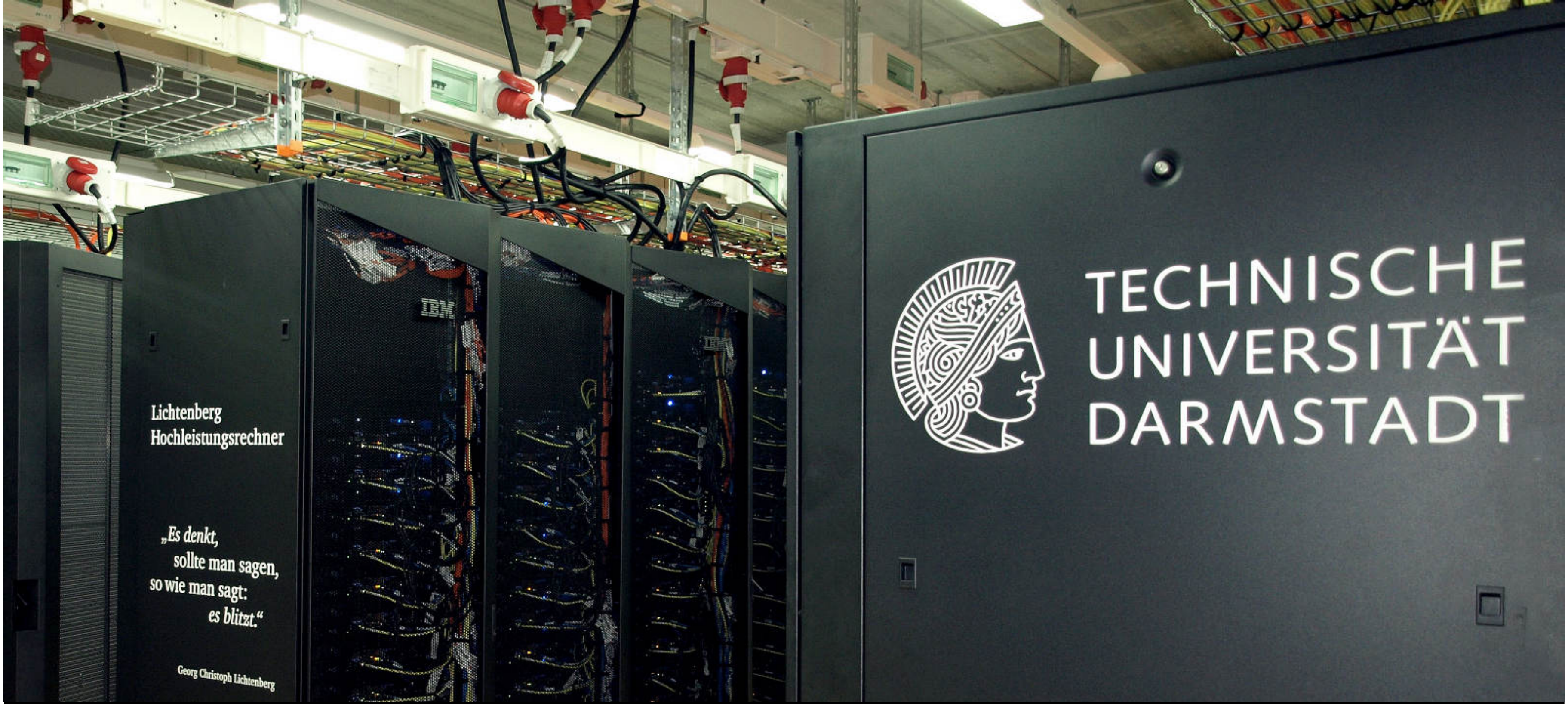
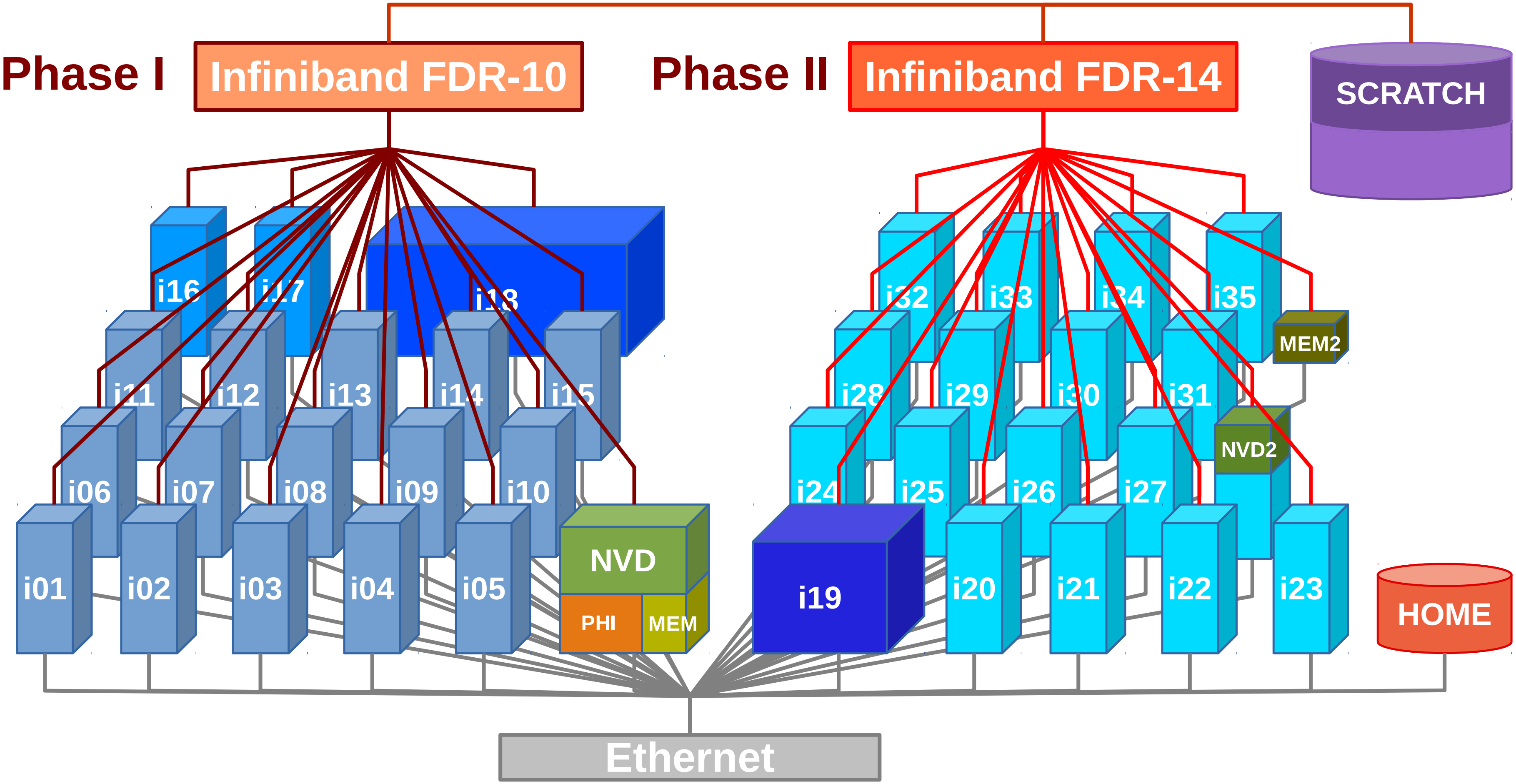
Introduction to the Lichtenberg Cluster

TECHNISCHE UNIVERSITÄT DARMSTADT

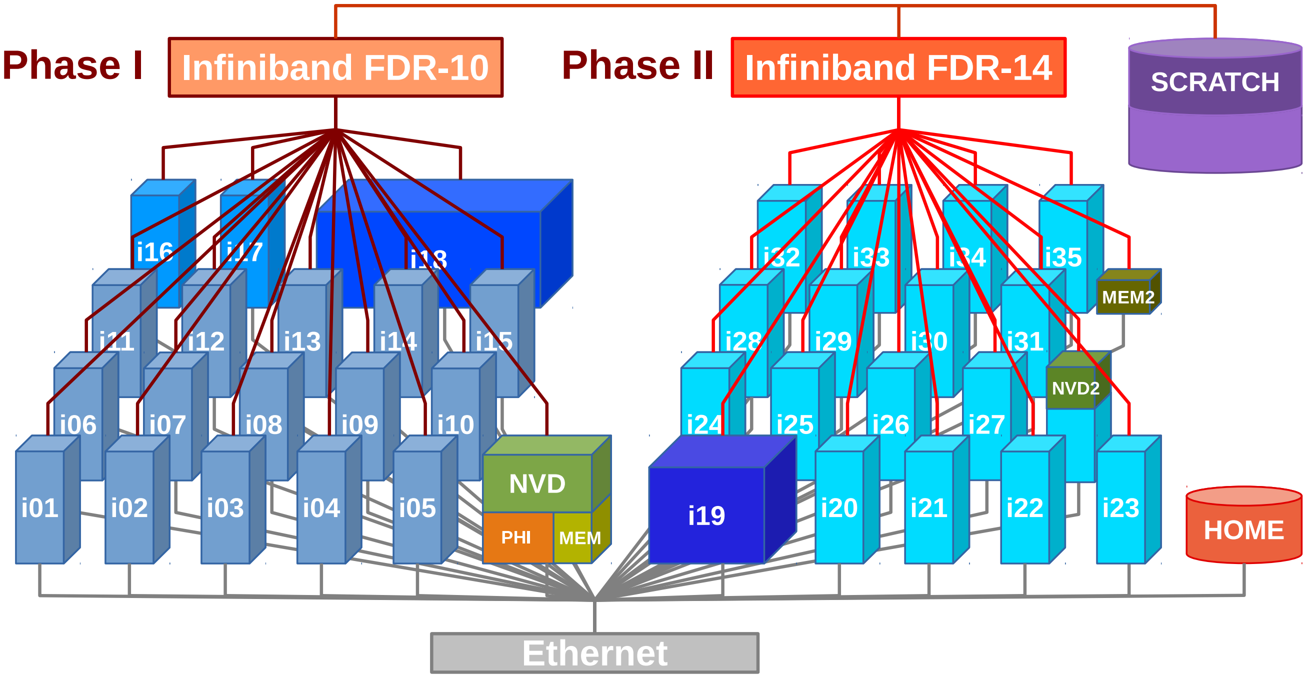


HRZ
Hochschulrechenzentrum

One Cluster – Multiple Islands



One Cluster – Multiple Islands (2)



- Cluster is divided into 2 phases
- Each phase is divided into several islands
- Rule of thumb:
  1 island ≙ 32 compute nodes ≙ 512 (phase I) / 768 (phase II) CPU cores

HARDWARE

Compute Nodes

Phase I (706+70 nodes):

- Processors:
  - 2 Intel Xeon E5-2670 (**Sandy Bridge**) processors ≙ 2 · 8 = **16 CPU cores**
  - **2.6 GHz** (up to 3.3 GHz in turbo mode)
- Main Memory:
  - **32 GB RAM** (some have 64 GB)
- Network:
  - Gigabit Ethernet
  - FDR-10 InfiniBand

Phase II (596+32 nodes):

- Processors:
  - 2 Intel Xeon E5-2680 v3 (**Haswell**) processors ≙ 2 · 12 = **24 CPU cores**
  - **2.5 GHz** (up to 3.3 GHz in turbo mode)
- Main Memory:
  - **64 GB RAM**
- Network:
  - Gigabit Ethernet
  - FDR-14 InfiniBand

Login Nodes

4 nodes (hardware similar to phase I):

- Processors:
  - 4 Intel Xeon E5-4650 (**Sandy Bridge**) processors ≙ 4 · 8 = **32 CPU cores**
  - **2.7 GHz** (up to 3.3 GHz in turbo mode)
- Main Memory:
  - **128 GB RAM**
- Network:
  - 2 · 10 Gigabit Ethernet
  - 2 · FDR-10 InfiniBand

8 nodes (hardware similar to phase II):

- Processors:
  - 2 Intel Xeon E5-2680 v3 (**Haswell**) processors ≙ 2 · 12 = **24 CPU cores**
  - **2.5 GHz** (up to 3.3 GHz in turbo mode)
- Main Memory:
  - **128 GB RAM**
- Network:
  - 2 · 10 Gigabit Ethernet
  - FDR-14 InfiniBand

Purpose of the Login Nodes

- Transferring files to the cluster
- Editing files on the cluster
- Compiling software
- Testing software (time limit: 30 minutes)
- Debugging small test cases

- Submitting computations to the batch system (explained later)
- Checking status of compute jobs

ACCESS TO THE CLUSTER

Set Password for Cluster Access

1. Go to https://www.idm.tu-darmstadt.de/ando

2. Log in with TU-ID

3. Click on tab “HLR”

4. Set password

Connecting to the Cluster

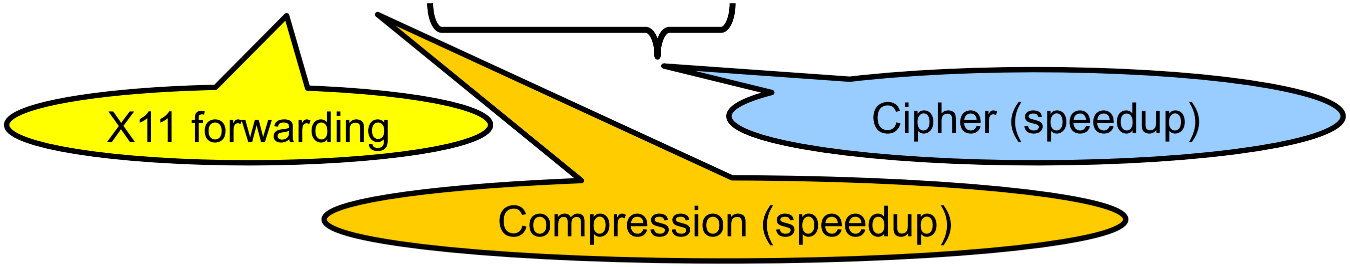
Open an SSH connection to one of the login nodes:

- lcluster1.hrz.tu-darmstadt.de – lcluster4.hrz.tu-darmstadt.de
  - Sandy Bridge, 32 cores, 128 GB RAM
- lcluster5.hrz.tu-darmstadt.de – lcluster11.hrz.tu-darmstadt.de
  - Haswell, 24 cores, 128 GB RAM

SSH Clients

Linux

- `ssh <tu-id>@lcluster9.hrz.tu-darmstadt.de`
- `ssh -X -C -c blowfish <tu-id>@lcluster9.hrz.tu-darmstadt.de`



X11 forwarding

Cipher (speedup)

Compression (speedup)

Windows

- Cygwin (https://www.cygwin.com): `ssh` as with Linux above
- PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

File Transfer to the Cluster

Copy files between own computer and cluster

- Linux
  - `scp <local_src> <tu-id>@lcluster9.hrz.tu-darmstadt.de:<remote_dst>`
  - `scp <tu-id>@lcluster9.hrz.tu-darmstadt.de:<remote_src> <local_dst>`

- Windows
  - Cygwin (https://www.cygwin.com): **`scp`** as with Linux above
  - PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

File Transfer to the Cluster (2)

Alternative to copying files between cluster and own computer is to mount remote cluster file system via SSH on own computer

- Linux
  - SSHFS
- Windows
  - Win-sshfs
  - NetDrive

SOFTWARE

Operating System

- Linux distribution CentOS 7.5
- Some programs are included in the operating system
  - You can just use these programs, e.g., text editors such as **`vim`**, **`nano`**, **`mc`**, **`gedit`** for directly editing files on the login nodes
  - Only a limited number of packages is provided by the operating system

Additional Software

- A lot of software is provided by the module system
  - Compilers (GCC, Intel, ...)
  - Libraries (MPI, CUDA, ...)
  - Tools (SVN, Git, GDB, ...)
  - Application software

- Use **`module avail`** or **`module whatis`** to obtain a complete list of all available modules
- Use **`module help <modulename>`** to get further details about specific module

Module System

- To load a module, use the command **`module load <modulename>`**
  - Read the hints that might appear
  - Sometimes you have to load additional modules
  - Typically, the module name has to contain the version number,

    e.g. **ansys/17.1**
  - For some modules, a default version is set

    E.g., it may be sufficient to write **`module load gcc`**, but not **`module load openmpi`** – choose module **`openmpi/gcc`** or **`openmpi/intel`**
- To show all currently loaded modules, use **`module list`**
- To remove a module, use **`module unload <modulename>`**
- To remove all modules, use **`module purge`**

QUESTIONS & DEMO LOGIN

USING THE CLUSTER

Batch System

- Do not confuse the login nodes with the cluster
  - There are only few login nodes but >1000 compute nodes
  - Login nodes are the point of access to the cluster
  - Connecting to compute nodes not possible

- **Use the batch system to submit jobs from the login nodes**
  - Write **job script** file with all job parameters and commands and submit the script

- The cluster runs 24/7 (including weekends)

Job Script Example

```
#!/bin/bash
```

```
#SBATCH -J my_job
#SBATCH --mail-user=my@mail.com
#SBATCH --mail-type=ALL
#SBATCH -e /home/<tu-id>/somewhere/Job_Name.err.%j
#SBATCH -o /home/<tu-id>/somewhere/Job_Name.out.%j
#SBATCH --mem-per-cpu=250
#SBATCH -t 00:05:00
#SBATCH -n 4
#SBATCH --account=kurs00025
#SBATCH --partition=kurs00025
#SBATCH --reservation=kurs00025
```

Job parameters

```
echo "This is Job $SLURM_JOB_ID"

module load gcc
module load openmpi/gcc
mpirun hostname
```

Commands to run your program

Job Script Example (2)

```
#!/bin/bash

#SBATCH -J my_job
#SBATCH --mail-user=my@mail.com
#SBATCH --mail-type=ALL
#SBATCH -e /home/<tu-id>/somewhere/Job_Name.err.%j
#SBATCH -o /home/<tu-id>/somewhere/Job_Name.out.%j
#SBATCH --mem-per-cpu=250
#SBATCH -t 00:05:00
#SBATCH -n 4
#SBATCH --account=kurs00025
#SBATCH --partition=kurs00025
#SBATCH --reservation=kurs00025

echo "This is Job $SLURM_JOB_ID"

module load gcc
module load openmpi/gcc
mpirun hostname
```

%j will be replaced
by the actual job id

Job Parameters

You should always set the following parameters:

- **-n <tasksnum>**: Number of tasks (processes)
  Default: one CPU core per task

- **--mem-per-cpu=<memory>**: Maximum memory per core in MB

- **-t <time>**: Time limit of the job (in minutes)
  Maximum time limit is 5 min for SPP

- **--partition=kurs00025**
- **--account=kurs00025**
- **--reservation=kurs00025**

Additional Job Parameters

- **-J <jobname>**: Name of the job (does not have to be unique)
- **--mail-user=<email>**: Email address
- **--mail-type=<BEGIN | END | ALL | […]>**: Send an email

- **-o <filename>**: Write standard output (‘stdout’) of the job to a file
- **-e <filename>**: Write standard error (‘stderr’) of the job to a file

- **-c <number>**: Number of CPU cores per task

  Important for OpenMP with several cores per process

Batch System Commands

- `sbatch <JobScriptFile>`: Submit the job

- `squeue`: List all your jobs
- `sjobs <JobID>`: Print detailed information about a job
- `scancel <JobID>`: Kill a particular job
- `scancel –u <TU-ID>`: Kill all your own jobs

Troubleshooting

Job was submitted but does not start

- Job does not start when it exceeds limits
- Time limit is 5 min
- Memory per core should not exceed 1600 MB
- Memory per task must not exceed 26 GB
- We have three exclusive nodes with 16 cores and 26 GB each

- When job limits are correct, job only starts when resources on the three nodes are available
- Job start might be delayed due to jobs of other students

Further Information

http://www.hhlr.tu-darmstadt.de

QUESTIONS & DEMO JOB