job status/cancelling*: sstat, scancel
  5. *Job and Cluster configuration*: scontrol

Launching interactive jobs - An Example

```
srun -p[standard|smp|gpu] -t20:00 --mem=2[T|G|M|K] -n8 -N4
--ntasks-per-node=2 --gres=gpu:tesla:1 --pty /bin/bash -il
```

1. --*mem*: resident memory per node in TiB, GiB, . . .
2. -*t*: wall time of your job
3. -*p*: selection of partition
4. --*mem*: resident memory per node
5. -*n*: overall number of tasks (processes) (digit, no range allowed)
6. -*N*: number of nodes (digit or range allowed; e.g.: -N2-4)
7. --*ntasks-per-node*: tasks (processes) per node (digit, no range allowed)
8. --*gres*: general resources (here GPU)
9. --*pty*: interactive shell on pseudo terminal

General batch job script

- **Writing scripts - Directives with sbatch:**

```
#!/bin/bash --login

#SBATCH -J JOB_NAME
# IMPORTANT: memory limit; otherwise all node memory allocated
#SBATCH --mem=1GB  # 1GiB resident memory per NODE
# #SBATCH --mem-per-cpu=1GB  # 1GiB resident memory per CORE
#SBATCH -N 4 # number of nodes (digit or range allowed)
#SBATCH -n 8 # overall number of tasks (digit; no range allowed)
# #SBATCH --ntasks-per-node=2 # tasks per node (digit; no range allowed)
#SBATCH -t 00:30:00  # 30 minutes wall clock time
# #SBATCH --gres=gpu:tesla:1  # no of GPU accelerators
#SBATCH -p smp  # runnable on hosts of partition smp
```

General batch job script

- **Continuation - Directives with sbatch:**

```
# setting CWD if
# different from directory where sbatch having been launched
# #SBATCH -D /path/to/target/dir

# general batch job output
# formatting:: %N: hostname; %j: $SLURM_JOBID; %x: jobname
#SBATCH -o stdout-%x-%j-%N.out
#SBATCH -e stderr-%x-%j-%N.out
# mail options
# #SBATCH --mail-type=<BEGIN|END|FAIL|ALL|NONE>
# #SBATCH --mail-user=name@postbox.de

# only if needed
# export MODULEPATH=/path/to/module/directory:$MODULEPATH
module load [your_module]
[your_command] < [input] > [output]
```

Advanced SLURM features

- *Array jobs* (available with sbatch): -a <id_range[:step]> (e.g. -a 0-12:4)
- *Job dependencies*: -d <[after|afterany|afterok|...]:job_id[:job_id]> (see `man srun`)
- *Exclusive Jobs and Job Steps*: --exclusive=<user|group> (read `man srun` carefully before using)
- *Heterogeneous job types*: see **SLURM** manpages and also https://slurm.schedmd.com/heterogeneous_jobs.html

Running batch jobs - An example with TURBOMOLE

Batch job script - An example with TURBOMOLE

- **TURBOMOLE with MPI:**

```
#!/bin/bash --login

#SBATCH -J dscf_MPI
# IMPORTANT: memory limit; otherwise all node memory allocated
#SBATCH --mem=2GB
# #SBATCH --mem-per-cpu=1GB
#SBATCH -N 4
#SBATCH -n 8
# #SBATCH --ntasks-per-node=2
#SBATCH -t 00:30:00
#SBATCH -p smp

export MODULEPATH=/home/units/Fak_II/quantenchemie/\
turbomole/modules:$MODULEPATH

export PARA_ARCH=MPI
module load turbomole_7.3.0
dscf control > dscf.out
```

Working on cluster - Best practice

Best practice on ZECM HPC

- write temporary data to directory */scratch/[your_user_name]* (**actually on BeeGFS, no node-local scratch**)
- Always define the number of tasks per job via *-N* - **even when running intra-node jobs** (cf. `man srun`)
- Always define your job‘s memory requirements via switches: *--mem=* or *--mem-per-cpu=* (standard for *--mem-per-cpu* predefined on this cluster)
- *new job types*: plan to run short test jobs in order to *acquire necessary amount of resident main memory and probable runtime*

Literature and references

Finding further information

- Further references ZECM HPC and **SLURM**

  Official documentation ZECM HPC (login with TUB account):
  https://hpc.tu-berlin.de

  Documentation ZECM HPC in team IT's Wiki:
  http://it.chem.tu-berlin.de/wiki/doku.php?id=hpc:hpc_ZECM:start

  ISIS course/forum (login with TUB account):
  https://isis.tu-berlin.de/course/view.php?id=13680

  Official **SLURM** documentation with online manual pages:
  https://slurm.schedmd.com

  Additional **SLURM** documentation:
  http://www.lrz.de/services/compute/linux-cluster/batch_parallel