Friedrich-Alexander-Universität
Technische Fakultät

Computer Architecture Practical Exercise

0 Cluster Introduction

Kenan Gündogan[1] **Philipp Gündisch**[1]

[1]Friedrich-Alexander Universität Erlangen-Nürnberg, Chair of Computer Science 3 (Computer Architecture)

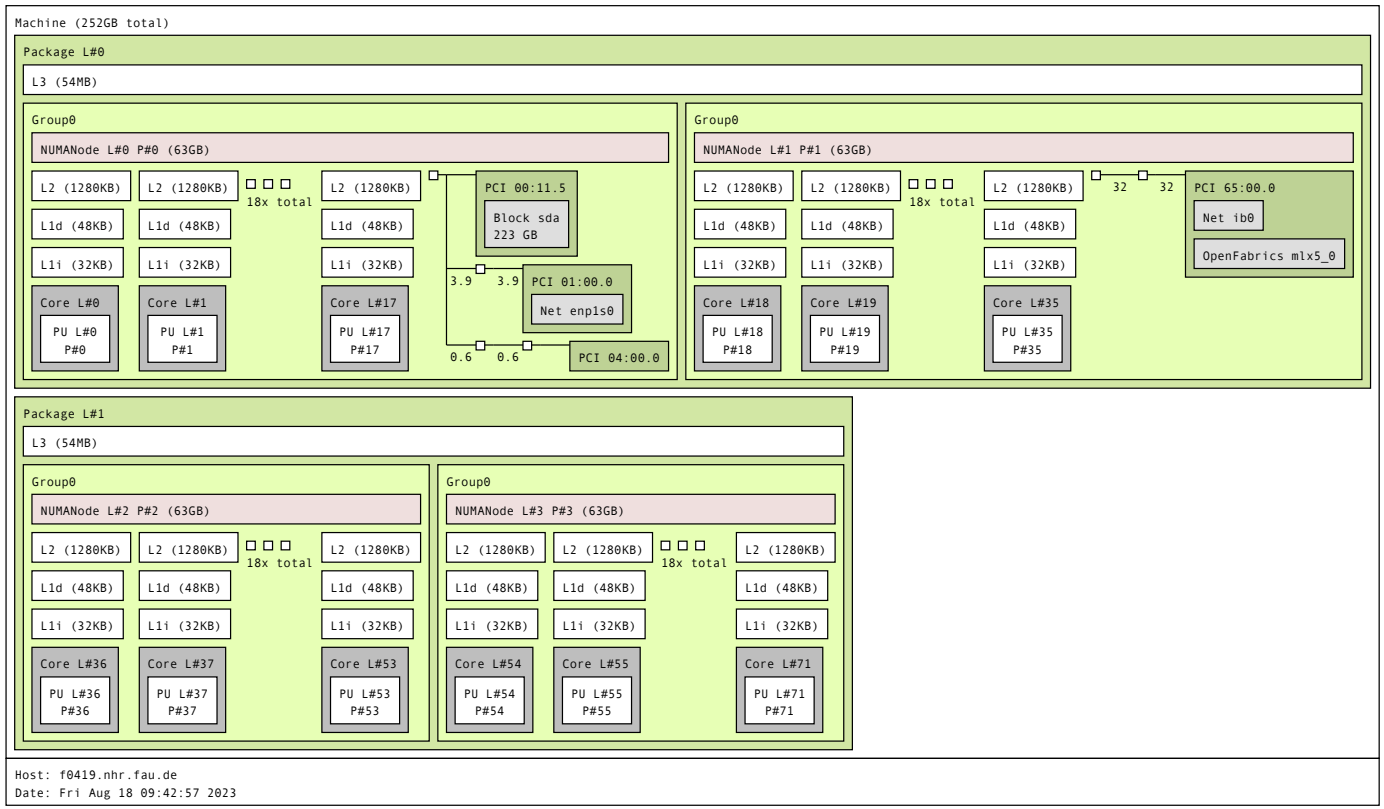
October 29, 2024

Cluster Resources

- The compute nodes of the **NHR** (Erlangen National High Performance Computing Center) are used in this exercise
- The ***fritz*** cluster consists of over 1000 compute nodes
  - 100 GBit/s HDR100 Infiniband interconnect
  - Dual-socket system with 16 x 16 GB of DDR4-3200 RAM
    - Ice Lake *(Intel Xeon Platinum 8360Y)*
    - 36 cores per chip
    - 2.4 GHz clock frequency
  - More information: see fritz

fritz Node

- Dual-socket Ice Lake node
  - 2x Xeon Platinum 8360Y Chips (CPUs, Packages) with 16 x 16 GB of RAM
  - 36 physical cores and 72 logical cores (SMT-Threads) per processor
  - CPU clock frequency of 2.4 GHz (nominal) and 3.5 GHz peak (turbo) frequency



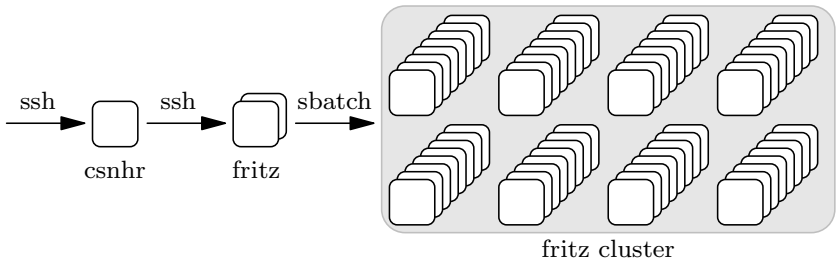
Cluster Access

- For each student an HPC account *(hpcvXXXv)* was created
- **You should have a mail invitation received**
- There is no password authentication possible
- For authentication to the server an ssh key needs to be generated and uploaded
- For configuration visit the HPC Portal

Note: The synchronization of the ssh public key to the servers can take up to two hours!

Log in to Cluster

- Access from outside of the university network is possible via dialog-server:
  `ssh hpcvXXXh@fritz.nhr.fau.de`



- There is no password authentication possible
- From the dialog server / university network the cluster head node is reachable via `ssh`
- To run a program on the cluster `slurm` tools need to be used

Slurm

salloc

- Slurm is a cluster management and job scheduling system
- Cluster access needs to be requested via Slurm
- A cluster node can be allocated for 1 hour with:

```
salloc
    --partition=singlenode    // fritz subset of nodes
    --time=01:00:00           // allocate for 1 hour
    --nodes=1                 // allocate 1 node
    --ntasks-per-node=1       // for 1 process
    --cpus-per-task=1         // for 1 thread
```

Slurm

sbatch

The `sbatch` command is able to allocate cluster resources and run a shell script. The resource allocation information is parsed by `sbatch` from the provided shell script (one argument per line).
An exemplary `myjob.sh` file header might look like this:

```
#/bin/bash -l

#SBATCH --partition=singlenode
#SBATCH --time=00:10:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --export=NONE

...
srun ./program
```

The shell script can be invoked with:
`sbatch myjob.sh`
You can find more information on the NHR website about slurm.

Responsible Usage of Resources

- Use the cluster responsibly!
- Allocate only one node at a time
- Calculate / predict the job runtime and specify `--time`
- Test your implementation and script before submitting a long running job

Shell Functions

You can write functions and invoke them in bash scripts very similar to other programming languages. The following example illustrates how it works.

```
#!/bin/bash -l

function multiply() {
    echo $(($1*$2))
}

echo $(multiply 2 10)
```

When invoked the example produces the following output.

```
$ ./example.sh
20
```

For-Loops

You can also write for-loops in bash scripts. The following example illustrates how for-loops work.

```
#!/bin/bash -l

for name in "Alf" "Joe"
do
    echo $name
done

for (( i = 0; i < 2; i += 1))
do
    echo $i
done
```

When invoked the example produces the following output.

```
$ ./example.sh
Alf
Joe
0
1
```

Exercise 0.1: Log in to cluster

- Configure the `ssh` access as described in the NHR website on SSH
- Use public key authentication
- Log in to the cluster with `ssh fritz`

Exercise 0.2: Running provided code on cluster

- To list available modules on a compute node run:
  - `module avail`
- To make `icx` available on a compute node run:
  - `module load intel`
- To compile the code run `icx -i src/hello.c -o bin/hello`
- Inspect current cluster load
  - HPC status page
  - To retrieve necessary login data please run `docpw` command on current cluster node
- Update the provided shell script to make use of functions and loops (e.g. run `bin/hello` 5 times)
- Use `sbatch` to run the shell script on the cluster
- Select walltime less than 1h: Devel queue (usually shorter waiting times)
- Analyze the produced `sbatch` output

Tasks

- E 0.1: Set Up
  - Configure ssh access
  - Log in to cluster
- E 0.2: Run code on cluster
  - Copy files via `scp`
  - Start an interactive session via `ssh` (terminate with `exit` or *ctrl+D*)
  - Implement on head node
  - Work on compute node only for benchmarking and testing
  - Compile `src/hello.c` with `icx`
  - Run `sbatch scripts/cluster.sh` on cluster