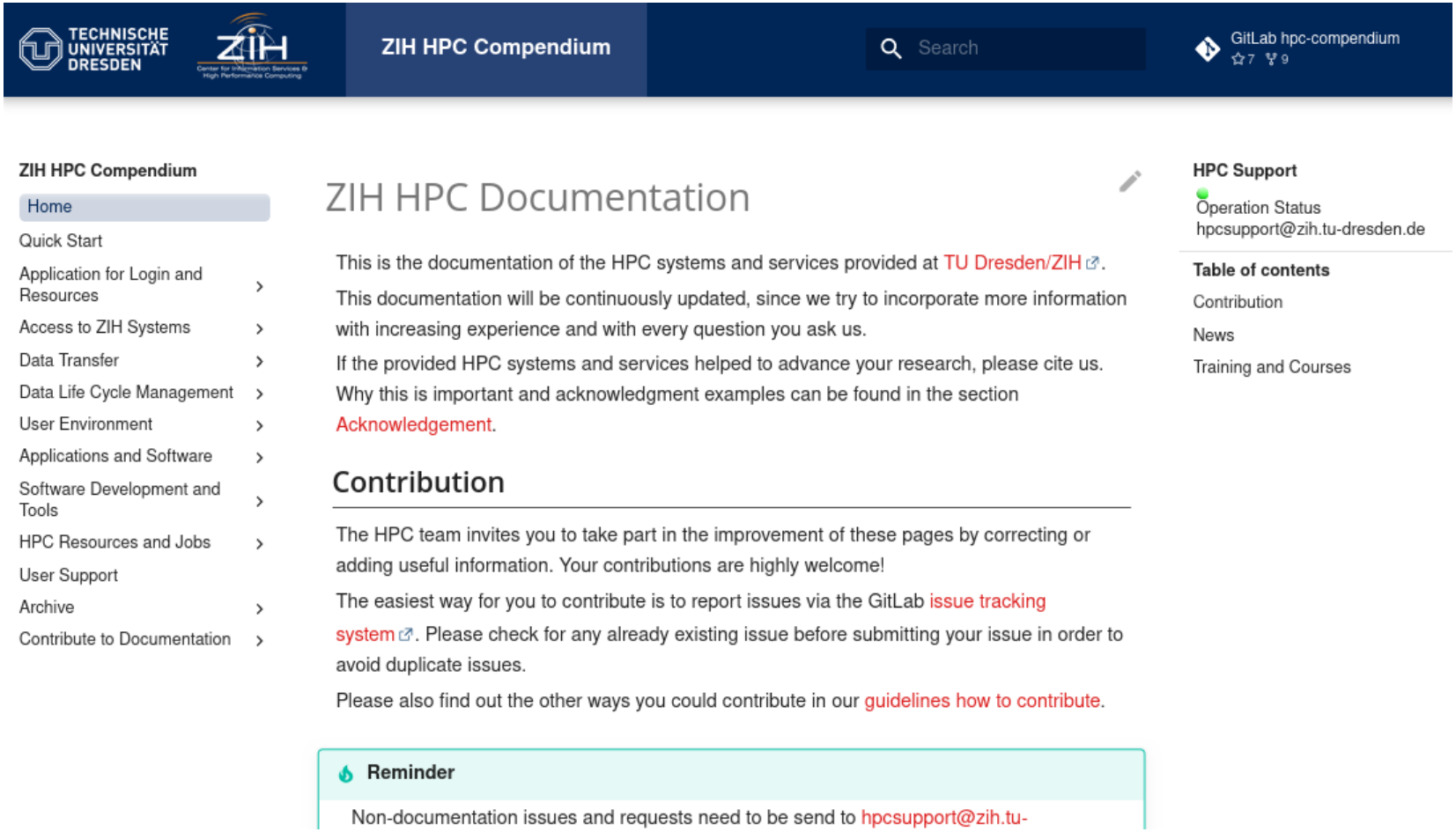
Dr. Ulf Markwardt
hpcsupport@zih.tu-dresden.de

Introduction to HPC at ZIH

Dresden, September 2023

HPC wiki has the answer

Please check our HPC wiki at `https://compendium.hpc.tu-dresden.de`



Agenda

General

- first version 1991, Linus Torvalds
- hardware-independent operating system
- 'Linux' is the name of the kernel as well as of the whole operating system
- since 1993 under GNU public license (GNU/Linux)
- various distributions for all purposes (OpenSuSE, SLES, Ubuntu, Debian, Fedora, RedHat,...) `http://www.distrowatch.com`

Tools for SSH access

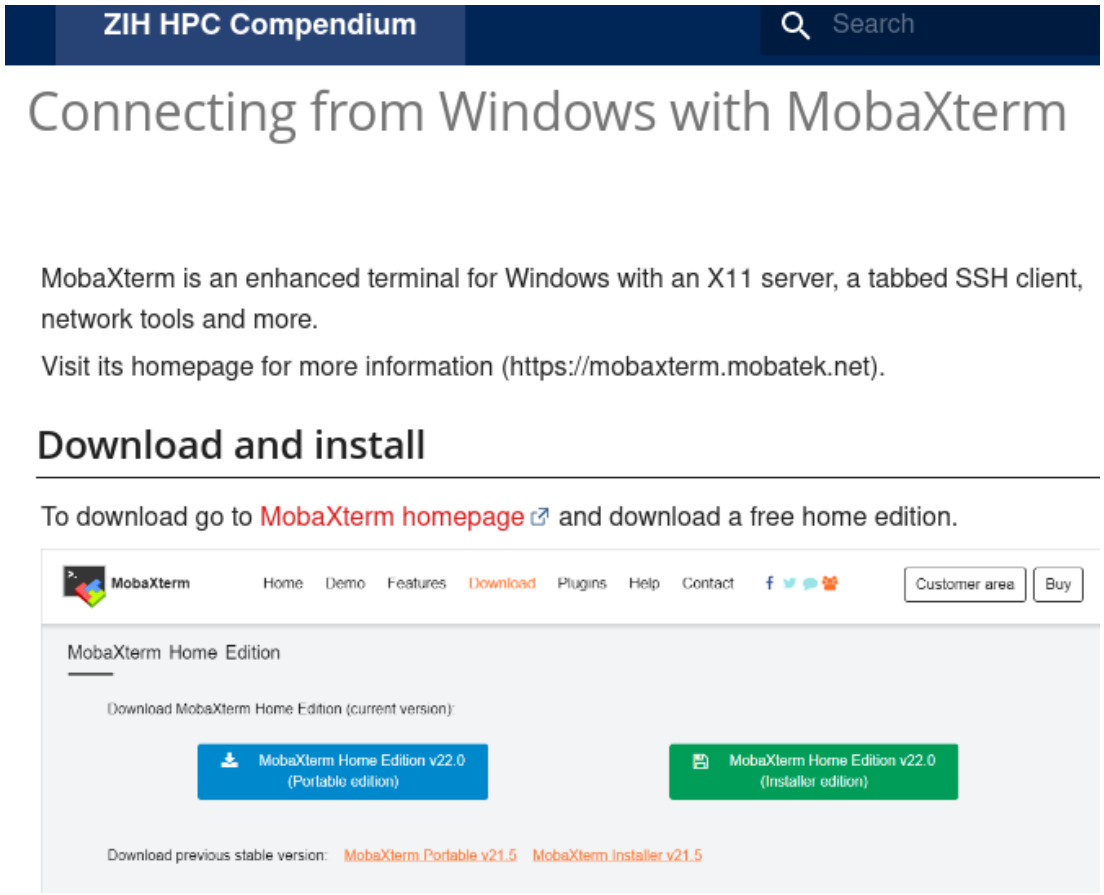
Tools to access HPC systems at ZIH from Windows systems
(see `https://compendium.../access/ssh_login`)

- command line login: PuTTY, Secure Shell
- file transfer: WinSCP, Secure Shell
- GUI transfer (XMing, XMing-Mesa, X-Win32)

- integrated solution: MobaXterm

MobaXterm step-by-step instructions

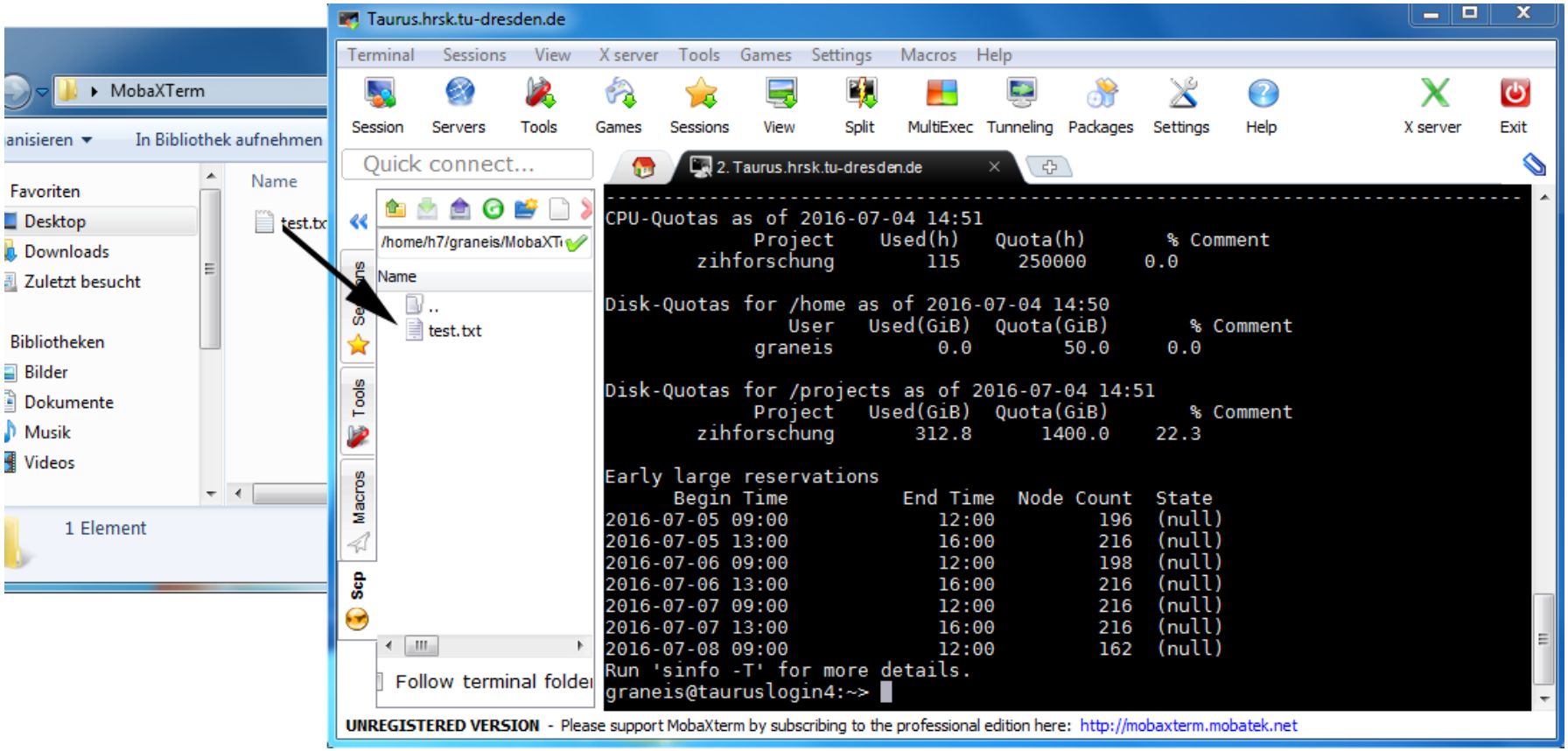
see our Wiki at `https://compendium.../access/ssh_mobaxterm`



or download PDF at `https://compendium.../access/misc/basic_usage_of_MobaXterm.pdf`

MobaXterm

- console to HPC systems (including X11 forwarding)
- transfer files to and from the HPC systems
- browse through the HPC file systems



Nelle's Pipeline

(https://software-carpentry.org)

In tedious field work **1520** jellyfish specimen were collected. Now the workflow in the lab is as follows:

- A scanner checks each sample for 300 different proteins
  Result: a file per specimen, one line per protein.
- For each protein, some software calculates statistics.
- Scientist writes up results for a paper.

Timeline – Publish within a month?

- Protein scanner: 2 weeks hard work in the lab
- Manually (GUI) select 1520 files in a file open dialog for analysis is boring and thus error-prone. (30s per "open" = 12h + processing time)

An adequate automation process for batch analysis would help.

Command shell - bash

"*Today, many end users rarely, if ever, use command-line interfaces and instead rely upon graphical user interfaces and menu-driven interactions. However, many software developers, system administrators and* ***advanced users*** *still rely heavily on command-line interfaces to perform tasks more efficiently...*" (Wikipedia)

The shell...

- tries to locate a program from an absolute (`/usr/bin/vi`) or relative (`./myprog`, or `bin/myprog`) path
- expands file names like `ls error*.txt`
- provides set of environment variables (`printenv [NAME]`) like...

`PATH`	search path for binaries	Program execution is controlled
`LD_LIBRARY_PATH`	search path for dynamic libraries	
`HOME`	path to user's home directory	

  by command line options.
- comes with a simple language for script execution.

Basic commands

Work with the filesystem from the command line:

`pwd`	print work directory
`ls`	list directory (`ls -ltrs bin`)
`cd`	change directory (`cd` = `cd $HOME`)
`mkdir`	create directory (`mkdir -p child/grandchild`)
`rm`	remove file/directory **Caution: No trash bin!** (`rm -rf tmp/*.err`)
`rmdir`	remove directory
`cp`	copy file/directory (`cp -r results ~/projectXY/`)
`mv`	move/rename file/directory (`mv results ~/projectXY/`)
`chmod`	change access properties (`chmod a+r readme.txt`)
`find`	find a file (`find . -name "*.c"`)
	or `find . -name "core*" -exec rm {} \;`

Basic commands

`echo`	display text to stdout `echo $PATH`
`cat`	display contents of a file `cat > newfile.txt`
`less`, `more`	pagewise display (`less README`)
`grep`	search for words/text (`grep result out.res`)
`file`	determine type of a file
`ps`	display running processes (`ps -axuf`)
`kill`	kill a process (`kill -9 12813`)
`top`	display table of processes (interactive per default)
`ssh`	secure shell to a remote machine (`ssh -X mark@taurus.hrsk.tu-dresden.de`)

Basic commands

`echo`	display text to stdout `echo $PATH`
`cat`	display contents of a file `cat > newfile.txt`
`less`, `more`	pagewise display (`less README`)
`grep`	search for words/text (`grep result out.res`)
`file`	determine type of a file
`ps`	display running processes (`ps -axuf`)
`kill`	kill a process (`kill -9 12813`)
`top`	display table of processes (interactive per default)
`ssh`	secure shell to a remote machine (`ssh -X mark@taurus.hrsk.tu-dresden.de`)

Editors:

- `vi` - a cryptic, non-intuitive, powerful, universal editor. The web has several "cheat sheets" of vi.
- `emacs` - a cryptic, non-intuitive, powerful, universal editor. But it comes with an X11 GUI.
- `nedit` - an inituitve editor with an X11 GUI. (`module load modenv/classic nedit`)

Help at the command line

Every Linux command comes with detailed manual pages. The command `man <program>` is the first aid kit for Linux questions.

```
CHMOD(1)                        User Commands                        CHMOD(1)

NAME
       chmod - change file mode bits

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This manual page documents the GNU version of chmod.  chmod changes the file
       mode bits of each given file according to mode, which can be either  a  sym-
       bolic representation of changes to make, or an octal number representing the
       bit pattern for the new mode bits.

       The format of a symbolic mode is [ugoa...][[+-=][perms...]...], where  perms
       is  either zero or more letters from the set rwxXst, or a single letter from
       the set ugo.  Multiple symbolic modes can be given, separated by commas.

       A combination of the letters ugoa controls which users' access to  the  file
       will  be  changed:  the user who owns it (u), other users in the file's group
       (g), other users not in the file's group (o), or all users (a).  If none  of
       these  are given, the effect is as if a were given, but bits that are set in
Manual page chmod(1) line 1
```

Linux file systems

- mounted remote file systems can be accessed like local resources.
- names are **case sensitive**
- system programs in `/bin`, `/usr/bin`
- third party applications, libraries and tools, special software trees e.g
  - normally in `/opt`
  - ZIH's HPC systems in `/software`
- every user has her own home directory
  - `/home/<login>`
  - e.g. `/home/mark`

Special directories:

- $\sim$ = home directory (`cd` $\sim$ or `cd $HOME`)
- `.` = current directory
- `..`=parent directory

Nelle's Pipeline II

Hypothetical look at the protein scans...

```
~ > ls
scan_results
```

Nelle's Pipeline II

Hypothetical look at the protein scans...

```
~ > ls
scan_results
```

```
~ > mkdir Jellyfish2020
~ > mv scan_results Jellyfish2020
~ > cd Jellyfish2020
```

```
~/Jellyfish2020 > ls scan_results
spec_0001.out spec_0002.out spec_0003.out spec_0004.out
```

Nelle's Pipeline II

Hypothetical look at the protein scans...

```
~ > ls
scan_results
```

```
~ > mkdir Jellyfish2020
~ > mv scan_results Jellyfish2020
~ > cd Jellyfish2020
```

```
~/Jellyfish2020 > ls scan_results
spec_0001.out spec_0002.out spec_0003.out spec_0004.out
```

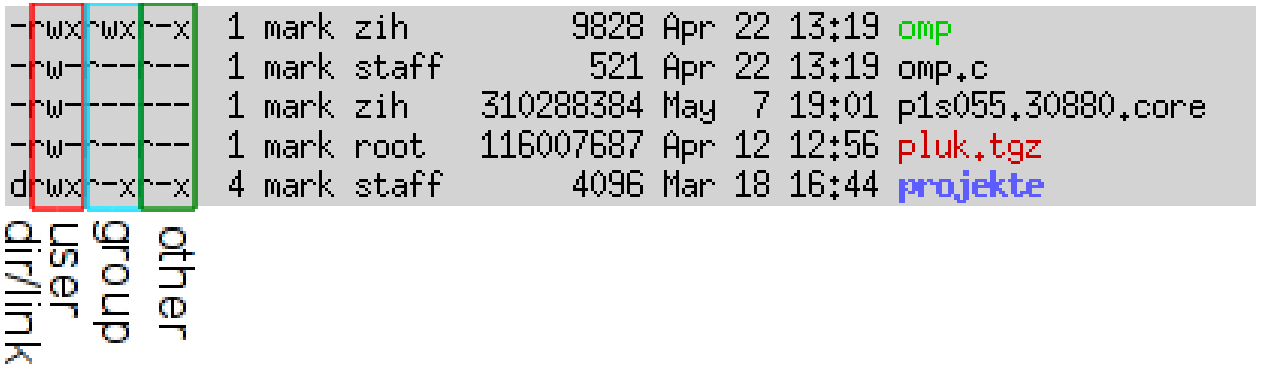
```
~/Jellyfish2020 > for f in scan_results/* ; do \
    calc_statistics $f ; done
```

Remark: Large computations not on the login nodes.

File properties

Every file or directory has its access properties:

- 3 levels of access: **u**ser, **g**roup, **o**ther
- 3 properties per level: **r**ead, **w**rite, e**x**ecute (for directories: execute = enter)
- list directory `ls -l .`



Default: User has all access rights in her `$HOME`-directory.
Which access rights shall be added/removed (easy way)

- set a file readable for all: `chmod a+r readme.txt`
- remove all rights for the group: `chmod g-rwx readme.txt`

Redirection of I/O

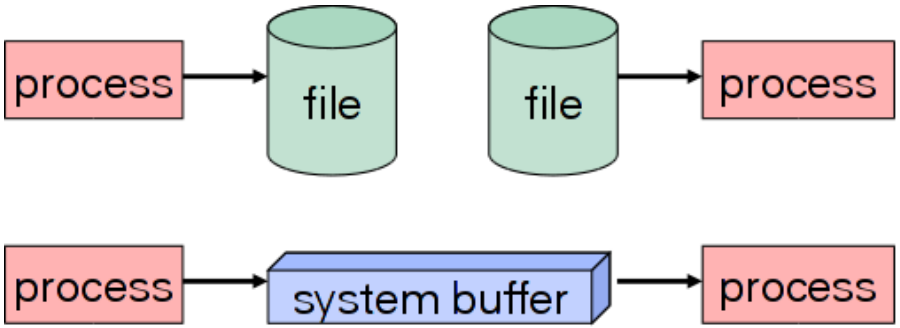
Linux is a text-oriented operating system. Input and output is 'streamable'.

- standard streams are: stdin, stdout, stderr
- streams can be redirected from/to files
  e.g. `myprog <in.txt >out.txt`
- error messages (warnings) are separated from normal program output
  e.g. `myprog 2>error.txt >out.txt`
- merge error messages and output: `myprog 2>&1 out_err.txt`

Attention:
The '>' operator will always empty an existing output file. For appending a stream output to a file use the '>>' operator. e.g. `myprog >>all_outs.txt`.

Command pipelines

Inputs and outputs can also be other programs.




```
ls -la | sort | more
echo 'Have fun!' | sed -s 's/fun/a break/g'
```

Versatility of Linux (and Linux like operating systems) comes from

- command line controlled program execution
- combining multiple programs in a pipelined execution
- mightful scripting, parsing, and little helper tools (shell, awk, sed, perl, grep, sort)

Hands-on training

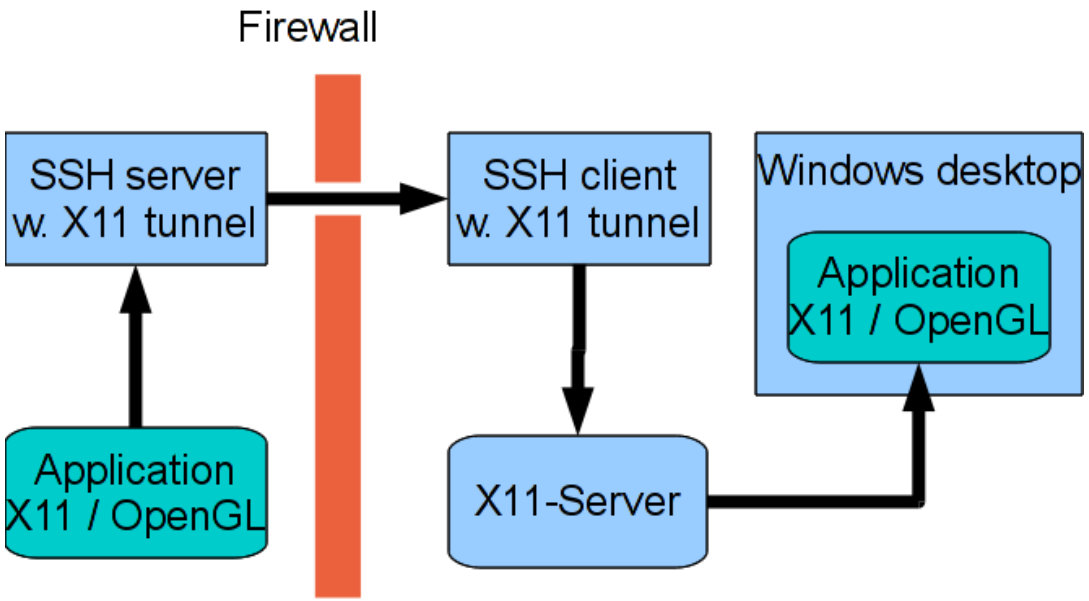
Recommended online material: `http://swcarpentry.github.io/shell-novice`

Introducing the Shell	What is a command shell and why would I use one?
Navigating Files and Directories	How can I move around on my computer? How can I see what files and directories I have? How can I specify the location of a file or directory on my computer?
Working With Files and Directories	How can I create, copy, and delete files and directories? How can I edit files?
Pipes and Filters	How can I combine existing commands to do new things?
Loops	How can I perform the same actions on many different files?
Shell Scripts	How can I save and re-use commands?
Finding Things	How can I find files? How can I find things in files?

X11 tunnel

Why do we need it?

- redirect graphic contents (GUI or images) to personal desktop system
- only SSH connections are allowed with HPC systems
- at desktop: X11 server to handle graphic input (mouse, keyboard) and output (window contents)

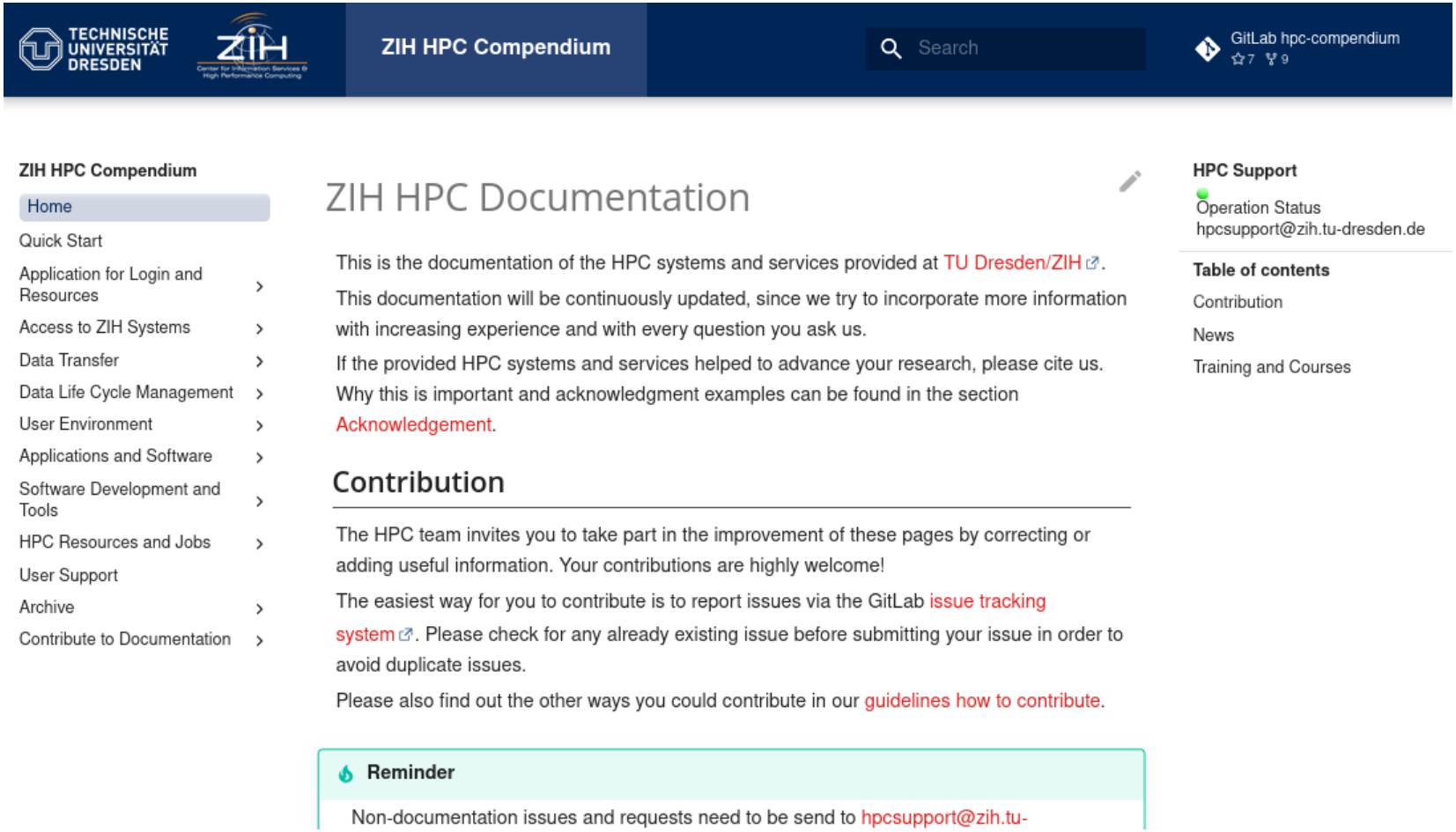


X11 forwarding

- Linux: `ssh -X ...`
- Mac OS X: `https://support.apple.com/downloads/x11`
- Windows:
  - Public Domain tool Xming/Xming-mesa: `http://www.straightrunning.com/XmingNotes` or similar product.
  - enable X11 forwarding in the SSH tool
  - integrated solution in MobaXterm
- OpenGL might be an issue

HPC wiki has the answer

Please check our HPC wiki at `https://compendium.hpc.tu-dresden.de`



Questionnaire

Are you already an HPC user...?

A yes

B no

Questionnaire

Which item describes your HPC-related research best...?

A chemistry and materials science

B life sciences

C physics

D mechanical engineering

E earth sciences

F computer science, mathematics

If none of the above matches: abstain.

Questionnaire

What kind of code do you use mostly (highest CPUh consumption)?

A commercial software

B community software

C “self” developed codes

Agenda

HPC Infrastructure at ZIH

HPC at ZIH

- state's computing center for HPC in Saxony
- HPC systems are funded by BMBF and SMWK
- services free of charge to
  - all universities in Saxony,
  - all listed research institutes (e.g. Leibniz, Max Planck, Fraunhofer institutes)
- active projects outside TUD: MPI-CBG, HZDR, IFW, Uni Leipzig, TUBAF

Nationales Hochleistungsrechnen - NHR

What is National HPC?

- 9 centers at universities
- restructuring (funding, application, workflow) since 2021
- collaboration on technical and organisational aspects (e.g. JARDS)
- better networking between HPC centers

NHR@TUD

- Main focus: life sciences and earth system science,
- Methodological focus:
  - Methods for Big Data, data analysis and data management
  - Machine Learning
  - Tiered storage architectures and I/O optimization
  - Performance and energy efficiency analysis and optimization.

HPC Infrastructure for Data Analytics

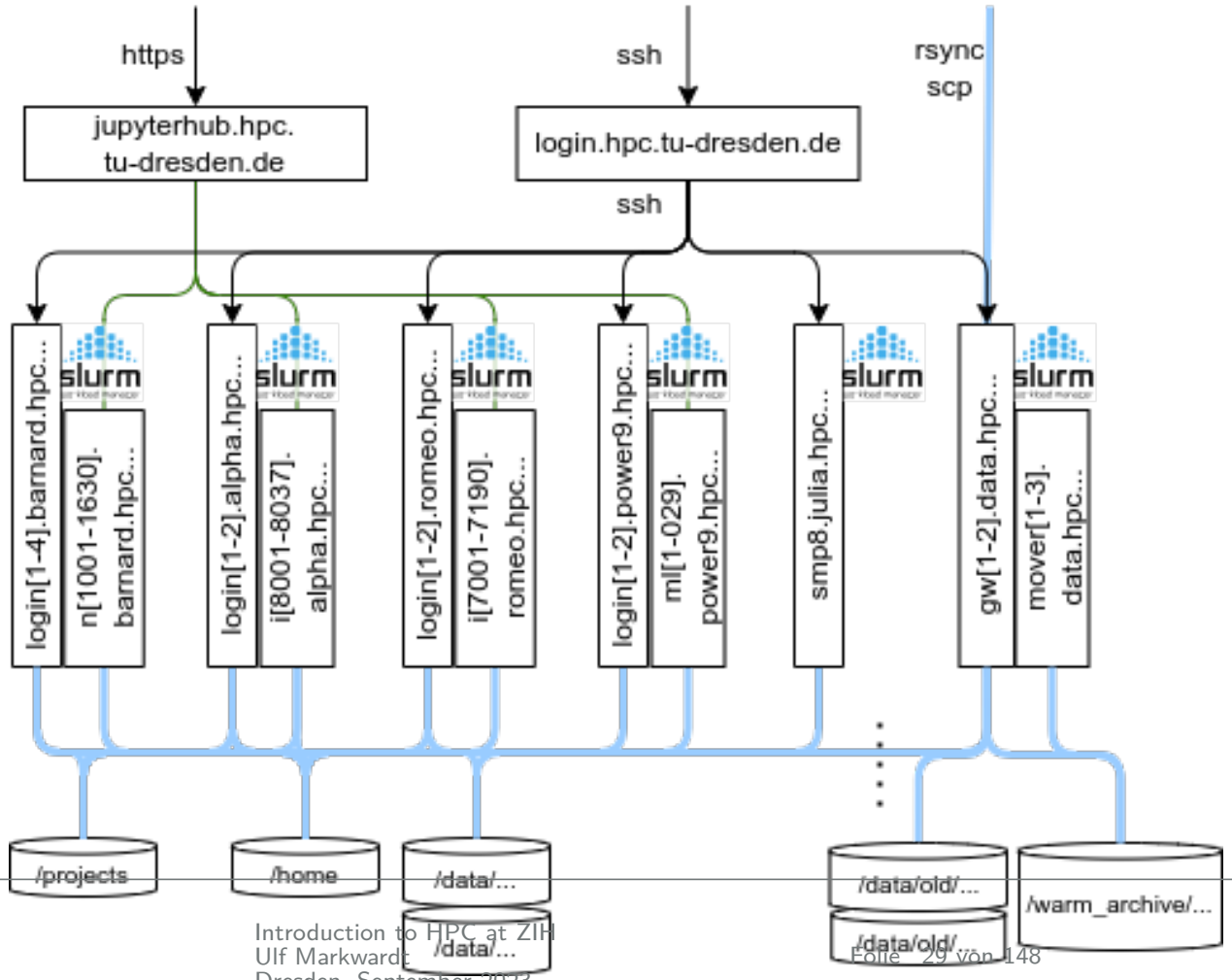
National competence center for data analytics
ScaDS.AI Dresden/Leipzig: Center for Scalable Data Analytics and Artificial Intelligence
`https://scads.ai`

- hardware extensions
  - NVMe nodes (fast storage over Infiniband),
  - nodes for machine learning,
  - “warm archive” for research data, VM images...
  - compute cluster (Romeo)
  - large SMP system (Julia)
  - GPU cluster (Alpha)
- new methods to access systems complementary to “classical” HPC mode
- large team for AI related research and support

Overview

Overview All HPC clusters...

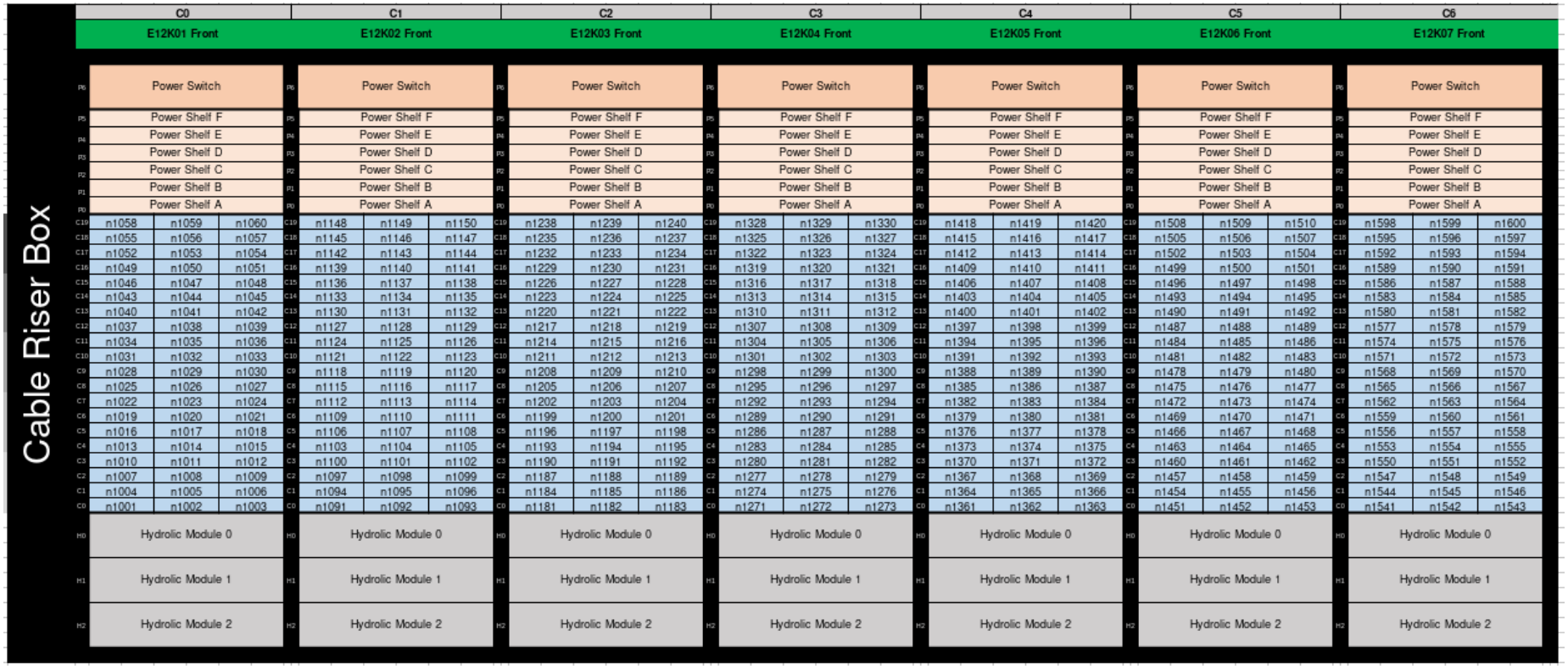
- run with RHEL 8.7 / Rocky 8.7
- have their own Slurm batchsystem,
- share the same parallel file systems with high bandwidth



Barnard

Barnard

Front view



Barnard

Rear view

Barnard

General Purpose Cluster (Bull)

Subdomain: barnard.hpc.tu-dresden.de

- Compute nodes: n[1001-1630]
  - 2x52 Cores Intel Sapphire Rapids
  - 512 GB RAM, diskless
  - Infiniband HDR100
- Login nodes: login[1-4]
  - 2x52 Cores Intel Sapphire Rapids
  - 1 TB GB RAM, 1.9 TB NVMe
  - Infiniband HDR200
- Visualization nodes: vis[1-4]
  - 2x52 Cores Intel Sapphire Rapids
  - 1 TB GB RAM, 1.2 TB NVMe
  - 2x Nvidia A40

Romeo

General Purpose Cluster (NEC)

Subdomain: romeo.hpc.tu-dresden.de

- Compute nodes: i[7001-7188]
  - 2x64 cores AMD Rome EPYC 7702
  - 512 GB RAM, local disk
- login nodes: login[1-2]
  - 2x64 cores AMD Rome EPYC 7702
  - 512 GB RAM, local disk
- use Intel compiler with `-mavx2 -fma`
- for Intel MKL set environment `export MKL_DEBUG_CPU_TYPE=5`

More information on `https://compendium.../jobs_and_resources/rome_nodes`

Alpha Centauri

ScaDS Cluster for Data Analysis and AI (NEC)

Subdomain: romeo.hpc.tu-dresden.de

- Comute nodes: n[8001-8037] beginitemize
- 8 x NVIDIA A 100-SXM4, 40GB RAM
- 2 x AMD EPYC CPU 7352, 1 TB RAM
- 3.5 TB local NVMe

login nodes: login[1-2]

- 8 x NVIDIA A 100-SXM4, 40GB RAM
- 2 x AMD EPYC CPU 7352, 1 TB RAM
- 3.5 TB local NVMe

More information on `https://compendium.../jobs_and_resources/alpha_centauri`

HPE Superdome Flex

Large shared-memory system (HPE Superdome Flex) for memory-intensive computing (2020)

Hostname: julia.hpc.tu-dresden.de

- 48 TB shared memory
- 10.6 TFlop/s peak performance
- 896 cores Intel 8276M CPU (Cascade Lake) 2.20GHz
- 370 TB local NVMe storage mounted at `/nvme`
- batch partition `julia`

Migration to new network and OS might take a bit more time.
More information on `https://compendium.../jobs_and_resources/sd_flex`

IBM Power9

ScaDS Cluster for Machine Learning (IBM)

Subdomain: power9.hpc.tu-dresden.de

- 29 Comute nodes: ml[1-29]
  - 2 x IBM Power9 CPU (2.80 GHz, 3.10 GHz boost, 22 cores)
  - 256 GB RAM DDR4 2666 MHz
  - 6 x NVIDIA VOLTA V100 with 32 GB HBM2
  - NVLINK bandwidth 150 GB/s between GPUs and host
- login nodes: login[1-2]

More information on `https://compendium.../software/machine_learning`

Agenda

Linux from the command line

HPC Environment at ZIH
- Compute hardware
- HPC file systems
- Software environment at ZIH
- Access to HPC systems at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Support

Migration

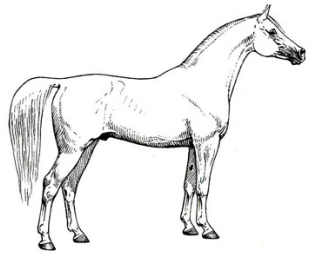
Overview

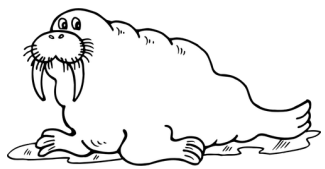
Properties of file systems:

- speed
  - bandwidth
  - IOPS
- size,
- backup, snapshot,
- technology
  - disk type (HDD, SSD, NVMe)
  - locality (local, network)
  - filesystem type (Lustre, NFS, WEKA, BeeGFS, Quobyte, XFS)
  - redundancy levels

Overview

- local SSD `/tmp`
- HPC global `/projects`, `/home`
- HPC global `/ssd`
- HPC global `/data/horse`
- fast IOPS `/data/weasel` - coming later
- interatcive jobs`/data/octopus`
- high capacity storage `/data/walrus`
- (TUD global intermediate archive)
- TUD long term storage for research data - `OPARA`





The **number of files** (billions) is critical for all file systems.

Local disk

- SSD: best option for lots of small I/O operations, limited size ($\sim$ 100GB),
- volatile: data will be deleted automatically after finishing the job,
- local disks only on a few nodes:
  - Rome, AlphaCentauri, smp8, ML
  - on Barnard use feature : `--constraint=local_disk`

Attention Multiple processes on the same node share their local disk.

Mounted at `/tmp`

High-IOPS file system

Coming by end 2023: Powered by WEKAio
Fastest parallel file systems (IOPS) at each* HPC system:

- large parallel file system for high number of I/O operations,
- management via workspaces,
- data may be deleted after 30 days,
- All* HPC nodes share this file system.

Attention Data might get lost.

Mounted at `/data/weasel`

* except Power9

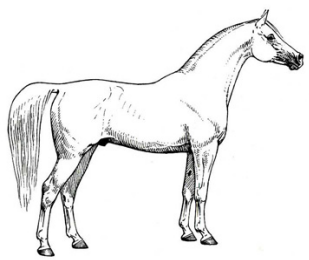
Scratch file system

Workhorse: powered by Lustre
Fastest parallel file systems (streaming) at each HPC machine:

- 20 PB parallel file system for high bandwidth,
- NVMe as caches,
- data may be deleted after 100 days,
- management via workspaces,
- All HPC nodes share this file system.

Attention: Data might get lost. Probably not.

Mounted at `/data/horse`



Permanent file systems

Common file system for all ZIH's HPC machines: powered by Lustre

- NVMe as caches
- Good IOPS rate
- Deleted files are accessible via the snapshots (available via ticket)
- Paths to permanent storage are
  - `/home/<login>` (20 GB !) and
  - `/projects/<projectname>`

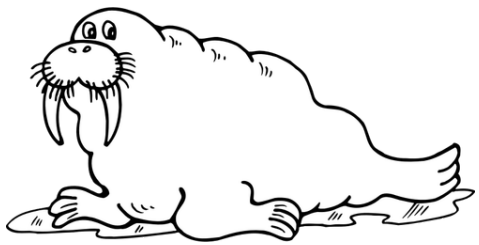
  with different access rights (cf. Terms of Use).
- All HPC systems of ZIH share these file systems.
- Daily tape backups are planned.

High-capacity storage

Large storage at each HPC machine: powered by Lustre

- 20 PB file system for moderately low bandwidth, low IOPS
- management via workspaces,
- all HPC nodes share this file system,
- mounted read-only on compute nodes (to avoid high IOPS)

Mounted at `/data/walrus`



Long-term archive

Common tape based file system:

- really slow and large,
- expected storage time of data: about 3 years,
- access under user's control.

Best practice:

- "Low" file count is important.
- Tar and zip your files. (Use datamover nodes.)
- LTO-6 tapes have a capacity of 2.5 TB. Please ask before you plan to archive files larger than 200 GB.

Data management

Automated workflows vs. ...

- A set of rules specifies how and when data is moved between storage systems.
- Who defines these rules? User or administrator?
- When are actions triggered?

...manual control

- User moves her own data.
- User knows when data can be stored away or have to be retrieved for next processing steps.

In general, users are responsible for their data.
Admins care for usability and data integrity.
See `https://compendium.../data_lifecycle/overview`

Workspaces

Tool for users to manage their storage demands

`https://compendium.../data_lifecycle/workspaces`

- In HPC, projects (and data) have limited lifetime.
- User creates a workspace with defined expiration date.
- User can get an email (or calender entry) before expiration.
- Data is deleted automatically (cf. comment).
- Life-span can be extended twice.

Maximum initial lifetime depends on file system:

Storage system	Duration	Remarks
weasel	30 days	High-IOPS file system, NVMe.
horse	100 days	High streaming bandwidth, disks.
walrus	1 year	Capacity file system, disks.

Workspace - examples

Available filesystems

```
~ > ws_find -l
available filesystems:
horse
walrus
octopus
```

Allocation

```
~ > ws_allocate -F walrus specimen 20
Info: creating workspace.
```

Notification:

```
~ > ws_send_ical -m nelle@tu-dresden.de -F walrus specimen
Sent reminder for workspace specimen to nelle@tu-dresden.de
please do not forget to accept invitation
```

→Calender invitation: “Workspace specimen will be deleted”

Workspace - examples

List all allocated workspaces

```
~ > ws_list -F walrus
id: specimen
     workspace directory  : /data/walrus/ws/nelle-specimen
     remaining time       : 19 days 23 hours
     creation time        : Wed Sep 13 13:21:19 2023
     expiration date      : Tue Oct  3 13:21:19 2023
     filesystem name      : walrus
     available extensions : 2
```

Extend the life time of a workspace

```
~ > ws_extend -F walrus specimen 10
Info: extending workspace.
/data/walrus/ws/nelle-specimen
remaining extensions  : 2
remaining time in days: 10
```

Attention: Extension starts **now**, not at the end of the life time

```
~ > ws_list -F walrus
id: specimen
     workspace directory  : /data/walrus/ws/nelle-specimen
     remaining time       : 9 days 23 hours
     creation time        : Wed Sep 13 13:25:35 2023
     expiration date      : Sat Sep 23 13:25:35 2023
     filesystem name      : walrus
     available extensions : 1
```

Workspace - examples

Workspace within a job

```
#!/bin/bash
#SBATCH -c 20
...
COMPUTE_WS=gaussian_$SLURM_JOB_ID
ws_allocate -F horse $COMPUTE_WS 7
export GAUSS_SCRDIR=/data/horse/ws/$USER-$COMPUTE_WS
srun g16 inputfile.gjf logfile.log

#Tell the "ws expirer" to delete without grace period
ws_release -F horse $COMPUTE_WS
```

Workspace

Expiration of workspaces

- expired workspaces are moved automatically to another location
- after a certain time span (30...60d) they are marked for deletion
- during this time workspaces can be restored by the user using `ws_restore`
- Deletion is final - pay attention to expiration date!

Data transfer

Special data transfer nodes are running in batch mode to comfortably transfer large data between different file systems:

- Commands for data transfer are available on all HPC systems with prefix **dt**: dtcp, dtls, dtmv, dtrm, dtrsync, dttar.
- The transfer job is then created, queued, and processed automatically.
- User gets an email after completion of the job.
- Aditional commands: dtinfo, dtqueue.

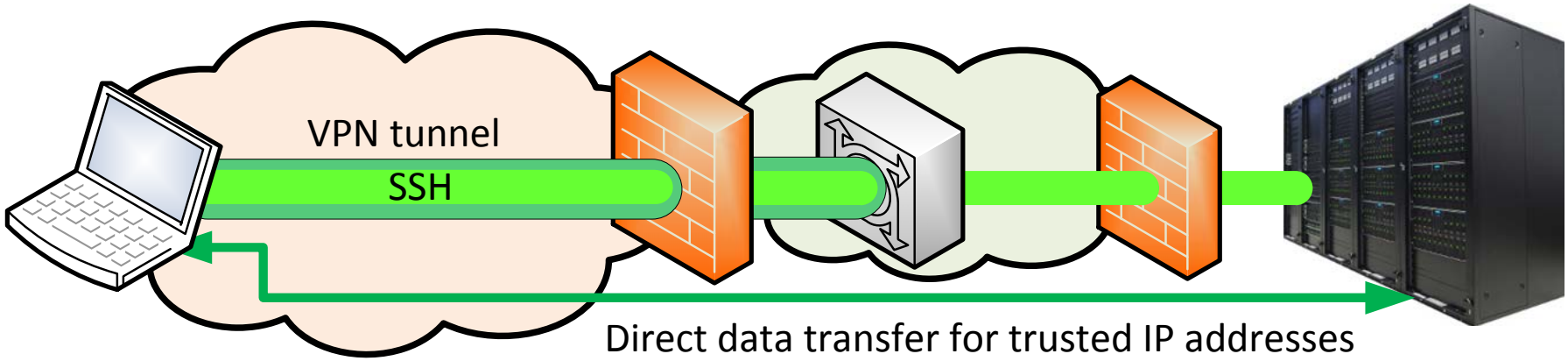
Very simple usage like

```
dttar -czf /warm_archive/ws/jellyfish-2020/results_20190820.tgz \
           /scratch/ws/jellyfish-2020/results
```

See `https://compendium.../data_transfer/overview`

External data transfer

The nodes taurusexport.hrsk.tu-dresden.de allow access with high bandwidth bypassing firewalls



Restrictions

- trusted IP addresses only
- protocols: sftp, rsync

Agenda

Linux from the command line

HPC Environment at ZIH
- Compute hardware
- HPC file systems
- Software environment at ZIH
- Access to HPC systems at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Support

Migration

Modules

Installed software is organized in modules.

A module is a user interface, that:

- allows you to easly switch between different versions of software
- dynamically sets up user's environment (`PATH`, `LD_LIBRARY_PATH`, ...) and loads dependencies.

Private modules files are possible (e.g. group-wide installed software).
`https://compendium.../software/modules`

Installed Software

At the moment 969 Packages!

abaqus, abinit, absl-py, ace, actc, adolc, advisor, afni, alabaster, alembic, algorithm, aliased, amber, amdlibm, amduprof, anaconda3, ansa, ansys, ansysem, ant, antlr, ants, anyevent, apipkg, app, appconfig, appdirs, apr, apr-util, archive, archspec, argparse, arpack-ng, array, arrow, asciidoc, ase, asm, asn1crypto, astunparse, async_generator, at-spi2-atk, at-spi2-core, atk, atomicwrites, atompaw, attr, attrs, authen, auto_ml, autoconf, autoloader, automake, autotools, avs-express, b, babel, backcall, backports.functools-lru-cache, basemap, batchspawner, bazel, bcrypt, bigdataframeworkconfigure, binutils, bison, bitstring, bleach, blist, blitz++, bokeh, boolean, boost, boost.python, bottleneck, bullxmpi, business, bzip2, cachecontrol, cachetools, cachy, cairo, canary, capture, carp, casita, cdo, ceph, cereal, certifi, certipy, cffi, cfitsio, cftime, cg, cgal, cgns, chardet, check, chrpath, clang, class, cleo, clfft, clhep, click, clikit, clone, cloudpickle, clustalw2, cmake, collectl, colorama, common, comsol, config, configparser, configurable-http-proxy, conn, constant, contextlib2, cp2k, cpanplus, cpufrequtils, crashtest, crypt, cryptography, ctags, ctool, cube, cubegui, cubelib, cubew, cubewriter, cuda, cudacore, cudnn, curl, cusp, cwd, cycler, cython, dalton, damask, darshan, dash, dask, dask-jobqueue, dask-mpi, data, dataheap, date, datetime, db, dbd, dbi, dbix, dbus, ddt, deap, decorator, defusedxml, delft3d, devel, devel-nytprof, dftb+, dftd3-lib, digest, dill, dist, distlib, distributed, dmtcp, docopt, docrep, docutils, dolfin, double-conversion, doxygen, dtcmp, dune, dyninst, easybuild, eccodes, ecdsa, eigen, eirods, elpa, elsi, emacs, email, eman2, emboss, encode, ensight, entrypoints, enum34, erlangotp, error, espresso, etsf_io, eval, exception, execnet, expat, expect, exporter, extrae, extutils, fennec, ffc, ffmpeg, fftw, fiat, file, filelock, firestarter, flair, flair-geoviewer, flann, flatbuffers, flatbuffers-python, flex, flink, flit, flit-core, fltk, fme, font, fontconfig, foss, fosscuda, fox, fplo, freecad, freeglut, freesurfer, freetype, fribidi, fsl, fsspec, funcsigs, functools32, future, futures, ga, gamess, gams, gast, gaussian, gaussview, gautomatch, gc, gcc, gcccore, gcccuda, gcl, gcovr, gctf, gdal, gdb, gdk, gdk-pixbuf, gdrcopy, geant4, geany, geos, getopt, gettext, gflags, ghc, ghostscript, giflib, git, git-cola, git-lfs, gitpython, gl2ps, glew, glib, glibmm, glm, glob2, globalarrays, glog, glpk, gmap, gmock, gmp, gmsh, gnuplot, go, gobject-introspection, golf, gomkl, gompi, gompic, google-auth, google-auth-oauthlib, google-pasta, gpaw, gpaw-setups, gperf, gperftools, gpi2, gpi2-mpi, gpudevkit, graph, graphicsmagick, grib_api, grid, groff, gromacs, grpcio, gsl, gsmpi, gsolf, gtk+, guile, gulp, gurobi, gviz-api, gzip, h5py, h5utils, hadoop, harfbuzz, hash, haskell-platform, hdeem, hdf, hdf5, hdfview, heapdict, help2man, hip, hoomd-blue, horovod, hpc-x, hpx, html, html5lib, htop, http, hwloc, hyperdex, hyperopt, hypothesis, hypre, icc, iccifort, icu, idna, if, ifort, iimpi, ima, imagemagick, imagesize, imkl, impi, import, importlib-metadata, importlib_metadata, importlib_resources, iniconfig, inline, inspector, intel, intelmpi, intervaltree, intltool, intreehooks, io, iomkl, iompi, iotop, iotrack, ipaddress, ipc, ipopt, ipykernel, ipython, ipython_genutils, ipywidgets, isl, isodate, itac, jasper, java, jedi, jeepney, jinja2, joblib, json, json5, jsoncpp, jsonschema, julia, junit, jupyter-telemetry, jupyter_client, jupyter_contrib_core, jupyter_core, jupyter_nbextensions_configurator, jupyterhub, jupyterhub-jwtauthenticator-v2, jupyterhub-ldapauthenticator, jupyterhub-nativeauthenticator, jupyterhub-simplespawner, jupyterhub-systemdspawner, jupyterlab, jupyterlab_server, keras, keras-preprocessing, keyring, keyrings.alt, kim-api, kiwisolver, knime, lame, lammps, lbfgsb, ldap3, liac-arff, libarchive, libcerf, libcint, libcircle, libdap, libdrm, libelf, libepoxy, libevent, libfabric, libffi, libgd, libgeotiff, libglu, libglvnd, libgridxc, libharu, libiconv, libint, libjpeg-turbo, libmatheval, libnbc, libpciaccess, libpng, libpsml, libreadline, libsigc++, libsndfile, libsodium, libssh2, libsvm, libtiff, libtirpc, libtool, libunistring, libunwind, libuuid, libvdwxc, libxc, libxml++, libxml2, libxslt, libxsmm, libyaml, liggghts, likwid, lingua, list, littlecms, llvm, lmdb, lo2s, locale, locket, lockfile, log, log4cxx, logger, lpsolve, ls-dyna, ls-dyna-usermat, ls-opt, ls-prepost, lumerical, lwgrp, lwp, lz4, m4, m4ri, magma, mail, make, makedepend, makeinfo, mako, map, markdown, markupsafe, math, mathematica, matio, matlab, matplotlib, maven, maxima, mce, mdanalysis, med, meep, mercurial, mesa, meson, mesquite, meta, metis, mime, miniconda2, miniconda3, mistune, mixin, mkl, mkl-dnn, mock, modenv, module, molmod, mongodb, moo, moose, moosex, more-itertools, motif, motioncor2, mouse, mozilla, mpb, mpc, mpfr, mpi4py, mpifileutils, mpirt, mpmath, mro, msgpack, mumps, must, mvapich2, mysql, namd, namespace, nasm, nbconvert, nbformat, nccl, ncdf4, nco, ncurses, nedit, nektar++, net, netaddr, netbeans, netcdf, netcdf-fortran, netcdf4-python, netifaces, netlogo, nettle, nextstrain, nfft, ninja, nlopt, nltk, nodejs, nose, notebook, nspr, nss, nsync, nsys, numactl, numba, numbapro, numdiff, numexpr, numpy, nvhpc, nvidia-nsight, nvtop, oauthlib, object, octave, octopus, onetimepass, opari2, openbabel, openblas, opencl, opencv, openems, openfoam, openfoam-extend, openmolcas, openmpi, openmx, openpgm, openssl, opentelemac, opt-einsum, orca, otf2, p7zip, package, packaging, pamela, pandas, pandoc, pandocfilters, pango, papi, parallel, paramiko, params, paraview, parent, parmetis,

Hierarchical module environment

Module hierarchy (at each hierarchy level,

- starting point: release version (e.g. 23.04)
  - will be updated cyclic (once/twice a year)
  - new software will be found in future-release versions (without guarantee)
- `module av` shows the next set of available modules

```
~> module av
--------------- Software build with Compiler GCC version 12.2.0 (HMNS Level Two) ----------------
   BLIS/0.9.0    FFTW/3.3.10    OpenBLAS/0.3.21    OpenMPI/4.1.4

------------- Software build with Compiler GCCcore version 12.2.0 (HMNS Level Two) --------------
   Autoconf/2.71           (D)      groff/1.22.4              numactl/2.0.16
   Automake/1.16.5         (D)      help2man/1.49.2           Perl/5.36.0
   Autotools/20220317      (D)      hwloc/2.8.0               pkgconf/1.9.3      (D)
   ...
 -------------------- Core Modules for rapids release r23.04 (HMNS Level One) --------------------
   ABAQUS/2022                          *GCCcore/11.3.0
   Anaconda3/2019.03                    *GCCcore/12.2.0          (L,D)
   Anaconda3/2022.05              (D)    gettext/0.19.8.1
...
```

Module usage

Use `module spider` to identify your desired module and version (case-sensitive):

```
~> module spider ParaView
----------------------------------------------------------------------------------------------------
      ParaView: ParaView/5.10.1-mpi
----------------------------------------------------------------------------------------------------
    Description:
      ParaView is a scientific parallel visualizer.

    You will need to load all module(s) on any one of the lines below before the "ParaView/5.10.1
          -mpi" module is available to load.

      release/23.04  GCC/11.3.0  OpenMPI/4.1.4

    Help:
      Description
      ===========
      ParaView is a scientific parallel visualizer.

      More information
      ================
       - Homepage: https://www.paraview.org
```

Module usage

Information from `module spider`

```
~> module spider SciPy-bundle/2022.05

----------------------------------------------------------------------------------------------------

  SciPy-bundle: SciPy-bundle/2022.05
----------------------------------------------------------------------------------------------------

    Description:
      Bundle of Python packages for scientific software

    You will need to load all module(s) on any one of the lines below before the "SciPy-bundle/2
          022.05" module is available to load.

      release/23.04  GCC/11.3.0  OpenMPI/4.1.4

    Help:
      Description
      ===========
      Bundle of Python packages for scientific software

      More information
      ================
       - Homepage: https://python.org/

      Included extensions
      ===================
      beniget-0.4.1, Bottleneck-1.3.4, deap-1.3.3, gast-0.5.3, mpi4py-3.1.3,
      mpmath-1.2.1, numexpr-2.8.1, numpy-1.22.3, pandas-1.4.2, ply-3.11,
      pythran-0.11.0, scipy-1.8.1
```

Modules for different architectures

Only on Taurus!

Not all software modules are available on all hardware platforms.
Information from `ml_arch_avail`

```
~>  ml_arch_avail CP2K
CP2K/6.1-foss-2019a: haswell, rome
CP2K/5.1-intel-2018a: sandy, haswell
CP2K/6.1-foss-2019a-spglib: haswell, rome
CP2K/6.1-intel-2018a: sandy, haswell
CP2K/6.1-intel-2018a-spglib: haswell
```

```
~> ml_arch_avail tensorflow|sort
TensorFlow/1.10.0-fosscuda-2018b-Python-3.6.6: sandy, haswell, rome
TensorFlow/1.14.0-PythonAnaconda-3.6: ml
TensorFlow/1.15.0-fosscuda-2019b-Python-3.7.4: haswell, rome, ml
TensorFlow/1.15.0-fosscuda-2019b-Python-3.7.4: haswell, rome, ml
TensorFlow/1.8.0-foss-2018a-Python-3.6.4-CUDA-9.2.88: sandy, haswell, rome
TensorFlow/2.0.0-foss-2019a-Python-3.7.2: sandy, haswell, rome
TensorFlow/2.0.0-fosscuda-2019b-Python-3.7.4: haswell, rome, ml
TensorFlow/2.0.0-fosscuda-2019b-Python-3.7.4: haswell, rome, ml
TensorFlow/2.0.0-PythonAnaconda-3.7: ml
TensorFlow/2.1.0-fosscuda-2019b-Python-3.7.4: haswell, rome, ml
TensorFlow/2.1.0-fosscuda-2019b-Python-3.7.4: haswell, rome, ml
TensorFlow/2.2.0-fosscuda-2019b-Python-3.7.4: ml
```

Module commands

`module avail` - lists all available modules (in the current module environment)
`module spider` - lists all available modules (across all module environments)
`module list` - lists all currently loaded modules
`module show <modname>` - display informations about `<modname>`
`module load <modname>` - loads module `modname`
`module save` - saves the current modules, to be reloaded at the next login
`module rm <modname>` - unloads module `modname`
`module purge` - unloads all modules

Modules for HPC applications

Loading compiler, MPI, and BLAS/LAPACK

```
~> module load foss/2022a
Module foss/2022a and 21 dependencies loaded.

~>mpicc --show
gcc -I/software/rapids/r23.04/OpenMPI/4.1.4-GCC-11.3.0/include -L/software/rapids/r23.04/OpenMPI
      /4.1.4-GCC-11.3.0/lib ... -lmpi

~> mpicc hello.c

~> srun -n 4 -t 1 -N 1 --mem-per-cpu=500 ./a.out
srun: job 444632 queued and waiting for resources
srun: job 444632 has been allocated resources
Hello world from processor n1630, rank 0 out of 4 processors
Hello world from processor n1630, rank 1 out of 4 processors
Hello world from processor n1630, rank 3 out of 4 processors
Hello world from processor n1630, rank 2 out of 4 processors
```

Remarks

Commercial codes requiring licenses (Matlab, Ansys)

- basic principle: do not uses thes extensively, we have only a limited number of licenses!
- Matlab: use the Matlab compiler `https://compendium.../software/mathematics/#matlab`

Containers

- Singularity as container environment on Taurus
- Docker containers can easily be converted
- more information at `https://compendium.../software/containers`

Agenda

Linux from the command line

HPC Environment at ZIH
- Compute hardware
- HPC file systems
- Software environment at ZIH
- Access to HPC systems at ZIH

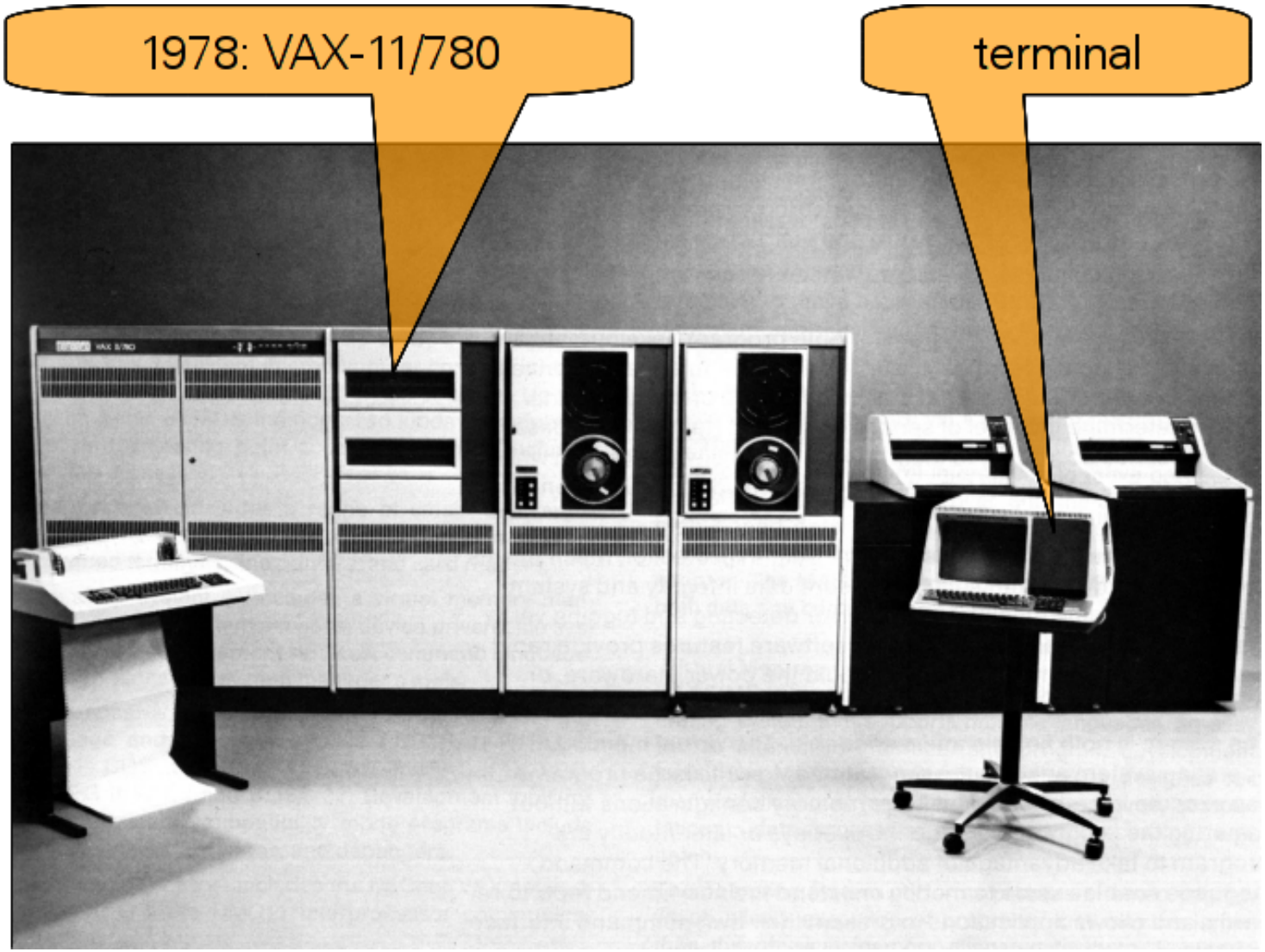
Batch System

Software Development at ZIH's HPC systems

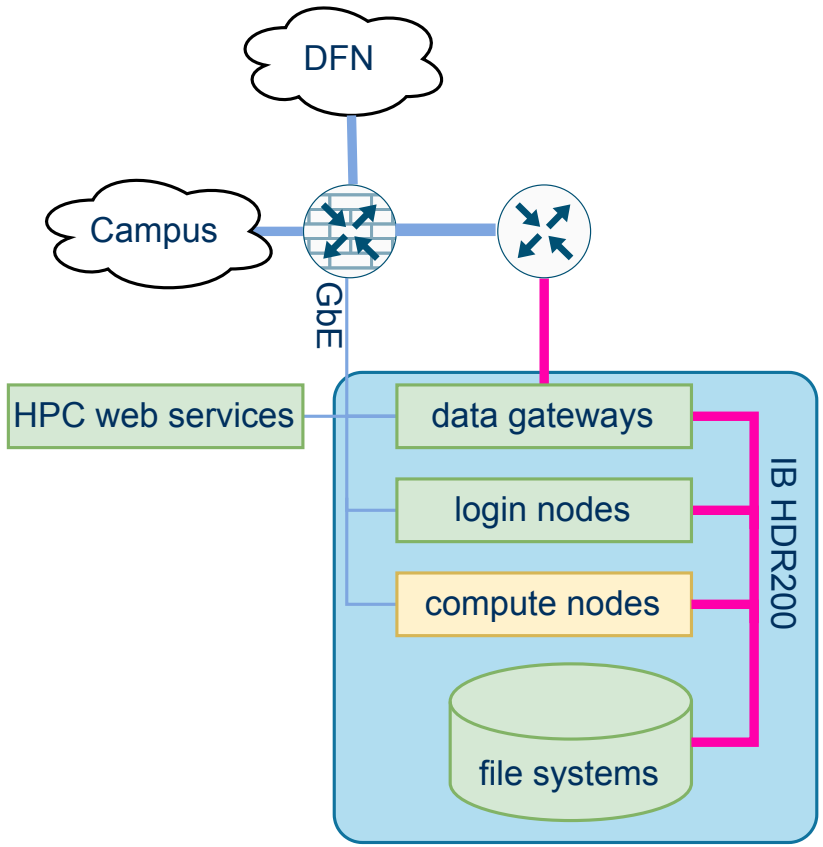
HPC Support

Migration

Computer and terminal



Network Overview



High-bandwidth data transfer to data gateways from Campus and acknowledged IP ranges (e.g. TUBAF: 139.20.0.0/16, TU Chemnitz 134.109.0.0/16, Uni Leipzig: 139.18.2.0/24)

VPN for external users

The only SSH access to ZIH's HPC systems is

- from within the TU Dresden campus
- via secure shell (ssh).

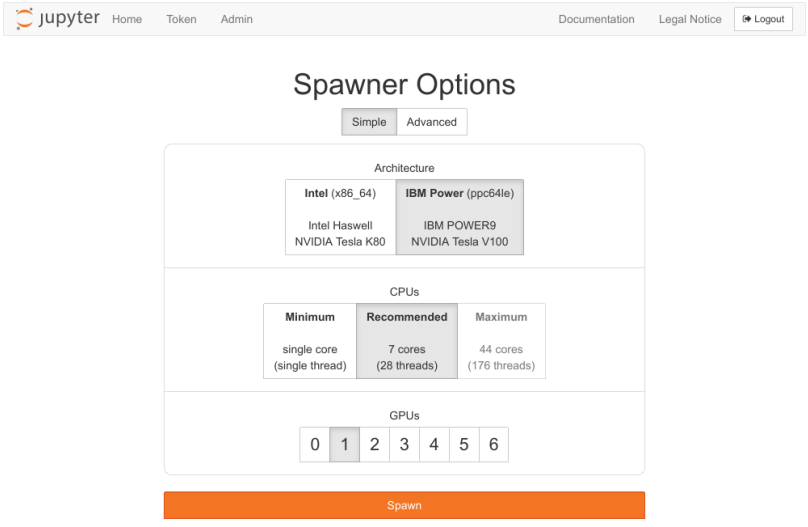
From other IP ranges: **V**irtual **P**rivate **N**etwork
How-To for Linux, Windows, Mac can be found here: `https://tu-dresden.de/zih/dienste/service-katalog/arbeitsumgebung/zugang_datennetz/vpn`

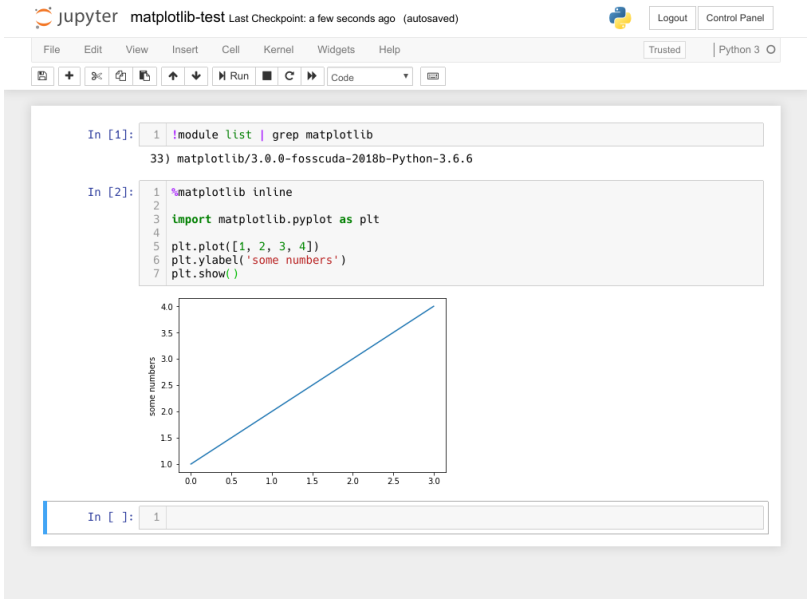
- install VPN tool at your local machine
  - OpenConnect (`http://www.infradead.org/openconnect`)
  - Cisco Anyconnect
- configuration

gateway	vpn2.zih.tu-dresden.de
group	TUD-vpn-all
username	<ZIH-LOGIN>@tu-dresden.de
password	<ZIH-PASSWORD>

Access to HPC

Use X11 forwarding with `ssh -X taurus.hrsk.tu-dresden.de`.
Or use a GUI from your Web browser → JupyterHub.





Detailled documentation can be found at `https://compendium.../access/jupyterhub` .

Agenda

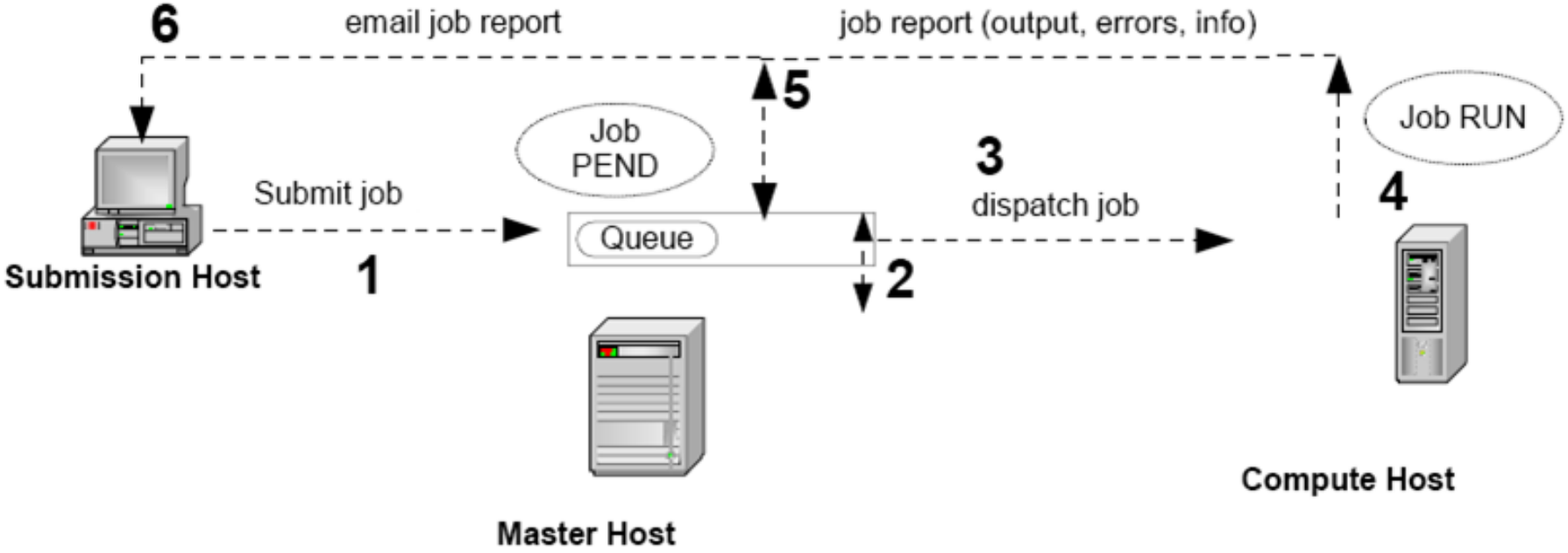
Overview

Why do we need a batchsystem?

- Find an adequate compute system for our needs.
- All resources in use? - The batch system organizes the queueing and messaging for us.
- Allocate the resource for us.
- Connect to the resource, transfer run-time environment, start the job.

Workflow of a batch system

Agreed, we need a batchsystem.



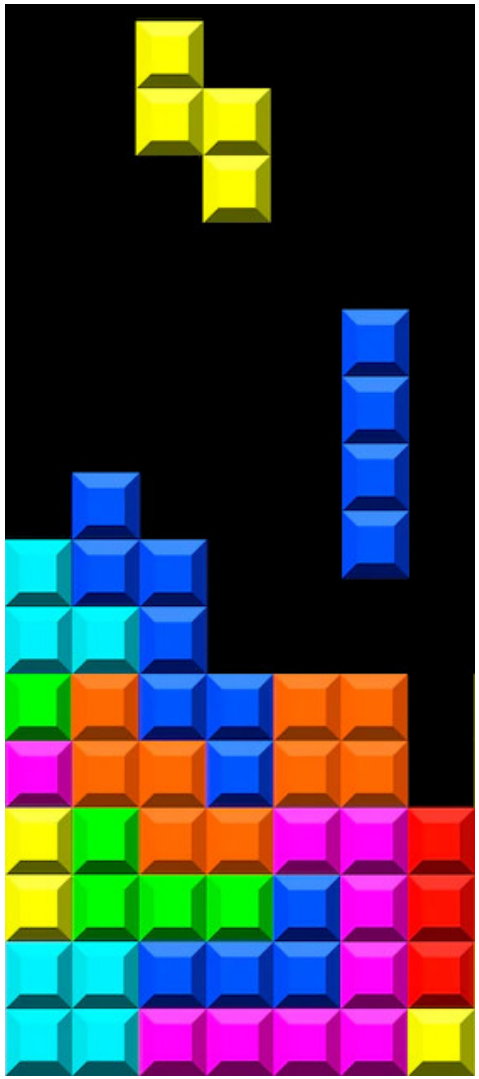
Multi-dimensional optimizations

Optimization goals:

- **Users want short waiting time.**

- Queueing priorities according to:
  - waiting time of the job (+),
  - used CPU time in the last 2 weeks (-),
  - remaining CPU time for the HPC project (+),
  - duration of the job (-)
- Limited resources require efficient job placement:
  - number of compute cores / compute nodes,
  - required memory per core for the job,
  - maximum wall clock time for the job

Optimization is NP-hard $\rightarrow$ heuristics allowed.



Useful functions of a batchsystem

Basic user functions:

- submit a job,
- monitor the status of my job (notifications),
- cancel my job

Additional functions:

- check the status of the job queue,
- handle job dependencies,
- handle job arrays

Job submission: required information

In order to allow the scheduler an efficient job placement it needs these specifications:

- requirements: cores, memory per core, (nodes), additional resources (GPU, local disk)
- maximum run-time,
- HPC project (normally use primary group which gives `id`),
- who gets an email on which occasion,

... to run the job:

- executable with path and command-line,
- environment is normally taken from the submit shell.

Queueing order

Factors that determine the position in the queue:

- **Total share of the project:**
  remaining CPU quota, new project starts with 100% (updated daily)
- **Share within the project:**
  balance equal chances between users of one project
- **Age:**
  the longer a job waits the higher becomes its priority
- **Recent usage:**
  the more CPU time a user has consumed recently the lower becomes her priority,
- **Quality of Service:**
  additional control factors, e.g. to restrict the number of long running large jobs

Pre-factors are subject to adaptations by the batch system administrators.

Overview Slurm

submit a job script	`sbatch`
run interactive job	`srun --pty ...`
monitor a job status	`squeue` - Not permanently!
kill a job	`scancel`
cluster status	`sinfo` - Not permanently!
host status	`sinfo -N`
max job time	`-t <[hh:]mm:ss>`
number of processes	`-n <N>`
number of nodes	`-N <N>`
MB per core	`--mem-per-cpu`
output file	`--output=result_%j.txt`
error file	`--error=error_%j.txt`
notification (TUD)	`--mail-user <email>`
notification reason	`--mail-type ALL`

Overview Slurm

job array	`--array 3-8`
job ID	`$SLURM_ARRAY_JOB_ID`
array idx	`$SLURM_ARRAY_TASK_ID`
redirect stdin and stdout (interactive jobs)	`--pty`
X11 forwarding	`--x11=first`

Examples for parameters for our batch systems can be found at `https://compendium.../jobs_and_resources/slurm` .

- job arrays,
- job dependecies,
- multi-threaded jobs

Slurm partitions - Taurus only

- `haswell` – largest compute partition, Intel x86_64 based, most software runs here. Differenz sizes of RAM managed by job submit plugin.
- `broadwell` – 32 nodes comparable to `haswell`. Intel x86_64 based. Most software runs here.
- `romeo` – powerful compute partition, AMD x86_64 based, most software should run here.
- `julia` – largest SMP node, Intel x86_64 based. For memory-consuming software. Don't use OpenMPI.
- `gpu2` – GPU partition, Intel x86_64 based. Most GPU software runs here.
- `ml` – powerful GPU partition for Machine Learning. IBM Power based. Only special software runs here.
- `hpdlf` – GPU partition for deep learning project, Intel x86_64 based. Most GPU software runs here.
- `alpha` – powerful GPU partition for ScaDS.AI. (Only short jobs (<24h).)
- `interactive` – `haswell` nodes for interactive jobs
- `gpu2-interactive` – `gpu2` nodes for interactive jobs
- `haswell64ht` – `haswell` nodes with activated HyperThreads

Slurm partitions - 2023

Less confusing - each cluster has its own batchsystem and only one partition.

Agenda

Linux from the command line

HPC Environment at ZIH

Batch System
- General
- Slurm examples

Software Development at ZIH's HPC systems

HPC Support

Migration

Slurm examples

Slurm interactive example:

```
srun --ntasks=1 --cpus-per-task=1 --time=1:00:00 \
     --mem-per-cpu=1000 --pty -p interactive bash
```

Slurm X11 example:

```
module load matlab
srun --ntasks=1 --cpus-per-task=8 --time=1:00:00 \
     --mem-per-cpu=1000 --pty --x11=first -p interactive
         matlab
```

Remarks:

- normally: shared usage of resources
- if a job asks for more memory it will be canceled by Slurm automatically
- a job is confined to its requested CPUs

After migration:

- QoS for interactive jobs will be set automatically. Highest prio for these.
- simply omit `-p interactive`

Slurm examples

Normal MPI parallel job `sbatch <myjobfile>`

```
#SBATCH --partition=haswell,romeo
#SBATCH --time=8:00:00
#SBATCH --ntasks=64
#SBATCH --mem-per-cpu=780
#SBATCH --mail-type=end
#SBATCH --mail-user=ulf.markwardt@tu-dresden.de
#SBATCH -o output_%j.txt
#SBATCH -e stderr_%j.txt
srun ./path/to/binary
```

Remark: The batch system is responsible to minimize number of nodes.

After migration: omit --partition...

Slurm examples

Requesting multiple GPU cards

```
#SBATCH --partition=alpha
#SBATCH --time=4:00:00
#SBATCH --job-name=MyGPUJob
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=2
#SBATCH --cpus-per-task=8
#SBATCH --gres=gpu:2
#SBATCH --mem-per-cpu=1200
#SBATCH --mail-type=END
#SBATCH --mail-user=ulf.markwardt@tu-dresden.de

#SBATCH -o stdout
#SBATCH -e stderr
echo 'Running program...'
```

After migration: omit --partition...

Slurm: Job monitoring

Basic question: Why does my job not start? Try: `whypending <jobid>`

```
> whypending 4719686
Reason Priority means that the job can run as soon as resources free up and the higher priority
      job start.
Position in queue: 5873
Estimated start time: Fri Sep 18 05:16:29 2020
==================================================
        Resource Availability Information:
==================================================
Your job is requesting:
    Time Limit: 6-20:00:00
    Nodes: 1
    Cores: 24
    Memory per core: 1500M
    Total Memory: 36000M
    QOS: long
    Features:
    Partitions: haswell64,broadwell

The following nodes are available in partition(s) haswell64,broadwell:
    Total: 28
    Fully Idle: 0
    Partially Idle: 28  (misleading... see note below)
         1 cores free: 5
         2 cores free: 5
         3 cores free: 4
         4 cores free: 7
```

Slurm: Fair share monitoring

Is my fair share really so low???

```
> sshare -u mark -A swtest
Accounts requested:      : swtest
Account  User  Raw Shares  Norm Shares  Raw Usage  Effectv Usage  FairShare
-------  ----  ----------  -----------  ---------  -------------  ---------
swtest                 0     0.000000     680889       0.000033   0.000000
swtest   mark      parent     0.000000      16789       0.000001  *0.000000*
```

Project information

Look at the login screen. Or `showquota`

```
CPU-Quotas as of 2020-09-14 10:54
    Project     Used(h)    Quota(h)        % Comment
    swtest      648440      300000    216.1 Limit reached (SOFT) *
* Job priority is minimal for this project

Disk-Quotas for /projects as of 2020-09-14 10:51
              Project    Used(GiB)  Quota(GiB)        % Comment
               swtest        157.5       300.0     52.5
```

As soon as a project reaches its CPU limit the share drops to 0.

As soon as a project reaches its DISK limit submission is blocked.
→ Clean up first!

What is fair...?

Fair share of a project is based on

- leftover CPU quota of the current month: *RawShare* $\rightarrow$ *NormShares*
- used resources "during the last few days" *RawUsage* $\rightarrow$ *EffektvUsage*
  CPUs usage is summed up with an exponential decay
  (half-value period 1 day)

Account	RawShares	NormShares	RawUsage	EffectvUsage	FairShare
p_abc	369	0.001355	123069773	0.034009	0.030841
p_def	342	0.001256	1962604	0.000546	0.941520

$FairShare = 2^{\frac{-EffektvUsage}{d \cdot NormShares}}$ (dampening factor $d = 5$).

See: `https://slurm.schedmd.com/priority_multifactor.html`

System information

Look at the login screen. Or `nodestat`

```
> nodestat
-------------------------------------------------------------------------------
nodes available:    1758/1967    nodes unavailable:  209/1967
gpus  available:    464/579      gpus  unavailable:  115/579
-----------------------------------+-------------------------------------------
jobs running:       849      |     cores in use:       54764
jobs pending:       3397     |     cores unavailable:  5884
jobs suspend:       0        |     gpus  in use:       258
jobs damaged:       1        |
-------------------------------------------------------------------------------
                            CORES / GPUS
                     free  |  resv  |  down  | total
                    -------+--------+--------+-------
Haswell 64GB:         405  | 10536  |   672  | 31248  (mem-per-cpu <= 2583)
Haswell 128GB:        369  |     0  |     0  |  2016  (mem-per-cpu <= 5250)
Haswell 256GB:        612  |     0  |     0  |  1056  (mem-per-cpu <= 10583)
                    -------+--------+--------+-------
Broadwell 64GB:        45  |     0  |     0  |   896  (mem-per-cpu <= 2214)
                    -------+--------+--------+-------
Rome 512GB:          4818  |  4480  |   768  | 24576  (mem-per-cpu <= 1972)
                    -------+--------+--------+-------
SMP 1TB:                0  |     0  |    64  |    64  (mem-per-cpu <= 31875)
SMP 2TB:              224  |     0  |     0  |   280  (mem-per-cpu <= 36500)
                    -------+--------+--------+-------
GPUs K20X:              0  |     0  |    64  |    64  (partition = gpu1)
GPUs K80:              19  |   208  |    12  |   248  (partition = gpu2)
GPUs V100:            142  |     6  |    12  |   192  (partition = ml)
-------------------------------------------------------------------------------
```

See also `sinfo -T`.

Simple job monitoring

Job information

```
~ > sjob 4843539
JobId=4843539 UserId=mark(19423) Account=hpcsupport JobName=bash
   TimeLimit=1-00:00:00 NumNodes=171 NumCPUs=4096
   TRES=cpu=4096,mem=1200G,node=1,billing=4096 Partition=
       haswell64,romeo
   JobState=PENDING Reason=Resources Dependency=(null)
   Priority=49533 QOS=normal
   StartTime=Unknown SubmitTime=2020-09-18T14:16:06
```

Detailled job monitoring

Detailled job information

```
~ > scontrol show job 4843539
JobId=4843539 JobName=bash
   UserId=mark(19423) GroupId=hpcsupport(50245) MCS_label=N/A
   Priority=49533 Nice=0 Account=hpcsupport QOS=normal
   JobState=PENDING Reason=Resources Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=0 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=1-00:00:00 TimeMin=N/A
   SubmitTime=2020-09-18T14:16:06 EligibleTime=2020-09-18T14:16:06
   AccrueTime=2020-09-18T14:16:06
   StartTime=Unknown EndTime=Unknown Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2020-09-18T14:16:26
   Partition=haswell64,romeo AllocNode:Sid=tauruslogin3:5741
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=(null)
   NumNodes=171 NumCPUs=4096 NumTasks=4096 CPUs/Task=1 ReqB:S:C:T=0:0:*:1
   TRES=cpu=4096,mem=1200G,node=1,billing=4096
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:1 CoreSpec=*
   MinCPUsNode=1 MinMemoryCPU=300M MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=bash
   WorkDir=/home/h3/mark
   Comment=<<<ZIH_JOB_STATS__REMOVE_HDF5>>>
   CPU_max_freq=Highm1
   Power=
```

Slurm tools

`scontrol show ...`

- `job <number>` – job information
- `reservation [ID]` – information on current and future reservations
- `node <name>` – status of a node

More tools

- `scancel` – cancel job
- `squeue` – show current queue jobs
- `sprio` – show priorities of current queue jobs
- efficiently distribute/collect data files to/from compute nodes: `sbcast`, `sgather`
- `sinfo` – cluster information (`-T` : reservations)

See man pages or documentation at `http://slurm.schedmd.com`

Still... not starting

The system looks empty, but no job starts. Especially not mine!

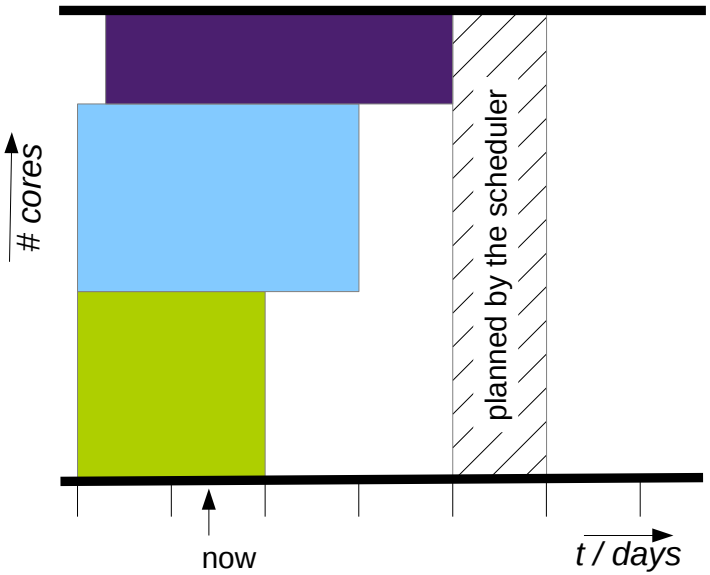
- Maybe a reservation prevents my job from starting (`sinfo -T`)
- Maybe an older large job is scheduled and waits for resources:

```
~ > sprio -S "-y" |head -n 20
    JOBID PARTITION PRIORITY SITE AGE FAIRSHARE JOBSIZE QOS
  4832990 haswell64    72001    0  11     26987       4   0
  4832990 broadwell    72001    0  11     26987       4   0
  4842303 haswell64    65993    0   3     26987       4   0
  4842303 broadwell    65993    0   3     26987       4   0
```

  Here is job 4832990 with a very high priority, scheduled for a certain time (see `scontrol show job`) . If my job would finish before that one it could be backfilled.

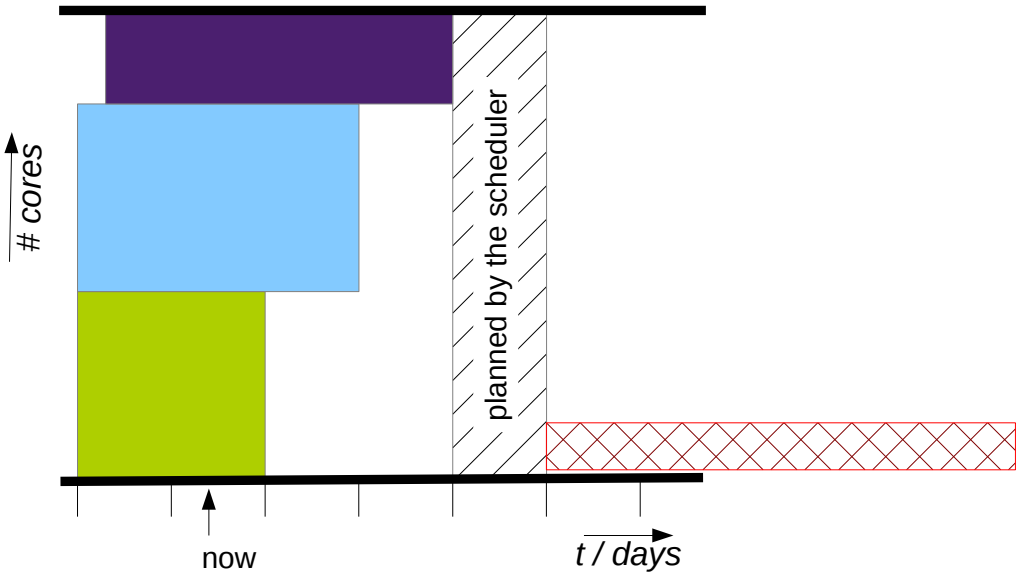
- Maybe fragmentation would be too high.

Backfilling

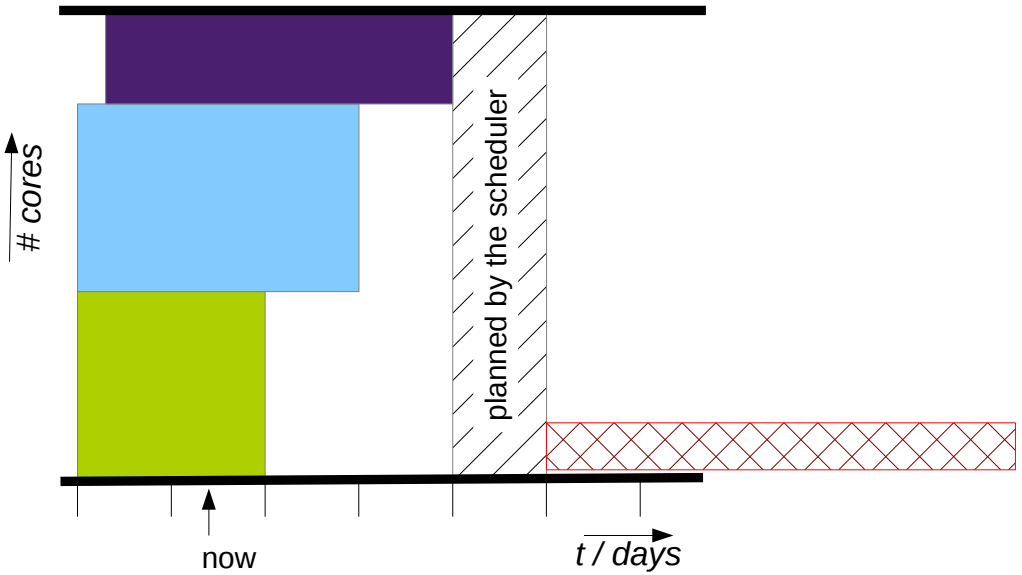


My job to be placed:

Backfilling

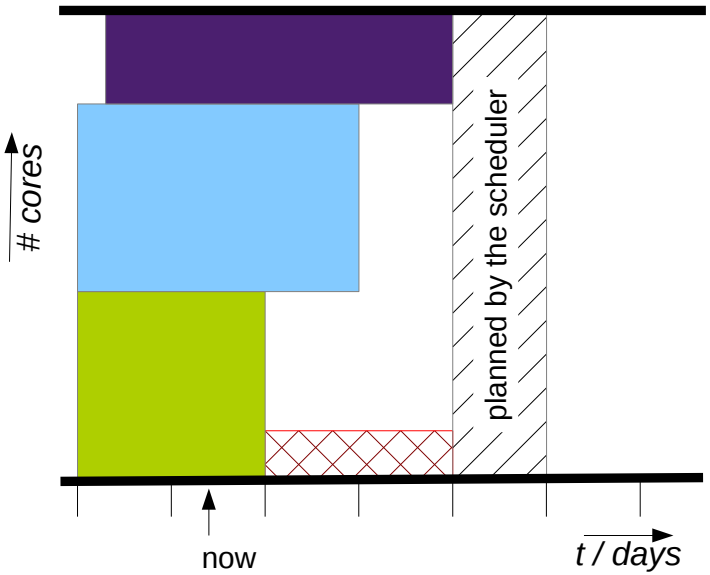


Backfilling



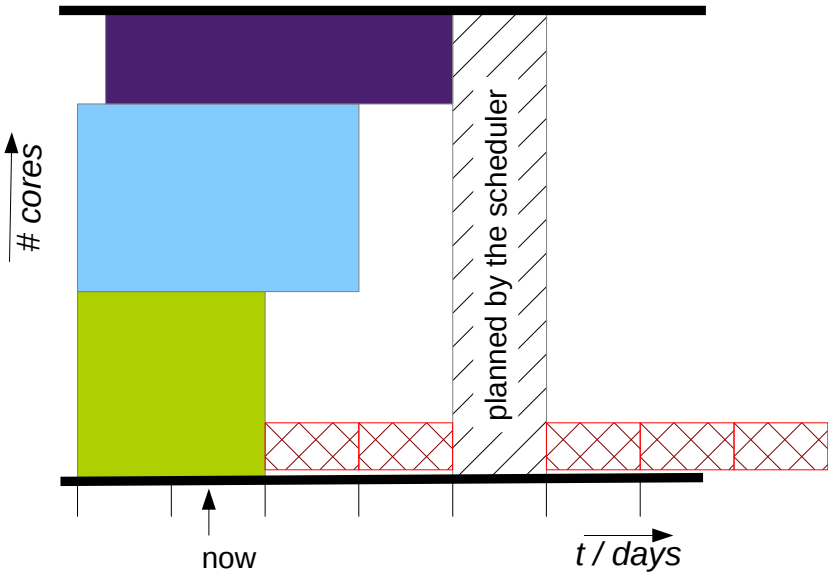
I know my job better:

Backfilling



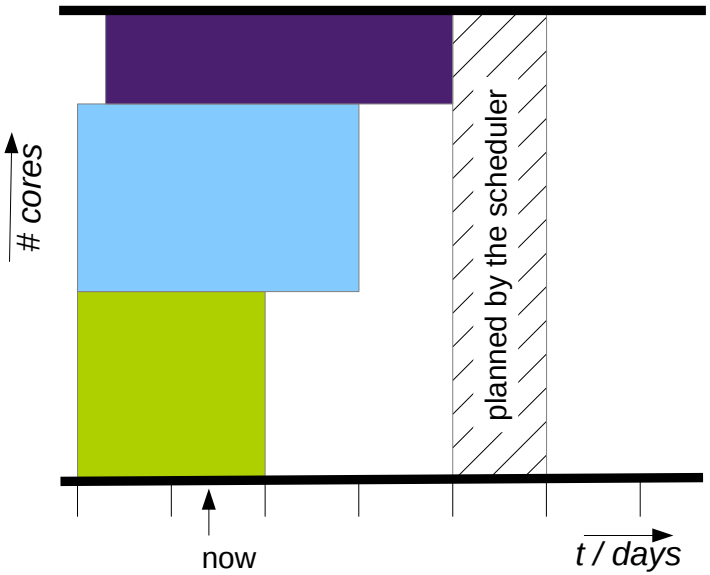
Estimate the maximum run-time of your job!

Backfilling



Try to use shorter jobs!

Backfilling



Allow checkpointing:

Checkpoint / restart

Self-developed code:

- identify best moment to dump “all” data to the file system
- implement data export and import
- implement restart

Commercial or community software

- Check if you can use built-in CR-capabilities of your application:
  (e.g. Abaqus, Amber, Gaussian, GROMACS, LAMMPS, NAMD, NWChem, Quantum Espresso, STAR-CCM+, VASP)

Efficient use of resources

Taurus only!

Make use of heterogeneity of the system

- number of cores per node differ (24, 32, 56, ...)
- memory per core available to the application is less then installed memory (OS needs RAM, too). Stay below the limits to increase the number of potential compute nodes for your job!
- Current numbers for Taurus (as of 2019):
  - 85% of the nodes have 2 GiB RAM per core. Slurm: 1875
  - 10% of the nodes have 4 GiB RAM per core. Slurm: 3995
  - 5% of the nodes have 8 GiB RAM per core. Slurm: 7942
  - 5 large SMP nodes have 56 cores, 2 TiB. Slurm: 36500
  - GPU nodes: 3/2.6 GiB. Slurm: 3000/2538
- AMD Rome nodes (128 cores, 512 GB): 3945
- HPE SDFlex (896 cores, 48 TB): 54006

Let Taurus work!

The batch system (Slurm) manages resources (heterogeneity - Taurus!) and job requirements (cores, RAM, runtime) to optimally use the system.

Normal jobs

- run without interaction (everything prepared in input data and scripts)
- start whenever resources for the particular jobs are available (+ priority)
- can run over hundreds of cores in parallel
- can run as a job array with thousands of independent single core jobs

Run-time considerations

- the larger a system the higher the chance of hitting a problem
- maximum run time: 7 days (today)
- use checkpoint / restart and chain jobs for longer computations
  - controlled by the application
  - controlled by Slurm + additional helper scripts

Nelle's Pipeline III

Let the batch system work... (analyze 1520 files)

```
~/Jellyfish2020 > ls scan_results
spec_0001.out spec_0002.out spec_0003.out spec_0004.out ...
```

Nelle's Pipeline III

Let the batch system work... (analyze 1520 files)

```
~/Jellyfish2020 > ls scan_results
spec_0001.out spec_0002.out spec_0003.out spec_0004.out ...
```

```
#!/bin/bash
#SBATCH -J Jellyfish
#SBATCH --array 1-1520
#SBATCH -o jellyfish-%A_%a.out
#SBATCH -e jellyfish-%A_%a.err
#SBATCH -n 1
#SBATCH -c 1
#SBATCH -p romeo
#SBATCH --mail-type=end
#SBATCH --mail-user=your.name@tu-dresden.de
#SBATCH --time=08:00:00
calc_statistics scan_results/spec_%4a.out
```

Nelle's Pipeline III

Let the batch system work... (analyze 1520 files)

```
~/Jellyfish2020 > ls scan_results
spec_0001.out spec_0002.out spec_0003.out spec_0004.out ...
```

```
#!/bin/bash
#SBATCH -J Jellyfish
#SBATCH --array 1-1520
#SBATCH -o jellyfish-%A_%a.out
#SBATCH -e jellyfish-%A_%a.err
#SBATCH -n 1
#SBATCH -c 1
#SBATCH -p romeo
#SBATCH --mail-type=end
#SBATCH --mail-user=your.name@tu-dresden.de
#SBATCH --time=08:00:00
calc_statistics scan_results/spec_%4a.out
```

```
~/Jellyfish2020 > sbatch jellyfish2020.slurm
```

Working with the Batch System

Interactive jobs

- for pre- or post- processing, compiling and testing / developmemt
- can use terminal or GUI via X11
- several partitions (e.g. `interactive`) are reserved for these jobs.
- "New" clusters come with separate login nodes (same hardware!) that can be used for interactive work.

For rendering applications with GPU support: Nice Desktop Cloud Virtualization (DCV)

- licensed product installed on Taurus
- e.g. rendering with ParaView using GPUs

Remember JupyterHub (`https://compendium.../access/jupyterhub`).

Availability

High utilization - good for “us” - bad for the users?

- short jobs lead to higher fluctuation (limits 1/2/7 days)
- interactive partition is nearly always empty
  - restricted to one job per user
  - default time 30 min, maximum time 8h
- plan resources in advance (publication deadline) - reserve nodes

Agenda

Software development

At `https://compendium.../software/software_development_overview` the following topics are addressed:

- compilers
- mathematical libraries
- debugging
- performance tuning

Available compilers

Which compilers are installed?

- Starting point: `https://compendium.../software/compiler`
- Up-to-date information: `module spider ...`

Available compilers

Which compilers are installed?

- Starting point: `https://compendium.../software/compiler`
- Up-to-date information: `module spider ...`

Which one is “the best”?

- Newer versions are better adapted to modern hardware.
- Newer versions implement more features (e.g. OpenMP, C++, Fortran).
- GNU compilers are most portable.
- Take hints from hardware vendors.

Available compilers

Which compilers are installed?

- Starting point: `https://compendium.../software/compiler`
- Up-to-date information: `module spider ...`

Which one is “the best”?

- Newer versions are better adapted to modern hardware.
- Newer versions implement more features (e.g. OpenMP, C++, Fortran).
- GNU compilers are most portable.
- Take hints from hardware vendors.

→ There is no such thing as “best compiler for all codes”.

Expensive operations

Time consuming operations in scientific computing:

- division, power, trigonometric and exponential functions,
- un-cached memory operations (bandwidth, latency)

Expensive operations

Time consuming operations in scientific computing:

- division, power, trigonometric and exponential functions,
- un-cached memory operations (bandwidth, latency)

How to find performance bottlenecks?

- Tools available at ZIH systems (PIKA, perf, hpctoolkit, Vampir, PAPI counters),
- see `https://compendium.../software/software_development_overview`
- additional courses in performance optimization
- Ask ZIH staff about your performance issues!

Low hanging fruits

What is the needed floating point precision?
32 bit vs. 64 bit impacts on

- memory footprint,
- computing speed.

Low hanging fruits

What is the needed floating point precision?
32 bit vs. 64 bit impacts on

- memory footprint,
- computing speed.

What is the needed floating point accuracy?

- very strict (replicable),
- slightly relaxed (numerical stability),
- very relaxed (aggressive optimizations)

→ see man pages!

Low hanging fruits

What is the needed floating point precision?
32 bit vs. 64 bit impacts on

- memory footprint,
- computing speed.

What is the needed floating point accuracy?

- very strict (replicable),
- slightly relaxed (numerical stability),
- very relaxed (aggressive optimizations)

→ see man pages!

Options for Intel compiler

- Romeo+Taurus: "-axavx" for Haswell and "-mavx2 -fma"
- Barnard+Romeo: "-Ofast -mavx -axCORE-AVX2,CORE-AVX512"

Or compile on the target system (login node).

Agenda

Linux from the command line

HPC Environment at ZIH

Batch System

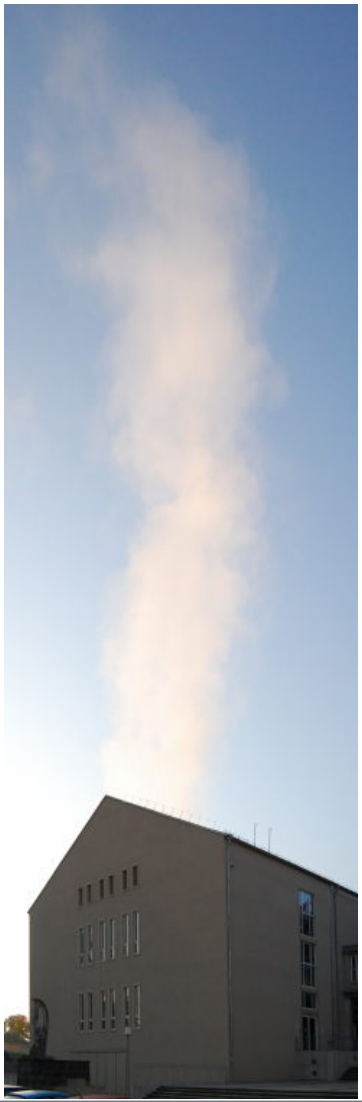
Software Development at ZIH's HPC systems
- Compiling
- Tools

HPC Support

Migration

On HPC systems: Efficient code is essential!

- the same code is running for several 1000 CPUh
- use of multiple CPUs sometimes does not help (wrong parallelization or job placement)
- parallel scalability



Profiling

. . . is a form of *dynamic program analysis*.

Profiling allows you to learn

- . . . where your (?) program has spent its time . . .
- . . . which functions have called which other functions . . .
- . . . how often each function is called . . .

while it was executing.

$\rightarrow$ Identify slow code – redesign it!

Profiling

. . . is a form of *dynamic program analysis*.

Profiling allows you to learn

- . . . where your (?) program has spent its time . . .
- . . . which functions have called which other functions . . .
- . . . how often each function is called . . .

while it was executing.

$\rightarrow$ Identify slow code – redesign it!

Profiling has an impact on performance, but relative performance should be consistent.

Using GNU's gprof

part of GCC available on most unix systems

- compiling and linking (`-pg`):
  ```
  g++ -pg my_prog.cpp -o my_prog
  ```
- execute to produce profiling information:
  ```
  ./my_prog
  ```
- get human readable information:
  ```
  gprof my_prog gmon.out > analysis.txt
  ```
- analysis: `vi analysis.txt`

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 34.70     16.42    16.42        1    16.42    16.42  func3
 33.52     32.29    15.86        1    15.86    15.86  func2
 26.97     45.05    12.76        1    12.76    29.19  func1
  0.13     45.11     0.06                             main
```

PIKA - Analyzing Job Performance

A hardware performance monitoring stack to identify inefficient HPC jobs

- statistics are collected with every job run (available for 14 days)
- a web portal allows eays access to own performance data
- graphs can be discussed with ZIH performance experts

`https://compendium.../software/pika`

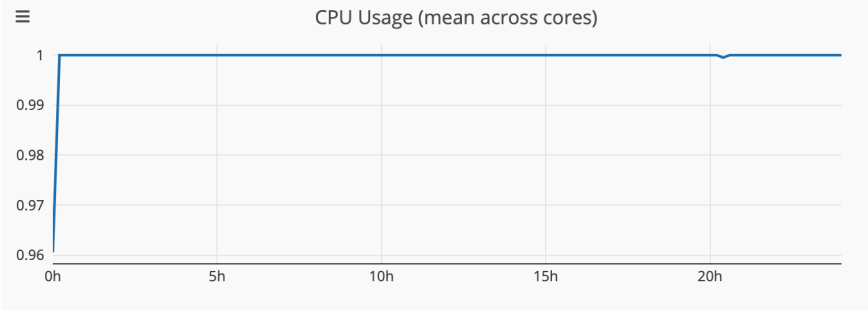
PIKA - Analyzing Job Performance

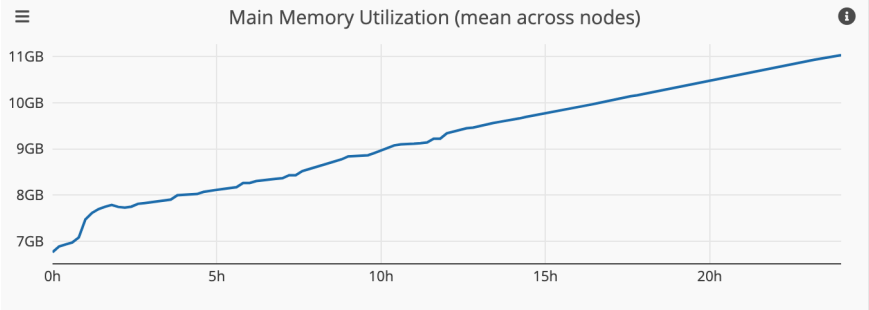
Potential memory leak

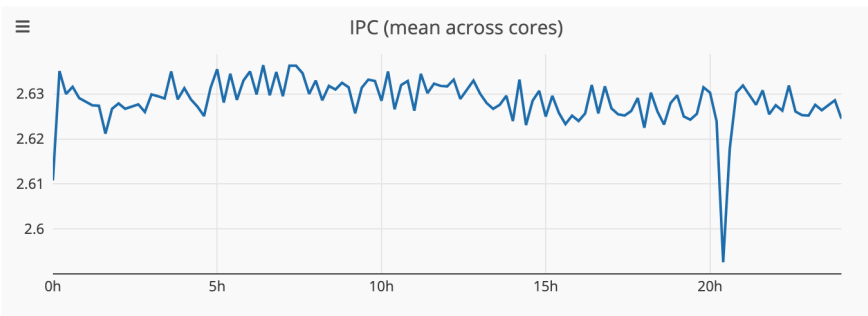
Job Name: **case_2** Job ID: **23482120**

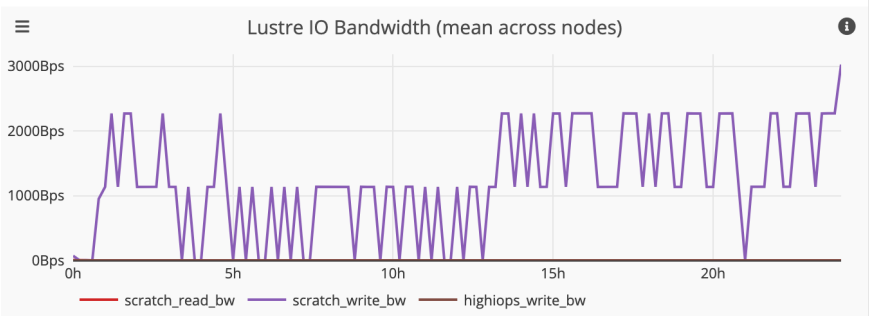
Project	Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
p_nhr_pika	09/02/22 13:03:42	10/02/22 13:04:00	timeout	4	96	0	01d 00:00:00h	00d 00:14:04h	01d 00:00:18h	0000y 096d 00:28h	100.02%	haswell64

compute-bound



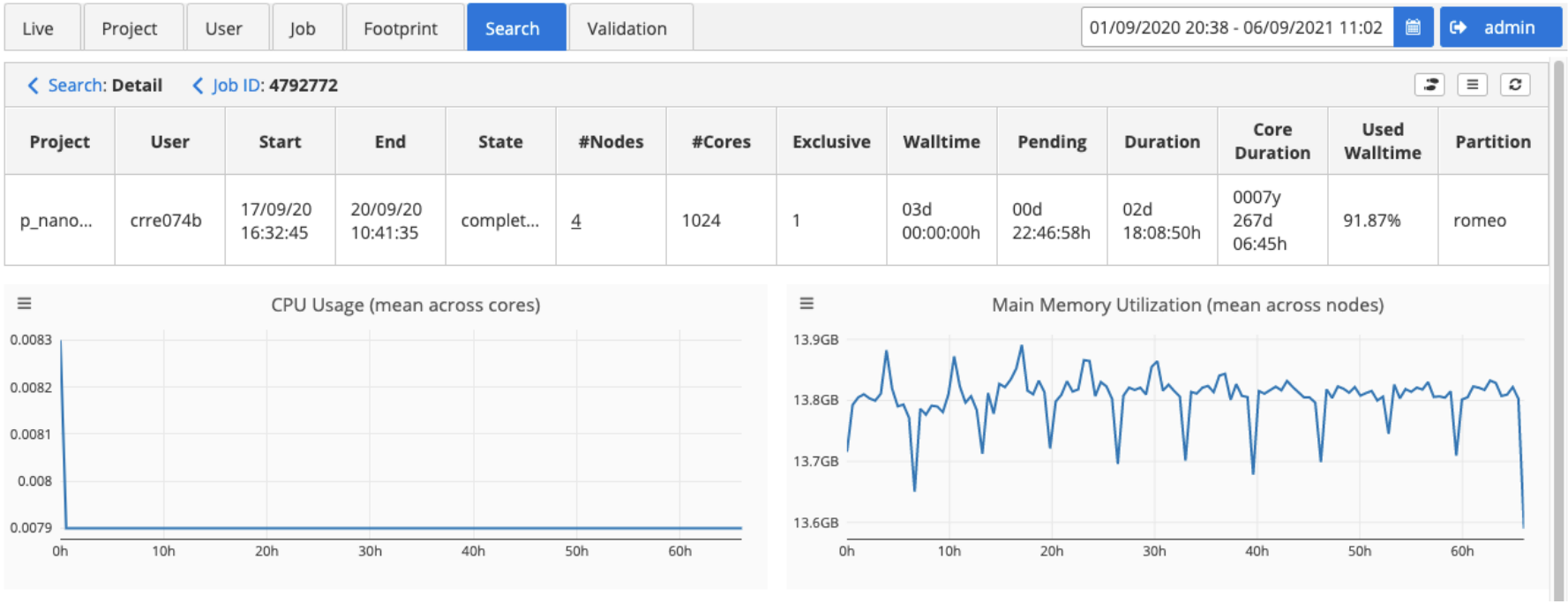






PIKA - Analyzing Job Performance

Low CPU usage

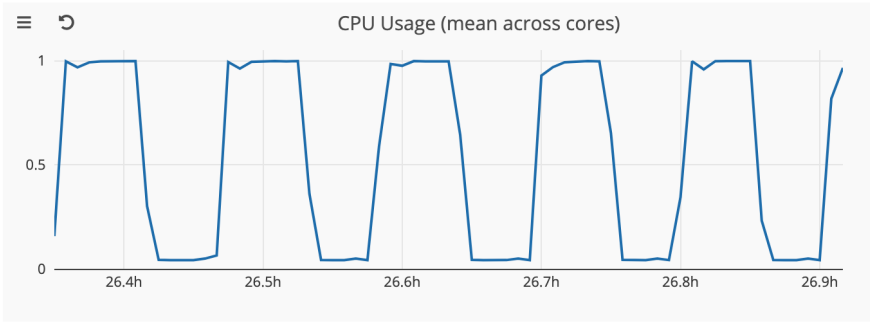


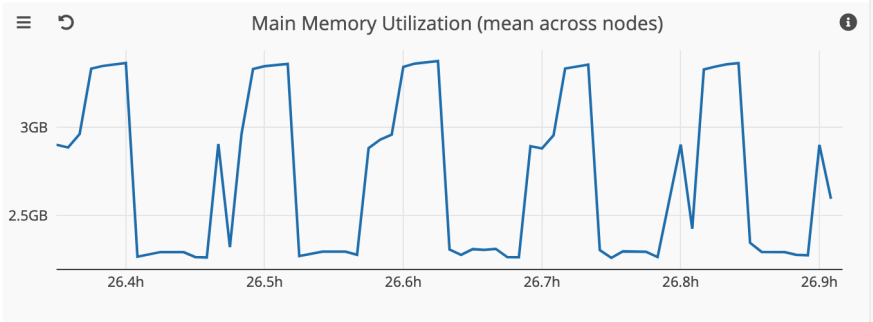
PIKA - Analyzing Job Performance

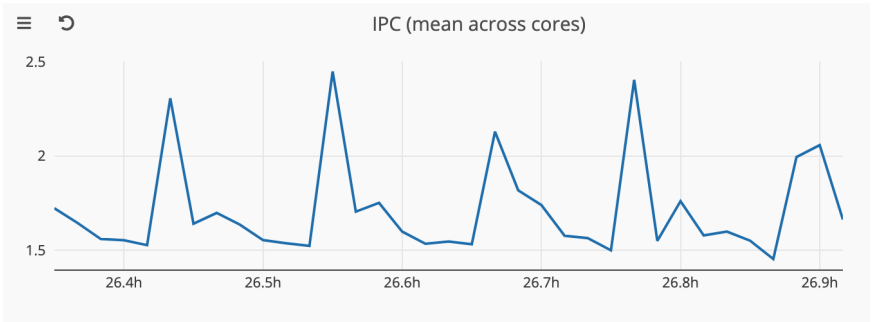
Alternating I/O and compute

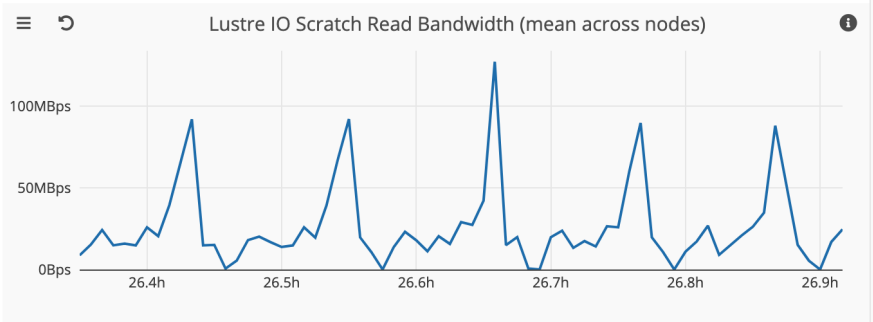
Job Name: **case_3** Job ID: **21087795**

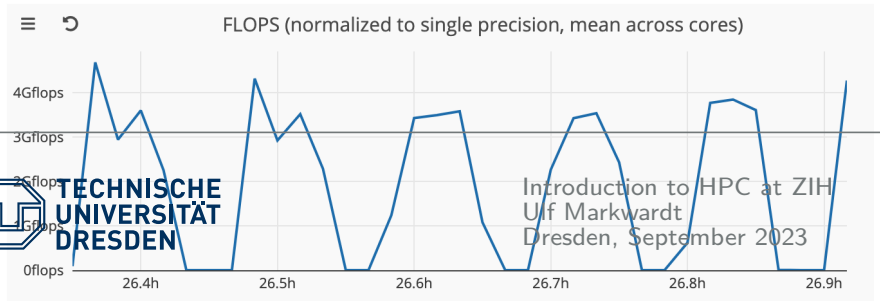
Project	Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
p_nhr_pika	01/12/21 22:33:21	04/12/21 22:33:52	timeout	1	24	0	03d 00:00:00h	06d 06:52:05h	03d 00:00:31h	0000y 072d 00:12h	100.01%	haswell64...

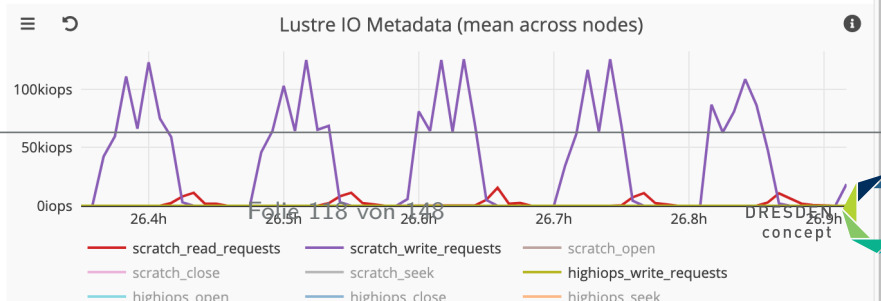












Agenda

Linux from the command line

HPC Environment at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Support
- Management of HPC projects
- Channels of communication
- Kinds of support
- Beyond support

Migration

Start a new project

Two steps for project application:

1. online application form
   - with or without existing ZIH login (select institute)
   - head of the project (universities: chair)
   - needed resources (CPUh per month, permanent disk storage...)
   - abstract

   After a technical review the project will be enabled for testing and benchmarking with up to 41000 CPUh/month.

Start a new project

Two steps for project application:

1. online application form
   - with or without existing ZIH login (select institute)
   - head of the project (universities: chair)
   - needed resources (CPUh per month, permanent disk storage...)
   - abstract
2. full application (3-4 pages pdf):
   - scientific description of the project
   - prelimiary work, state of the art...
   - objectives, used methods
   - software, estimation of needed resources and scalability

Management of HPC projects

Who...

- project leader (normally chair of institute) $\rightarrow$ accountable
- project administrator (needs HPC login) $\rightarrow$ responsible

What...

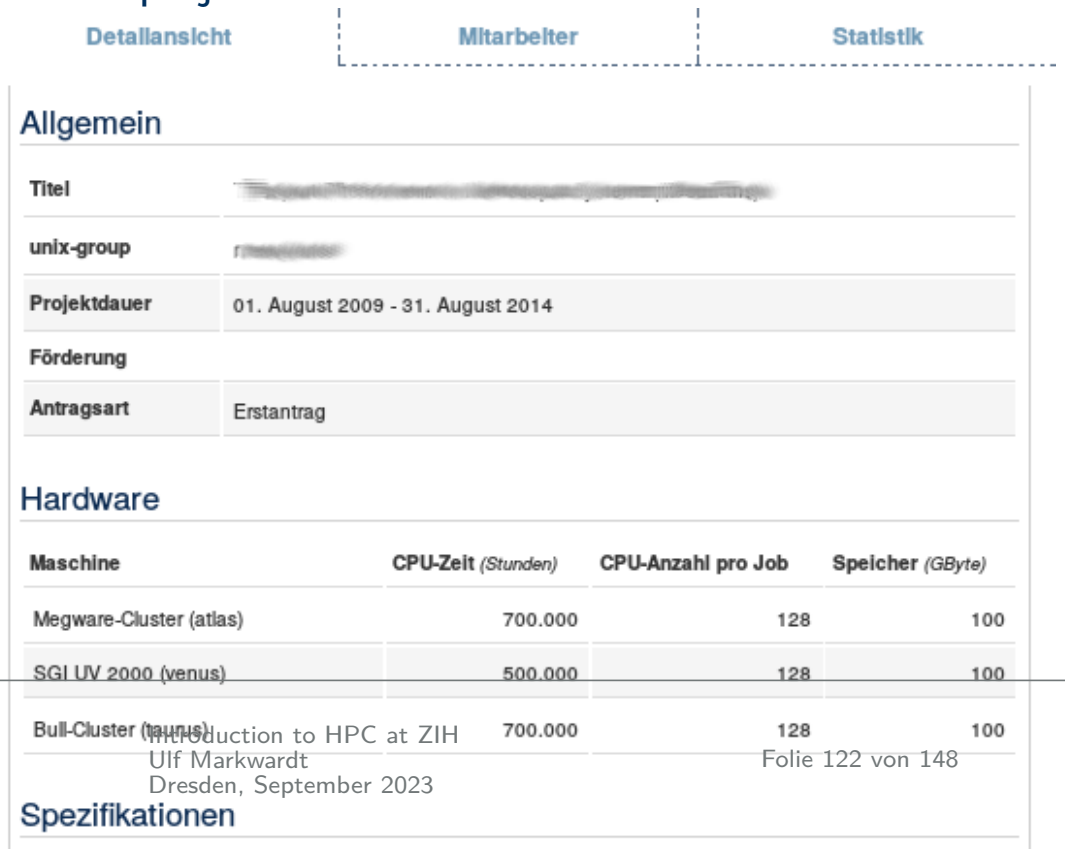
- manage members of the project (add + remove)
  (remark: external users need login..)
- check storage consumption within the project,
- retrieve data of retiring members
- contact for ZIH

Online project management

Web access: `https://hpcprojekte.zih.tu-dresden.de/managers`

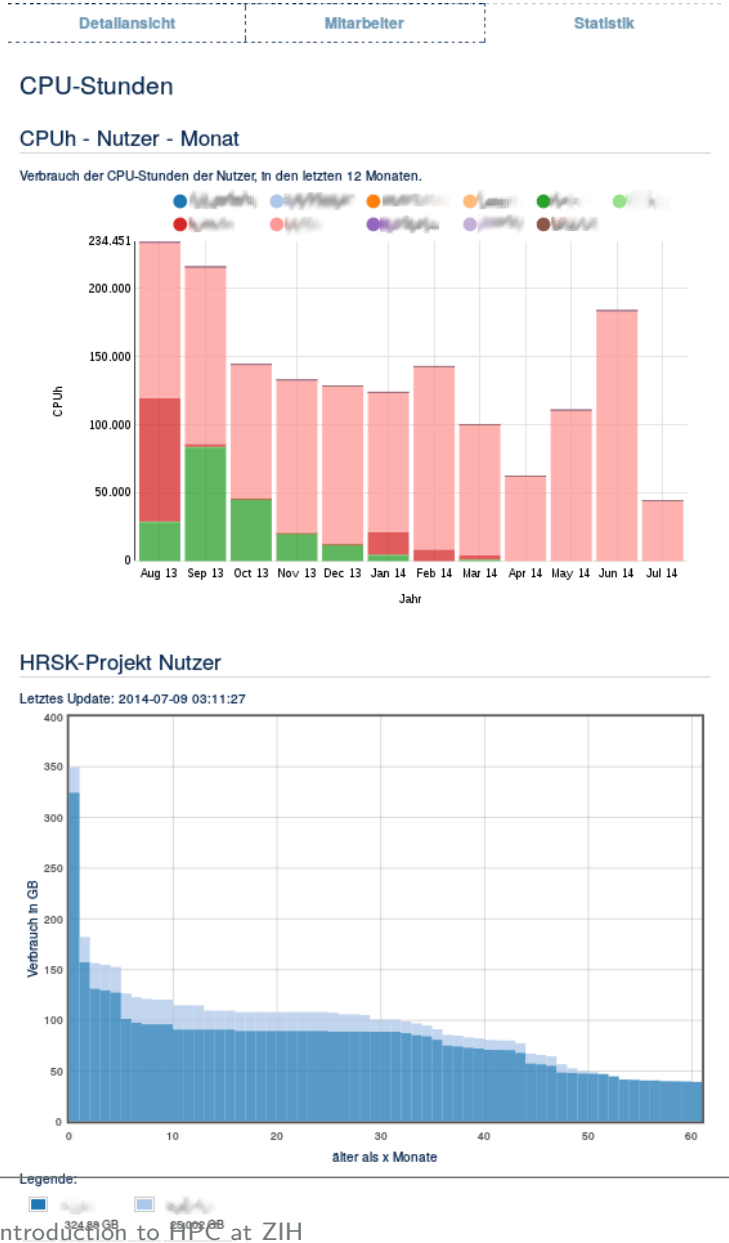
The front-end to the HPC project database enables the project leader and the project administrator to

- add and remove users from the project,
- define a technical administrator,
- view statistics (resource consumption),
- file a new HPC proposal,
- file results of the HPC project.



Online project management

Online project management



Agenda

Linux from the command line

HPC Environment at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Support
- Management of HPC projects
- Channels of communication
- Kinds of support
- Beyond support

Migration

Channels of communication

ZIH $\rightarrow$ users:

- training course "Introduction to HPC at ZIH"
- HPC wiki: `https://compendium.hpc.tu-dresden.de`
  - link to the operation status,
  - knowledge base for all our systems, howtos, tutorials, examples...
- mass notifications per signed email from the sender "[ZIH] HPC Support" to your address `...@mailbox.tu-dresden.de` or `...@tu-dresden.de` for:
  - problems with the HPC systems,
  - new features interesting for all HPC users,
  - training courses
- email, phone - in case of requests or emergencies
  (e.g. user stops the file system).

Channels of communication

User → ZIH

- If the machine feels "completely unavailable" please check the operation status first.
  (Support is notified automatically in case a machine/file system/batch system goes down.)
- Trouble ticket system:
  - advantages
    - reach group of supporters (independent of personal availability),
    - issues are handled according to our internal processes,
  - entry points
    - email: **`servicedesk@tu-dresden.de`** or
      `hpcsupport@zih.tu-dresden.de`
      please: use your **`...@tu-dresden`** address as sender and voluntarily include: name of HPC system, job ID...
    - phone: service desk (0351) 463 40000
    - planned: self service portal
- personal contact
  - phone call, email, talk at the Mensa
  - socializing is fine... but: risk of forgetting

Agenda

Linux from the command line

HPC Environment at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Support
- Management of HPC projects
- Channels of communication
- Kinds of support
- Beyond support

Migration

Kinds of support

HPC management topics:

- HPC project proposal,
- login,
- quota, accounting etc.

HPC usage requests:

- Why does my job not start? - and other questions concerning the batch system
- Why does my job crash?
- How can I ...

Kinds of support

HPC Software questions:

- help with the compiling of a new software
- installation of new applications, libraries, tools
- update to a newer / different version

$\rightarrow$ restrictions of this support:

- only if several user groups need this
- no support for a particular software
- allow for some time

Kinds of support

Performance issues

- joint analysis of a piece of SW
- discussion of performance problems
- detailed inspection of self-developed code
- in the long run: help users to help themselves

Storage and workflow issues

- joint analysis of storage capacity needs
- joint development of a storage strategy
- joint design of workflows

Kinds of support

Scalable Data Services and Solutions – Dresden-Leipzig
ScaDS support for data analytics:

- data analysis tools (parallel R/Python, RStudio, Jupyter, etc.)
- Big Data Frameworks (Apache Hadoop, Spark, Flink, etc.)
- software for Deep Learning (TensorFlow, Keras, etc.)
- survey of performance optimization of the mentioned software

`https://www.scads.de/services` or `services@scads.de`

HPC Support Team

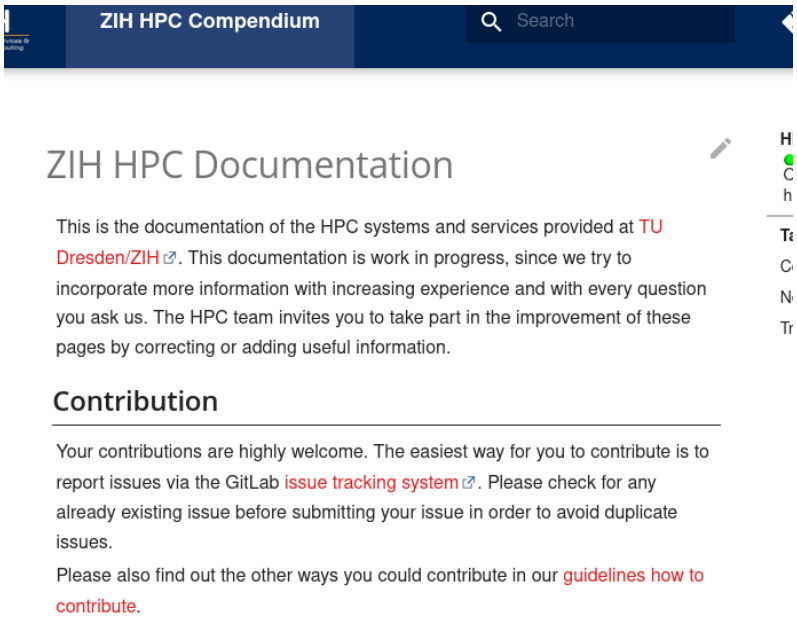
HPC support group

- Anja Gerbes, Claudia Schmidt (project management)
- Matthias Kräußlein (accounting and project infrastructure)
- Guilherme Calandrini, Etienne Keller (technical support)
- Danny Rotscher (Slurm, senior technical support)
- Ulf Markwardt (Slurm, senior technical support... head of the group)

Contribute to HPC Compendium

Help us out. Simply file issues for HPC compendium:

- Point out mistakes or unclear phrasing.
- Contribute with your expert software knowledge to help researchers of your field in the future.



Or open a ticket via `hpcsupport@zih.tu-dresden.de`.

Agenda

Linux from the command line

HPC Environment at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Support
- Management of HPC projects
- Channels of communication
- Kinds of support
- Beyond support

Migration

Beyond support

ZIH is state computing centre for HPC

- hardware funded by DFG and SMWK
- collaboration between (non-IT) scientists and computer scientists
- special focus on data-intensive computing

Joint research projects

- funded by BMBF or BMWi
- ScaDS.AI Dresden Leipzig

We are there to help you with your workflows.

- But not under pressure.
- Should be planned before data come in.

Research topics

Scalable software tools to support the optimization of applications for HPC systems

- Data intensive computing and data life cycle
- Performance and energy efficiency analysis for innovative computer architectures
- Distributed computing and cloud computing
- Data analysis, methods and modeling in life sciences
- Parallel programming, algorithms and methods

You can help

If you plan to publish a paper with results based on the used CPU hours of our machines please acknowledge ZIH like...

The authors gratefully acknowledge the GWK support for funding this project by providing computing time through the Center for Information Services and HPC (ZIH) at TU Dresden.

The authors are grateful to the Center for Information Services and High Performance Computing [Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH)] at TU Dresden for providing its facilities for high throughput calculations.

Recapitulation

Most important topics:

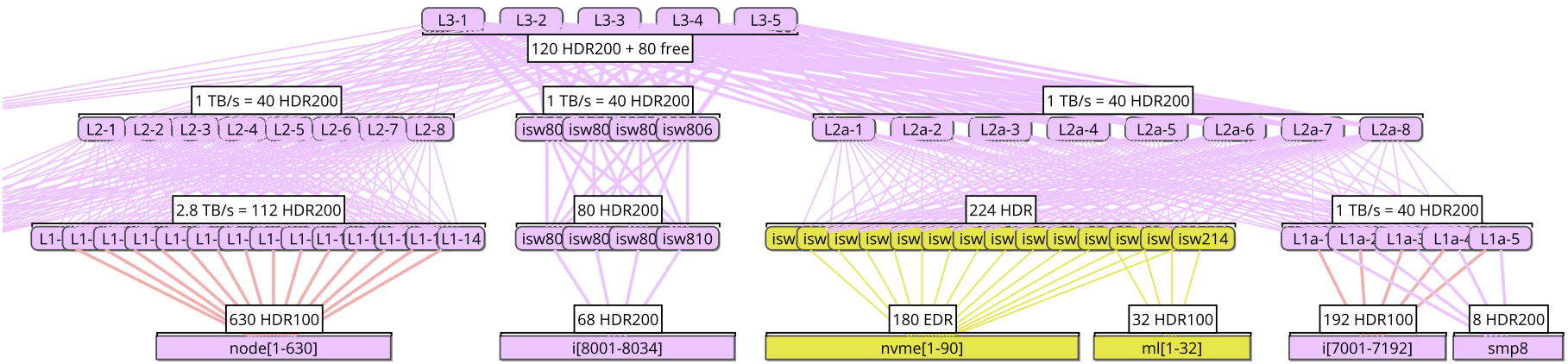
- Use the correct file system.
- Hand over the requirements of your application to the batch system.
- Plan your needed resources (machine and human) in advance.

- You are responsible for your application and your data.
  We can help you.
- Please acknowledge ZIH and send us the publication.

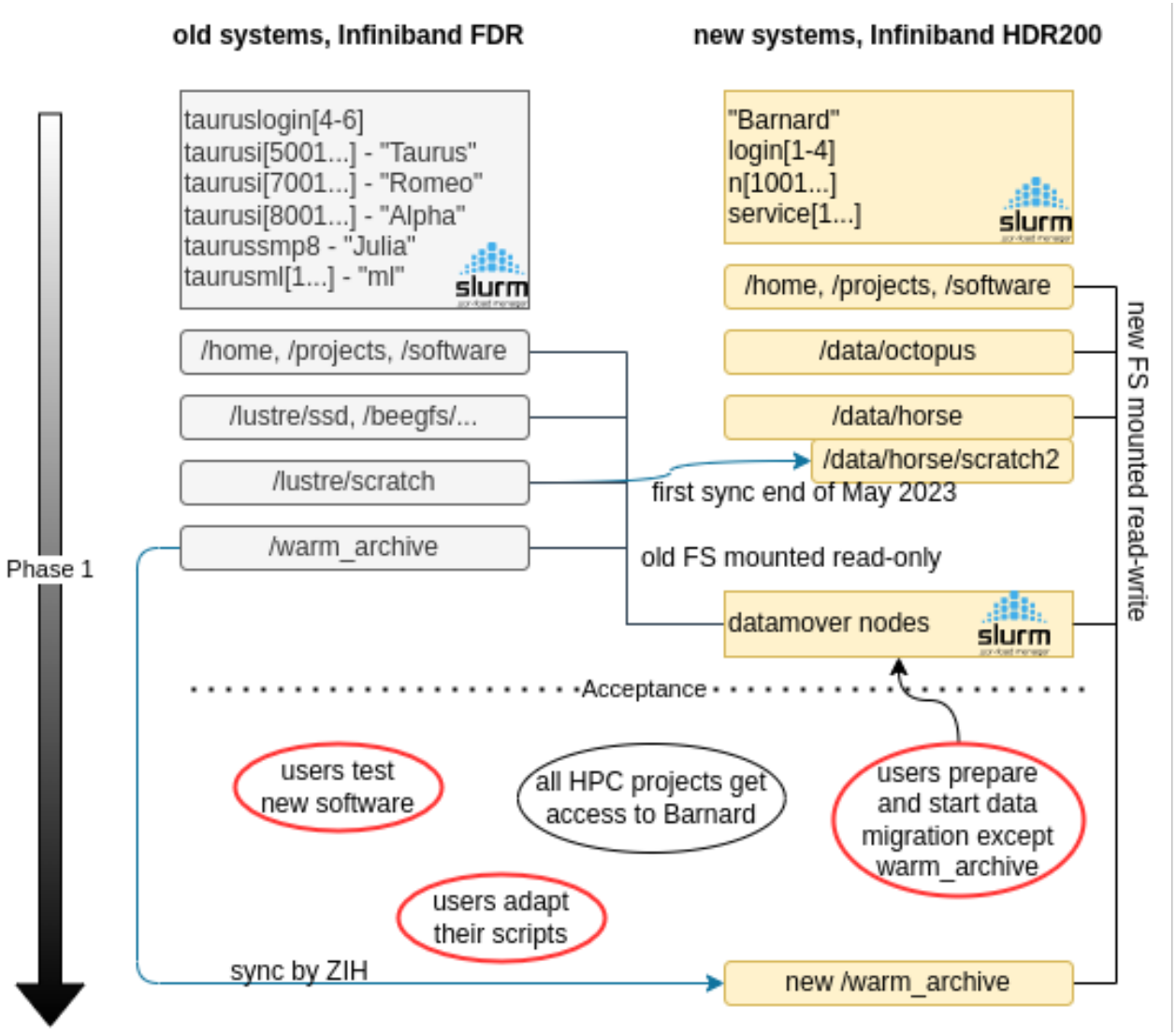
Why?

Main reasons:

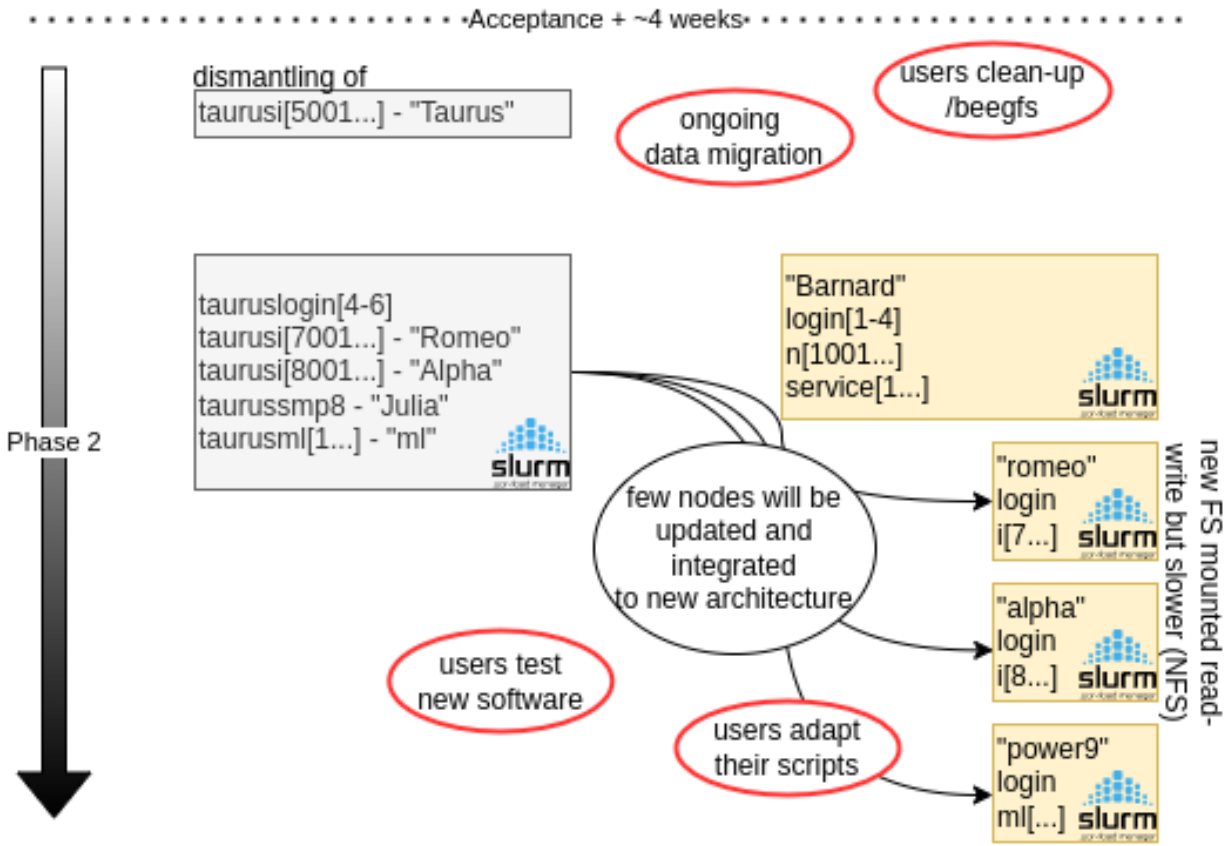
- new storage systems
- HPC architecture changes from heterogeneous cluster/partitions to homogeneous clusters (better to use)
- new IB backbone (from FDR/EDR to HDR/EDR)
- Recable several hundreds IB connections



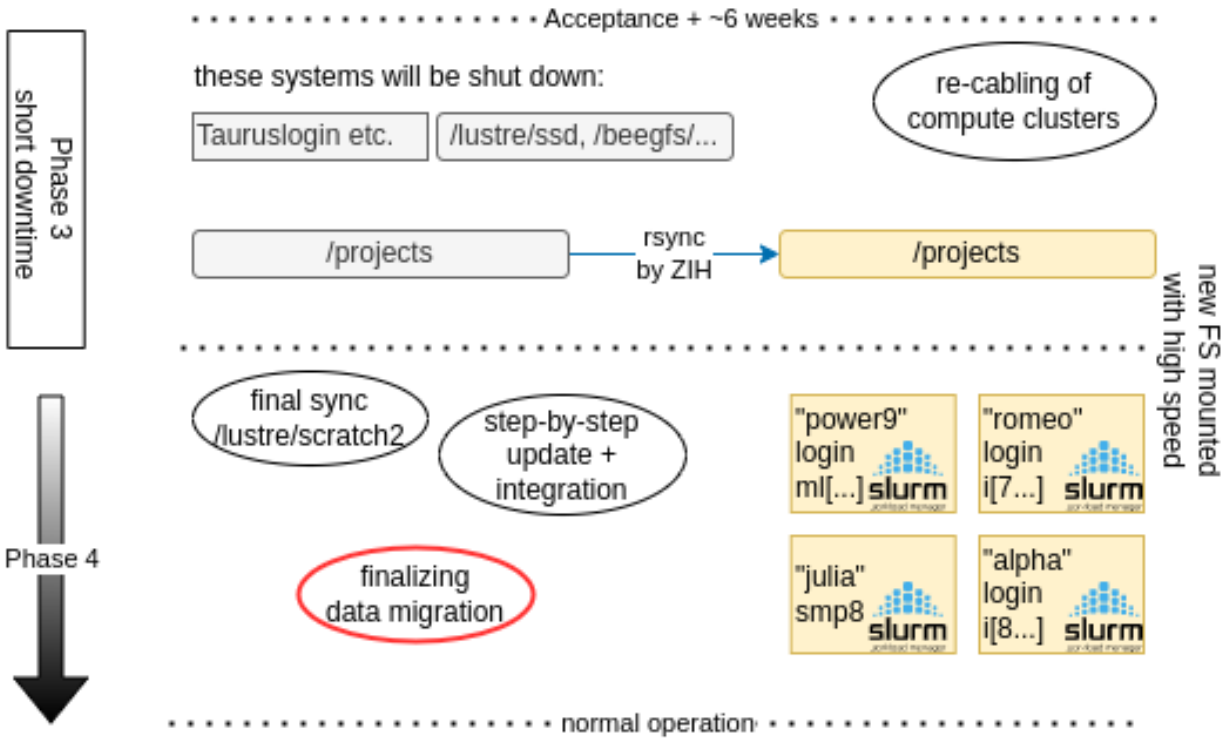
Migration - Phase 1



Migration - Phase 2



Migration - Phase 3+



Datamover cluster

Datamover cluster si still in preparation. Current configuration

```
PARTITION  AVAIL  TIMELIMIT  NODES  STATE NODELIST
datamover*    up   infinite      1   unk* service5
datamover*    up   infinite      4    mix service[1-4]

----------------------------------------------------------------
MACHINE     LOCAL DIR           FILE SYSTEM
Barnard     /home               Lustre
            /projects           NFS
            /data/horse         Lustre
            /data/octopus       Lustre
            /data/walrus        Lustre
            /data/new/projects  Lustre
            /data/old/home      NFS
----------------------------------------------------------------
```

Ongoing:

- second rsync from Quobyte `/warm_archive`
- first rsync from `/projects`
- second rsync from `/projects` - veeery carefully.

Next steps

...for ZIH

- prepare data migration workflows
- acceptance tests
- Slurm changes
  - version update
  - customization according to specific needs (QoS, energy efficiency)
- IB partitions, routing via SkyWay Gateways
- consolidate multi-cluster JupyterHub
- migrate clusters
  - consolidate multi-cluster management
  - new networks (recabling IB + ethernet)
  - update to Rocky 8
  - software tests
- adapt tons of smaller tools (`whypending`, `nodestat`)

Next steps

...for users

- check for missing software (`module spider`)
- clean-up as much as possible in Taurus (all filesystems)
  - reduce number of files
  - evacuate (`/lustre/ssd` and `/beegfs`
- use login nodes to prepare own software (in `$HOME`)

Keep fingers crossed for `/scratch`!

ZIH will provide hints and tools for data migration in time.
Do NOT start before our "go".

And then... TODOs

Data Management

- roll out Weka as high-IOPS file system
- after complete evacuation of the former warm archive: operate these servers as Ceph storage,
- consolidate data mangement workflows, esp. for data archiving

Hardware

- dismantle all old HPC systems
- install and integrate new GPU cluster

Software

- finish migration of all tools and helpers
- build a new HPC software release version based on RHEL 9
- update all systems to RHEL9

Thank you!

This presentation - and much more - can be found at

`https://compendium.hpc.tu-dresden.de`