
# Introduction to Slurm

RWTH AACHEN UNIVERSITY

## General Introduction

- What is a batch system and why do I need it?

## General Introduction

- What is a batch system and why do I need it?
- The login-nodes are NOT „the cluster“.

## General Introduction

- What is a batch system and why do I need it?
- The login-nodes are NOT „the cluster“.
- So, what is the cluster?

# Batch system for CLAIX and beyond

## General Introduction

- What is a batch system and why do I need it?
- The login-nodes are NOT „the cluster“.
- So, what is the cluster?
- Use the batch system for calculations!

# Slurm as the batch system for CLAIX

## First batchscript

- Slurm needs a batchscript
- Submit batchscript with „sbatch <batchscript>“
- SHEBANG (very first line) of the batchscript should be

#!/usr/local_rwth/bin/zsh

## Sessions learned from the output

- sbatch: [I] No output file given, set to: output_%j.txt
  - no jobname was given within the jobscript
  - Slurm set the name for you
  - the %j is a placeholder for the jobid

## Sessions learned from the output

- sbatch: [I] No output file given, set to: output_%j.txt
  - no jobname was given within the jobscript
  - Slurm set the name for you
  - the %j is a placeholder for the jobid

- sbatch: [I] No runtime limit given, set to: 15 minutes
  - Slurm needs a runtime limit, as none was given, Slurm set it to 15 minutes

## Sessions learned from the output

- sbatch: [I] No output file given, set to: output_%j.txt
  - no jobname was given within the jobscript
  - Slurm set the name for you
  - the %j is a placeholder for the jobid

- sbatch: [I] No runtime limit given, set to: 15 minutes
  - Slurm needs a runtime limit, as none was given, Slurm set it to 15 minutes

- Submitted batch job 12757911
  - that is the jobid
  - the name of the output file will be output_12757911.txt

# Slurm as the batch system for CLAIX

## Sessions NOT learned from the output

- the job gets one core
- the job gets 3900 MB memory
- the partition was set to „c18m“
  - the default partition for jobs without a project

# Slurm as the batch system for CLAIX

## Partitions

| name | #nodes | cpu arch | #cores / node | #mem / node | #mem / core | accel |
|---|---|---|---|---|---|---|
| c18m | 1240 | Skylake | 48 | 187.200 | 3.900 | |
| c18g | 54 | Skylake | 48 | 187.200 | 3.900 | 2 * Volta |
| c16m | 608 | Broadwell | 24 | 124.800 | 5.200 | |
| c16s | 8 | Broadwell | 144 | 1.020.000 | 7.050 | |
| c16g | 9 | Broadwell | 24 | 124.800 | 5.200 | 2 * Pascal |

## Influence job behaviour

- Use the magic cookie „#SBATCH“ in the jobscript
  - We would like to strongly encourage you, to NOT use commandline parameters, as this makes support harder for us

## Influence job behaviour

- Use the magic cookie „#SBATCH“ in the jobscript
  - We would like to strongly encourage you, to NOT use commandline parameters, as this makes support harder for us

- Parameters from sessions learned (or not)
  - Jobname
    - #SBATCH -j <jobname> or #SBATCH --job-name=<jobname>

## Influence job behaviour

- Use the magic cookie „#SBATCH“ in the jobscript
  - We would like to strongly encourage you, to NOT use commandline parameters, as this makes support harder for us

- Parameters from sessions learned (or not)
  - Jobname
    - #SBATCH -j <jobname> or #SBATCH --job-name=<jobname>
  - Runtime limit
    - #SBATCH -t <d-hh:mm:ss> or #SBATCH --time=<d-hh:mm:ss>

## Influence job behaviour

- Use the magic cookie „#SBATCH“ in the jobscript
  - We would like to strongly encourage you, to NOT use commandline parameters, as this makes support harder for us

- Parameters from sessions learned (or not)
  - Jobname
    - #SBATCH -j <jobname> or #SBATCH --job-name=<jobname>
  - Runtime limit
    - #SBATCH -t <d-hh:mm:ss> or #SBATCH --time=<d-hh:mm:ss>
  - Number of requested tasks (processes)
    - #SBATCH -n <numtasks> or #SBATCH --ntasks=<numtasks>

# Slurm as the batch system for CLAIX

## Influence job behaviour

- Use the magic cookie „#SBATCH“ in the jobscript
  - We would like to strongly encourage you, to NOT use commandline parameters, as this makes support harder for us

- Parameters from sessions learned (or not)
  - Jobname
    - #SBATCH -j <jobname> or #SBATCH --job-name=<jobname>
  - Runtime limit
    - #SBATCH -t <d-hh:mm:ss> or #SBATCH --time=<d-hh:mm:ss>
  - Number of requested tasks (processes)
    - #SBATCH -n <numtasks> or #SBATCH --ntasks=<numtasks>
  - Amount of memory per requested core
    - #SBATCH --mem-per-cpu=<mem in MB>

# Slurm as the batch system for CLAIX

## Influence job behaviour

- Use the magic cookie „#SBATCH“ in the jobscript
  - We would like to strongly encourage you, to NOT use commandline parameters, as this makes support harder for us

- Parameters from sessions learned (or not)
  - Jobname
    - #SBATCH -j <jobname> or #SBATCH --job-name=<jobname>
  - Runtime limit
    - #SBATCH -t <d-hh:mm:ss> or #SBATCH --time=<d-hh:mm:ss>
  - Number of requested tasks (processes)
    - #SBATCH -n <numtasks> or #SBATCH --ntasks=<numtasks>
  - Amount of memory per requested core
    - #SBATCH --mem-per-cpu=<mem in MB>
  - For the sake of completeness, the partition
    - This should not be needed, as the job modifier chooses the partition for you
    - #SBATCH -p <partition> or #SBATCH --partition=<partition>

# Slurm as the batch system for CLAIX

## Further parameters

| | |
|---|---|
| project (account) | -A <account> or --account=<account> |
| shared memory job | --ntasks=1 --cpus-per-task=<numthreads> |
| distributed memory job | --ntasks=<numtasks> |
| hybrid job, „r“ MPI ranks, „p“ tasks per node, „t“ threads per task | --ntasks=<r> --tasks-per-node=<p><br>--cpus-per-task=<t> |
| gpu | pascal: --gres=gpu:pascal:<numgpus per node><br>volta: --gres=gpu:volta:<numgpus per node> |

## Job control

- Show job details
  - scontrol show job <jobid>
- Show queue of jobs
  - squeue –u <username>
- Cancel job
  - scancel <jobid>

# Slurm as the batch system for CLAIX

## Accounts

- Accounts have a default partition and „allowed“ partitions
  - The account „default“ has „c18m“ as default partition and is allowed to use additionally „c16g“ and „c18g“
  - Accounts are able to switch to another allowed partition
- Submission without a project results in submission to the „default“ account
- In which projects am I involved?
  - Use „r_wlm_usage -q“
  - It also tells you
    - which partitions you can use with your project
    - the max usable cores per job
    - the max runtime limit per job
    - consumed corehours of the last 4 weeks
    - consumed corehours up to now and total granted corehours

## Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now

## Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now
- Priority
  - There exists at least one other job, which has a higher priority and will run first

## Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now
- Priority
  - There exists at least one other job, which has a higher priority and will run first
- Resources
  - The job is waiting for resources to become free

## Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now
- Priority
  - There exists at least one other job, which has a higher priority and will run first
- Resources
  - The job is waiting for resources to become free
- AssocMaxWallDurationPerJobLimit
  - The job requested a longer runtime than it is allowed

## Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now
- Priority
  - There exists at least one other job, which has a higher priority and will run first
- 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

## Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now
- Priority
  - There exists at least one other job, which has a higher priority and will run first
- 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 you allowed

## Pending Reasons

- None
  - The job has not been in a schedule run of Slurm up to now
- Priority
  - There exists at least one other job, which has a higher priority and will run first
- 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 you allowed
- Dependency
  - The job is waiting for another specific job to end

# Slurm as the batch system for CLAIX

## Quota

- Projects have a contingent of corehours they are **granted** to use per month
  - This is NOT a bank account, you cannot save corehours, as you cannot save time to use it later
- Scientific workloada often do not match this
  - Introduction of the so called „3-month-window“
- It basically means, you could use unused quota from the last month and „borrow“ quota from the next month

E.g. you are allowed to use 1000 corehours per month, the actual usage and the usage from the last month are added and if this is more than three times of the grant, you will be scheduled to the so called „low“ queue
This means, your jobs will only start, if they do not hinder „normal“ jobs from starting

# Slurm as the batch system for CLAIX

## Fair use of compute hardware

- corehours vs. billingvalue
- Example node:
  - 10 cores, 100 GB memory, 2 GPU cards
    -> 10 GB is worth 1 core, 1 GPU is worth 5 cores
- Job uses 1 core, 1 GB memory and 0 GPUs -> billing is 1

# Slurm as the batch system for CLAIX

## Fair use of compute hardware

- corehours vs. billingvalue
- Example node:
  - 10 cores, 100 GB memory, 2 GPU cards
    -> 10 GB is worth 1 core, 1 GPU is worth 5 cores
- Job uses 10 cores, 1 GB memory and 0 GPUs -> billing is 10

# Slurm as the batch system for CLAIX

## Fair use of compute hardware

- corehours vs. billingvalue
- Example node:
  - 10 cores, 100 GB memory, 2 GPU cards
    -> 10 GB is worth 1 core, 1 GPU is worth 5 cores
- Job uses 1 core, 1 GB memory and 1 GPUs -> billing is 5

# Slurm as the batch system for CLAIX

## Fair use of compute hardware

- corehours vs. billingvalue
- Example node:
  - 10 cores, 100 GB memory, 2 GPU cards
    -> 10 GB is worth 1 core, 1 GPU is worth 5 cores
- Job uses 1 core, 90 GB memory and 0 GPUs -> billing is 9

# Slurm as the batch system for CLAIX

## Fair use of compute hardware

- corehours vs. billingvalue
- Example node:
  - 10 cores, 100 GB memory, 2 GPU cards
    -> 10 GB is worth 1 core, 1 GPU is worth 5 cores
- Job uses 7 cores, 80 GB memory and 1 GPUs -> billing is 8

## X11-Forwarding

- Sad to say, but X11 forwarding is not working for us for now, maybe with a later Slurm version
- Yet, we have „some kind“ of X11 forwarding for you
  - Write your jobscript, but don‘t execute your application, insert a sleep according to the --time parameter
  - Use „guishell batchscript“, a new xterm will be started for you on the starting computenode as soon as the job begins running
    - Please remark, that this is a pure ssh-session, that implies that no SLURM variables are set
    - Your ssh session will still be restricted to the requested resources though
  - Still useful for example for debugging with totalview

# Thank you for your attention.
# Any questions?