Short Slurm Introduction

Run programs on CLAIX using the batch system SLURM

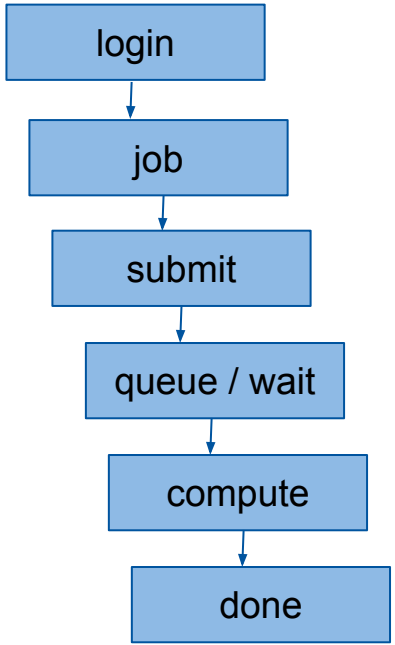
NHR4 CES
NHR for Computational Engineering Science

slurm workload manager

RWTH AACHEN UNIVERSITY

Batch system for CLAIX

- SLURM Batch system:
  - We all share hardware and time resources “fairly”
  - **Queue**’s user programs as **jobs**
  - Considers **priority** to decide order
  - **Allocates** time and resources to **“jobs”** from users.
  - Starts, executes, and monitors **jobs**.
  - Jobs end, users see their results.

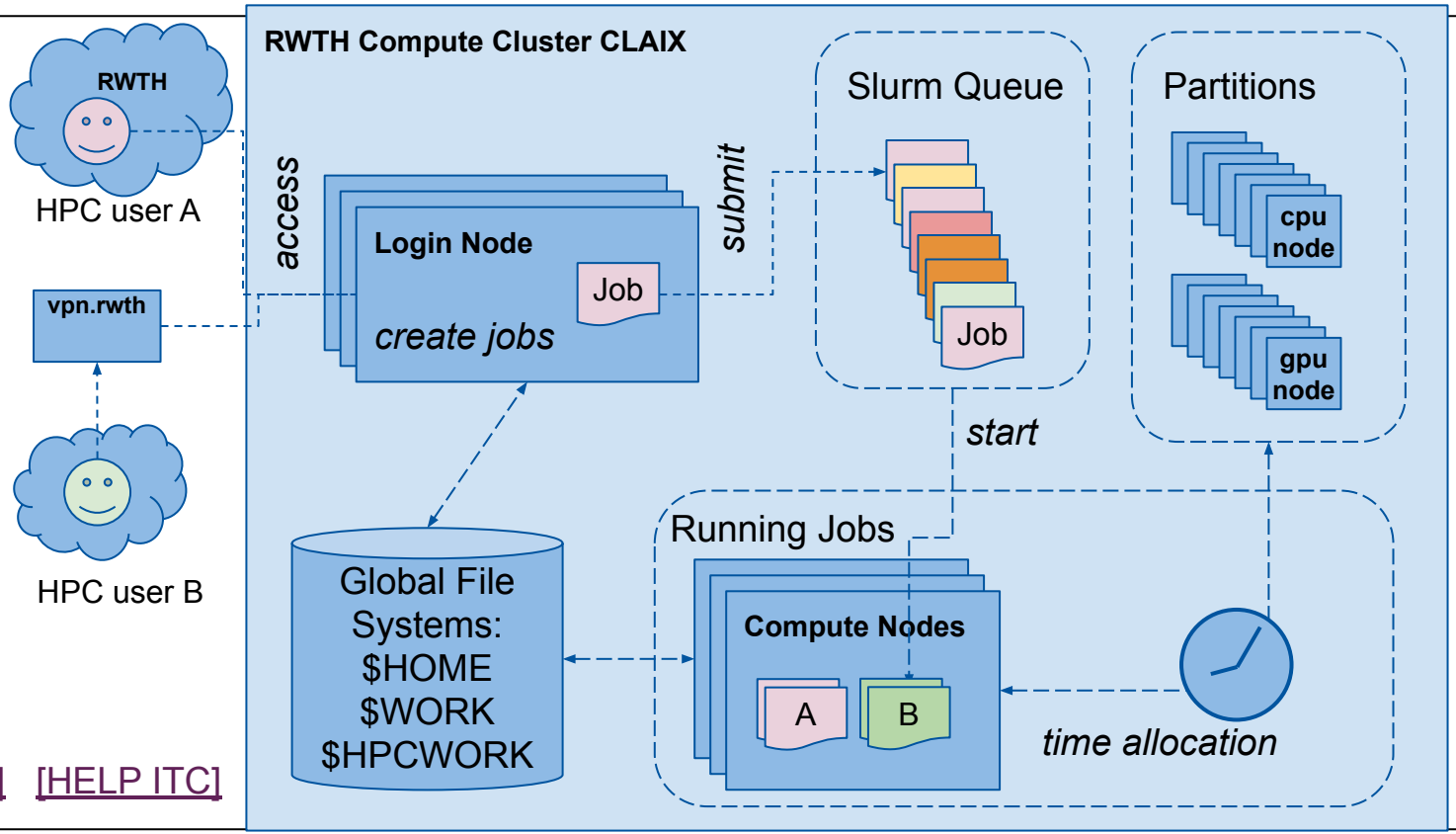


22.05.4

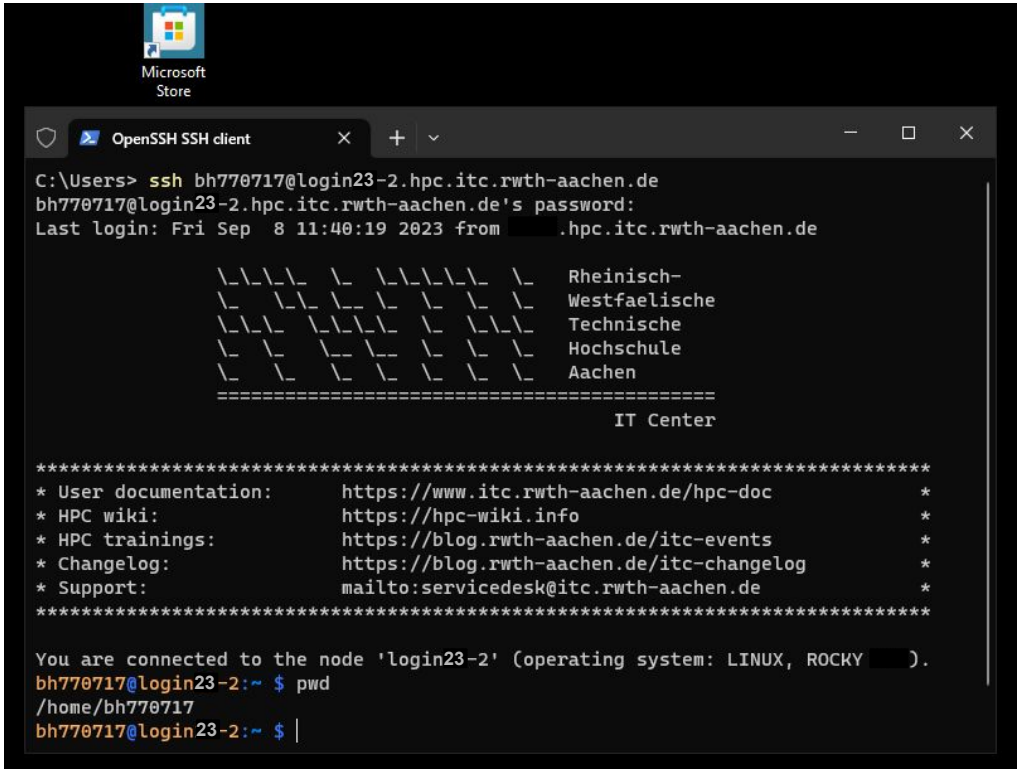
[HELP ITC]

SLURM: Resource Manager + Scheduler

Batch system for CLAIX



Login - SSH - NOT SLURM



Login Nodes Do's:

- Code, Compile, Edit
- SSH to other nodes
- Slurm commands
- Basic GUI programs

Login Nodes Dont's:

- Large data processing
- MPI, PARALLEL, intensive programs
- Use all resources
- Monitoring

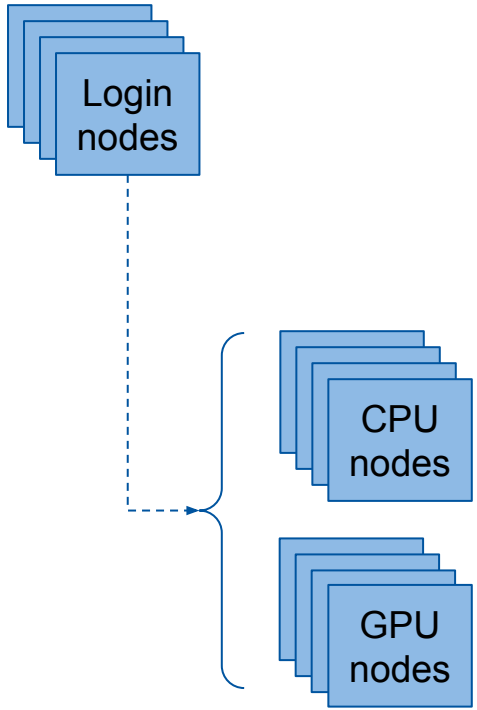
- **We Limit Cores + Mem**
- **We will kill the programs and you will lose the work**
- **Login nodes are restarted mondays**

Compute nodes

- Submit jobs:
  - **Submit** programs to compute nodes
  - **Compute nodes** is where computations/tests/work are done
  - Queue/wait for requested resources

- Commands
  - **salloc** <args> run cmd's interactively
  - **sbatch** script.sh submit batch script job

- The **login-nodes** are NOT the "compute cluster"
  - Still you can:
    - Compile, run, test programs and work



Compute nodes - salloc

- An interactive session with **salloc** needs:
  - **Resources:** CPUs, GPUs, Nodes, Time
  - **Optional partition**: Group of nodes: use **devel** for quick tests!

```
[redacted] ~ $ salloc -n 8 -N 2 -p cXXm --time=01:00:00 --mem=7G --account=supp0001
salloc: [I] No output file given, set to: output_%j.txt
salloc: Pending job allocation 39234937
salloc: job 39234937 queued and waiting for resources
salloc: job 39234937 has been allocated resources
salloc: Granted job allocation 39234937
salloc: Waiting for resource configuration
salloc: Nodes ncm[0707,0713] are ready for job
```

Partition devel (no account):

```
[redacted]:~ $ salloc -n 8 -N 2 -p devel
```

Compute nodes - salloc

```
                    :~ $ salloc -n 8 -N 2 --time=01:00:00 --mem=7G
salloc: [I] No output file given, set to: output_%j.txt
salloc: Pending job allocation 39214003
salloc: job 39214003 queued and waiting for resources
salloc: job 39214003 has been allocated resources
salloc: Granted job allocation 39214003
salloc: Waiting for resource configuration
salloc: Nodes ncm[0229,0238] are ready for job
```

```
You are connected to the node 'ncm0229' (operating system: LINUX, ROCKY 8.8).
bh770717@ncm0229:~ $
```

Compute nodes - salloc

```
You are connected to the node 'ncm0229' (operating system: LINUX, ROCKY 8.8).
bh770717@ncm0229:~ $ exit
-------------------------------------------------------------------------------
Bis zum naechsten Mal, bh770717!
-------------------------------------------------------------------------------
salloc: Relinquishing job allocation 39214003
bh770717@login[redacted]-1:~ $
```

Compute nodes - sbatch

- Define a Slurm Job (a file):
  - **Resources:** CPUs, GPUs, Nodes, Time
  - **Partition**: Group of nodes
  - **Account**: Project account, resources get billed
  - **Modules:** Our provided libraries and programs
  - **Your Program:** commands in a script file.
- Code it using either:
  - Console editors: vim, nano, pico, emacs, …
  - GUI editors, PLUMA, EMACS, …

Sample batch script for sbatch

batch_script.sh

```
1   #!/usr/bin/zsh
2
3   ### Job Parameters
4   #SBATCH --ntasks=8                # Ask for 8 MPI tasks
5   #SBATCH --time=00:15:00           # Run time of 15 minutes
6   #SBATCH --job-name=example_job    # Sets the job name
7   #SBATCH --output=stdout.txt       # redirects stdout and stderr to stdout.txt
8   #SBATCH --account=<project-id>    # Replace with your project-id or delete the line
9
10  ### Program Code
11  srun hostname
```

Submit batch job to sbatch

```
> sbatch batch_script.sh
Submitted batch job 12345678
```

```
> squeue --me
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          12345678     c23ms example_job  AB123456  R       0:02      1 n23m0023
```

[HELP ITC]

DEMO

Important!

- User programs should run as slurm jobs.
- Slurm jobs can be interactive jobs or batch (non-interactive) jobs.
- The primary interface with SLURM is the text console!

Quota system

Slurm Accounts (not = your user)

- Accounts have a **default** partition and **„allowed**“ partitions
  - The account **„default**“ has **„cXXm**“ as default partition and is allowed **„cXXg**“
- Submission without a project results in submission to the **„default**“ account
- In which projects am I involved?
  - Use „**r_wlm_usage -p <projectname/account>**“
  - Shows
    - Allowed partitions
    - Max usable cores per job
    - Max runtime limit per job
    - Consumed corehours of the last 4 weeks
    - Consumed corehours up to now and total granted corehours
- **r_wlm_usage -q** shows user quota information

Quota system

Quota

- Projects have a quota of corehours **granted** to be used use **per month FOR the month**

- But: we provide a „**3-month-window**“ to use the quota:
  - This month you could use unused quota from the previous month and „borrow“ quota from the next month.

- WARNING! Using more than 3x your Monthly Quota:
  - Your jobs will **only** start, if no one else is using the CLAIX.

Slurm as the batch system for CLAIX

Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now
- Priority
  - At least one other job has a higher priority and will run first on the same resources
- Resources
  - The job is waiting for resources to become free
- AssocMaxWallDurationPerJobLimit
  - The job requested a longer runtime than it is allowed
- AssocMaxCpuPerJobLimit
  - The job requested more cpus than it is allowed
- JobArrayTaskLimit
  - The job array has more running tasks than allowed
- Dependency
  - The job is waiting for another specific job to end

Best Practices

- Avoid loading modules in your `.zshrc` or `.bashrc`.
  - Do not automatically load conda environments.
- Use the **module** version of Programs provided by us when possible.
  - Avoid your own conda environments when possible.
  - Use module spider <name> to search for already installed software
- Purge modules (module purge) when uncertain of library dependencies
- Test your application on **devel** nodes with smaller problem sizes using **salloc**.
- Put all commands the program at the **end** of the batch job script file.
- Don't forget the module commands and the shebang (#!):
  - `#!/usr/bin/zsh`
- If you use relative paths: know the directory in which the job starts.
- Make the scripts executable (`chmod +x myscript.sh`)
- Confirm your program's memory / stack requirements fit in your job memory request.
- Avoid moving millions of tiny files between **GFS** and compute nodes.
  - Read once, distribute within the program.
- Do not run out of disk quotas!
- Use your project quota!