

Cluster Deployment Guidelines

Robotics & Artificial Intelligence (RAI)

About Multi-node Computation

A computer cluster consists of multiple computers (nodes) connected via high-speed networks and working together as a single system. Clusters are cost-effective alternatives to large single computers, offering improved performance and availability.

A node is an individual computer within a cluster, typically containing one or more CPUs (with multiple cores) and possibly GPUs. While memory is shared between cores within the same CPU, it is not shared across different nodes.

Jobs on a cluster are managed through a batch system. Users log in to a "login node" and submit job scripts, which specify requirements like the number of nodes, CPUs, GPUs, memory, runtime, and input data. These scripts enable non-interactive job execution, ideal for resource-intensive tasks that run without user interaction.

HPC2N & Alvis

There are two clusters available to the RAI group.

1. High Performance Computing Center North (HPC2N) is a national center for Scientific and Parallel Computing. This collaboration and coordination between universities and research institutes form a competence network for high performance computing (HPC), scientific visualization, and virtual reality (VR) in Northern Sweden.

2. The Alvis cluster is a national NAISS resource dedicated for Artificial Intelligence and Machine Learning research. The system is built around Graphical Processing Units (GPUs) accelerator cards, and consists of several types of compute nodes with multiple NVIDIA GPUs.

Cluster Resource Allocation

Batch or scheduling systems are essential for managing multi-user jobs on clusters or supercomputers. These systems track available resources, enforce usage policies, and schedule jobs efficiently by organizing them into priority queues. Jobs are submitted using job scripts, which specify resource requirements (e.g., nodes, cores, GPUs, memory) and include commands to execute tasks. Outputs and error logs are generated after job completion.

salloc is a scheduler command used to allocate a job, which is a set of resources (nodes), possibly with some set of constraints (e.g. number of processors per node). If no command is specified, then by default salloc starts the user's default shell on the same machine.

```
1 # Allocate 1 node with 4 workers for 1 hour and 30 minutes
2 salloc --account <your project> --nodes=1 --ntasks-per-node=4 --time=1:30:00
3
4 # You must use srun to run your job on the allocated resources
5 srun --ntasks 2 python program.py <ARGS>
```

SLURM Workload Manager

SLURM (Simple Linux Utility for Resource Management) is a widely used open-source job scheduling system for Linux and Unix-like environments. It plays a crucial role in managing resources and scheduling on clusters, including many supercomputers.

SLURM provides three main functions:

- Resource Allocation: Grants users exclusive or shared access to nodes for a set duration.
- Job Management: Offers a framework for starting, executing, and monitoring parallel job.
- Queue Management: Manages job queues, resolving contention for resources.

Jobs and the resources associated are requested and controlled through the following commands. The `job.sh` file contains all the specifications for the allocation and the programs to be executed.

```
1 sbatch job.sh           # Submit a job
2 squeue -u USERNAME -j JOBID # Job status
3 scontrol show job JOBID    # Job information
4 scancel JOBID             # Cancel a job
```

SLURM directives in job scripts are prefixed with `#SBATCH`, while general comments are prefixed with `#`. This system enables efficient resource utilization and streamlined job execution on high-performance computing systems.

```
1 ''' Job Script Example file: job.sh '''
2
3 #SBATCH --account hpc2nXXX-YYY      # The name of the account you are running in, mandatory.
4 #SBATCH --job-name my_job_name     # Give a sensible name for the job
5 #SBATCH --time=00:15:00            # Request runtime for the job (HHH:MM:SS)
6
7 #SBATCH --error=job.%J.err         # Set the names for the error and output files
8 #SBATCH --output=job.%J.out        # %J is equivalent to the specified job name
9
10 # The following directives set up two nodes with 1 V100 GPU each
11 # *****
12 #SBATCH --ntasks 2                 # Number of workers (recommended 1 worker per GPU)
13 #SBATCH --nodes 2                  # Number of nodes
14 #SBATCH --gpus-per-node=v100:1    # Number of GPU cards needed per node
15 # *****
16
17 srun python program.py <ARGS>     # Run program on allocated resources
```

GPU types include **v100**, **a40**, **a6000**, **l40s** and **h100**. When no GPU type is specified the scheduler will allocate any free GPU in the cluster. For more information read the clusters' documentation indicated in the references.

The following directive allocates a node exclusively for a job even if there is enough resources for another job.

```
1 #SBATCH --exclusive
```

The following directive allows the selection of the type of instance of the nodes allocated on the cluster.

```
1 #SBATCH --constraint=skylake # HPC2N example
```

Sample Code

Available code and job scripts with examples on (1) single-machine multi-gpu and (2) cluster deployment can be found in the following repository: <https://github.com/fernand0labra/rai-cluster-deployment.git>
The repository showcases code changes in Pytorch and Tensorflow frameworks for distributed NN training.

References

About HPC2N

HP2CN Information <https://www.hpc2n.umu.se/about>

HP2CN Documentation <https://docs.hpc2n.umu.se/tutorials/clusterguide/>

About Alvis

Alvis Information <https://www.c3se.chalmers.se/about/Alvis/>

Alvis Documentation https://www.c3se.chalmers.se/documentation/for_users/intro-alvis/slides/

SLURM Documentation (HPC2N)

Basic Commands https://docs.hpc2n.umu.se/documentation/batchsystem/basic_commands/

Basic Examples https://docs.hpc2n.umu.se/documentation/batchsystem/basic_examples/

Submit File Design https://docs.hpc2n.umu.se/documentation/batchsystem/submit_file_design/

Job Submission https://docs.hpc2n.umu.se/documentation/batchsystem/job_submission/

Batch Scripts https://docs.hpc2n.umu.se/documentation/batchsystem/batch_scripts/