HPS

https://hps.vi4io.org

Patrick Höhn, Ruben Kellner, Markus Boden, Sebastian Krey, Azat Khuziyakhmetov

High-Performance System Administration

Introduction to Slurm

Table of contents

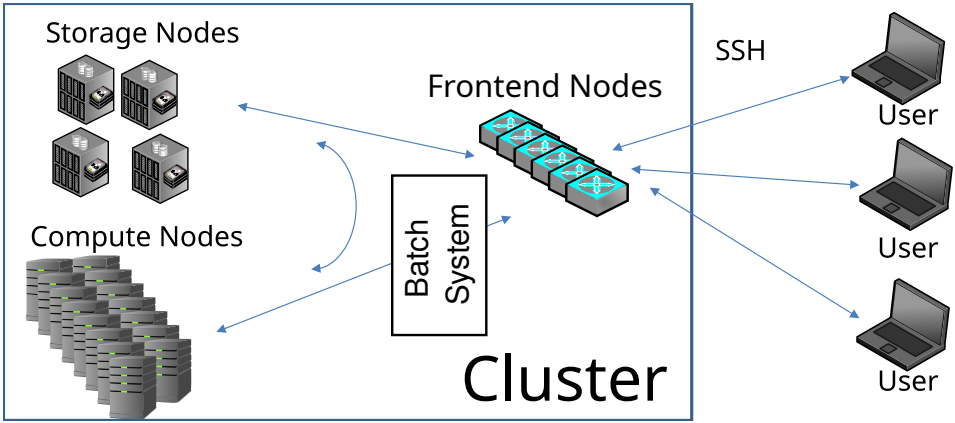
Learning Objectives

After the course the students should be able to:

- Comprehend there are login and compute nodes with different functionality
- Understand the basics of the HPC infrastrucutre
- Use a workload manager like SLURM to allocate HPC resources (e.g. CPUs) and to submit a batch job.
- Run parallel programs in an HPC environment.
- Compile programs on the HPC Cluster
- Run Interactive Sessions with a graphical user interface

Table of Contents, Part II

What is a Cluster?



Basic Concepts

Cluster	A collection of networked computers intended to provide compute capabilities.
Node	One of these computers, also called host or server.
Frontend	Special node provided to interact with the cluster. `login-mdc.hpc.gwdg.de` in our case.
Job	Execution of a program consisting of one or several parallel tasks.
Partition	A pool of nodes of equal or similar type.
Batch System	Management system distributing job tasks across job slots.

Batch Systems

- Moab Cluster Manager.
- PBS Pro.
- Slurm.
- Spectrum LSF.
- Oracle Grid Engine.
- TORQUE Resource Manager.

Batch Systems

- Moab Cluster Manager.
- PBS Pro.
- **Slurm.**
- Spectrum LSF.
- Oracle Grid Engine.
- TORQUE Resource Manager.

Basic Concepts 2

Serial job Job consisting of one task using one job slot.

SMP job Job with shared memory parallelization, i.e. all processes need access to the memory of the same node. $\Rightarrow$ several job slots **on the same node**.

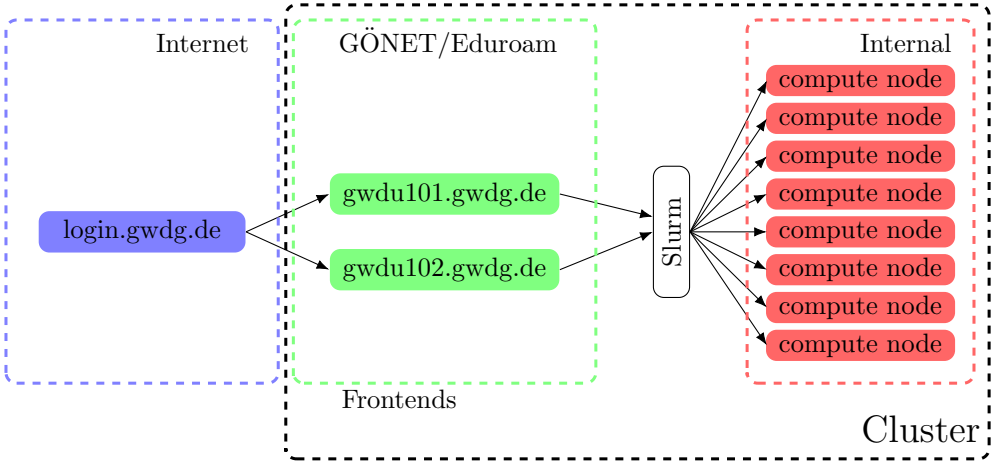
MPI job Job with distributed memory parallelization, realized with MPI. Can use several job slots on several nodes. Needs to be started with a helper program, e.g., `mpirun or srun`.

Single node vs. Multi node computations

- MPI jobs are a lot of independent tasks that usually use one core each.
  - Slurm calls these `tasks` with `srun`.
- Single node jobs are usually just one task with many cores.
- Both can be combined into hybrid jobs: multiple tasks using multiple cores each.

Getting started with Slurm

SCC Configuration



Available Partitions at SCC

General purpose partitions:

medium General purpose partition, well suited for most jobs. Up to 1024 cores per job.

fat Up to 512 GB in one host.

fat+ For extreme memory requirements. Up to 2048GB per host.

Special purpose partitions:

gpu For jobs using GPU acceleration.

int For interactive jobs, i.e. jobs which require a shell or a GUI.

The fat+ partition

The `fat+` partition contains:

- 5 nodes with 1.5Tb Memory
- 1 node with 2Tb Memory

Usage recommendations:

- Work your way up. Start in `fat` and only use `fat+` if your jobs runs out of memory.
- Use `reportseff`, see if your job really is memory bound
- When unsure, ask us!
- `--mem` or `--mem-per-cpu` is mandatory

Telling Slurm what to do

- `srun` submits information on your job to Slurm.
  - What is to be done? (path to your program and required parameters)
  - What are its requirements? (e.g. which nodes, number of tasks, maximum runtime)
- Slurm matches the jobs requirements against the capabilities of the nodes.
- When suitable free resources are found, the job is started.
- Slurm prioritizes the jobs based on a number of factors.

Your first job

Use `srun` to submit a job to Slurm.

```
srun <program>
```

Example:

```
gwdu101:27 12:53:50 ~ > hostname
gwdu101
gwdu101:27 12:53:53 ~ > srun hostname
amp078
gwdu101:27 12:53:56 ~ > srun hostname -f
amp078.global.gwdg.cluster
```

Submitting a job

```
srun <parameters> <program>
```

common parameters

-p <partition>	partition.
-t <hh:mm:ss>	maximum runtime. If this is exceeded the job is killed.
-n <i >	number of tasks.

Interactive Jobs

srun: Interactive jobs

--x11	adds X11 (GUI) forwarding. This requires that you connect to the frontend with `ssh -Y` and your local machine supports X-Windows.
-p int	use the interactive partition. In `int` the nodes have no slot limit. They will take jobs until their load crosses a specified threshold, so jobs start immediately.
--pty	interactive mode

Interactive X11 Job

Running Matlab

```
> ssh -Y login-mdc.hpc.gwdg.de
> module load matlab
> srun --x11 -p int matlab -desktop
```

- The job will be dispatched and as soon as an available node is found and the Matlab interface will start.

Interactive Console Job

Running python interactively

```
> ssh login-mdc.hpc.gwdg.de
> module load anaconda3
> srun --pty -p int python3
>>> import socket, os
>>> print(socket.gethostname())
>>> print(os.system("slurm_resources"))
```

Resource selection: CPU

srun options for parallel (SMP or MPI) jobs.

-N <min>-<max>, --nodes=<min>-<max>	Minimum and maximum node count. You can also specify the exact number.
-n,--ntasks=<n>	Number of tasks (not equally distributed!)
--tasks-per-node=<n>	Tasks per node. If used with -n it denotes the maximum number of tasks per node.
-c,--cpus-per-task=<n>	CPUs per tasks.

A note on -n vs. -c

Rule of thumb

- -c for single node jobs
- -n for MPI jobs

A note on -n vs. -c

Rule of thumb

- -c for single node jobs
- -n for MPI jobs

Rule of thumb 2

If you are unsure if your program uses MPI, then it does not.

Resource Selection: Memory

`srun` options

`--mem <size[K|M|G|T]>`	Memory per node.
`--mem-per-cpu<size[K|M|G|T]>`	Memory per core.

- without options:
  - each partition has a `DefMemPerCPU` option
  - can be retrieved via `scontrol show partition <name>`

Non interactive Jobs

Problem

- if you have big jobs, your queue time will be long
- `srun` needs you to stay logged in
- jobs can run for days

Non interactive Jobs

Problem

- if you have big jobs, your queue time will be long
- `srun` needs you to stay logged in
- jobs can run for days

Solution

Batch Jobs!

sbatch: Using Job Scripts

A job script is a shell script with a special comment section.

The #SBATCH lines have to come first!

sbatch: Basic job script example

```
1 #!/bin/bash
2 #SBATCH -p medium
3 #SBATCH -t 10:00
4 #SBATCH -o job-%J.out
5
6 slurm_resources
```

Submit with:

```
sbatch <script name>
```

Jobscripts

- a job script is essentially a normal script
- usually bash/shell, but can be any scripting language (R, python, perl)
- #SBATCH lines need to be at the top!
- only lines starting exactly with #SBATCH are parsed
- you can copy files, load modules, and do any scripting you want
- for MPI, use `srun` or `mpirun` to start your program

Useful options for batch job submssion

```
sbatch <slurm options> jobscript
```

--mail-type=<TYPE>	get mail notifications (type: `BEGIN`, `END`, etc.)
--mail-user=<address>	Default: `${USER}@gwdg.de`
-o/-e <file>	Store job output in file (slurm-<jobid>.out by default). `%J` in the filename stands for the jobid.

Slurm Commands

sinfo Info about the system and partitions.

```
sinfo -p <partition>, -t <state>
```

squeue Show the job queue.

```
squeue -p <partition>, --me
```

scancel Cancel Job

```
scancel <JobID>
scancel -p <partition>|-u $USER
```

Batch Script for Task Distribution in `medium` partition

```
#SBATCH -p medium
#SBATCH -N 10
#SBATCH --ntasks-per-node 24
#SBATCH -o job-%J.out

module purge
module load intel-oneapi-compilers  intel-oneapi-mkl intel-oneapi-mpi namd

srun namd2 +setcpuaffinity apoa1.namd
```

Memory is faster then network!

Try to spread your tasks to as little nodes as possible.

Job Disk Space Usage Options

/local	Local hard disk of the node. SSD based and therefore a very fast option for storing temporary data. Automatic file deletion. A temporary directory is created on all nodes at `$TMP_LOCAL`.
/scratch	Shared scratch space, available on most nodes, but there are two instances (use `-C scratch`). Very fast, no automatic file deletion, but also no backup! Files may have to be deleted manually when we run out of space.
$HOME	Available everywhere, permanent, with backup. Personal disk space can be increased. Comparably slow.

Recipe: Combine shared memory and MPI

Running hybrid jobs

```
#SBATCH --partition medium
#SBATCH --nodes 5
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=6
#SBATCH --output job-%J.out

module purge
module load openmpi/gcc

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

srun hybrid_job
```

Longer or shorter jobs

The `--qos` parameter

- Default maximum runtime: 2 days
- `--qos=` <qos> can select a QoS
- Two extra QoS available:

short for shorter jobs (max. 2h), has higher priority, limited job slots

long longer jobs (max. 5d), limited job slots.

But my job is even longer

- try parallelizing more
- break it down into smaller steps
- check, if your software supports checkpoints
- check again!
- contact us

More Slurm Commands

`scontrol show [partition|node|job] <x>` where x should be a node name, JobID or partition name.

`sprio` Priority information about pending jobs

`sacct` Get information about a job after it finished

```
-j <jobid>
--format=JobID,User,JobName,MaxRSS,Elapsed,Timelimit
```

Using the gpu partition

GPU parameters

-G | --gpus=[type:]<n>	requests n GPUs of type
--gpus-per-task=[type:]<n>	requests n GPUs of type per task
--gpus-per-node=[type:]<n>	requests n GPUs of type per node

- CPUs are evenly distributed for every GPU
- Available types are:
  - rtx5000
  - v100
  - gtx1080
- See: `sinfo -p gpu --format=%N,%G`

Debugging

- take a look at your output files, while the job is running:
  - tail -f /path/to/output
- take a look at the jobs, while it is running
  - you can `ssh` into every node that currently calculates your job
  - use `htop` to see the processor and ram usage

Debugging

Read the extra job information

```
================================================================
JobID = 4383174
User = mboden, Account = admin
Partition = gpu, Nodelist = dge[001,006]
================================================================
[job output]
============ Job Information ===========================================
Submitted: 2020-04-24T17:35:41
Started: 2020-04-24T17:35:41
Ended: 2020-04-24T17:45:45
Elapsed: 10 min, Limit: 60 min, Difference: 50 min
CPUs: 2, Nodes: 2
============= ProfiT-HPC ===============================================
To generate the ProfiT-HPC text report, run the following command
profit-hpc 4383174
================================================================
```

Take a look at all the information. Is it as expected?

Debugging

Read your errors!

```
slurmstepd: error: Detected 1064 oom-kill event(s) in step XXXXXX.0 cgroup.
Some of your processes may have been killed by the cgroup out-of-memory handler.
srun: error: gwda024: task 3: Out Of Memory
```

Might have something to do with memory!

Have a look at your jobs memory with:

`reportseff JOBID`

Information sources

- man pages
- Slurm online help
  - For example: `sbatch --help`
- GWDG scientific compute cluster documentation
  - https://docs.gwdg.de/doku.php?id=en:services:application_services:high_performance_computing:start
- GWDG scientific compute cluster user wiki
  - https://info.gwdg.de/wiki/doku.php?id=wiki:hpc:start
- HPC announce mailing list
  - https://listserv.gwdg.de/mailman/listinfo/hpc-announce

Rocket.Chat

- Rocket.Chat Channel at: `https://chat.gwdg.de/channel/hpc-users`
- Mainly for quick questions
- Usually faster response but we might tell you to open a ticket

Using the GWDG Support Ticket System

- Write an email to *hpc-support@gwdg.de*
- State your user id ($USER)
- If you have a problem with jobs, **always** include:
  - Job IDs
  - standard output (`-o <file>`)
  - standard output (`-e <file>`)
- If you have a lot of failed jobs send at least two outputs. You may also list the jobid's of all failed jobs.
- If you don't mind us looking at your files, please state this in your request
  - You may limit your permission to specific directories or files

GöHPCoffee

- Bi-weekly meeting between users and admins.
- Format: Short focus session and open discussions and Q&A afterwards
- More Information at https://docs.gwdg.de/doku.php?id=en:services:application_services:high_performance_computing:hpc_coffee

The End

Thank you all for your attention!

Questions?

- We connected to the cluster
- We managed sessions
- Do you need additional information?