

Wikiprint Book

Title: System usage

Subject: DEEP - Public/User_Guide/DEEP-EST_DAM

Version: 24

Date: 17.04.2025 14:23:34

Table of Contents

System usage	3
Persistent Memory	3
Using Cuda	3
Using FPGAs	3
Filesystems and local storage	4
Multi-node Jobs	4

System usage

The DEEP-EST Data Analytics Module (DAM) can be used through the SLURM based batch system that is also used for (most of) the Software Development Vehicles (SDV). You can request a DAM node (dp-dam[01-16]) with an interactive session like this:

```
srun -A deepsea -N 1 --tasks-per-node 4 -p dp-dam --time=1:0:0 --pty --interactive /bin/bash
kreutzl@dp-dam01 ~]$ srun -n 8 hostname
dp-dam01
dp-dam01
dp-dam01
dp-dam01
```

When using a batch script, you have to adapt the partition option within your script: `--partition=dp-dam` (or short form: `-p dp-dam`)

Persistent Memory

Each of the DAM nodes is equipped with [?Intel's Optane DC Persistent Memory Modules](#) (DCPMM). All DAM nodes (dp-dam[01-16]) expose 3 TB of persistent memory.

The DCPMMs can be driven in different modes. For further information of the operation modes and how to use them, please refer to the following [?information](#)

Currently all nodes are running in "App Direct Mode".

Using Cuda

The first 12 DAM nodes are equipped with GPUs

- dp-dam[01-08]: 1 x Nvidia V100
- dp-dam[09-12]: 2 x Nvidia V100

Please use the `gres` option with `srun` if you would like to use GPUs on DAM nodes, e.g. in an interactive session:

```
srun -A deepsea -p dp-dam --gres=gpu:1 -t 1:0:0 --interactive --pty /bin/bash # to start an interactive session on an DAM
srun -A deepsea -p dp-dam --gres=gpu:2 -t 1:0:0 --interactive --pty /bin/bash # to start an interactive session on an DAM
```

To compile and run Cuda applications on the Nvidia V100 cards included in the DAM nodes, it is necessary to load the CUDA module. It's advised to use the 2022 Stage to avoid [Nvidia driver mismatch](#) issues.

```
module --force purge
ml use $OTHERSTAGES
ml Stages/2022
ml CUDA
[kreutzl@deepv ~]$ ml

Currently Loaded Modules:
  1) Stages/2022 (S)   2) nvidia-driver/.default (H,g,u)   3) CUDA/11.5 (g,u)

Where:
S:  Module is Sticky, requires --force to unload or purge
g:  built for GPU
u:  Built by user
```

Using FPGAs

Nodes dp-dam[13-16] are equipped with 2 x Stratix 10 FPGAs each ([?Intel PAC d5005](#)).

It is recommended to do the first steps in an interactive session on a DAM node. Since there is (currently) no FPGA resource defined in SLURM for these nodes, please use the `--hostlist=` option with `srun` to open a session on a DAM node equipped with FPGAs, for example:

```
srun -A deepsea -p dp-dam --odelist=dp-dam13 -t 1:0:0 --interactive --pty /bin/bash
```

For getting started using OpenCL with the FPGAs you can find some hints as well as the slides and exercises from the Intel FPGA workshop held at JSC in:

```
/usr/local/software/legacy/fpga/
```

More details to follow.

Filesystems and local storage

The home filesystem on the DEEP-EST Data Analytics Module is provided via GPFS/NFS and hence the same as on (most of) the remaining compute nodes. The DAM is connected to the all flash storage system (AFSM) system via Infiniband. The AFMS runs BeeGFS and provides a fast local work filesystem at

```
/work
```

In addition, the older SSSM storage system provides the `/usr/local` filesystem on the DAM compute nodes running BeeGFS as well.

There is node local storage available for the DEEP-EST DAM node (2 x 1.5 TB NVMe SSD), it is mounted to `/nvme/scratch` and `/nvme/scratch2`. Additionally, there is a small (about 380 GB) scratch folder available in `/scratch`. Remember that the three **scratch folders** are not persistent and **will be cleaned after your job has finished !**

Please, refer to the [system overview](#) and [filesystems](#) pages for further information of the CM hardware, available filesystems and network connections.

Multi-node Jobs

The latest `pscom` version used in ParaStation MPI provides support for the Infiniband interconnect used in the DEEP-EST Data Analytics Module. Hence, loading the most recent ParaStationMPI module will be enough to run multi-node MPI jobs over Infiniband:

```
module load ParaStationMPI
```

For using Cluster nodes in heterogeneous jobs with DAM and ESB nodes no gateway has to be used (anymore), since all 3 compute modules (as well as the login and file servers) are using EDR Infiniband as interconnect.

TBD:

- Cuda aware MPI with GPU DAM nodes

For using DAM nodes in heterogeneous jobs together with CM and/or ESB nodes no gateway has to be used (anymore), since all 3 compute modules (as well as the login and file servers) are using EDR Infiniband as interconnect. For further information, please also take a look at [heterogeneous jobs](#).