Wikiprint Book

Title: Information about the batch system (SLURM)

Subject: DEEP - Public/User_Guide/Batch_system

Version: 63

Date: 19.05.2024 17:55:57

Table of Contents

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)
knl	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 ?Slurm documentation 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 `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.:

```
[kreutz1@deepv ~]$ srun -A deep -N 1 -n 8 -p dp-cn -t 00:30:00 --pty --interactive bash
[kreutz1@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:

```
[kreutz1@deepv Temp]$ salloc -A deep -p dp-cn -N 4 -n 8 -t 00:30:00 srun --pty --interactive /bin/bash
[kreutz1@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-cn02
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:

```
[kreutz1@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`.

```
[kreutz1@deepv Temp]$ ml GCC/10.3.0 ParaStationMPI/5.4.9-1
[kreutz1@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-cn02
Hello World from rank 4 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:

```
[kreutz1@deepv Temp]$ salloc -A deep -p dp-cn -N 4 -n 8 -t 00:30:00
salloc: Granted job allocation 69263
[kreutz1@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
...
[kreutz1@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 Temp]$ squeue -u $USER
            JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            69264     dp-cn hello_cl  kreutz1 CG       0:04      4 dp-cn[01-04]
```

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

```
[kreutz1@deepv Temp]$ cat /p/project/cdeep/kreutz1/hello_cluster-69264.out
Hello World from rank 7 of 8 on dp-esb37
Hello World from rank 3 of 8 on dp-esb35
Hello World from rank 5 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 0 of 8 on dp-esb34
```