VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

ADOPTING SLURM: TRANSITIONING FROM PBS SCHEDULER

Vojtěch Gubani
Ondřej Meca

INTRODUCTION - AIMS

- SLURM is a workload manager used to allocate jobs on Barbora and Complementary systems. It will be used also on Karolina since September 2023.
- This introductory course is designed to help the users seamlessly migrate from the PBS scheduler to this newly installed job management system.
- The course describes the SLURM fundamental concepts, its job submission process, terminology, and environment variables.
- After the course, attendees should be able to create, submit, monitor, and manage computational jobs using SLURM efficiently.
- Special focus will be put on the description of differences between PBS and SLURM and how to transform PBS scripts to SLURM.

CONTENT

Theory

- Motivation
- Differences between PBS and Slurm
- Slurm setting (submitting a script to Slurm, job management, interactive jobs, environment variables)

Demonstrations

- Basic work with Slurm (basic commands, creating scripts, interactive jobs)
- Transitioning PBS scripts to Slurm
- Starting parallel jobs (environment variables, threads, MPI, mapping, pinning)

SLURM – WHAT IS SLURM?

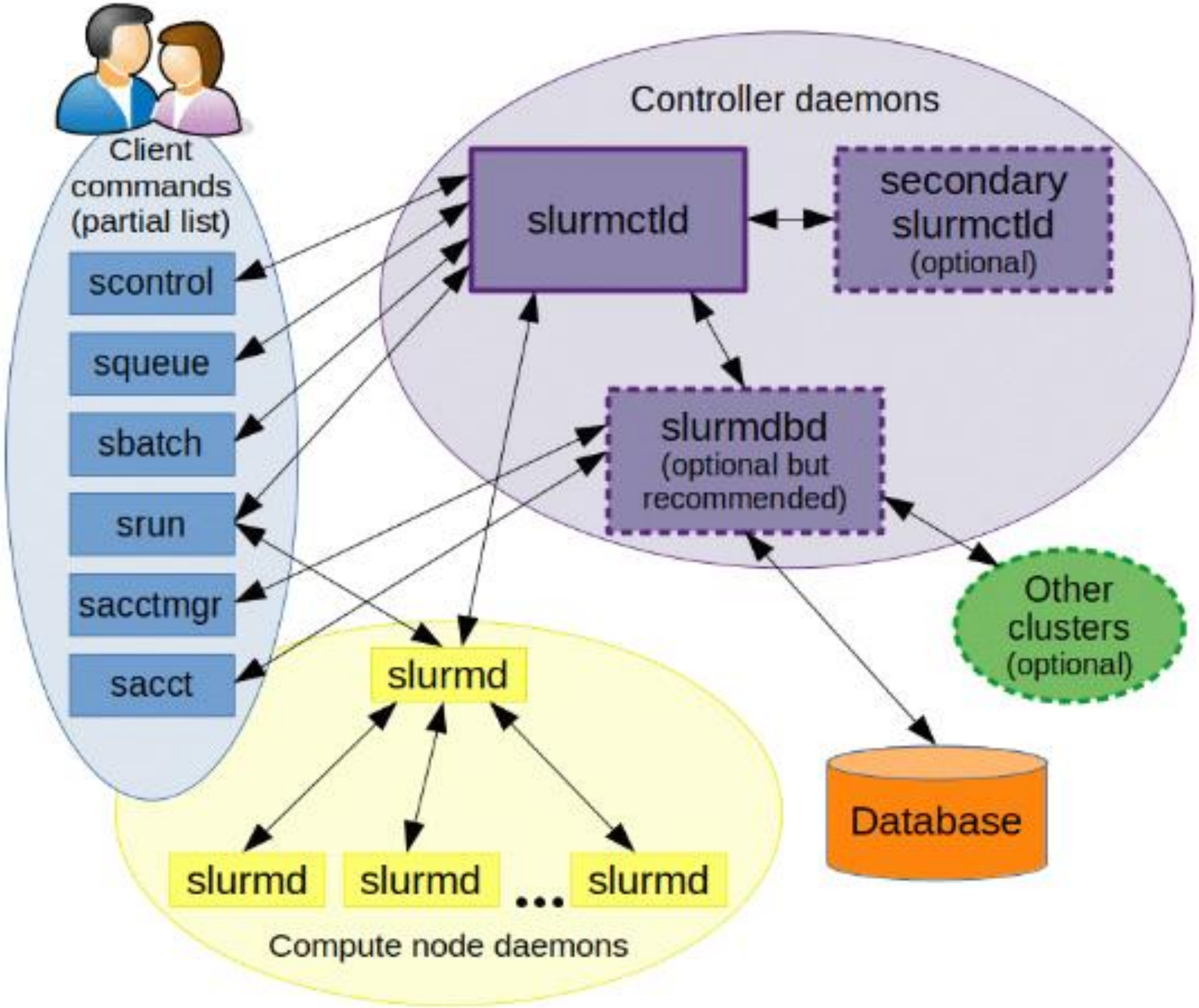
Simple **L**inux **U**tility for **R**esource **M**anagement

- Development started in 2002 at Lawrence Livermore National Laboratory as a simple resource manager for Linux clusters
- Has evolved into a capable job scheduler through use of optional plugins
- More than 500,000 lines of C code.
- Supports AIX, Linux, Solaris, other Unix variants
- Used on many of the world's largest computers

SLURM – ADVANTAGE

- **Scalability:** Slurm is designed to be scalable to large clusters. It can be used to manage a cluster with thousands of nodes and millions of cores (widely used by Supercomputing communty – trainigs)
- **Robustness:** Slurm is a robust scheduler that can handle a variety of failures. It has built-in mechanisms for detecting and recovering from failures (more stables than PBS). More friendy for admins using.
- **Efficiency:** Slurm is an efficient scheduler that can minimize the amount of time that jobs spend waiting in the queue. It does this by using a variety of techniques, such as backfilling and preemption (scheduler managing jobs quicker – large allocation).
- **Open source:** Slurm is an open-source scheduler that is freely available to use and modify.
- Growing **community**, materials, documentations

SLURM – ARCHITECTURE – SLURM COMPONETS


Client commands (partial list)
scontrol
squeue
sbatch
srun
sacctmgr
sacct
Controller daemons
slurmctld
secondary slurmctld (optional)
slurmdbd (optional but recommended)
Other clusters (optional)
Database
slurmd
slurmd
slurmd
...
slurmd
Compute node daemons

SLURM USER COMMANDS – BASIC

Slurm Command	What it does
sinfo	reports the state of partitions and nodes managed by Slurm. It has a wide variety of filtering, sorting, and formatting options.
squeue	reports the **state** of **jobs** or job steps. It has a wide variety of filtering, sorting, and formatting options. By default, it reports the running jobs in priority order and then the pending jobs in priority order.
sbatch	is used to **submit** a job script for later execution. The script will typically contain one or more srun commands to launch parallel tasks.
scancel	is used to **cancel** a pending or running job or job step. It can also be used to send an arbitrary signal to all processes associated with a running job or job step.
sacct	is used to report job or job **step** accounting information about active or completed jobs.
srun	is used to submit a **job for execution** or initiate job steps in real time. srun has a wide variety of options to specify resource requirements, including minimum and maximum node count, processor count, specific nodes to use or not use, and specific node characteristics (so much memory, disk space, certain required features, etc.). A job can contain multiple job steps executing sequentially or in parallel on independent or shared nodes within the job's node allocation.
salloc	Create job allocation and start a shell to use it (interactive mode)

https://slurm.schedmd.com/quickstart.html

SLURM X PBS – USER COMMANDS

User Commands	PBS	Slurm
Job submission	qsub [script_file]	sbatch [script_file]
Job deletion	qdel [job_id]	scancel [job_id]
Job status (by job)	qstat [job_id]	squeue [job_id]
Job status (by user)	qstat -u [user_name]	squeue -u [user_name]
Job hold	qhold [job_id]	scontrol hold [job_id]
Job release	qrls [job_id]	scontrol release [job_id]
Queue list	qstat -Q	sinfo -s
Node list	pbsnodes -l	sinfo -N OR scontrol show nodes
Cluster status	qstat -a	sinfo

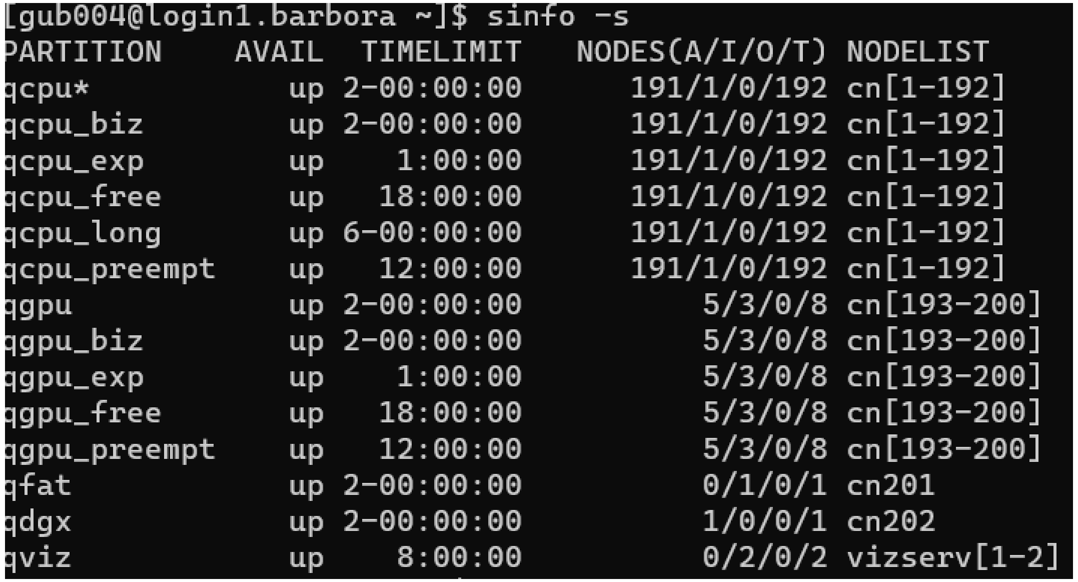
SLURM X PBS – USER COMMANDS - JOBS

Job Specification	PBS	Slurm
Script directive	#PBS	#SBATCH
Queue/Partition	-q [name]	-p [name]
Node Count	-l nodes=[count]	-N [min[-max]]
Total Task Count	-l ppn=[count] OR -l mppwidth=[PE_count]	--ntasks-per-node=[count]
Wall Clock Limit	-l walltime=[hh:mm:ss]	-t [min] OR -t [days-hh:mm:ss]
Standard Output File	-o [file_name]	-o [file_name]
Standard Error File	-e [file_name]	-e [file_name]
Job Name	-N [name]	--job-name=[name]
Job Restart	-r [y | n]	--requeue OR --no-requeue
Resource Sharing	-l naccesspolicy=singlejob	--exclusive OR --shared
Memory Size	-l mem=[MB]	--mem=[mem][M | G | T] OR --mem-per-cpu=[mem][M | G | T]
Tasks Per Node	-l mppnppn [PEs_per_node]	--ntasks-per-node=[count]
Job Arrays	-t [array_spec]	--array=[array_spec]

PARTITION INFORMATION - SINFO

$ sinfo –s qstat -Q

report the state of partitions and nodes managed by Slurm. It has a wide variety of filtering, sorting, and formatting options.



```
[gub004@login1.barbora ~]$ sinfo -s
PARTITION     AVAIL  TIMELIMIT   NODES(A/I/O/T) NODELIST
qcpu*            up 2-00:00:00      191/1/0/192 cn[1-192]
qcpu_biz         up 2-00:00:00      191/1/0/192 cn[1-192]
qcpu_exp         up    1:00:00      191/1/0/192 cn[1-192]
qcpu_free        up   18:00:00      191/1/0/192 cn[1-192]
qcpu_long        up 6-00:00:00      191/1/0/192 cn[1-192]
qcpu_preempt     up   12:00:00      191/1/0/192 cn[1-192]
qgpu             up 2-00:00:00          5/3/0/8 cn[193-200]
qgpu_biz         up 2-00:00:00          5/3/0/8 cn[193-200]
qgpu_exp         up    1:00:00          5/3/0/8 cn[193-200]
qgpu_free        up   18:00:00          5/3/0/8 cn[193-200]
qgpu_preempt     up   12:00:00          5/3/0/8 cn[193-200]
qfat             up 2-00:00:00          0/1/0/1 cn201
qdgx             up 2-00:00:00          1/0/0/1 cn202
qviz             up    8:00:00          0/2/0/2 vizserv[1-2]
```

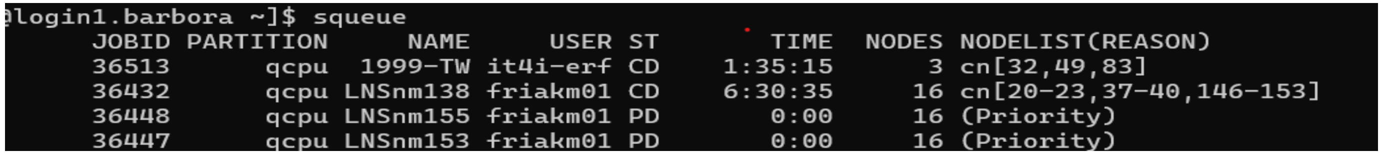
Nodes status: allocated/idle/other/total
Graphical representation of clusters' usage

Barbora at https://extranet.it4i.cz/rsweb/barbora
Complementary Systems at https://extranet.it4i.cz/rsweb/compsys

On Complementary systems, only some queues/partitions provide full node allocation.
https://slurm.schedmd.com/sinfo.html

JOB INFORMATION - SQUEUE

$ squeue   qstat



```
@login1.barbora ~]$ squeue
     JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
     36513      qcpu  1999-TW it4i-erf CD    1:35:15      3 cn[32,49,83]
     36432      qcpu LNSnm138 friakm01 CD    6:30:35     16 cn[20-23,37-40,146-153]
     36448      qcpu LNSnm155 friakm01 PD       0:00     16 (Priority)
     36447      qcpu LNSnm153 friakm01 PD       0:00     16 (Priority)
```

- reports the state of jobs or job steps. It has a wide variety of filtering, sorting, and formatting options. By default, it reports the running jobs in priority order and then the pending jobs in priority order.

Examples:
Show my jobs: $ squeue --me
Show my jobs using a long output format which includes time limit: $ squeue --me -l
Show my jobs in running state: $ squeue --me -t running
Show my jobs in pending state: squeue --me -t pending
Show jobs for a given project: squeue -A PROJECT-ID

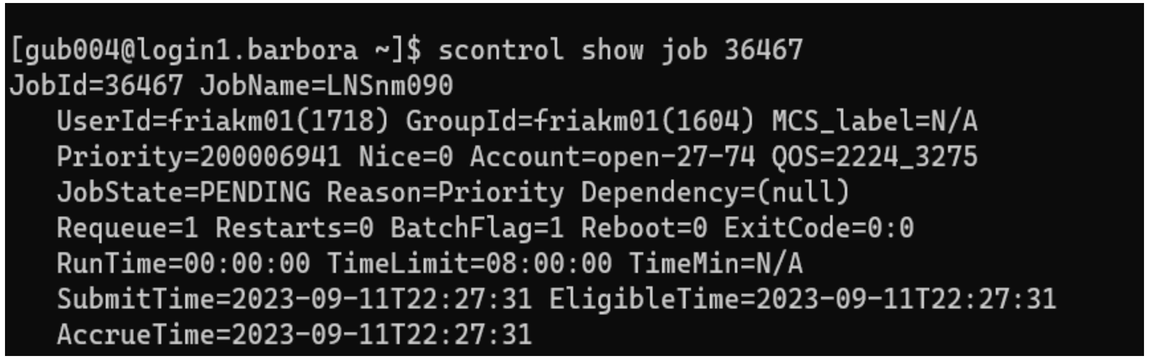
ST - status: PD - pending, R – running, CD – completed, F-Fail, CA - cancelled, ST - stopped,
S - susspended, TO tiemout, atc.

TIME - hows how long the jobs have run
NODELIST(REASON) - Resources (waiting for resources to become available) and Priority (queued behind a higher priority job), Dependency

https://slurm.schedmd.com/squeue.html
https://docs.it4i.cz/general/slurm-job-submission-and-execution/#job-states

JOB INFORMATION - SCONTROL

$ scontrol qrls, qmgr, qhold



```
[gub004@login1.barbora ~]$ scontrol show job 36467
JobId=36467 JobName=LNSnm090
   UserId=friakm01(1718) GroupId=friakm01(1604) MCS_label=N/A
   Priority=200006941 Nice=0 Account=open-27-74 QOS=2224_3275
   JobState=PENDING Reason=Priority Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=08:00:00 TimeMin=N/A
   SubmitTime=2023-09-11T22:27:31 EligibleTime=2023-09-11T22:27:31
   AccrueTime=2023-09-11T22:27:31
```

scontrol command can be used to
report more detailed information about nodes,
partitions, jobs, job steps, and configuration.
It can also be used by system administrators to make configuration changes.

Examples:
Show job details for a specific job:scontrol show job JOBID
Modify job's time limit: $ scontrol update JobId=JOBID timelimit=4:00:00
Set/modify job's comment:$ scontrol update JobId=JOBID Comment='The best job ever'
Show information about nodes: $ scontrol show nodes
To kill a job: $scontrol kill JOBID*

Note:
*Scontrol is the administrative tool used to view and/or modify Slurm state (root)

https://slurm.schedmd.com/scontrol.html

DELETE JOBS- SCANCEL

$ scancel qdel

scancel is used to cancel a pending or running job or job step.
It can also be used to send an arbitrary signal to all processes associated with a running job or job step.

Delete a job by job ID:$ scancel JOBID
Delete all my jobs:$ scancel --me
Delete all my jobs in interactive mode, confirming every action:$ scancel --me -i
Delete all my running jobs:$ scancel --me -t running
Delete all my pending jobs:$ scancel --me -t pending
Delete all my pending jobs for a project PROJECT-ID:$ scancel --me -t pending -A PROJECT-ID

https://slurm.schedmd.com/scancel.html

INTERACTIVE JOBS- SALLOC

```
$ salloc qsub –I
```

salloc is used to allocate resources for a job in real-time.
Typically, this is used to allocate resources and a shell (debugging, testing).

Run interactive job - queue qcpu_exp, one node by default, one task by default:

```
$ salloc -A PROJECT-ID -p qcpu_exp
```

Example:
Run interactive job on four nodes, 36 tasks per node
(Barbora cluster, CPU partition recommended value based on node core count), two hours:

```
$ salloc -A PROJECT-ID -p qcpu -N 4 --ntasks-per-node 36 -t 2:00:00
$ qsub -I -l nodes=4:ncpus=36,walltime=2:00:00
```

Note:
Do not use srun for initiating interactive jobs.

https://slurm.schedmd.com/salloc.html

JOB SCRIPT – COMPARATION

PBS	Slurm
#!/bin/bash	#!/bin/bash
#PBS -N hello_world	#SBATCH --job-name="hello_world"
#PBS -q batch	#SBATCH -p batch
#PBS -l select=2:mpiprocs=8	#SBATCH -N 2
	#SBATCH -n 16
#PBS -l walltime=49:00:00	#SBATCH -t 2-01:00:00
#PBS -j oe	
#PBS -o $PBS_JOBNAME-$PBS_JOBID.log	#SBATCH --output=%x-%j.log
cd $PBS_O_WORKDIR	
ml openmpi	ml openmpi
mpirun -n16 hello_world	srun hello_world

Example:

Here is a quick example of converting a simple PBS submission script which runs an OpenMPI rendition of "Hello World!" to a Slurm submission script.

https://slurm.schedmd.com/sbatch.html

FILENAME PATTERN

variables in naming your output files, you
will need to use Slurm's file patterns shown below.

Variable Name	File Pattern
Job name	%x
Job id	%j
Job array id	%a
Username	%u
Hostname (This will create a separate I/O file per node)	%N

https://slurm.schedmd.com/srun.html

JOB SCRIPT – CONDITIONS

Script	Slurm
#!/usr/bin/bash	• use bash shell interpreter
#SBATCH --job-name MyJobName	• use MyJobName as job name
#SBATCH --account PROJECT-ID	• use project PROJECT-ID for job access and accounting
#SBATCH --partition qcpu	• use partition/queue qcpu
#SBATCH --nodes 4	• use four nodes
#SBATCH --ntasks-per-node 36	• use 36 tasks per node - value used by MPI
#SBATCH --time 12:00:00	• set job time limit to **12 hours**
ml purge	• load appropriate module
ml OpenMPI/4.1.4-GCC-11.3.0	
srun hostname | sort | uniq	run command, srun serves as Slurm's native way of executing MPI-enabled applications, hostname is used in the example just for sake of simplicity

Submit directory will be used as working directory for submitted job, so there is no need to change directory in the job script. Alternatively you can specify job working directory using sbatch --chdir (or shortly -D) option.

https://slurm.schedmd.com/sbatch.html

https://docs.it4i.cz/general/slurm-job-submission-and-execution/#job-script

SBATCH - OPTIONS

Command Short, long version	Comment
J, --job-name=<jobname>	Specify a name for the job allocation. The specified name will appear along with the job id number when querying running jobs on the system.
N, --nodes=<minnodes>[-maxnodes]|<size_string>	Request that a minimum of minnodes nodes be allocated to this job
-A, --account=<account>	Charge resources used by this job to specified account.
-a, --array=<indexes>	Submit a job array, multiple jobs to be executed with identical parameters
--comment=<string>	An arbitrary comment enclosed in double quotes if using spaces or some special characters
-J, --job-name=<jobname>	specify a name for the job allocation. The specified name will appear along with the job id number when querying running jobs on the system.
-t, --time=<time>	Set a limit on the total run time of the job allocation. If the requested time limit exceeds the partition's time limit, the job will be left in a PENDING state.
-o, --output=<filename_pattern>	Instruct Slurm to connect the batch script's standard output directly to the file name specified in the "filename pattern"
--ntasks-per-node=<ntasks>	Request that ntasks be invoked on each node
-p, --partition=<partition_names>	Request a specific partition for the resource allocation.
-n, --ntasks=<number>	sbatch does not launch tasks, it requests an allocation of resources and submits a batch script.

https://slurm.schedmd.com/sbatch.html

SUBMIT BATCH JOB - SBATCH

$ sbatch qsub

is used to submit a job script for later execution.
The script will typically contain one or more srun commands to launch parallel tasks.

Submit batch job:

```
$ cd my_work_dir
$ sbatch script.sh
```

Run batch job (options, on complementary system)

```
 $ sbatch -A PROJECT-ID -p p01-arm ./script.sh
```

Job output is stored in a file called slurm-JOBID.out.
Job standard output and error output.
This can be changed using sbatch options --output (shortly -o) and --error (shortly -e).

https://slurm.schedmd.com/sbatch.html
https://docs.it4i.cz/cs/job-scheduling/

SBATCH-ERRORS

Error status: sbatch: error: Batch job submission failed: Invalid account or account/partition combination specified

Possible causes:

1)Invalid account (project) was specified in job submission.
2)User does not have access to given account/project.
3)Given account/project does not have access to given partition.
4)Access to given partition was retracted due to the project's allocation exhaustion.

Slurm support – Bug report: https://bugs.schedmd.com/

JOB ENVIRONMENT VARIABLES (OUTPUTS)

Slurm provides useful information to the job via environment variables.
Environment variables are available on all nodes allocated to a job.

Variable name	description	example
SLURM_JOB_ID	job id of the executing job	593
SLURM_JOB_NODELIST	nodes allocated to the job	cn[101-102]
SLURM_JOB_NUM_NODES	number of nodes allocated to the job	2
SLURM_STEP_NODELIST	nodes allocated to the job step	cn101
SLURM_STEP_NUM_NODES	number of nodes allocated to the job step	1
SLURM_JOB_PARTITION	name of the partition	qcpu
SLURM_SUBMIT_DIR	submit directory	/scratch/project/open-xx-yy/work

https://docs.it4i.cz/general/slurm-job-submission-and-execution/#job-environment-variables
https://slurm.schedmd.com/sbatch.html

JOB ENVIRONMENT VARIABLES (INPUTS)

Upon startup, sbatch will read and handle the options set in the following environment variables
The sbatch command honors the following environment variables,
when present (these override any inline directives within your batch script,
but will be overridden by those also specified on the sbatch command line.

Variable name	description
SBATCH_ACCOUNT	-A, --account
SBATCH_ACCTG_FREQ	--acctg-freq
SBATCH_JOB_NAME	-J, --job-name
SBATCH_PARTITION	-p, --partition
SBATCH_REQUEUE	--requeue
SBATCH_RESERVATION	--reservation
SBATCH_THREADS_PER_CORE	--threads-per-core

Prirority of variables
1. Comand line
2. Environment (input/output)
3. Batch script

NOTE: Environment variables will override any options set in a batch script, and command line options will override any environment variables.

https://slurm.schedmd.com/sbatch.html

Demonstrations

DOCUMENTATIONS/RESOURCES

Slurm documentation: https://slurm.schedmd.com/documentation.html
Basic commands in the Slurm: https://slurm.schedmd.com/pdfs/summary.pdf

Tutorials (videos) from The University of Utah:
https://www.chpc.utah.edu/documentation/software/slurm.php
Video: Slurm Workload Manager Architecture, Configuration and Use(intended for developers)
https://www.open-mpi.org/video/?category=slurm

IT4I documentations:
https://docs.it4i.cz/general/karolina-slurm/
https://docs.it4i.cz/general/slurm-job-submission-and-execution/
https://docs.it4i.cz/cs/job-scheduling/

CONCLUSION

Questions, comments?

Thank you for your time.

I hope you found this presentation informative and helpful.