Kiel University
Christian-Albrechts-Universität zu Kiel

HPC introduction course

University Computing Centre (RZ)
16 October 2019

Support and contact

HPC support team

- User consulting
  - Dr. Karsten Balzer (physics)
  - Dr. Simone Knief (chemistry)
  - Dr. Michael Kisiela (bioinformatics)
- System administration
  - Dr. Cebel Kücükkaraca
  - Dr. Holger Naundorf

Email support (RZ ticket system)

- hpcsupport@rz.uni-kiel.de

Outline

Part I

- High performance computing (HPC)
- HPC systems @ RZ CAU
- HPC systems @ HLRN

Part II

- Short tour through server rooms

Part III

- Working on the local HPC systems

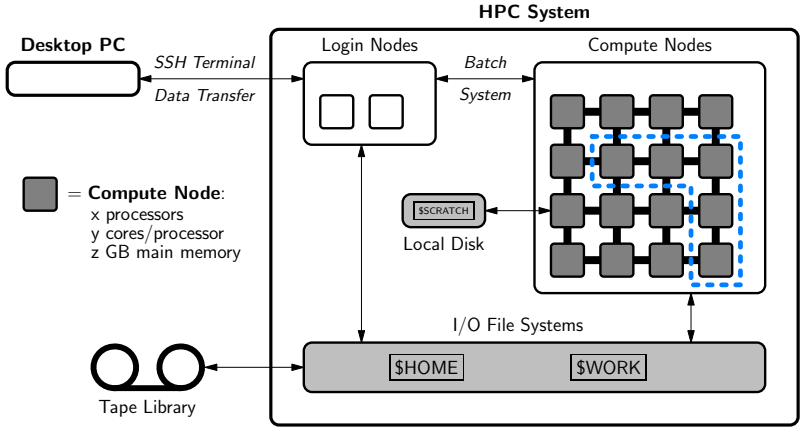
High performance computing (HPC)

What ist high performance computing?

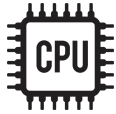
- **High performance computing**
  - = computer aided computation
  - HPC is not a very well defined term
  - But mainly synonymous with "parallel computing"
  - Due to the fact that typical HPC hardware allows for parallel computations
  - Even desktop PCs are nowadays multi-core platforms
- Typical ingredients for a **HPC machine**
  - Many compute nodes with multi-core processors
  - Fast communication network between nodes
  - Powerful I/O file systems

Parallel HPC systems

- **Typical machine layout**:



Scalar and vector processors I

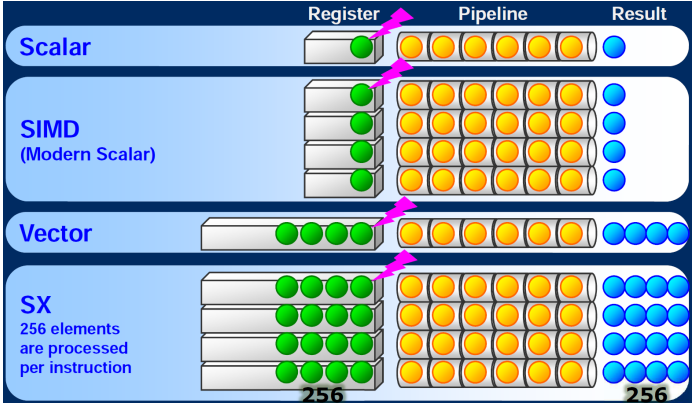


- **Scalar CPUs**:
  - Is the standard instruction set architecture (x86 family)
  - Instructions operate on single data items
  - Nowadays even include vector extensions (e.g., SIMD, AVX)
- **Vector CPUs**:
  - Different characteristics: graphics card (GPUs), special vector processors (VPUs), manycore processors (e.g., Intel Xeon Phi)
  - VPUs:
    - On-chip parallel processing
    - A vector processor can work on an entire vector in one instruction
    - It operates on one-dimensional arrays instead on single data items
    - Suitable for vectorizable code, e.g., matrix-vector operations

Scalar and vector processors II

- **Scalar / modern scalar** versus **vector / modern vector**

SX-ACE:



HPC services of the CAU Computing Centre

- Provisioning of compute resources of different performance classes:
  - **Local compute resources @ RZ**
    - cau- and rzcluster (scalar [x86-based] Linux clusters)
    - Hybrid NEC HPC system (scalar and vector)
  - **Regional (non-local) compute resources @ HLRN**
    - North-German Supercomputing Alliance (HLRN)
      = Norddeutscher Verbund zur Förderung des Hoch- und Höchstleistungsrechnens, Hochleistungsrechner Nord
    - Component of the national HPC infrastructure

HPC services of the CAU Computing Centre

- Provisioning of compute resources of different performance classes:
  - **Local compute resources @ RZ**
    - cau- and rzcluster (scalar [x86-based] Linux clusters)
    - Hybrid NEC HPC system (scalar and vector)
  - **Regional (non-local) compute resources @ HLRN**
    - North-German Supercomputing Alliance (HLRN)
      = Norddeutscher Verbund zur Förderung des Hoch- und Höchstleistungsrechnens, Hochleistungsrechner Nord
    - Component of the national HPC infrastructure
- All these resources are accessible to all CAU and GEOMAR members

HPC services (cont.)

- **HPC user consulting**
  - We provide user support for the aforementioned HPC systems
  - In particular:
    - Choice of appropriate computer architecture (x86/vector, RZ/HLRN)
    - Assistance in software installation and porting
    - Assistance in program and code optimization
    - Support on parallelization and vectorization issues
    - Support on HLRN project proposals

Kiel University
Christian-Albrechts-Universität zu Kiel
University Computing Centre (RZ)

HPC systems @ RZ CAU

Overview

- **rzcluster**
  - x86-based Linux cluster
- **caucluster**
  - x86-based Linux cluster

- **Hybrid NEC HPC system**
  - NEC x86-based Linux cluster
  - NEC SX-Aurora TSUBASA vector systems

cau- and rzcluster

- Currently, two separate Linux clusters which however share the user file systems
- Hardware:
  - **rzcluster**
    - 1 login node: 12 cores, 48 GB main memory
    - ≈140 compute nodes: ≈1900 cores in total (Intel and AMD CPUs); very heterogeneous configuration (8 - 32 cores/node, 32 GB/node - 1 TB/node)
    - Only 40 compute nodes available for the general user (8-12 cores/node, 32-48 GB/node)
    - Operating system: Scientific Linux 6 (part. CentOS 7); batch system: SLURM
  - **caucluster**
    - 1 login node: 8 cores (64 GB main memory)
    - 7 compute nodes: 280 cores in total (Intel Haswell CPUs, 40 cores/node, 256 GB/node)
    - Operating system: CentOS 7; batch system: SLURM + fairshare scheduling ($\neq$ first-in first out)

NEC HPC system

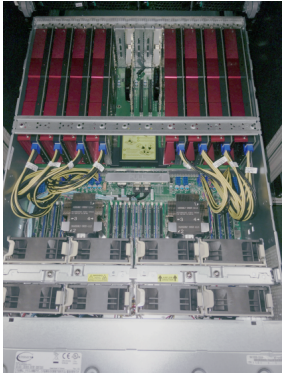
- **Hybrid HPC system**, consisting of
  - Scalar x86 Linux cluster
  - SX-Aurora TSUBASA vector system
- **Uniform system environment**:
  - 4 login nodes: 32 cores (Intel Skylake CPUs, 768 GB main memory)
  - Shared, global I/O file systems:
    - Home file system: 64 TB NEC GxFS, 2 partitions a 32 TB
    - Work file system: 5 PB = 5000 TB NEC ScaTeFS, 2 partitions a 2/3 PB
  - Operating systems: Red Hat Enterprise Linux 7 (Linux cluster + SX-Aurora)
  - Single batch system to organize compute jobs: NQSV
  - Temporarily a separate login node for SX-Aurora cross-compiling

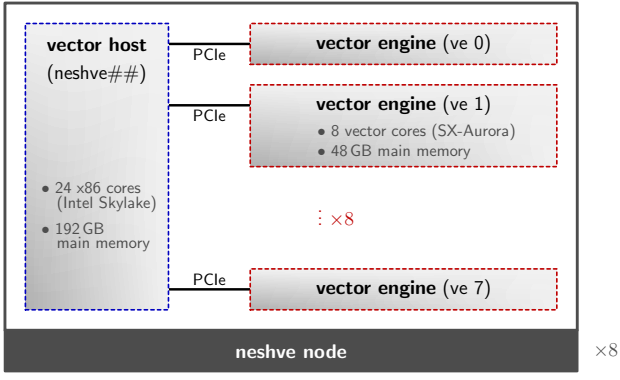
NEC HPC system (cont.)

- Compute hardware:
  - **Scalar x86 Linux cluster**
    - 180 compute nodes: 32 cores each (Intel Skylake CPUs, 192 GB resp. 384 GB [8 "bigmem" nodes])
    - 18 compute nodes: 24 cores each (Intel Haswell CPUs, 128 GB)
    - Fast EDR infiniband network between nodes
  - **SX-Aurora TSUBASA vector system**
    - 8 compute nodes [called "vector hosts" (VH)] with eight SX vector engines (VE) each (A300-8, type 10B), which are connected via PCIe to the host
    - VE: 8 SX vector cores with 48 GB, 1.4 GHz, 64 logical vector registers per core (length: 256 x 64 Bits)
    - VE Memory bandwidth: 1.2 TB/s
    - Infiniband network between vector hosts

NEC SX-Aurora TSUBASA

- Hardware layout and characteristics

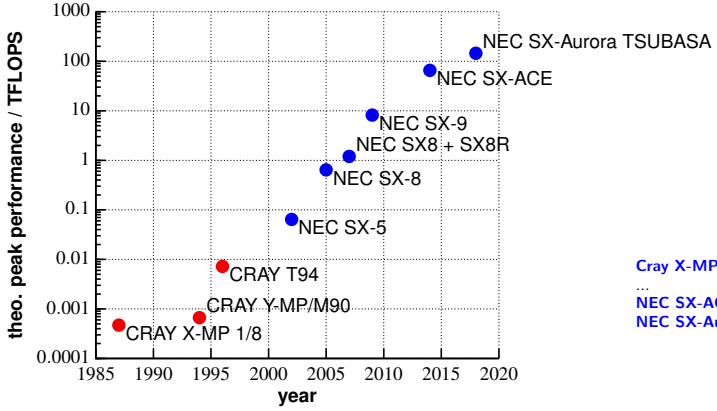




- At least one complete vector engine per user
- Planned in near future: Hybrid use of VE and VH (e.g.. offloading)

Vector system performance

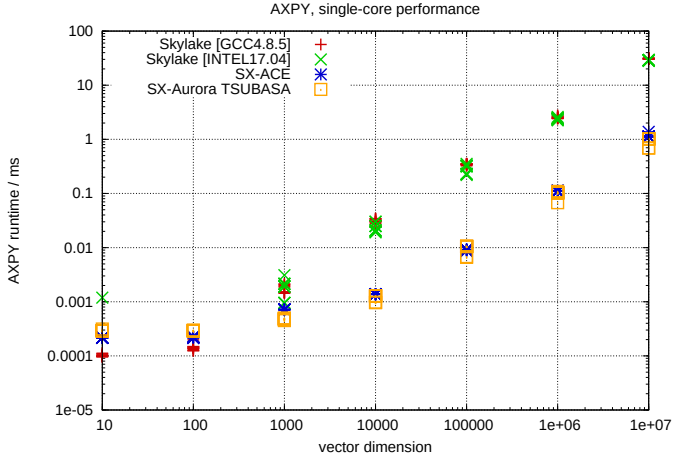
- History of systems installed at RZ CAU Kiel:



Cray X-MP 1/8 (1987): 470 MFLOPS
...
NEC SX-ACE: 65.5 TFLOPS
NEC SX-Aurora TSUBASA: 145.2 TFLOPS

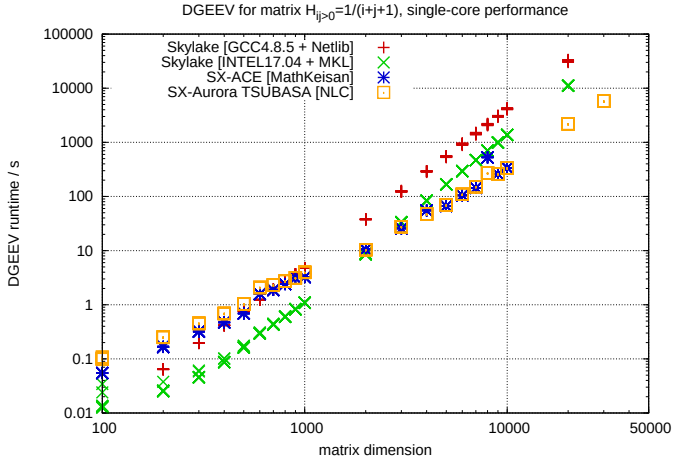
NEC HPC system

- **Linux cluster** versus **SX-Aurora TSUBASA**
  - A) Simple axpy operation: $\mathbf{z} = a \cdot \mathbf{x} + \mathbf{y}$ with vectors **x**, **y**, **z**



NEC HPC system

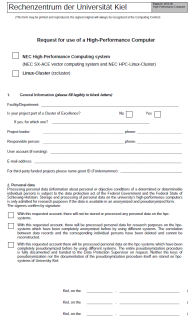
- **Linux cluster** versus **SX-Aurora TSUBASA**
  - B) Simple matrix diagonalization with Lapack routine DGEEV



RZ CAU: Account application

Kiel University
Christian-Albrechts-Universität zu Kiel
University Computing Centre (RZ)

- **How to get access?**
  - Access may be requested at any time
  - Use form no. 3:
    "Request for use of a High-Performance Computer"
  - Link: https://www.rz.uni-kiel.de/en/about-us/terms-and-policies/forms/sign-up
  - Return completed form to RZ user administration
  - After registration, login details will be sent in written form to the responsible person (via in-house CAU mail)



Rechenzentrum der Universität Kiel

Request for use of a High-Performance Computer

NEC High-Performance Computing system
(NEC SX-ACE vector computing system and NEC HPC-Linux-Cluster)

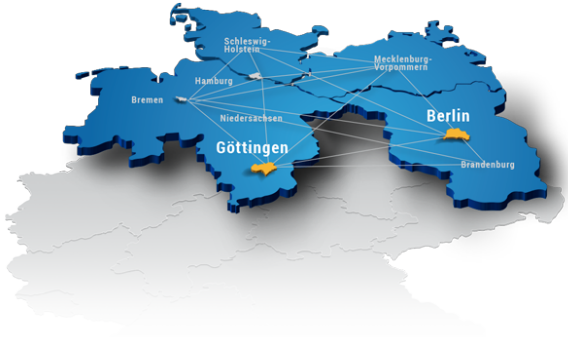
Linux-Cluster (exclusive)

- User account comprises subscription to HPC mailing lists:
  - nesh_user@lists.uni-kiel.de resp. rzcluster_user@lists.uni-kiel.de:
    HPC system specific information: downtimes, problems, new software, ...
  - hpc_user@uni-kiel.de:
    General information: courses, events, new installations, ...

Kiel University
Christian-Albrechts-Universität zu Kiel
University Computing Centre (RZ)

HPC systems @ HLRN

HLRN: North-German Supercomputing Alliance



- https://www.hlrn.de

HLRN: North-German Supercomputing Alliance

- German: Norddeutsche Verbund für Hoch- und Höchstleistungsrechnen (**H**och**l**eistungs**r**echner **N**ord, HLRN)
- Component of the national HPC infrastructure, formed by seven North-German states: Berlin, Brandenburg, Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen and Schleswig-Holstein

- In operation since 2001
- Aims:
  - Joint procurement and operation of a Tier-2 HPC centre
  - Establishment and development of a HPC competence network

HLRN: Site locations

- Zuse Institute Berlin (ZIB)
- University of Göttingen, since 09/2018 (former site: University Hannover)

- Current expansion stage:
  Göttingen: **HLRN IV phase 1 (Emmy)**
  ZIB: Installation of **HLRN IV (Lise)** in progress

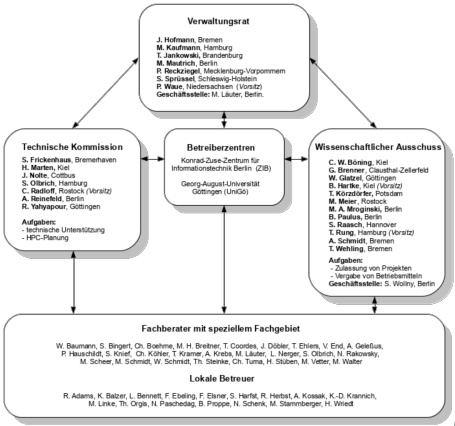
- Full expansion stage of HLRN IV: 16 PFLOPS, about 244000 cores (Intel-CPUs) over both sites

HLRN: Currently available resources

- **HLRN IV phase 1 (Göttingen)**
  - **432 'medium' compute nodes with**:
    - 2 x Intel Skylake Gold 6148 CPUs (40 cores per nodes), 192 GB memory
    - 1 x 480 GB SSD hard disk
    - 1 x Intel Omni-Path host fabric adapter
  - **16 'large' compute nodes with**:
    - 2 x Intel Skylake Gold 6148 CPUs (40 cores per nodes), 768 GB memory
    - 1 x 480 GB SSD hard disk
    - 1 x Intel Omni-Path host fabric adapter
  - **1 GPU node with**:
    - 2 x Intel Skylake Gold 6148 CPUs (40 cores per nodes), 192 GB memory
    - 1 x 480 GB SSD hard disk
    - 1 x Intel Omni-Path host fabric adapter
    - 4 x NVIDIA Tesla V100, 32 GB

HLRN: Organization chart and access

Kiel University
Christian-Albrechts-Universität zu Kiel
University Computing Centre (RZ)



(09/2018)

- Utilization possible for all university employees in Schleswig-Holstein, incl. universities of applied sciences
- Access in two steps:
  i. Application for a test account (called “Schnupperkennung”)
  ii. Application for a research project (“Großprojekt”)
- More infos:
  https://www.hlrn.de/home/view/Service/ApplicationHowTo

HLRN: Account application

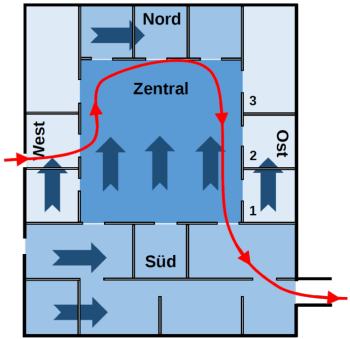
- **Schnupperkennung**
  - This account should be used to prepare a research project ("Großprojekt")
  - Apply at any time
  - Expires after 3 quarters
  - Limited computing time: 2500 NPL per quarter
    (increase to a maximum of 10000 NPL possible)

  - NPL = "Norddeutsche Parallelrechner Leistungseinheit"
    Depending on machine: 2 - 6 NPL per hour and node

HLRN: Account application

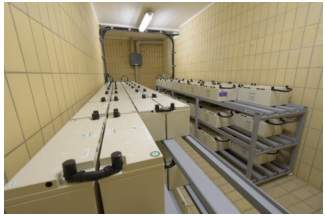
- **Großprojekt**
  - Deadlines for proposal submission:
    28.01., **28.04.**, **28.07.**, **28.10.**
  - Project duration: Total NPL are granted for 1 year and are distributed on a quaterly basis
  - Proposals will be assessed by the scientific council of HLRN
  - Prerequisite: a valid HLRN user account
  - Minimum demands for a project proposal:
    - Project abstract
    - Project description or status report (for project extension)
    - Justification of requested NPL
    - Other resource demands (such as main memory, disk space)
    - More infos: https://www.hlrn.de/home/view/Service/ProjektBeschreibungHinweise
  - Contact local HPC consultants (us!) prior to proposal submission

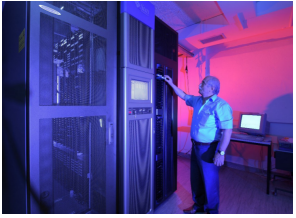
Short tour through server rooms



HPC systems ...

Not shown ...





Working on the local HPC systems

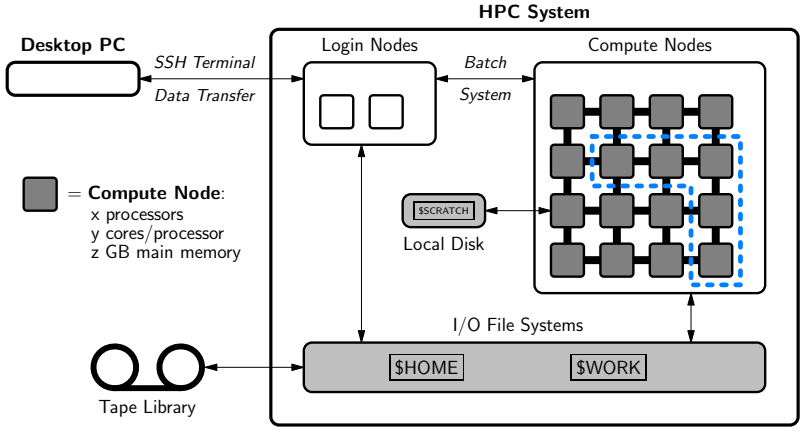
Working on the local HPC systems

Kiel University
Christian-Albrechts-Universität zu Kiel
University Computing Centre (RZ)

- Login and data transfer
- File systems, data protection and integrity
- Software environment
- Interactive work and batch processing

Parallel HPC systems

- **Typical machine layout**:



Login

- Access to the login (front end) nodes
  - Connect via SSH from within the networks of CAU, GEOMAR and UKSH
  - From outside: Establish first a VPN connection to above networks (with RZ account)

- **How to establish an ssh connection?**
  - One needs an SSH client
  - Linux and Mac:
    - Available through the standard distribution
  - Windows:
    - Putty (simple, no x11 window forwarding by default)

      https://www.putty.org
    - MobaXterm (preferred, versatile ssh client incl. x11)

      https://mobaxterm.mobatek.net
    - SmarTTY (ssh client incl. scp capability)

      http://smartty.sysprogs.com
    - X-Win 32 (alternative solution, campus license)

      https://www.rz.uni-kiel.de/de/angebote/software/x-win32

SSH login commands

- NEC HPC system (**Linux cluster** and **SX-Aurora**)

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

- NEC HPC system (**temporary SX-Aurora cross-compile node**)

```
ssh -X username@nesh-fe4.rz.uni-kiel.de
```

- **caucluster** and **rzcluster**

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

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

- -X option enables X11 forwarding

- Command for password change: `passwd`

Data transfer

- Data transfer to/from the HPC system:
  - Linux and Mac:
    - Use secure copy command

      ```
      scp file username@nesh-fe.rz.uni-kiel.de:/path-to-destination-dir
      ```

  - Windows:
    - Install WinSCP, which is a gui-based scp client

      https://winscp.net
    - SmarTTY

      http://smartty.sysprogs.com

File systems

- There are **four different file systems** available:
  - HOME file system (with the user's **HOME directory**)
  - WORK file system (with the user's **WORK directory**)
  - Local file system on the compute node (**local disk space**)
  - **Tape library file system** (which is part of the magnetic tape robot system)

HOME directory

- **The directory after ssh login**
- Accessible via the environment variable $HOME
- Total available disk space (for all users)
  - NEC HPC system: 60 TB
  - cau- and rzcluster: 13 TB
- Mounted on all compute nodes
- Regular backup (daily)
- Currently no user quotas

- For important data which need a backup, e.g.,
  software, programs, code, scripts, small amount of results
- NOT for the execution of production runs (batch jobs)!
- Slow access times

WORK directory

- **Global parallel file system**
- Accessible via the environment variable $WORK
  (older accounts on cau- and rzcluster: use path /work_beegfs/*username*)
- Total available disk space (for all users)
  - NEC HPC system: 5 PB (ScaTeFS)
  - cau- and rzcluster: 350 TB (BeeGFS)
- No backup!
- User quotas for disk space and inodes (#directories + #files)
- Must be used for the execution of production runs (batch jobs)!

User quotas on WORK directory

- **NEC HPC system**:
  - Disk space:
    - Soft limit: 1.8 TB (CAU), 4.5 TB (GEOMAR)
    - Hard limit: 2.0 TB (CAU), 5.0 TB (GEOMAR)
  - Inodes:
    - Soft limit: 225000
    - Hard limit: 250000
- **cau- and rzcluster**:
  - Disk space: 1 TB
  - Inodes: 250000

- Display quota usage: `workquota`

Local disk space

- The disk space directly on the compute nodes
- Only temporarily available within a batch job
- Access via environment variables:
  - cau- and rzcluster: $TMPDIR
  - NEC HPC-system: $SCRATCH after defining

```
export $SCRATCH="/scratch/"‘echo $PBS_JOBID | cut -f2 -d\:‘
```

- Advantage:
  - Fast access times
  - Faster I/O for read- and write-intensive computations

Tape library

- "Disk space" with **automatic file relocation onto magnetic tape**
- Accessibility:
  - NEC HPC system: Environment variable $TAPE_CACHE
  - cau- and rzcluster: Path /nfs/tape_cache/*username*
- Tape cache directory is only mounted on the login nodes
- NEC HPC system: Also available in the special batch queue "feque"

- Use the tape library to store non-active data
- Transfer and store only archived data, e.g., tar-files
  (max. 1 TB, recommended: 3-50 GB)
- Do not store many small files on the tape library
- Slow access times
- Never work interactively within the tape cache directory!
- Deleted data cannot be recovered (single copy, no archive system)

Data protection and integrity

- **HOME directory**:
  - Regular backup (daily)
  - Backup history comprises the last 8 weeks
  - For last 2 week, files can be recovered on a daily basis, prior to that on a weekly basis
  - Each user can access its own backup directory:
    - cau and rzcluster: /mybackup-0
    - NEC HPC system: /nfs/mybackup-0 or /nfs/mybackup-1
- **WORK directory**:
  - No backup!
- **Tape library**:
  - No backup!
- **Some remarks**:
  - User data is only accessible from an active account
  - The computing center does currently not provide a long-term data archiving service
  - Check your data stock from time to time

Software I

- There is a whole bunch of user software available:
  - **(Standard) user software**
    - Python, Perl, R, Matlab, Gaussian, Turbomole, Quantum Espresso, Gnuplot, Xmgrace, ...
  - **Compilers**
    - GNU compilers
    - Intel compilers
    - Portland compilers (cau- and rzcluster only)
    - Cross compilers for using the NEC vector systems (SX-Aurora)
  - **MPI environment**
    - Intel MPI (all Linux clusters)
    - Special (SX) MPI (NEC vector systems)
  - **Libraries**
    - NetCDF, HDF5, FFTW, MKL, PETSc, GSL, Boost, ...

Software II

- **Available editors**:
  - nedit
  - GNU Emacs
  - GNU nano
  - vi
  - mc
- A remark regarding files transfered from a Microsoft Windows environment:
  - Windows-based editors generally put an extra "carriage return" character (also referred to as control-M/^M) at the end of each line of text.
  - This will cause problems for most Linux-based applications. Appearance of errors like

    /bin/bash^M: bad interpreter: no such file or directory
  - For correction, execute the built-in utility dos2unix on the ASCII file:

    `dos2unix` *filename*

Software environment

- User software is provided in **two ways**
  i. There is *some* software which comes with the distribution of the operating system (and is thus available in the standard path)
  ii. Specific software and specific versions of software are provided via so called modulefiles

- **Modulefiles** allow for a simplified software access:
  - After loading a specific module, software, programs, libraries etc. are usable without specifying the full path ($PATH)
  - Moreover, also other required search paths and environment variables are automatically set ($LD_LIBRARY_PATH, $MANPATH, ...)
  - There exist simple commands to load and unload modulefiles

Module commands I

Kiel University
Christian-Albrechts-Universität zu Kiel
University Computing Centre (RZ)

Command	Explanation
module avail	Shows 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

Module commands II

- **Matlab**
  - For CAU users:

```
module load matlab2017a
matlab
matlab -nodisplay
```

  - For GEOMAR users (special [toolbox] licenses):

```
module load matlab2017a_geomar
```

- **Intel compilers**

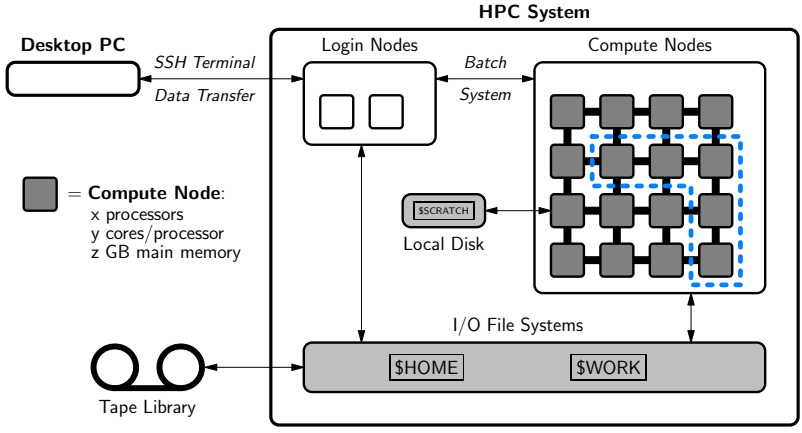
```
module load intel17.0.4
ifort ...
icc ...
icpc ...
```

```
module load intel17.0.4 intelmpi17.0.4
mpiifort ...
mpiicc ...
mpiicpc ...
```

```
------------------- /sfs/fs5/sw/modules/sxcrosscompiling -------------------
crosscompiler           MathKeisan/4.0.3(default) sxf03/rev061(default)
crosscompiler.nec       mpisx/10.2.4(default)     sxf90/rev534(default)
crosskit/r211(default)  netcdf_4.1.1_sx
fvlew                   sxc++/rev112(default)

----------------------- /sfs/fs5/sw/modules/x86compilers ----------------------
gcc5.3.0      intel16.0.4    intelmpi16.0.4 java1.8.0      llvm6.0.0
gcc7.2.0      intel17.0.4    intelmpi17.0.4 llvm4.0.1

------------------------ /sfs/fs5/sw/modules/x86libraries ---------------------
boost1.65.0             hdf5-1.8.19intel        pcre2-10.21
curl7.55.1              hdf5parallel-1.8.19     pcre8.41
eigen3.3.4              hdf5parallelintel-1.8.19 pnetcdf1.8.1
fftw3.3.6               jags4.3.0               pnetcdf1.8.1intel
fftw3.3.6intel          lapack3.8.0             proj4.9.3
gdal2.2.3               ncurses6.0              readline7.0
geos3.6.2               netcdf4.4.1             szip2.1.1
glib2.52.3              netcdf4.4.1intel        udunits2.2.25
gsl2.4                  netcdf4.4.1paraintel    xz5.2.3
hdf5-1.8.19             openssl1.0.2            zlib1.2.11

------------------------ /sfs/fs5/sw/modules/x86software ----------------------
abaqus2018              g16a03                  opennwolcas4.2017serial
adf2017.110             glpk4.61                perl5.26.0
adf2017.110intel        gnts.4.2                perl5.26.0threads
allpathslg52488         gnuplot5.0.7            petsc3.6.1intel
amos3.1.0               grace5.1.9              petsc3.6.1intel-debug
beat1.0                 grib_api1.23.1          petsc3.7.6
blender2.79             hatll0.2                petsc3.7.6debug
blender2.79a            interproscan5.30-69.0   petsc3.7.6intel
blender2.79b            jasper2.0.14            petsc3.7.6intel-debug
bowtie2-2.3.3           lammps17                plumed2.4.0
bwa0.7.17               matlab2011b_geomar      plumed2.4.0intel
cactus7.2018            matlab2015a             pyferret7.4.3
cdo1.9.0                matlab2015a_geomar      python2.7.13
consol5.3a-tetra        matlab2017a             python3.6.2
consol5.4-tetra         matlab2017a_geomar      R3.4.1
cp2k5.1                 megahit1.1.3            R3.5.1
cplex                   metawrap1.0.2           R3.5.2
cufflinks2.2.1          molpro2015              salmon0.12.0
espresso5.4.0           mothur1.39.5            samtools1.5
espresso6.2.1           mothur1.40.0            spades3.12.0
espresso6.2plumed       mumer3.23               specfem3d3.0
fastqualr12.2017        nclplot3.0              specfem3d3.0npi
ferret6.72              ncl6.4.0                star2.6.0a
ferret6.82              nco4.6.8                transrate1.0.3
ferret7.2               ncview1.2.7             turbomole7.2
ferret7.4test           octave4.2.1             turbomole7.2mpi
fonosto-qssp2017        octopus7.1              turbomole7.2smp
g09d01                  opennwolcas4.2017       yambo4.2.0

------------------------ /sfs/fs5/sw/modules/x86tools -------------------------
bzip2-1.0.6  cmake3.9.1  imake1.0.7  miniconda3   sensors3.4.0
cmake3.12.1  git2.14.1   mc4.8.19    parallel     use.own
```

Parallel HPC systems

- **Typical machine layout**:



Interactive work

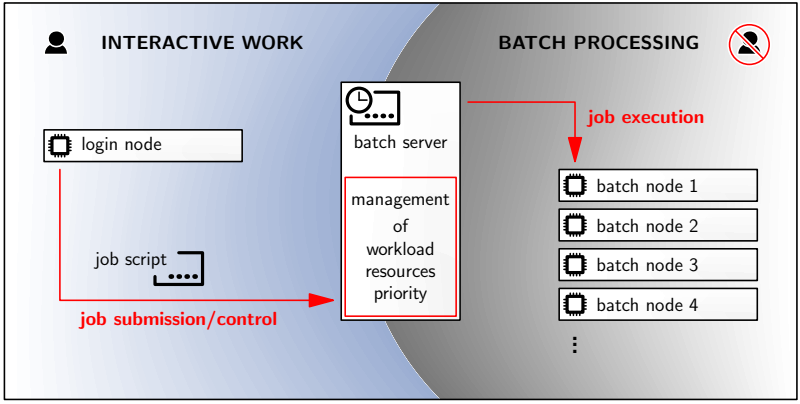
- = **Working on the login nodes**
- Usually covers the following activities:
  - Data transfer (desktop PC ↔ HPC systems ↔ tape library)
  - Software provisioning and installation
  - Program development and compilation
  - Preparation of batch scripts and batch jobs
  - Submission and control of batch jobs
  - Small pre- and postprocessing
- Do not start long and recource demanding calculations on the login node!
- Also be careful with small test calculations
  ⇒ monitor CPU and memory consumption, as well as runtime!

  Simple, useful command: `top`

Batch processing

- = **Running calculations (jobs) on the compute node**
- How to do this?:
  - One does not start the job from the command line
  - Instead:
    1. Prepare a small ASCII file (called batch or job script) which contains all necessary information
    2. Submit this script to the batch system
  - Typical information provided in a batch script:
    - Required compute resources (# cores, main memory, walltime, ...)
    - Software environment (module load ..., if required)
    - Program call

Principle of batch processing



- Batch server takes the job script, searches for free, appropriate compute resources and then executes the actual computation or queues the job.

Advantages of batch processing

- **Very efficient** and **well organized** use of available compute resources
- In particular:
  - Larger throughput
  - Fair distribution of resources
  - Every user can execute multiple jobs in parallel
  - Presence of requested resources is ensured

Batch scripts and batch commands

- Different syntax and commands for different batch systems:
  - NEC HPC system: NQSV
  - cau- and rzcluster: SLURM
- Different forms for different types of calculations:
  - Serial calculation
  - Using multiple cores on a single node (e.g., via OpenMP threads)
  - Using multiple nodes (e.g., via MPI)
  - Hybrid schemes (e.g., MPI+OpenMP)
  - **Remark**: For distributed MPI calculations, batch nodes need to communicate without password! To this end, generate a key pair once:
    1. Execute the ssh-keygen command without setting a passphrase

       ```
       ssh-keygen -t rsa
       ```

    2. Copy the public key into the authorized keys file

       ```
       cp $HOME/.ssh/id_rsa.pub $HOME/.ssh/authorized_keys
       ```

Batch script examples I

- A serial calculation [NEC HPC Linux cluster (NQSV)]

```
#!/bin/bash
#PBS -b 1
#PBS -l cpunum_job=1
#PBS -l elapstim_req=01:00:00
#PBS -l memsz_job=20gb
#PBS -N test
#PBS -o test.out
#PBS -j o
#PBS -q clmedium

# Change into qsub directory
cd $PBS_O_WORKDIR

# Start of the computation
module load intel17.0.4
time ./program.ex

# Output of used resources (computation time, main memory) after the job
qstat -f ${PBS_JOBID/0:}
```

Batch script examples II

- NEC HPC Linux cluster (NQSV): A parallel, multinode MPI calculation

```
#!/bin/bash
#PBS -T intmpi
#PBS -b 4
#PBS -l cpunum_job=32
#PBS -l elapstim_req=10:00:00
#PBS -l memsz_job=256gb
#PBS -N test
#PBS -o test.out
#PBS -j o
#PBS -q clbigmem

# Change into qsub directory
cd $PBS_O_WORKDIR

# Start of the computation
module load intel17.0.4 intelmpi17.0.4
time mpirun $NQSII_MPIOPTS -np 128 ./program.ex

# Output of used resources (computation time, main memory) after the job
qstat -f ${PBS_JOBID/0:}
```

Batch script examples III

- NEC SX-Aurora TSUBASA (NQSV): A single-host MPI calculation

```
#!/bin/bash
#PBS -T necmpi
#PBS -b 1
#PBS --venum_lhost=4
#PBS -l elapstim_req=01:00:00
#PBS -N test
#PBS -o test.out
#PBS -j o
#PBS -q vequeue

# Change into qsub directory
cd $PBS_O_WORKDIR

# Display performance summary (compile with -proginf flag)
export VE_PROGINF=DETAIL

# Start of the computation
mpirun -nn 1 -nnp 32 -ve 0-3 ./program.sx-aurora
```

Batch script examples IV

- rzcluster (SLURM): An OpenMP parallel calculation

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=4
#SBATCH --mem=1000
#SBATCH --time=01:00:00
#SBATCH --job-name=test
#SBATCH --output=test.out
#SBATCH --error=test.err
#SBATCH --partition=small

export OMP_NUM_THREADS=4

module load intel16.0.0
time ./program.ex
```

Batch script examples V

- rzcluster (SLURM): A multinode hybrid MPI+OpenMP calculation

```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --tasks-per-node=3
#SBATCH --cpus-per-task=4
#SBATCH --mem=1000
#SBATCH --time=01:00:00
#SBATCH --job-name=test
#SBATCH --output=test.out
#SBATCH --error=test.err
#SBATCH --partition=small

export OMP_NUM_THREADS=4

module load intel16.0.0 intelmpi16.0.0
time mpirun -np 6 ./program.ex
```

Job submission and control

NEC HPC system (NQSV)	cau-/rzcluster (SLURM)	
qsub *jobscript*	sbatch *jobscript*	Submission of a new batch job
qstatall	squeue	List all jobs currently in the system
qstat	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	sinfo	Get information about queues/partitions
qstat -J *jobid*		Lists on which nodes the job is running

Interactive batch jobs

- **How to request an interactive session on a compute node?**
- cau- and rzcluster (SLURM)

```
srun --pty --nodes=1 --cpus-per-task=1 --time=00:10:00 --partition=small /bin/bash
```

- NEC SX-Aurora TSUBASA (NQSV)

```
qlogin -q veinteractive -l cpunum_jobs=1 -l elapstim_req=1:00:00
```

Appropriate use of the batch system

- Do not request more nodes and cores than are required by the computation
- Adapt the walltime and memory resources to the need of the program
  - Note that a more accurate resource specification can lead to smaller waiting times and increased throughput
  - 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 on $WORK

Documentation

RZ HPC web pages

- https://www.rz.uni-kiel.de/en/our-portfolio/hiperf
  - System-specific documentation
  - Presentation slides

Email support (RZ ticket system)

- hpcsupport@rz.uni-kiel.de

Thank you for your attention