UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

ziti

Tools 2025 – Cluster Introduction

Shigehiko Schamoni

12.06.2025

Overview

- Introduction
- Basics
- Slurm Usage
- Exercise

- Introduction
- Basics
- Slurm Usage
- Exercise

Why to Use a Compute Cluster

- Processes require more resources than a single server or a workstation provides
- Parallel processing of large amounts of data
- Huge models that require distributed computation
- Tuning of model parameters
- Typical properties of a cluster:
  - Sharing of resources
  - Uses a batch system s.t. a user's tasks don't interfere
  - Load of RAM, CPU cores, GPU resources

Clusters for ZITI

- bwUniCluster
  - Accessible to (practically) every university member after registration
  - `https://wiki.bwhpc.de/e/BwUniCluster2.0`
  - `https://wiki.bwhpc.de/e/BwUniCluster3.0`
- bwForCluster
  - More complex application process: needs an entitlement
  - `https://wiki.bwhpc.de/e/Registration/bwForCluster`
- ZITI's research groups
  - HAWAII-Cluster
  - ASC-Cluster
  - NCA-Cluster
- Common to all clusters above:
  - Slurm batch system
  - `https://slurm.schedmd.com/documentation.html`

- Introduction
- Basics
- Slurm Usage
- Exercise

Login via SSH

- Secure shell (SSH) is the most common way to connect to remote machines.
- Linux, Mac, and BSD users have the command `ssh` installed by default.
- Windows users can use WSL or PuTTy[1]
- The command `ssh` expects a username and a hostname seperated by an @-sign:

```
:~$ ssh <user>@zitigate.ziti.uni-heidelberg.de
```

  `<user>` is typically your UniID.
- The example above creates an ssh-connection to a gate server which enables access to ZITI's compute resources.
- The gate's system is very restricted and should only be used as a *jump host*.

[1] `https://www.chiark.greenend.org.uk/~sgtatham/putty/`

Login via SSH contd.

- To jump to a different server, run `ssh` on the gate:

```
:~$ ssh <user>@<hostname>
```

  If `hostname` is part of the ZITI domain, you don't need the fully qualified domain name (FQDN).
- The jump host configuration can be automated by editing the client's config file `$HOME/.ssh/config`:

```
Host zitigate
HostName zitigate.ziti.uni-heidelberg.de
User <username>
Host headnode
HostName csg-headnode
User <username>
ProxyJump zitigate
```

- Now one can directly connect to `csg-headnode` by

```
:~$ ssh headnode
```

- Hint: use an ssh-key to avoid being locked out by mistyped passwords!

- Introduction
- Basics
- Slurm Usage
- Exercise

Slurm

- Users don't have direct access to cluster's compute nodes.
- All computations need to go through the Slurm batch system.
- The batch system collects information on all jobs and distributes them among the nodes.
- There exist several queues (partitions) with different properties:
  - Compute resources (CPU cores, GPUs)
  - File systems
  - Installed software
  - Maximum job's run time
  - etc.
- If the job exceeds its limits it is killed by the Slurm system.

Basic Slurm Commands (I)

- The `sinfo` command returns general information about all accessible queues

```
:~$ sinfo
```

- Jobs are started by using the `srun` command:

```
:~$ srun [-p <partition >] [-w <node >] --pty
-- <command>
```

  If the requested resources are currently not available, the job is queued.

- The `squeue` command shows the status of all active queues

```
:~$ squeue
```

Basic Slurm Commands (II)

- Additional resources are be requested via command line parameters, e.g.
  - Allocate a single GPU for a job:

    ```
    :~$ srun --gres=gpu:1 <...>
    ```
  - Allocate a eight CPUs for a job:

    ```
    :~$ srun --cpus-per-task 8 <...>
    ```
- To cancel a running job, look up its job ID:

  ```
  :~$ squeue -u <username>
  ```
- Then issue the `scancel` command

  ```
  :~$ scancel <jobid>
  ```

  Note: you can only cancel your own jobs unless you have superuser privileges.

Advanced Slurm Usage

- Slurm supports special shell scripts using an `sbatch` mechanism
- Command line parameters and shell commands are combined in a single file:

```
#!/bin/bash
#SBATCH --job-name=my_batch_job
#SBATCH --output=my_result.txt
#SBATCH --mail-user=UniID@stud.uni-heidelberg.de
#SBATCH --mail-type=ALL
#SBATCH --partition=compute

# JOB STEPS
srun hostname # example job step
srun echo $CUDA_VISIBLE_DEVICES # another example job step
```

- Handy for starting multiple jobs with different parameters
- Good introduction: `https://hpc.nmsu.edu/discovery/slurm/slurm-commands/`

- Introduction
- Basics
- Slurm Usage
- Exercise

SSH Exercise

- Log in to `zitigate` using your UniID
- Generate a keypair on your client machine (e.g. the pool computer or your laptop):

```
:~$ ssh-keygen -t ed25519
```

- Add the public key to `$HOME/.ssh/authorized_keys` on `zitigate` and your target machine (e.g. `physik1.kip.uni-heidelberg.de`).
- Edit your ssh-configuration to use zitigate as a jump host to connect to your target machine.
- Verify if you can directly reach the target without a password.

Slurm Demo

- Demonstration of basic slurm commands.