C | A | U
Christian-Albrechts-Universität zu Kiel
Rechenzentrum

Computing systems of the University Computing Centre (RZ)

Dr. Karsten Balzer
University Computing Centre (RZ)

07 September 2018
09:00 - 10:30

Outline

I. HPC systems of the Computing Centre

- **Linux cluster**
  - rzcluster
  - caucluster
- **Hybrid NEC HPC system**
  - NEC HPC Linux cluster
  - NEC SX-ACE vector system (until end 2018)
  - NEC SX-Aurora Tsubasa vector system

II. Practical work on the HPC systems

- **Login and interactive working**
  - Login nodes (front ends) and file systems
  - Software, compilers, libraries ...
- **Batch processing**
  - Batch nodes, batch system und commands
  - Batch scripts to start computations

III. Account application, documentation and HPC support

I. HPC systems of the Computing Centre

- **x86-based Linux clusters: rzcluster and caucluster**
  - **For rather general purpose: serial and not highly parallel applications**
  - **Partly for high-memory applications**

- **rzcluster**
  - **1 login node (12 cores Intel-Westmere, 48 GB main memory)**
  - **$\approx$ 140 compute (batch) nodes, $\approx$1900 cores**
    - Grown step-by-step, some resources age $>$ 5 years
    - Different processors and processor generations (AMD and Intel)
    - Cores per node: 8 - 32
    - Main memory per node: 32 GB - 1 TB
    - Restricted general resources:
      - 40 batch nodes (388 cores in total)
      - 23 nodes with 8 cores and 32 GB
      - 17 nodes with 12 cores and 48 GB
  - **Operating system: Scientific Linux 6.x, partly CentOS 7.x**
  - **Batch system: SLURM**

C | A | U
Christian-Albrechts-Universität zu Kiel
Rechenzentrum

- caucluster
  - **1 login node (8 cores Intel-Haswell, 64 GB)**
  - **7 batch nodes, 280 cores**
    - 40 cores Intel-Haswell nodes (4-socket), 1.9 GHz
    - Main memory per node: 256 GB
  - **Operating system: CentOS 7.x**
  - **Batch system: SLURM**
  - **caucluster is/shall be the nucleus of a "new" Linux cluster**

- Shared resources
  - **rz- and caucluster share the same I/O file systems**

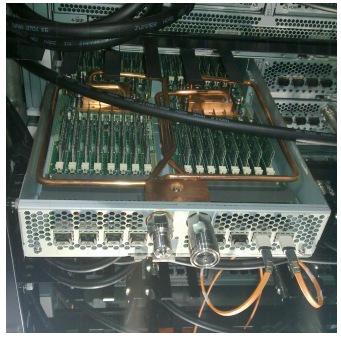
NEC HPC system I

- **Hybrid HPC system**
  - **Ultimately intended for (highly) parallel applications**
  - **Use different CPU architectures on one system without data transfer!**
  - **Hardware**
    - Scalar NEC HPC Linux cluster
    - NEC SX-ACE vector system (until end 2018)
    - NEC SX-Aurora Tsubasa vector system (since August 2018)
  - **Details**
    - 4 Login nodes (32 cores Intel Skylake, 768 GB)
    - NEC HPC Linux cluster:
      - 180 batch nodes: Intel-Skylake (32 cores, 192 GB, 2.1 GHz)
        8 nodes thereof with even 384 GB (clbigmem)
      - 18 batch nodes: Intel-Haswell (24 cores, 128 GB, 2.5 GHz)
      - Fast EDR infiniband network between nodes
    - NEC SX-ACE:
      - 256 SX-ACE batch nodes (4 vector cores, 64 GB
        and 256 GB/s total memory bandwidth)
      - Fast node interconnect (IXS crossbar switch)
    - NEC SX-Aurora:
      - 8 vector hosts (VH) with 8 SX vector engines (VE) each
      - VE: 8 SX vector cores, 48 GB, 1.2 TB/s memory bandwidth

NEC HPC system II

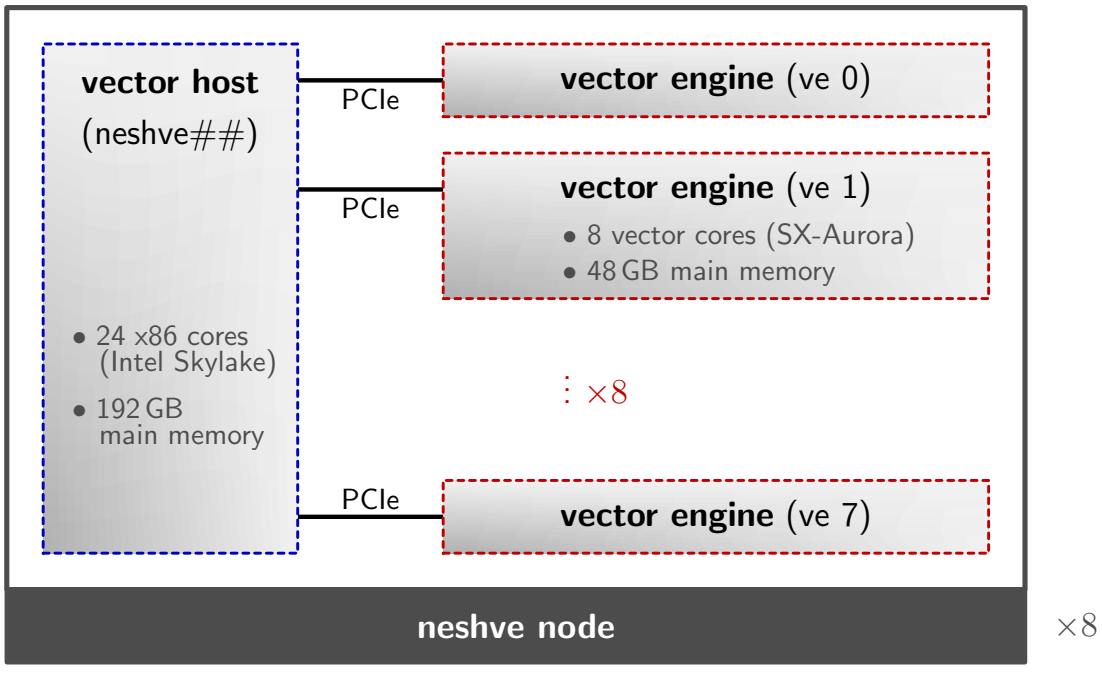
- **System environment**
  - **Operating system(s):**
    - HPC Linux cluster, SX-Aurora (VH): Red Hat Enterprise Linux 7.x
    - SX-ACE: Super-UX

- **File systems**
  - **Global parallel file sytems for $HOME and $WORK**
    - $HOME: 60 TB (NEC GxFS)
    - $WORK: 5 PB = 5000 TB (NEC ScaTeFS, very suitable for parallel I/O)
  - **Access to the tape-library (on all login nodes and via a special batch queue [feque])**



NEC HPC system III

- **SX-Aurora Tsubasa vector system**
  - **Layout:**



  - At least one complete vector engine per user
  - Planned in the future: Hybrid use of VE and VH

II. Practical work on the HPC systems

Login to the HPC systems

- **Login to the front ends with ssh**
  - **Direct access within the campus network**
  - **rzcluster**

```
ssh -X user@rzcluster.rz.uni-kiel.de
```

  - **caucluster**

```
ssh -X user@caucluster.rz.uni-kiel.de
```

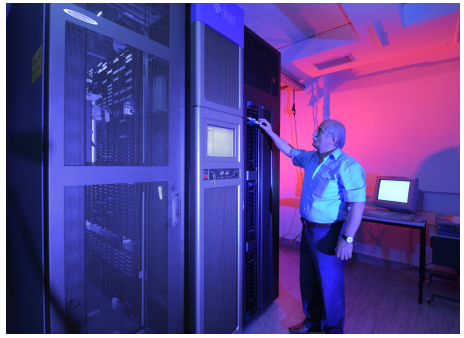
  - **NEC HPC system**
    - Generic login name with automatic redirection on one of the available front ends depending on load

```
ssh -X user@nesh-fe.rz.uni-kiel.de
```

- **Test accounts for this CSC block course:**
  - **20 accounts**
    - *user* $\in \{\text{suabc116}, \ldots, \text{suabc135}\}$
    - Accounts expire on 2. October 2018

File systems

- **Home directory**
  - **Environment variable $HOME**
    - Disk space: 60 / 13 TB [1]
    - For software, programs, code, scripts, etc. No batch production here!
    - Daily backup, see paths (/nfs)/mybackup-*
- **Work directory**
  - **$WORK** (/work_beegfs)
    - Parallel file system: 5 PB ScaTeFS / 350 TB BeeGFS [1]
    - Use this file system for batch production. No backup!
    - User quotas for disk space and inodes, see command workquota
- **Local disk space on compute nodes (scratch)**
  - **$SCRATCH (resp. $TMPDIR)**
    - $\approx$ 500 GB disk space. For fast IO
    - Only available during a batch computation!
- **Tape library (tape robot system)**
  - **$TAPE_CACHE** (/nfs/tape_cache)
    - Automatic file relocation onto magnetic tapes (compressed archives $<$ 1 TB)
    - Only available on login nodes!



[1] NEC HPC system / rzcluster, caucluster

C | A | U
Christian-Albrechts-Universität zu Kiel
Rechenzentrum

- **Software provision via modules**
  - **There exist modules for user software, compilers, libraries and tools**
    - Compilers: GNU, Intel, Portland (rz-/caucluster), SX cross compilers (NEC)
    - Libraries: NetCDF, HDF5, FFTW, MKL, PETSc, GSL, Boost, Lapack/Blas ...
    - Supported MPI environment: Intel-MPI (SX-MPI)

Commands	
module av	Displays all available modules
module load *name*	Loads the module *name* and performs all required settings
module list	Lists all modules which are currently loaded
module unload *name*	Removes the module *name*, i.e., resets all corresponding settings
module purge	Removes all currently loaded modules (module list becomes empty)
module show *name*	Displays the settings which are performed by the module

- **Examples (NEC HPC system):**

```
module load matlab2017a
matlab [-nodisplay]
```

```
module load intel17.0.4 intelmpi17.0.4
mpiicc [-xhost -O2 ...]  main.c -o program.ex
```

Multinode calculations

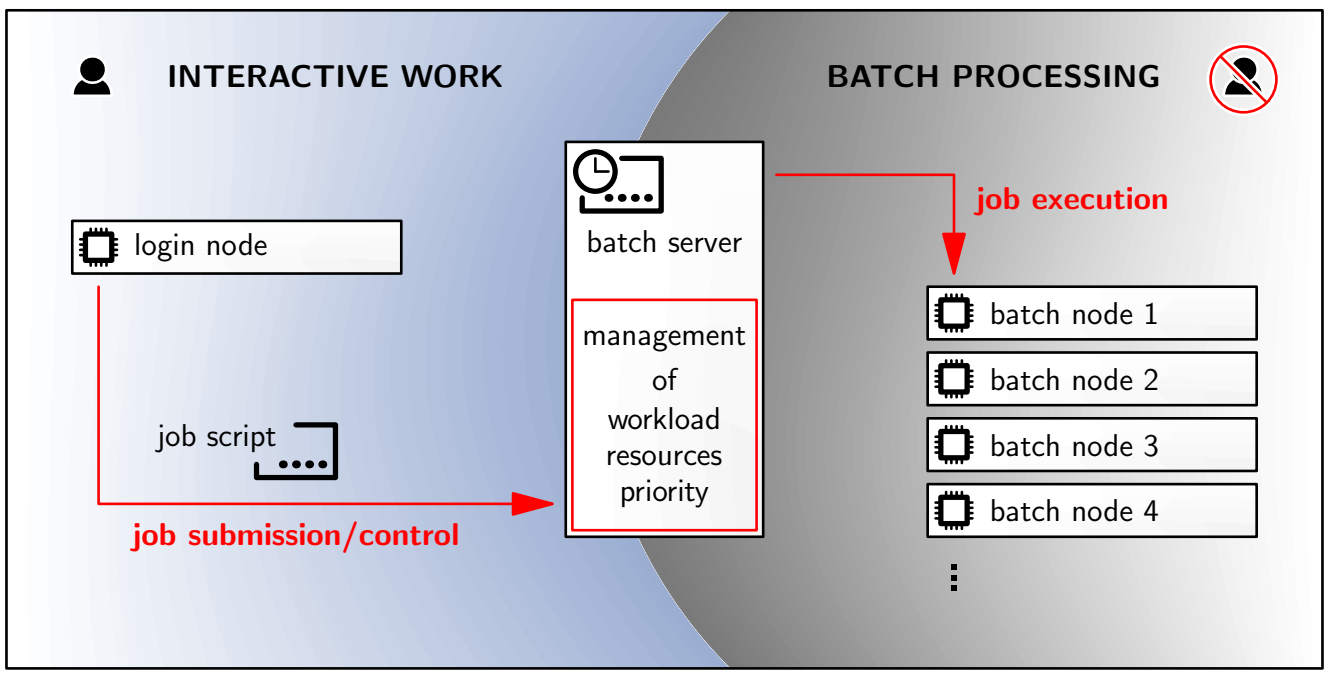
- **Requirement**
  - **Inter-node communication without password**
    - Create once a key pair and add it to the authorized keys
    - Approve keygen requests just with ENTER

```
ssh-keygen -t rsa
cp $HOME/.ssh/id_rsa.pub $HOME/.ssh/authorized_keys
```

C | A | U
Christian-Albrechts-Universität zu Kiel
Rechenzentrum

- **How to perform computations (jobs) on the compute (batch) nodes?**
  - **No job start directly from the command line**
  - **Instead, prepare a so called job (batch) script and submit it to the batch system. Content:**
    - Requested resources, such as #nodes, #cores/node, max. main memory/node, max. elapse time (walltime)
    - Program call

- **Principle of batch processing**



Batch systems and commands

- **Different batch systems:**
  - **rzcluster: SLURM with "first in, first-out" (FIFO) scheduling policy + backfill**
  - **caucluster: SLURM with a fairshare scheduling policy (simple fairshare tree)**
  - **NEC HPC system: NQSII/V with FIFO + backfill**

- **Important batch commands:**

NEC HPC system (NQSII/V)	rz-/caucluster (SLURM)	
qsub *jobscript*	**sbatch** *jobscript*	Submission of a new batch job
qstatall	**squeue**	List all jobs currently in the system
qstat / **qstatcl** / **qstatace**	**squeue -u** *username*	List only the own jobs
qdel *jobid*	**scancel** *jobid*	Delete or terminate a batch job
qstat -f *jobid*	**scontrol show job** *jobid*	Show details of a specific job
qcl / **qace**	**sinfo**	Get information about queues/partitions
qstat -J *jobid*		List on which nodes the job is running

- **Batch queues/partitions:**
  - **"waiting lines" for batch jobs, staggered by walltime and/or resources**

C | A | U
Christian-Albrechts-Universität zu Kiel
Rechenzentrum

- **A simple serial calculation**

```
#!/bin/bash
#PBS -b 1
#PBS -l cpunum_job=1
#PBS -l memsz_job=10gb
#PBS -l elapstim_req=01:00:00
#PBS -N test
#PBS -o test.out
#PBS -j o
#PBS -q cltestque

# Change into qsub directory
cd $PBS_O_WORKDIR

# Program call
module load intel17.0.4
time ./program.ex

# Output of used resources after the job
qstat -f ${PBS_JOBID/0:}
```

C | A | U
Christian-Albrechts-Universität zu Kiel
Rechenzentrum

- **An OpenMP parallel calculation**

```
#!/bin/bash
#PBS -b 1
#PBS -l cpunum_job=8
#PBS -l memsz_job=10gb
#PBS -l elapstim_req=01:00:00
#PBS -N test
#PBS -o test.out
#PBS -j o
#PBS -q cltestque

# Change into qsub directory
cd $PBS_O_WORKDIR

# Program call
module load intel17.0.4
export OMP_NUM_THREADS=8
time ./program.ex

# Output of used resources after the job
qstat -f ${PBS_JOBID/0:}
```

Batch scripts III

- **An MPI parallel calculation**

```
#!/bin/bash
#PBS -T intmpi
#PBS -b 1
#PBS -l cpunum_job=32
#PBS -l memsz_job=10gb
#PBS -l elapstim_req=01:00:00
#PBS -N test
#PBS -o test.out
#PBS -j o
#PBS -q cltestque

# Change into qsub directory
cd $PBS_O_WORKDIR

# Program call
module load intel17.0.4 intelmpi17.0.4
time mpirun $NQSII_MPIOPTS -np 32 ./program.ex

# Output of used resources after the job
qstat -f ${PBS_JOBID/0:}
```

Batch scripts IV

- **A multi-node MPI parallel calculation**

```
#!/bin/bash
#PBS -T intmpi
#PBS -b 2
#PBS -l cpunum_job=32
#PBS -l memsz_job=10gb
#PBS -l elapstim_req=01:00:00
#PBS -N test
#PBS -o test.out
#PBS -j o
#PBS -q cltestque

# Change into qsub directory
cd $PBS_O_WORKDIR

# Program call
module load intel17.0.4 intelmpi17.0.4
time mpirun $NQSII_MPIOPTS -np 64 ./program.ex

# Output of used resources after the job
qstat -f ${PBS_JOBID/0:}
```

Batch scripts V

- **A multi-node hybrid OpenMP+MPI parallel calculation**

```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --tasks-per-node=10
#SBATCH --cpus-per-task=4
#SBATCH --mem=10000
#SBATCH --time=01:00:00
#SBATCH --job-name=test
#SBATCH --output=test.out
#SBATCH --error=test.err
#SBATCH --partition=fairq

# Program call
module load intel17.0.0 intelmpi17.0.0
export OMP_NUM_THREADS=4
time mpirun -np 20 ./program.ex
```

- **Appropriate use of the batch system**
  - **Some remarks**
    - Do not request more nodes and cores that are required by the computation
    - Adapt the walltime and memory resources to the need of the program
    - A more accurate specification can lead to smaller waiting times!
      But do not plan too restrictive
    - Try to save intermediate results! Particularly during longer calculations
    - Check if the program has a restart option
    - The stdout of a batch job should be kept small, otherwise
      redirect it to a file on the local disk or $WORK

- **Summary**
  - **Very elegant and user-friendly way to distribute workload**
  - **Advantages**
    - Fair, controlled & structured distribution of compute resources
    - Presence of requested compute resources are ensured
    - Every user can submit and thus also execute multiple jobs in parallel
    - Larger throughput
  - **Disadvantages**
    - At first glance, unusual work flow
    - Sometimes jobs do not start immediately

III. Account application, documentation and HPC support

Account application

- **HPC application form**
  - **Form no. 3: "Request for use of a High-Performance Computer" [1]**
    - Return completed form to RZ user administration
    - After registration, login details will be sent to you by letter
    - User account comprises subscription to HPC mailing lists
      hpc_user@uni-kiel.de
      nesh_user@uni-kiel.de
      rz_user@uni-kiel.de

- **Remote access**
  - **Sign up for dial-up services (form no. 1)**
  - **How to connect to the HPC systems?:**
    - First, establish a VPN connection to the campus network [2] (use the same user account!)
    - Second, connect, as usual, to the HPC system

[1] https://www.rz.uni-kiel.de/en/about-us/terms-and-policies/forms/sign-up
[2] https://www.rz.uni-kiel.de/en/hints-howtos/vpn

Documentation and HPC support

- **Documentation**
  - **Web pages of the University Computing Centre [1]**
    - System specific informations, batch script examples etc.
    - Slides/material of HPC (introductory) courses and workshops

- **HPC support team**
  - **User consulting**
    - Dr. Karsten Balzer
    - Dr. Simone Knief
    - Dr. Michael Kisiela
  - **System administration**
    - Dr. Cebel Kücükkaraca
    - Dr. Holger Naundorf
    - Alfred Wagner

+ HLRN
North-German Supercomputing Alliance

- **E-mail support (ticket system)**
  - **hpcsupport@rz.uni-kiel.de**

[1] https://www.rz.uni-kiel.de/de/angebote/hiperf

INTENTIONALLY LEFT BLANK!