# Wikiprint Book

Title: Information about the batch system (SLURM)

Subject: DEEP - Public/User\_Guide/Batch\_system

Version: 63

Date: 19.05.2024 18:16:35

# **Table of Contents**

Information about the batch system (SLURM)	3
Overview	3
Available Partitions	3
Remark about environment	3
An introductory example	3
From a shell on a node	3
Running directly from the front ends	4
Batch script	5
Job chains	5
Information on past jobs and accounting	6
Advanced topics	6
FAQ	6
Is there a cheat sheet for all main Slurm commands?	6
Why's my job not running?	6
How can I check which jobs are running in the machine?	6
How do I do chain jobs with dependencies?	6
How can check the status of partitions and nodes?	7
Can I join stderr and stdout like it was done with -joe in Torque?	7

# Information about the batch system (SLURM)

The DEEP prototype systems are running SLURM for resource management. Documentation of Slurm can be found ?here.

# Overview

Slurm offers interactive and batch jobs (scripts submitted into the system). The relevant commands are srun and sbatch. The srun command can be used to spawn processes (**please do not use mpiexec**), both from the frontend and from within a batch script. You can also get a shell on a node to work locally there (e.g. to compile your application natively for a special platform or module).

# **Available Partitions**

Please note that there is no default partition configured. In order to run a job, you have to specify one of the following partitions, using the --partition=... switch:

Name	Description
dp-cn	dp-cn[01-50], DEEP-EST Cluster nodes (Xeon Skylake)
dp-dam	dp-dam[01-16], DEEP-EST Dam nodes (Xeon Cascadelake + 1 V100 + 1 Stratix 10)
dp-esb	dp-esb[log:@26-75 "[01-75]"], DEEP-EST ESB nodes connected with IB EDR (Xeon Cascadelake + 1 V100)
dp-sdv-esb	dp-sdv-esb[01-02], DEEP-EST ESB Test nodes (Xeon Cascadelake + 1 V100)
ml-gpu	ml-gpu[01-03], GPU test nodes for ML applications (4 V100 cards)
kni	knl[01,04-06], KNL nodes
knl256	knl[01,05], KNL nodes with 64 cores
knl272	knl[04,06], KNL nodes with 68 cores
snc4	knl[05], KNL node in snc4 memory mode
debug	all compute nodes (no gateways)

Anytime, you can list the state of the partitions with the sinfo command. The properties of a partition (.e.g. the maximum walltime) can be seen using

scontrol show partition <partition>

# **Remark about environment**

By default, Slurm passes the environment from your job submission session directly to the execution environment. Please be aware of this when running jobs with srun or when submitting scripts with sbatch. This behavior can be controlled via the --export option. Please refer to the <u>?Slurm</u> <u>documentation</u> to get more information about this.

In particular, when submitting job scripts, it is recommended to load the necessary modules within the script and submit the script from a clean environment.

# An introductory example

 $Suppose you have an mpi executable named {\tt hello\_mpi}. There are three ways to start the binary.$ 

### From a shell on a node

If you just need one node to run your interactive session on you can simply use the srun command (without salloc), e.g.:

```
[kreutzl@deepv ~]$ srun -A deep -N 1 -n 8 -p dp-cn -t 00:30:00 --pty --interactive bash
[kreutzl@dp-cn22 ~]$ srun -n 8 hostname
dp-cn22
```

dp-cn22 dp-cn22 dp-cn22 dp-cn22 dp-cn22 dp-cn22 dp-cn22

The environment is transported to the remote shell, no .profile, .bashrc, ... are sourced (especially not the modules default from /etc/profile.d/modules.sh). As of March 2020, an account has to be specified using the --account (short -A) option, which is "deepsea" for DEEP-SEA project members. For people not included in the DEEP-SEA project, please use the "Budget" name you received along with your account creation.

Assume you would like to run an MPI task on 4 cluster nodes with 2 tasks per node. It's necessary to use salloc then:

[kreutzl@deepv Temp]\$ salloc -A deep -p dp-cn -N 4 -n 8 -t 00:30:00 srun --pty --interactive /bin/bash [kreutzl@dp-cn01 Temp]\$ srun -N 4 -n 8 ./MPI\_HelloWorld Hello World from rank 3 of 8 on dp-cn02 Hello World from rank 7 of 8 on dp-cn04 Hello World from rank 2 of 8 on dp-cn04 Hello World from rank 6 of 8 on dp-cn04 Hello World from rank 0 of 8 on dp-cn01 Hello World from rank 4 of 8 on dp-cn03 Hello World from rank 1 of 8 on dp-cn01 Hello World from rank 5 of 8 on dp-cn03

Once you get to the compute node, start your application using srun. Note that the number of tasks used is the same as specified in the initial srun command above (4 nodes with two tasks each). It's also possible to use less nodes in the srun command. So the following command would work as well:

```
[kreutzl@dp-cn01 Temp]$ srun -N 1 -n 1 ./MPI_HelloWorld
Hello World from rank 0 of 1 on dp-cn01
```

### Running directly from the front ends

You can run the application directly from the frontend, bypassing the shell. Do not forget to set the correct environment for running your executable on the login node as this will be used for execution with srun.

```
[kreutzl@deepv Temp]$ ml GCC/10.3.0 ParaStationMPI/5.4.9-1
[kreutzl@deepv Temp]$ srun -A deep -p dp-cn -N 4 -n 8 -t 00:30:00 ./MPI_HelloWorld
Hello World from rank 7 of 8 on dp-cn04
Hello World from rank 3 of 8 on dp-cn02
Hello World from rank 6 of 8 on dp-cn04
Hello World from rank 2 of 8 on dp-cn03
Hello World from rank 0 of 8 on dp-cn01
Hello World from rank 1 of 8 on dp-cn01
Hello World from rank 5 of 8 on dp-cn03
```

It can be useful to create an allocation which can be used for several runs of your job:

```
[kreutzl@deepv Temp]$ salloc -A deep -p dp-cn -N 4 -n 8 -t 00:30:00
salloc: Granted job allocation 69263
[kreutzl@deepv Temp]$ srun ./MPI_HelloWorld
Hello World from rank 7 of 8 on dp-cn04
Hello World from rank 3 of 8 on dp-cn02
Hello World from rank 6 of 8 on dp-cn04
Hello World from rank 2 of 8 on dp-cn02
Hello World from rank 5 of 8 on dp-cn03
```

```
Hello World from rank 1 of 8 on dp-cn01
Hello World from rank 4 of 8 on dp-cn03
Hello World from rank 0 of 8 on dp-cn01
...
# several more runs
...
[kreutzl@deepv Temp]$ exit
exit
salloc: Relinquishing job allocation 69263
```

Note that in this case the -N and -n options for the srun command can be skipped (they default to the corresponding options given to salloc).

# **Batch script**

As stated above, it is recommended to load the necessary modules within the script and submit the script from a clean environment.

The following script hello\_cluster.sh will unload all modules and load the modules required for executing the given binary:

```
#!/bin/bash
#SBATCH --partition=dp-esb
#SBATCH -A deep
#SBATCH -N 4
#SBATCH -n 8
#SBATCH -o /p/project/cdeep/kreutz1/hello_cluster-%j.out
#SBATCH -e /p/project/cdeep/kreutz1/hello_cluster-%j.err
#SBATCH --time=00:10:00
ml purge
ml GCC ParaStationMPI
srun ./MPI_HelloWorld
```

This script requests 4 nodes of the ESB module with 8 tasks, specifies the stdout and stderr files, and asks for 10 minutes of walltime. You can submit the job script as follows:

[kreutz1@deepv Temp]\$ sbatch hello\_cluster.sh Submitted batch job 69264

... and check what it's doing:

[kreutz1@deepv Ter	np]\$ squeue -u \$	ŞUSER			
JOBID	PARTITION N	NAME USER	ST TIM	E NODES	NODELIST(REASON)
69264	dp-cn hello	o_cl kreutz1	CG 0:0	4 4	dp-cn[01-04]

Once finished, you can check the result (and the error file if needed)

```
[kreutzl@deepv Temp]$ cat /p/project/cdeep/kreutzl/hello_cluster-69264.out
Hello World from rank 7 of 8 on dp-esb37
Hello World from rank 3 of 8 on dp-esb36
Hello World from rank 1 of 8 on dp-esb34
Hello World from rank 6 of 8 on dp-esb37
Hello World from rank 2 of 8 on dp-esb35
Hello World from rank 4 of 8 on dp-esb36
Hello World from rank 4 of 8 on dp-esb36
Hello World from rank 0 of 8 on dp-esb36
```

# Job chains

Please refer to the <u>FAQ</u> for creation of job chains and implementing job dependencies. If you would like to implement workflows, take a look at the <u>Workflows</u> section.

# Information on past jobs and accounting

The sacct command can be used to enquire the Slurm database about a past job.

[kreutz1@deep	pv Temp]\$ sa	acct -j 6920	58			
JobID	JobName	Partition	Account	AllocCPUS	State E	ExitCode
69268+0	bash	dp-cn	deepest-a+	96	COMPLETED	0:0
69268+0.0	MPI_Hello+		deepest-a+	2	COMPLETED	0:0
69268+1	bash	dp-dam	deepest-a+	384	COMPLETED	0:0

On the Cluster (CM) nodes it's possible to query the consumed energy for a certain job:

[kreutz1@deepv	kreutz1]\$	sacct -o Consu	medEnergy,	JobName,JobID,CPUTime,AllocNodes -j 69326
ConsumedEnergy	JobName	e JobID	CPUTime	AllocNodes
496.70K	hpl_MKL_O+	69326	16:28:48	1
0	batch	69326.batch	16:28:48	1
496.70K	xlinpack_+	69326.0	08:10:24	1

This feature will also be for the ESB nodes.

# **Advanced topics**

For further details on the batchsystem and psslurm which is used on DEEP as well as on the JSC production systems, please refer to the in-depth description for using the <u>Batchsystem on Jureca</u>. Among extended examples for allocation of nodes you can find information on job steps, dependency chains and multithreading there. If you are interested in pinning of threads and tasks to certain CPUs or cores, please also take a look into the <u>Processor Anffinity</u> sections of the Jureca documentation. Most of the information provided there will also refer to the DEEP System.

# FAQ

#### Is there a cheat sheet for all main Slurm commands?

Yes, it is available ?here.

### Why's my job not running?

You can check the state of your job with

scontrol show job <job id>

In the output, look for the Reason field.

You can check the existing reservations using

scontrol show res

#### How can I check which jobs are running in the machine?

Please use the squeue command ( the "-u \$USER" option to only list jobs belonging to your user id).

#### How do I do chain jobs with dependencies?

Please confer the sbatch/srun man page, especially the

-d, --dependency=<dependency\_list>

entry.

Also, jobs can be chained after they have been submitted using the scontrol command by updating their Dependency field.

# How can check the status of partitions and nodes?

The main command to use is sinfo. By default, when called alone, sinfo will list the available partitions and the number of nodes in each partition in a given status. For example:

Ideamical Jetes       AVALL       TIMELINIT       NODES       STATE NODELIST         PARTITION       AVALL       TIMELINIT       NODES       STATE NODELIST         sdv       up       20:00:00       11       idle deper-sdv[06-16]         kn1       up       20:00:00       3       idle kn104         kn1256       up       20:00:00       1       idle kn105         kn1272       up       20:00:00       1       idle kn105         kn1272       up       20:00:00       1       idle kn105         extol       up       20:00:00       1       idle deper-sdv[06-16]         m1-gpu       up       20:00:00       1       idle deper-sdv[06-16]         m1-gpu       up       20:00:00       1       drain dp-cn33         dp-cn       up       20:00:00       5       resv dp-cn[01-08,11-24,26-32,34-48]         dp-dam       up       20:00:00       3       resv dp-calm08         dp-dam       up       20:00:00       3       resv dp-dam08         dp-dam       up       20:00:00       2       resv dp-dam08         dp-dam       up       20:00:00       2       resv dp-dam09         dp-dam						
sdv         up         20:00:00         1         idle deeper-sdv(06-16]           kn1         up         20:00:00         1         drain kn101           kn1256         up         20:00:00         1         drain kn101           kn1256         up         20:00:00         1         idle kn105           kn1272         up         20:00:00         1         idle kn105           snc4         up         20:00:00         1         idle kn105           extoll1         up         20:00:00         1         idle kn105           extoll4         up         20:00:00         1         idle deeper-sdv(06-16]           ml-gpu         up         20:00:00         1         idle deeper-sdv(06-16]           ml-gpu         up         20:00:00         1         idle deeper-sdv(06-16]           dp-cn         up         20:00:00         5         resv dp-cn(09-10,25,49-50]           dp-cn         up         20:00:00         5         resv dp-cdm(03,07]           dp-dam         up         20:00:00         3         resv dp-dam(03,07]           dp-dam         up         20:00:00         3         resv dp-dam(03,06]           dp-dam         up	[deamicis1@d	eepv hy	/bridhello]\$	sinfo		
knl       up       20:00:00       1       drain knl01         knl       up       20:00:00       1       idle knl04-06         knl256       up       20:00:00       1       idle knl05         knl272       up       20:00:00       2       idle knl05         knl272       up       20:00:00       1       idle knl05         snc4       up       20:00:00       1       idle deeper-sdv(06-16]         ml-gpu       up       20:00:00       1       idle deeper-sdv(06-16]         ml-gpu       up       20:00:00       1       drain dp-cn33         dp-cn       up       20:00:00       1       drain dp-dam08         dp-dam       up       20:00:00       2       drain dp-dam(05,09-10]         dp-dam       up       20:00:00       2       alloc dp-dam(10,04]         dp-dam       up       20:00:00       2       alloc dp-dam(10,04]         dp-dam       up       20:00:00       2       alloc dp-dam(10,04]         dp-dame.xt       up       20:00:00       2       alloc dp-dam(10,04]         dp-dame.xt       up       20:00:00       2       alloc dp-dam(11.6]         dp-eab       up	PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
kn1         up         20:00:00         3         idle kn1[04-06]           kn1256         up         20:00:00         1         drain kn101           kn1256         up         20:00:00         1         idle kn105           sn24         up         20:00:00         1         idle kn105           snc4         up         20:00:00         1         idle kn106           extol1         up         20:00:00         1         idle kn107           dp-cn         up         20:00:00         1         drain dp-cn33           dp-cn         up         20:00:00         1         drain dp-cn34           dp-dan         up         20:00:00         1         drain dp-dm08           dp-dan         up         20:00:00         1         drain dp-dm103,07]           dp-dan         up         20:00:00         2         drain dp-dm103,07]           dp-dan         up         20:00:00         2         drain dp-dm103,07]           dp-dan         up         20:00:00         2         arew dp-dm102,06,11-16]           dp-dan         up         20:00:00         2         rew dp-dm102,06,11-16]           dp-dan         up         20:00:00	sdv	up	20:00:00	11	idle	deeper-sdv[06-16]
kn1256up20:00:001drain kn101kn1256up20:00:001idle kn105kn1272up20:00:001idle kn105extollup20:00:001idle kn105extollup20:00:001idle deper-sdv[06-16]ml-gpuup20:00:001idle deper-sdv[06-16]dp-cnup20:00:001idle deper-sdv[06-16]dp-cnup20:00:001idle deper-sdv[06-16]dp-cnup20:00:0044idle deper-sdv[06-10,25,49-50]dp-cnup20:00:001drain* dp-dam08dp-damup20:00:002drain dp-dam0(3,07]dp-damup20:00:002aresv dp-dam(05,09-10]dp-damup20:00:002alle dp-dam(02,06,11-16]dp-damup20:00:002alle dp-dam(02,06,11-16]dp-damup20:00:002alle dp-dam(1,04]dp-damup20:00:002alle dp-dam(1,04]dp-damup20:00:002drain* dp-esb(1,26-75)dp-damup20:00:002alloc dp-esb(09-10]dp-esbup20:00:002alloc dp-esb(09-10]dp-esbup20:00:002idle dp-dam(1-01,02]dp-esbup20:00:002alloc dp-esb(09-10]dp-esbup20:00:002resv dp-cd-esb(10-2]dp-esbup20:00:002resv	knl	up	20:00:00	1	drain	kn101
kn1256       v       20:00:00       1       idle kn105         kn1272       vp       20:00:00       2       idle kn106         snc4       vp       20:00:00       1       idle kn105         extol1       vp       20:00:00       1       idle deeper-adv[06-16]         al-gpu       vp       20:00:00       3       idle ml-gpu[01-03]         dp-cn       vp       20:00:00       5       resv dp-cn[09-10,25,49-50]         dp-dn       vp       20:00:00       1       drain dp-cn33         dp-dn       vp       20:00:00       1       drain dp-cn3[         dp-dn       vp       20:00:00       1       drain dp-cn3[         dp-dan       vp       20:00:00       1       drain* dp-dam0         dp-dan       vp       20:00:00       2       drain dp-cam[03,07]         dp-dan       vp       20:00:00       3       resv dp-dam[02,06,11-16]         dp-dan       vp       20:00:00       2       resv dp-dam[02,06,11-16]         dp-dam-ext       vp       20:00:00       2       resv dp-esb[01,02]         dp-esb       vp       20:00:00       2       drain dp-esb[01,02,12-22,24-25]         dp-esb	knl	up	20:00:00	3	idle	kn1[04-06]
kn1272up20:00:002idle kn104,06]snc4up20:00:001idle kn105extol1up20:00:001idle kn105anl-gpuup20:00:003idle nl-gpu01-03]dp-cnup20:00:001drain dp-cn3dp-cnup20:00:005resv dp-cn(09-10,25,49-50]dp-damup20:00:001drain dp-cn3dp-damup20:00:001drain dp-dam(03,07]dp-damup20:00:002drain dp-dam(03,07]dp-damup20:00:002atloc dp-dam(02,06,11-16]dp-damup20:00:002atloc dp-dam(02,06,11-16]dp-dam-extup20:00:002resv dp-dam(02,06,11-16]dp-esbup20:00:002atloc dp-esb(09-10]dp-esbup20:00:002atloc dp-esb(0-10]dp-esbup20:00:002atloc dp-esb(0-10]dp-esbup20:00:002atloc dp-esb(0-102]psw-bosterup20:00:001idle nfgw01psw-bosterup20:00:001idle nfgw01psw-bosterup20:00:001idle ncp-cam(03,07],kn101debugup20:00:001idle ncp-cam(03,07],kn101debugup20:00:001idle ncp-cam(03,07],kn101debugup20:00:001idle ncg-cam(03,07],kn101debugup20:00:001	knl256	up	20:00:00	1	drain	kn101
snc4       up       20:00:00       1       idle kn105         extol1       up       20:00:00       3       idle m1-gpu(10-3]         dp-cn       up       20:00:00       1       drain dp-cn33         dp-cn       up       20:00:00       1       drain dp-cn34         dp-cn       up       20:00:00       44       idle dp-cn[09-10,25,49-50]         dp-dam       up       20:00:00       1       drain* dp-dam08         dp-dam       up       20:00:00       2       drain dp-dam08         dp-dam       up       20:00:00       2       drain dp-dam03,07]         dp-dam       up       20:00:00       2       allce dp-dam[03,07]         dp-dam       up       20:00:00       2       allce dp-dam[03,07]         dp-dam       up       20:00:00       2       allce dp-dam[03,07]         dp-dam       up       20:00:00       3       idle dp-dam[03,07]         dp-dam       up       20:00:00       2       allce dp-dam[02,06,11-16]         dp-dam-ext       up       20:00:00       2       resv dp-dam[02,06,11-16]         dp-esb       up       20:00:00       2       drain dp-esb[04,23]         dp-esb	knl256	up	20:00:00	1	idle	kn105
extollup20:00:0011idle deeper-sdv[06-16]ml-gpuup20:00:003idle ml-gpu[01-03]dp-cnup20:00:001drain dp-cn33dp-cnup20:00:005resv dp-cn[09-10,25,49-50]dp-damup20:00:0044idle dp-cn[11-08,11-24,26-32,34-48]dp-damup20:00:002drain $dp$ -dam[03,07]dp-damup20:00:002drain $dp$ -dam[03,07]dp-damup20:00:002alloc dp-dam[01,04]dp-damup20:00:002alloc dp-dam[01,04]dp-damup20:00:002resv dp-dam[09-10]dp-damup20:00:002resv dp-dam[09-10]dp-damup20:00:002resv dp-dam[09-10]dp-dam-extup20:00:002resv dp-dam[09-10]dp-esbup20:00:002drain $dp$ -eab[11,26-75]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002idle dp-esb[01-07,12-22,24-25]dp-sdv-esbup20:00:001idle nfgw01psgw-blosterup20:00:001idle nfgw02debugup20:00:001idle nfgw02debugup20:00:001drain $dp$ -dam[03,07],kn101debugup </td <td>knl272</td> <td>up</td> <td>20:00:00</td> <td>2</td> <td>idle</td> <td>kn1[04,06]</td>	knl272	up	20:00:00	2	idle	kn1[04,06]
n1-gpuup20:00:003idle n1-gpu[01-03]dp-cnup20:00:001drain dp-cn33dp-cnup20:00:005resvdp-cn[01-08,11-24,26-32,34-48]dp-damup20:00:001drain*dp-dam08dp-damup20:00:001drain*dp-dam08dp-damup20:00:002drain dp-dam06dp-damup20:00:003resvdp-dam(01,04]dp-damup20:00:002alloc dp-dam[02,06,11-16]dp-damup20:00:002resvdp-dam[02,06,11-16]dp-damup20:00:002resvdp-dam[02,06,11-16]dp-dam-extup20:00:002resvdp-dam[02,06,11-16]dp-aemextup20:00:002resvdp-edm[02,06,11-16]dp-esbup20:00:002resvdp-edm[02,06,11-16]dp-esbup20:00:002resvdp-edm[02,06,11-16]dp-esbup20:00:002resvdp-edm[02,05]dp-esbup20:00:002drain dp-esb[02,23]dp-esbup20:00:002alloc dp-esb[01-02]psgw-clusterup20:00:002resvpsgw-clusterup20:00:001idle nfgw02debugup20:00:001idle nfgw02debugup20:00:001idle nfgw04debugup20:00:001 </td <td>snc4</td> <td>up</td> <td>20:00:00</td> <td>1</td> <td>idle</td> <td>kn105</td>	snc4	up	20:00:00	1	idle	kn105
dp-cnup20:00:001drain dp-cn33dp-cnup20:00:005resv dp-cn[09-10,25,49-50]dp-dnup20:00:001drain * dp-dam08dp-damup20:00:002drain dp-dam[03,07]dp-damup20:00:002aresv dp-dam[05,09-10]dp-damup20:00:002aresv dp-dam[02,06,11-16]dp-damup20:00:002resv dp-dam[09-10]dp-dam-extup20:00:002resv dp-dam[09-10]dp-dam-extup20:00:005resv dp-dam[1-16]dp-esbup20:00:002drain * dp-esb[1,26-75]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002resv dp-esb[01-02]psgw-clusterup20:00:002resv dp-esb[01-02]psgw-boosterup20:00:001idle nfgw01gesw-boosterup20:00:001idle nfgw02debugup20:00:001idle nfgw02debugup20:00:001idle nfgw02debugup20:00:001idle	extoll	up	20:00:00	11	idle	deeper-sdv[06-16]
dp-cn         up         20:00:00         5         resv         dp-cn[09-10,25,49-50]           dp-cn         up         20:00:00         44         idle dp-cn[01-08,11-24,26-32,34-48]           dp-dam         up         20:00:00         1         drain * dp-dam08           dp-dam         up         20:00:00         2         drain * dp-dam[03,07]           dp-dam         up         20:00:00         2         alloc dp-dam[03,07]           dp-dam         up         20:00:00         2         alloc dp-dam[03,07]           dp-dam         up         20:00:00         2         alloc dp-dam[05,09-10]           dp-dam         up         20:00:00         2         alloc dp-dam[02,06,11-16]           dp-dam-ext         up         20:00:00         2         resv dp-dam[09-10]           dp-dam-ext         up         20:00:00         5         drain * dp-esb[11,26-75]           dp-esb         up         20:00:00         2         alloc dp-esb[01-07,12-22,24-25]           dp-esb         up         20:00:00         2         alloc dp-esb[01-02]           psgw-cluster         up         20:00:00         1         idle nfgw01           gp-dabug         up         20:00:00         <	ml-gpu	up	20:00:00	3	idle	ml-gpu[01-03]
dp-cmup20:00:0044idle dp-cm[01-08,11-24,26-32,34-48]dp-damup20:00:001drain* dp-dam08dp-damup20:00:002drain dp-dam[03,07]dp-damup20:00:003resv dp-dam[05,09-10]dp-damup20:00:002alloc dp-dam[02,06,11-16]dp-dam-extup20:00:002resv dp-dam[01-04]dp-dam-extup20:00:002resv dp-dam[11-16]dp-dam-extup20:00:002drain* dp-esb[11,26-75]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-sbup20:00:002alloc dp-esb[01-07,12-22,24-25]dp-sbup20:00:002resv dp-sdv-esb[01-02]psgw-boosterup20:00:001idle nfgw01psgw-boosterup20:00:001idle nfgw02debugup20:00:001drain* dp-dam8debugup20:00:001idle ncg-dam8, dp-dam[03,07], kn101debugup20:00:001drain* dp-dam8debugup20:00:002alloc dp-dam8, dp-dam[03,07], kn101debugup20:00:001drain* dp-dam8debugup20:00:002alloc dp-dam[01,04]	dp-cn	up	20:00:00	1	drain	dp-cn33
dp-damup20:00:001drain*dp-dam08dp-damup20:00:002draindp-dam[03,07]dp-damup20:00:003resvdp-dam[05,09-10]dp-damup20:00:002allocdp-dam[01,04]dp-dam-extup20:00:002resvdp-dam[09-10]dp-dam-extup20:00:006idledp-dam[09-10]dp-dam-extup20:00:0051drain*dp-esb[11,26-75]dp-esbup20:00:002allocdp-esb[08,23]dp-esbup20:00:002allocdp-esb[09-10]dp-esbup20:00:002allocdp-esb[09-10]dp-esbup20:00:002allocdp-esb[01-07,12-22,24-25]dp-esbup20:00:002resvdp-esb[01-02]psgw-lusterup20:00:001idle nfgw01psgw-boosterup20:00:001drain*debugup20:00:004draindebugup20:00:004drain*debugup20:00:004draindebugup20:00:002allocup20:00:004drain*dp-dam[03,07],kn101debugup20:00:004drain*debugup20:00:002allocup20:00:002allocdp-chageup20:00:001<	dp-cn	up	20:00:00	5	resv	dp-cn[09-10,25,49-50]
dp-damup20:00:002drain dp-dam[03,07]dp-damup20:00:003resv dp-dam[05,09-10]dp-damup20:00:002alloc dp-dam[01,04]dp-damup20:00:008idle dp-dam[02,06,11-16]dp-dam-extup20:00:002resv dp-dam[09-10]dp-dam-extup20:00:006idle dp-dam[11-16]dp-eabup20:00:0051drain* dp-esb[11,26-75]dp-esbup20:00:002alloc dp-esb[08,23]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[01-07,12-22,24-25]dp-esbup20:00:002resv dp-sdv-esb[01-02]psgw-clusterup20:00:001idle nfgw01psgw-boosterup20:00:001idle nfgw02debugup20:00:001idle nfgw02debugup20:00:002alloc dp-cn3,dp-dam[03,07],kn101debugup20:00:001resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]	dp-cn	up	20:00:00	44	idle	dp-cn[01-08,11-24,26-32,34-48]
dp-damup20:00:003resvdp-dam[05,09-10]dp-damup20:00:002allocdp-dam[01,04]dp-damup20:00:008idledp-dam[02,06,11-16]dp-dam-extup20:00:002resvdp-dam[09-10]dp-dam-extup20:00:006idledp-dam[11-16]dp-esbup20:00:0051drain*dp-esb[11,26-75]dp-esbup20:00:002drain dp-esb[08,23]dp-esbup20:00:002allocdp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[01-02]psgw-clusterup20:00:002resv dp-sdv-esb[01-02]psgw-boosterup20:00:001idle nfgw02debugup20:00:001drain* dp-cn33,dp-dam[03,07],kn101debugup20:00:0010resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-dam[01,04]	dp-dam	up	20:00:00	1	drain*	dp-dam08
dp-damup20:00:002alloc dp-dam[01,04]dp-damup20:00:008idle dp-dam[02,06,11-16]dp-dam-extup20:00:002resv dp-dam[09-10]dp-dam-extup20:00:006idle dp-dam[11-16]dp-esbup20:00:0051drain* dp-esb[11,26-75]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:002alloc dp-esb[01-07,12-22,24-25]dp-esbup20:00:002resv dp-sdv-esb[01-02]psgw-clusterup20:00:001idle nfgw01psgw-boosterup20:00:001idle nfgw02debugup20:00:001drain dp-cn33,dp-dam[03,07],kn101debugup20:00:0010resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-cn33,dp-dam[03,07],kn101debugup20:00:0010resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-dam[01,04]	dp-dam	up	20:00:00	2	drain	dp-dam[03,07]
dp-damup20:00:008idle dp-dam[02,06,11-16]dp-dam-extup20:00:002resvdp-dam[09-10]dp-dam-extup20:00:006idle dp-dam[11-16]dp-eam-extup20:00:0051drain*dp-esb[11,26-75]dp-esbup20:00:002drain * dp-esb[08,23]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:0020idle dp-esb[01-07,12-22,24-25]dp-esbup20:00:002resv dp-sdv-esb[01-02]dp-sdv-esbup20:00:001idle nfgw01psgw-clusterup20:00:001idle nfgw02debugup20:00:001drain* dp-dam08debugup20:00:004drain dp-cn33,dp-dam[03,07],knl01debugup20:00:0010resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-cam[01,04]	dp-dam	up	20:00:00	3	resv	dp-dam[05,09-10]
dp-dam-extup20:00:002resvdp-dam[09-10]dp-dam-extup20:00:006idledp-dam[11-16]dp-esbup20:00:0051drain*dp-esb[11,26-75]dp-esbup20:00:002draindp-esb[08,23]dp-esbup20:00:002allocdp-esb[09-10]dp-esbup20:00:0020idledp-esb[01-07,12-22,24-25]dp-sdv-esbup20:00:002resvdp-sdv-esb[01-02]psgw-clusterup20:00:001idlenfgw02debugup20:00:001drain*dp-dam08debugup20:00:004draindp-cn33,dp-dam[03,07],kn101debugup20:00:0010resvdp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002allocdp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002allocdp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002allocdp-dam[01,04]	dp-dam	up	20:00:00	2	alloc	dp-dam[01,04]
dp-dam-extup20:00:006idle dp-dam[11-16]dp-esbup20:00:0051drain* dp-esb[11,26-75]dp-esbup20:00:002drain dp-esb[08,23]dp-esbup20:00:002alloc dp-esb[09-10]dp-esbup20:00:0020idle dp-esb[01-07,12-22,24-25]dp-sdv-esbup20:00:002resv dp-sdv-esb[01-02]psgw-clusterup20:00:001idle nfgw01psgw-boosterup20:00:001idle nfgw02debugup20:00:001drain* dp-dam08debugup20:00:004drain dp-cn33,dp-dam[03,07],knl01debugup20:00:0010resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002alloc dp-dam[01,04]	dp-dam	up	20:00:00	8	idle	dp-dam[02,06,11-16]
dp-esbup20:00:0051 drain* dp-esb[11,26-75]dp-esbup20:00:002 drain dp-esb[08,23]dp-esbup20:00:002 alloc dp-esb[09-10]dp-esbup20:00:0020 idle dp-esb[01-07,12-22,24-25]dp-sdv-esbup20:00:002 resv dp-sdv-esb[01-02]psgw-clusterup20:00:001 idle nfgw01psgw-boosterup20:00:001 idle nfgw02debugup20:00:001 drain* dp-dam08debugup20:00:0010 resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:0020 resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:0020 resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:0020 resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:0020 resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:0020 resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002 alloc dp-dam[01,04]	dp-dam-ext	up	20:00:00	2	resv	dp-dam[09-10]
dp-esbup20:00:002draindp-esb[08,23]dp-esbup20:00:002allocdp-esb[09-10]dp-esbup20:00:0020idledp-esb[01-07,12-22,24-25]dp-sdv-esbup20:00:002resvdp-sdv-esb[01-02]psgw-clusterup20:00:001idlenfgw01psgw-boosterup20:00:001idlenfgw02debugup20:00:001drain*dp-dam08debugup20:00:0010resvdp-cn33, dp-dam[03,07], kn101debugup20:00:0010resvdp-cn[09-10,25,49-50], dp-dam[05,09-10], dp-sdv-esb[01-02]debugup20:00:002allocdp-dam[01,04]	dp-dam-ext	up	20:00:00	6	idle	dp-dam[11-16]
dp-esbup20:00:002allocdp-esb[09-10]dp-esbup20:00:0020idledp-esb[01-07,12-22,24-25]dp-sdv-esbup20:00:002resvdp-sdv-esb[01-02]psgw-clusterup20:00:001idlenfgw01psgw-boosterup20:00:001idlenfgw02debugup20:00:001drain*dp-dam08debugup20:00:004draindp-cn33,dp-dam[03,07],kn101debugup20:00:0010resvdp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]debugup20:00:002allocdp-dam[01,04]	dp-esb	up	20:00:00	51	drain*	dp-esb[11,26-75]
dp-esb       up       20:00:00       20       idle dp-esb[01-07,12-22,24-25]         dp-sdv-esb       up       20:00:00       2       resv dp-sdv-esb[01-02]         psgw-cluster       up       20:00:00       1       idle nfgw01         psgw-booster       up       20:00:00       1       idle nfgw02         debug       up       20:00:00       1       drain* dp-dam08         debug       up       20:00:00       4       drain dp-cn33,dp-dam[03,07],kn101         debug       up       20:00:00       10       resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]         debug       up       20:00:00       2       alloc dp-dam[01,04]	dp-esb	up	20:00:00	2	drain	dp-esb[08,23]
dp-sdv-esb       up       20:00:00       2       resv       dp-sdv-esb[01-02]         psgw-cluster       up       20:00:00       1       idle nfgw01         psgw-booster       up       20:00:00       1       idle nfgw02         debug       up       20:00:00       1       drain*       dp-dam08         debug       up       20:00:00       4       drain dp-cn33,dp-dam[03,07],kn101         debug       up       20:00:00       10       resv       dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]         debug       up       20:00:00       2       alloc       dp-dam[01,04]	dp-esb	up	20:00:00	2	alloc	dp-esb[09-10]
psgw-cluster       up       20:00:00       1       idle nfgw01         psgw-booster       up       20:00:00       1       idle nfgw02         debug       up       20:00:00       1       drain* dp-dam08         debug       up       20:00:00       4       drain dp-cn33,dp-dam[03,07],knl01         debug       up       20:00:00       10       resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]         debug       up       20:00:00       2       alloc dp-dam[01,04]	dp-esb	up	20:00:00	20	idle	dp-esb[01-07,12-22,24-25]
psgw-booster       up       20:00:00       1       idle nfgw02         debug       up       20:00:00       1       drain* dp-dam08         debug       up       20:00:00       4       drain dp-cn33,dp-dam[03,07],knl01         debug       up       20:00:00       10       resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]         debug       up       20:00:00       2       alloc dp-dam[01,04]	dp-sdv-esb	up	20:00:00	2	resv	dp-sdv-esb[01-02]
debug       up       20:00:00       1 drain* dp-dam08         debug       up       20:00:00       4 drain dp-cn33,dp-dam[03,07],knl01         debug       up       20:00:00       10 resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]         debug       up       20:00:00       2 alloc dp-dam[01,04]	psgw-cluster	up	20:00:00	1	idle	nfgw01
debug       up       20:00:00       4       drain dp-cn33,dp-dam[03,07],knl01         debug       up       20:00:00       10       resv dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]         debug       up       20:00:00       2       alloc dp-dam[01,04]	psgw-booster	up	20:00:00	1	idle	nfgw02
debug       up       20:00:00       10       resv       dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]         debug       up       20:00:00       2       alloc       dp-dam[01,04]	debug	up	20:00:00	1	drain*	dp-dam08
debug up 20:00:00 2 alloc dp-dam[01,04]	debug	up	20:00:00	4	drain	dp-cn33,dp-dam[03,07],knl01
	debug	up	20:00:00	10	resv	dp-cn[09-10,25,49-50],dp-dam[05,09-10],dp-sdv-esb[01-02]
debug up 20:00:00 69 idle deeper-sdv[06-16],dp-cn[01-08,11-24,26-32,34-48],dp-dam[02,06,11-16],kn1[04-0	debug	up	20:00:00	2	alloc	dp-dam[01,04]
	debug	up	20:00:00	69	idle	deeper-sdv[06-16],dp-cn[01-08,11-24,26-32,34-48],dp-dam[02,06,11-16],knl[04-0

Please refer to the man page for  $\mathtt{sinfo}$  for more information.

#### Can I join stderr and stdout like it was done with -joe in Torque?

Not directly. In your batch script, redirect stdout and stderr to the same file:

#!sh ... #SBATCH -o /point/to/the/common/logfile-%j.log #SBATCH -e /point/to/the/common/logfile-%j.log ...

(The %j will place the job id in the output file). N.B. It might be more efficient to redirect the output of your script's commands to a dedicated file.