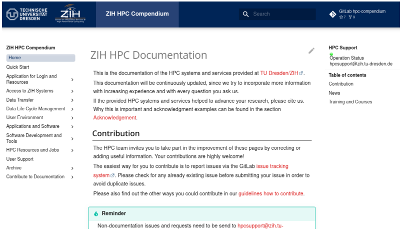
Dr. Ulf Markwardt, Dr. Natalie Breidenbach, Noah Löwer, Yan Ge (PhD)
hpc-support@tu-dresden.de

Introduction to HPC at ZIH

Dresden, May 2025

Main steps of working on HPC systems

Main steps and ...	the ZIH wiki has the answer
↓ **Project application**	`https://doc.../application/overview/`
↓ **Data management plan**	`https://doc.../data_lifecycle/overview/`
↓ **Access HPC system**	`https://doc.../access/overview/`
↓ **HPC Environment**	`https://doc.../software/overview/`
↓ **Transfer data**	`https://doc.../data_transfer/overview/`
↓ **Start a job**	`https://doc.../jobs_and_resources/overview/`
↓ **Problem solutions**	`https://doc.../support/support/`



ZIH HPC Documentation

This is the documentation of the HPC systems and services provided at TU Dresden/ZIH.

This documentation will be continuously updated, since we try to incorporate more information with increasing experience and with every question you ask us.

If the provided HPC systems and services helped to advance your research, please cite us. Why this is important and acknowledgment examples can be found in the section Acknowledgement.

Contribution

The HPC team invites you to take part in the improvement of these pages by correcting or adding useful information. Your contributions are highly welcome!

The easiest way for you to contribute is to report issues via the GitLab issue tracking system. Please check for any already existing issue before submitting your issue in order to avoid duplicate issues.

Please also find out the other ways you could contribute in our guidelines how to contribute.

Reminder

Non-documentation issues and requests need to be send to hpcsupport@zih.tu-

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware
- HPC file systems
- Software environment at ZIH

Batch System
- General
- Slurm examples

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

HPC Projects
- Project application
- Management of HPC projects

HPC Support
- Channels of communication
- Support types and topics

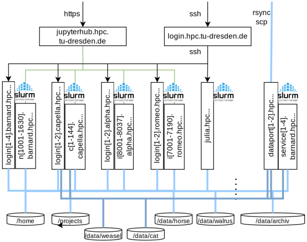
General HPC system

- High performance computing (HPC) systems are optimal research environment, for data analytics, artificial intelligence and machine learning applications. In particular perfect for highly scalable, data-intensive and compute-intensive applications
- capabilities for energy measurement and performance monitoring
- HPC system is based on Linux
- various distributions for all purposes (OpenSuSE, SLES, Ubuntu, Debian, Fedora, RedHat,. . .) `http://www.distrowatch.com`

Overview HPC system TU Dresden

- consists of five homogeneous clusters: **Barnard, Alpha Centauri, Capella, Romeo, Julia**
- each cluster has their own Slurm instance
- they share one file system (/home, /projects, /data/horse, /data/walrus, ...)
- possible to switch easily between components
- PIKA to monitor your slurm jobs

More details about the infrastructure later during this presentation



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
`LD_LIBRARY_PATH`	search path for dynamic libraries
`HOME`	path to user's home directory

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

Basic commands I

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 empty 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 II

`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 marie@login2.barnard.hpc.tu-dresden.de`)

Basic commands II

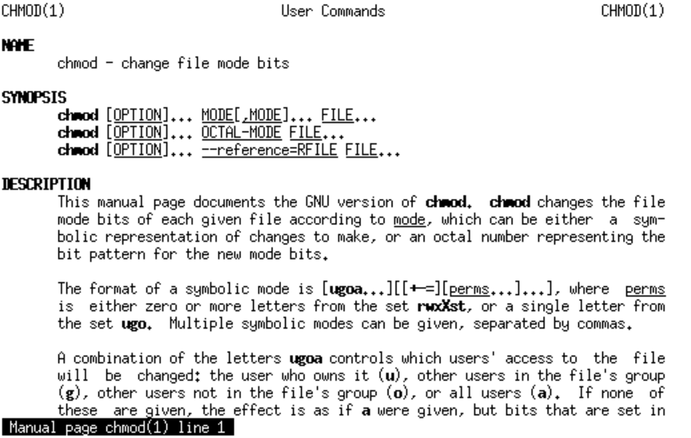
`echo`	display text to stdout `echo $PATH`
`cat`	display contents of a file `cat > newfile.txt`
`less`, `more`	pagewise display (`less README`)
`grep`	search for words/text (`grep result out.res`)
`file`	determine type of a file
`ps`	display running processes (`ps -axuf`)
`kill`	kill a process (`kill -9 12813`)
`top`	display table of processes (interactive per default)
`ssh`	secure shell to a remote machine (`ssh -X marie@login2.barnard.hpc.tu-dresden.de`)

Editors:

- `vi/vim` - a cryptic, non-intuitive, powerful, universal editor. The web has several “cheat sheets” of vi/vim.
- `emacs` - a cryptic, non-intuitive, powerful, universal editor. But it comes with an X11 GUI.
- `[g,m,n,x]edit` - inituitve editors with an X11 GUI.

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**, don't use spaces!
- 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/h9/marie`

Special directories:

- $\sim$ = home directory (`cd ~` 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
```

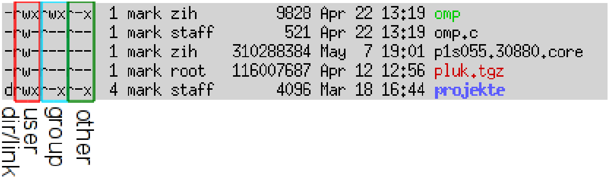
```
~/Jellyfish2020 > for f in scan_results/* ; do \
    calc_statistics $f ; done
```

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 .`



```
-rwxrwxr-x  1 mark zih        9828 Apr 22 13:19 omp
-rw-r--r--  1 mark staff       521 Apr 22 13:19 omp.c
-rw-------  1 mark zih   310288384 May  7 19:01 p1s055.30880.core
-rw-r--r--  1 mark root  116007687 Apr 12 12:56 pluk.tgz
drwxr-xr-x  4 mark staff      4096 Mar 18 16:44 projekte
```

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

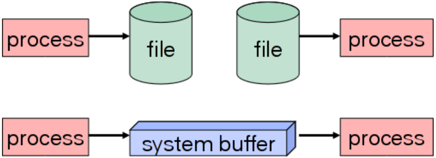
Linux is a text-oriented operating system. Input and output is 'streamable'.

- 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?

Tools for access

Detailed information in the ZIH wiki (see `https://doc.../access/ssh_login`)

Linux:

- command line
- GUI JupyterHub

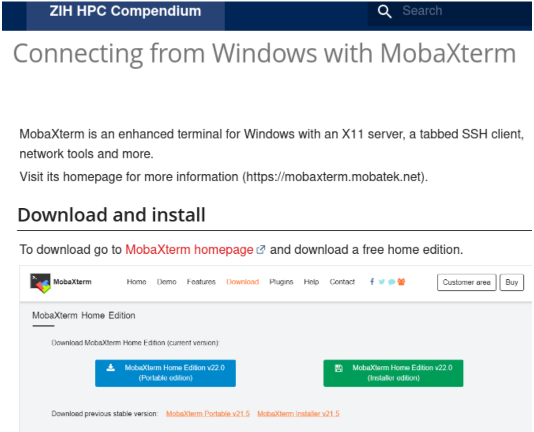
Windows:

- command line login: PuTTY, Secure Shell (SSH)
- file transfer: WinSCP, Secure Shell
- GUI transfer (XMing, XMing-Mesa, X-Win32)
- integrated solution: MobaXterm
- GUI JupyterHub

```
#login via terminal (Linux) or 'Windows PowerShell' (Windows)
     to e.g. cluster Barnard
marie@local$ ssh marie@login2.barnard.hpc.tu-dresden.de
#login with X11 fowarding to e.g. cluster Romeo
marie@local$ ssh -X marie@login1.romeo.hpc.tu-dresden.de
```

MobaXterm Installation

see our Compendium for step by step instructions `https://doc.../access/ssh_mobaxterm`



Connecting from Windows with MobaXterm

MobaXterm is an enhanced terminal for Windows with an X11 server, a tabbed SSH client, network tools and more.

Visit its homepage for more information (https://mobaxterm.mobatek.net).

Download and install

To download go to MobaXterm homepage and download a free home edition.

MobaXterm Home Edition

Download MobaXterm Home Edition (current version):

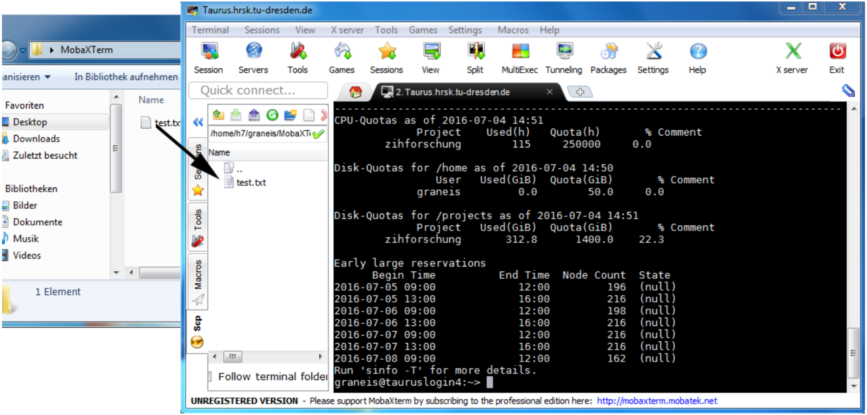
MobaXterm Home Edition v22.0 (Portable edition)
MobaXterm Home Edition v22.0 (Installer edition)

Download previous stable version: MobaXterm Portable v21.5 MobaXterm Installer v21.5

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

MobaXterm

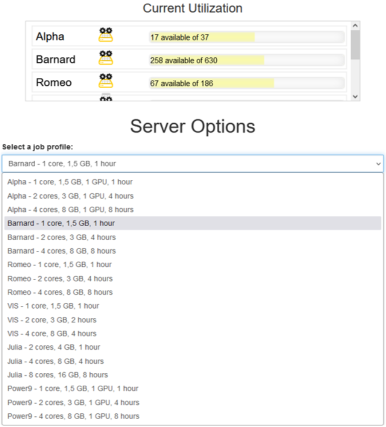
- console to HPC systems (including X11 forwarding)
- transfer files to and from the HPC systems
- browse through the HPC file systems



```
-------------------------------------------------------------------------------
CPU-Quotas as of 2016-07-04 14:51
             Project     Used(h)    Quota(h)       % Comment
        zihforschung         115      250000     0.0

Disk-Quotas for /home as of 2016-07-04 14:50
                User    Used(GiB)  Quota(GiB)       % Comment
             graneis          0.0        50.0     0.0

Disk-Quotas for /projects as of 2016-07-04 14:51
             Project    Used(GiB)  Quota(GiB)       % Comment
        zihforschung        312.8      1400.0    22.3

Early large reservations
      Begin Time           End Time  Node Count  State
2016-07-05 09:00              12:00         196  (null)
2016-07-05 13:00              16:00         216  (null)
2016-07-06 09:00              12:00         198  (null)
2016-07-06 13:00              16:00         216  (null)
2016-07-07 09:00              12:00         216  (null)
2016-07-07 13:00              16:00         216  (null)
2016-07-08 09:00              12:00         162  (null)
Run 'sinfo -T' for more details.
graneis@tauruslogin4:~>
```

Access to HPC via JupyterHub

Use a GUI from your Web browser → JupyterHub.
https://jupyterhub.hpc.tu-dresden.de/



Current Utilization

Alpha	17 available of 37
Barnard	258 available of 630
Romeo	67 available of 186

Server Options

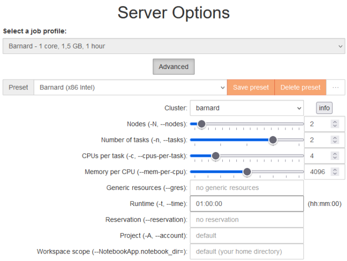
Select a job profile:

Barnard - 1 core, 1,5 GB, 1 hour

- Alpha - 1 core, 1,5 GB, 1 GPU, 1 hour
- Alpha - 2 cores, 3 GB, 1 GPU, 4 hours
- Alpha - 4 cores, 8 GB, 1 GPU, 8 hours
- Barnard - 1 core, 1,5 GB, 1 hour
- Barnard - 2 cores, 3 GB, 4 hours
- Barnard - 4 cores, 8 GB, 8 hours
- Romeo - 1 core, 1,5 GB, 1 hour
- Romeo - 2 cores, 3 GB, 4 hours
- Romeo - 4 cores, 8 GB, 8 hours
- VIS - 1 core, 1,5 GB, 1 hour
- VIS - 2 core, 3 GB, 2 hours
- VIS - 4 core, 8 GB, 4 hours
- Julia - 2 cores, 4 GB, 1 hour
- Julia - 4 cores, 8 GB, 4 hours
- Julia - 8 cores, 16 GB, 8 hours
- Power9 - 1 core, 1,5 GB, 1 GPU, 1 hour
- Power9 - 2 cores, 3 GB, 1 GPU, 4 hours
- Power9 - 4 cores, 8 GB, 1 GPU, 8 hours

Detailed documentation can be found at `https://doc.../access/jupyterhub` .

Advanced options JupyterHub



Server Options

Select a job profile: Barnard - 1 core, 1,5 GB, 1 hour

Preset: Barnard (x86 Intel)

Cluster: barnard

Nodes (-N, --nodes): 2

Number of tasks (-n, --tasks): 2

CPUs per task (-c, --cpus-per-task): 4

Memory per CPU (--mem-per-cpu): 4096

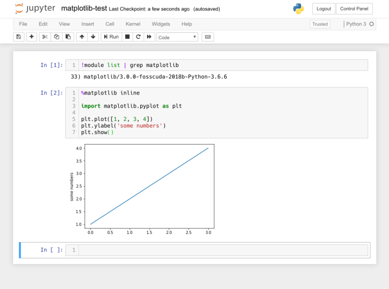
Generic resources (--gres): no generic resources

Runtime (-t, --time): 01:00:00 (hh:mm:00)

Reservation (--reservation): no reservation

Project (-A, --account): default

Workspace scope (--NotebookApp.notebook_dir=): default (your home directory)



jupyter matplotlib-test Last Checkpoint: a few seconds ago (autosaved)

In [1]:

```
!module list | grep matplotlib
```

```
33) matplotlib/3.0.0-fosscuda-2018b-Python-3.6.6
```

In [2]:

```
%matplotlib inline

import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

Detailed documentation can be found at `https://doc.../access/jupyterhub` .

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

Starting from May 30 2025, the general VPN access will only be possible via the eduVPN software:

- installation at your local machine required: `https://tu-dresden.de/.../vpn`
- Start eduVPN tool
  - Select Institute: Technische Universität Dresden
  - authentication via web browser (session expires automatically after 15h)
  - select (any) VPN profile

Virtual machines

VM's are not required to access the HPC system but may be helpful for Windows users for preprocessing , e.g. building a singularity container.

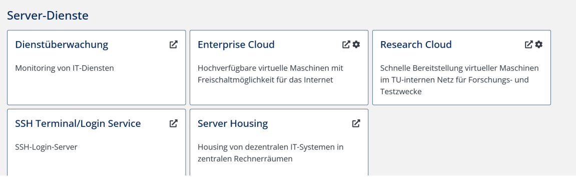
TU Dresden provides virtual machines with Linux and Windows OS

- Research Cloud
  for testing purposes ready to use in approximately 15 minutes
- Enterprise Cloud
  with individual configuration (more resources) takes a several days to activate

`https://tu-dresden.de/zih/dienste/service-katalog/zusammenarbeiten-und-forschen/server_hosting`

Using a virtual machine I

- create a virtual machine in your Self Service Portal



Server-Dienste

Dienstüberwachung
Monitoring von IT-Diensten

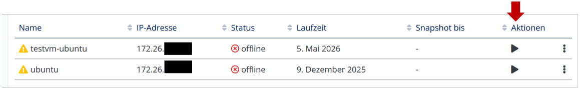
Enterprise Cloud
Hochverfügbare virtuelle Maschinen mit Freischaltmöglichkeit für das Internet

Research Cloud
Schnelle Bereitstellung virtueller Maschinen im TU-internen Netz für Forschungs- und Testzwecke

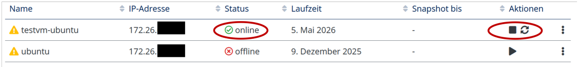
SSH Terminal/Login Service
SSH-Login-Server

Server Housing
Housing von dezentralen IT-Systemen in zentralen Rechnerräumen

- start the VM



Name	IP-Adresse	Status	Laufzeit	Snapshot bis	Aktionen	
⚠ testvm-ubuntu	172.26.[redacted]	⊗ offline	5. Mai 2026	-	▶	⋮
⚠ ubuntu	172.26.[redacted]	⊗ offline	9. Dezember 2025	-	▶	⋮

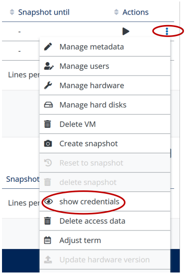


Name	IP-Adresse	Status	Laufzeit	Snapshot bis	Aktionen	
⚠ testvm-ubuntu	172.26.[redacted]	✓ online	5. Mai 2026	-	■ ⟳	⋮
⚠ ubuntu	172.26.[redacted]	⊗ offline	9. Dezember 2025	-	▶	⋮

Using a virtual machine II

connect via ssh

```
# Open a terminal or 'Windows PowerShell'. add the IP address
      of the VM
marie@local$ ssh service@IP
# Enter password of the virtual machine
service@VM-name$
# start your work. e.g. login to Barnard cluster
service@VM-name$ ssh marie@login2.barnard.hpc.tu-dresden.de
```



Use what you have learned so far

- login to the Romeo cluster using command line, Windows PowerShell or Mobaxterm. If you want to use the command line but have a Windows OS, you may use the command line in a virtual machine.
- list the content of your /home directory
- create a file named test-script.sh and write 'hello' inside
- open the file with an editor and change the text, save and close the file.

```
# login to Romeo
marie@local$ ssh marie@login2.romeo.hpc.tu-dresden.de
# list the files in your home directory
marie@romeo$ ls
# create the file test-script.sh with hello
marie@romeo$ echo hello >> test-script.sh
```

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 mostly use (highest CPUh consumption)?

A commercial software

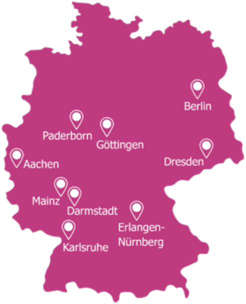
B community software

C "self" developed codes

Nationales Hochleistungsrechnen - NHR

What is National HPC?

- political action to support research at universities with strong HPC infrastructure
- structural federal and state funding for infrastructure, hardware, staffing
- better networking between the 9 NHR centers
- collaboration on technical and organisational aspects (e.g. JARDS)



NHR@TUD

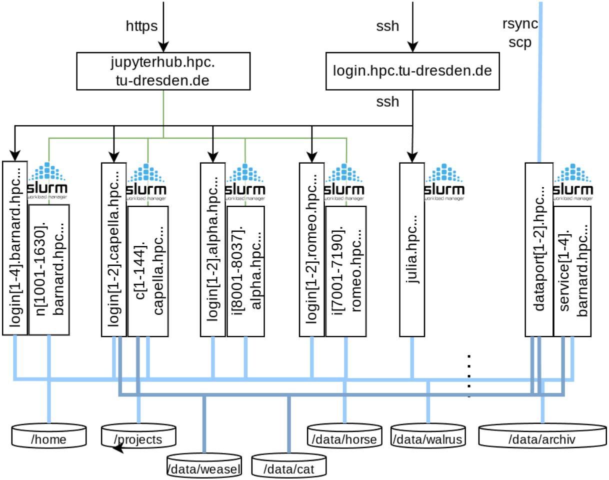
- Main focus: life sciences and earth system science
- Methodological focus:
  - Methods for Big Data, data analysis and data management
  - Machine Learning
  - Tiered storage architectures and I/O optimization
  - Performance and energy efficiency analysis and optimization.

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
  - GPU clusters (Alpha, Power9)
  - compute cluster (Romeo)
  - large SMP system (Julia)
  - NVMe nodes (fast storage over Infiniband),
- new methods to access systems complementary to “classical” HPC mode
- large team for AI related research and support

Infrastructure Overview



- shared parallel file systems with high bandwidth
- separate Slurm batch systems

Barnard

Barnard

General Purpose Cluster (Bull)

Subdomain: barnard.hpc.tu-dresden.de
OS: RedHat 8.9

- Compute nodes: n[1001-1720]
  - 2x52 Cores Intel Sapphire Rapids
  - 512 GB RAM (90 nodes: 1 TB)
  - **diskless** (12 nodes: 1.8 TB)
- Login nodes: login[1-4]
  - 2x52 Cores Intel Sapphire Rapids
  - 1 TB RAM, 1.9 TB NVMe
- Visualization nodes: vis[1-4]
  - 2x52 Cores Intel Sapphire Rapids
  - 1 TB RAM, 1.2 TB NVMe
  - 2x Nvidia A40

More information on `https://doc.../jobs_and_resources/hardware_overview/barnard`

Romeo

General Purpose Cluster (NEC)

Subdomain: romeo.hpc.tu-dresden.de
OS: RockyLinux 8.9

- 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://doc.../jobs_and_resources/romeo`

Julia

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

Hostname: julia.hpc.tu-dresden.de
OS: RockyLinux 8.9

- 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`

More information on `https://doc.../jobs_and_resources/julia`
At the moment 1/4 of the system is used by DZA.

Alpha Centauri

ScaDS Cluster for Data Analysis and AI (NEC)

Subdomain: alpha.hpc.tu-dresden.de
OS: RockyLinux 8.9

- Compute nodes: n[8001-8037]
  - 8 x NVIDIA A 100-SXM4, 40GB VRAM
  - 2 x AMD EPYC CPU 7352
  - 1 TB RAM
  - 3.5 TB local NVMe
- login nodes: login[1-2]
  - same configuration as compute nodes

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

Capella

GPU Cluster for AI and ML (Megware)

OS: AlmaLinux 9.4

- Compute nodes: c[1-156]
  - 4 x NVIDIA H100-SXM5 Tensor Core-GPUs, 94 GB VRAM
  - 2 x AMD EPYC CPU 9334 (32 cores)
  - 768 GB RAM (8 nodes: 1.5 TB)
  - 800 GB local NVMe
- login nodes: login[1-2]
  - no GPU available
  - constrained part of nodes c[1,2] : interactive partition with 28 (smaller) GPUs each

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

- HPC file systems
- Software environment at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Projects

HPC Support

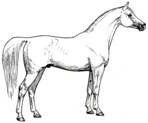
HPC Filesystems Overview

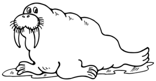
Properties of file systems:

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

HPC Filesystems Overview

- local SSD/NVMe/RAM `/tmp`
- HPC global `/projects`, `/home`
- HPC global scratch `/data/horse`
- high capacity storage `/data/walrus`
- fast IOPS for Capella `/data/cat`
- fast IOPS for Alpha Centauri `/data/weasel`





For archiving outside HPC:

- 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

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

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

Mounted at `/tmp`

High-IOPS file systems

powered by WEKAio

Capella: 2 PB at `/data/cat`

Alpha Centauri: 800 TB at `/data/weasel`

Fastest parallel file systems (IOPS) at the GPU clusters:

- large parallel file system for high number of I/O operations,
- management via workspaces,
- data may be deleted after 30 days,
- **cluster-specific** file system; use datamovers and dataports

Attention Data might get lost.

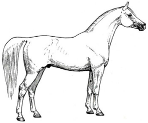
Scratch file system

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

- 18 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 NFS

- Deleted files are accessible via the snapshots (available via ticket)
- Paths to permanent storage are
  - `/home/<login>` and
  - `/projects/<projectname>`

  with different access rights (cf. Terms of Use).
- Exeeding the quota automatically blocks job submits. (Updating nightly.)
- All HPC systems of ZIH share these file systems.
- Daily tape backups.

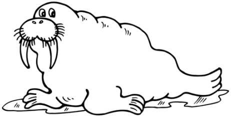
Do not use as work directories.

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`



Data management

Users are responsible for their data.
We highly recommend to plan data management prior project start, considering:

- set of software
  - different levels: modules, containers, Python virtual environments
- data life cycle management, including data storage
  - folder structure, naming the data
  - data organization depending on type of data and analysis
  - organization using workspaces
  - create README file to provide an overview of the project and software
  - backup of important data to a safe location
- access rights to project data

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.

see `https://doc.../data_lifecycle/overview`

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.

Workspaces

Tool for users to manage their storage demands

`https://doc.../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.

Attention: No backups are generated from Workspaces

Maximum initial lifetime depends on file system:

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

Workspace

Expiration of workspaces

- expired workspaces are moved automatically to another location
- after a certain time (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!

Workspace - examples

Available filesystems

```
marie@login$ ws_find -l
available filesystems:
horse
walrus
```

Allocation

```
marie@login$ ws_allocate -F walrus specimen 20
Info: creating workspace.
```

Notification:

```
marie@login$ 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

```
marie@login$ 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
```

Caution

Has the potential to cause file system failure if workspaces are not deleted after job (e.g. jobs run into timeout)!

Storage utilization

The `show_resources` command on login nodes gives an overview of storage utilization:

```
marie@login$ show_resources

Workspaces
-----------------
Own data in workspaces                  files       size    last access
    /data/horse/ws/marie-number_crunch    165   576.09 kB    13 days ago
   ==> Total:                             165   576.09 kB

Project data in workspaces
                       filesystem          files        size
   p_number_crunch     horse             767,125     5.47 TB
   p_number_crunch     walrus                 72   203.14 GB

Permanant Storage
-----------------
Own data in /home:   221.36 MB in 6,216 files ( 0.4% of   50.00 GB
     limit)

Projects                         files       size       limit       %
   p_number_crunch             148,962  250.06 GB   322.12 GB    77.6
```

See `https://doc.hpc...` .

Internal data transfer

Datamover are special nodes, which 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.
- Additional commands: dtinfo, dtqueue.

Very simple usage like

```
dttar -czf /walrus/ws/jellyfish-2020/results_20190820.tgz \
           /horse/ws/jellyfish-2020/results
```

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

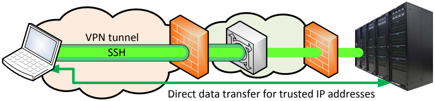
External data transfer

The nodes dataport[1,2].hpc.tu-dresden.de allow access with high bandwidth bypassing firewalls to transfer data from your local machine to the HPC file system

Restrictions are: trusted IP addresses only and using the protocols: sftp, rsync

Examples using rsync, assuming SSH configuration is set up:

```
#transfer file from local workstation to your workspace
marie@local$ rsync spec_0001.out dataport1:/data/walrus/ws/
    nelle-specimen

#transfer folder to your workspace
marie@local$ rsync -r scan_results datapor1t:/data/walrus/ws/
    nelle-specimen

# transfer from your workspace to your local workstation
# Hint: In Windows, directory paths are determined with backslashes
marie@local$ rsync -r dataport:/data/walrus/ws/nelle-specimen/
    scan_results ~/Documents/results
```



https://doc.../data_transfer/dataport_nodes

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

- HPC file systems
- Software environment at ZIH

Batch System

Software Development at ZIH's HPC systems

HPC Projects

HPC Support

Modules

Installed software is organized in modules.

A module is a user interface, that:

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

Users have several options:

- use pre-installed software
- install modules in Python virtual environment
- install private modules in a workspace or /projects (for a group-wide installation).
- containers

See `https://doc.../software/modules`

Installed Software

Various software is already installed on each cluster. Some of them with several versions. Software and versions may differ among clusters

What information is important to use the available software?

- version
- dependencies
- extensions

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

Hint: `module` and `ml` can be used as synonyms

Hierarchical module environment

Level One: login node of the particular cluster
Level Zero: release with specific module versions (e.g. release/24.04.)

- module av: lists available modules within its respective hierarchical environment level
- each release consists of different module versions.
- Update cycle is once/twice a year
- new software will be found in future release version (no guarantee)

```
marie@login2.barnard ~$ 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 r24.04 (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
...
```

Attention: Available modules differ among clusters

Module usage

Use `module spider` to identify your desired module and version (case-sensitive). Additionally the module dependencies are listed, which may help during the installation process.

```
marie@login2.barnard ~$ module spider ParaView
---------------------------------------------------------------------------------------------- \\
      ParaView: ParaView/5.10.1-mpi
ParaView: ParaView/5.12.0
----------------------------------------------------------------------------------------------
    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
```

To retrieve the details for each version you may specify the version

```
marie@login2.barnard ~$ module spider ParaView/5.12.0
```

Bundled modules

SciPy is part of Python package bundle, to retrieve detailed information you have to provide the bundle name

```
marie@login2.barnard.~$ module spider SciPy
---------------------------------------------------------------------------------------------
  SciPy-bundle:
---------------------------------------------------------------------------------------------
    Description:
      Bundle of Python packages for scientific software
     Versions:
        SciPy-bundle/2020.11
        SciPy-bundle/2021.05
        SciPy-bundle/2021.10
        SciPy-bundle/2022.05
        SciPy-bundle/2023.07
        SciPy-bundle/2023.11
---------------------------------------------------------------------------------------------
  For detailed information about a specific "SciPy-bundle" package (including how to load the
        modules) use the module's full name.
  Note that names that have a trailing (E) are extensions provided by other modules.
  For example:
     $ module spider SciPy-bundle/2023.11
-----------------------------------------------------------------------
```

Bundled modules

Specifying the bundle version to receive more detailed information:

```
marie@login2.barnard ∼$ module spider SciPy-bundle/2023.11
---------------------------------------------------------------------------------------------
  SciPy-bundle: SciPy-bundle/2023.11
---------------------------------------------------------------------------------------------
    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
          023.11" module is available to load.
      development/24.04  GCC/13.2.0
      release/24.04  GCC/13.2.0
   Help:
      Description
      ===========
      Bundle of Python packages for scientific software
      More information
      ================
       - Homepage: https://python.org/
      Included extensions
      ===================
      beniget-0.4.1, Bottleneck-1.3.7, deap-1.4.1, gast-0.5.4, mpmath-1.3.0,
      numexpr-2.8.7, numpy-1.26.2, pandas-2.1.3, ply-3.11, pythran-0.14.0,
      scipy-1.11.4, tzdata-2023.3, versioneer-0.29
```

Modules within a Toolchain

Working with several module versions, may result into wrong dependency versions. Toolchains avoid this problem by having a set of modules, which are building the software for other modules, e.g. `foss`-toolchain consisting of `GCC,OpenMPI,OpenBLAS & FFTW`

Loading compiler, MPI, and BLAS/LAPACK

```
marie@login2.barnard ~$ module load foss/2023b
Module foss/2023b and 21 dependencies loaded.

marie@login2.barndard ~$ 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

marie@login2.barndard ~$ mpicc hello.c

marie@login2.barndard ~$ 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 these extensively, we have only a limited number of licenses!
- Matlab: use the Matlab compiler `https://doc.../software/mathematics/#matlab`

Containers

- Singularity as container environment on the used Cluster (e.g. Barnard)
- Docker containers can easily be converted
- more information at `https://doc.../software/containers`

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

- General
- Slurm examples

Software Development at ZIH's HPC systems

HPC Projects

HPC Support

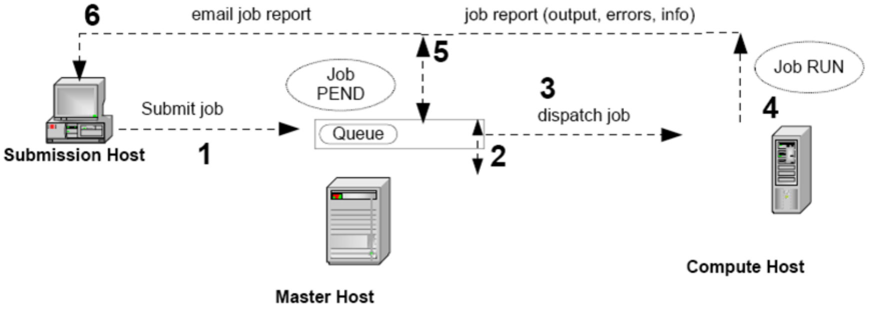
Overview

Why do we need a batch system?

- 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

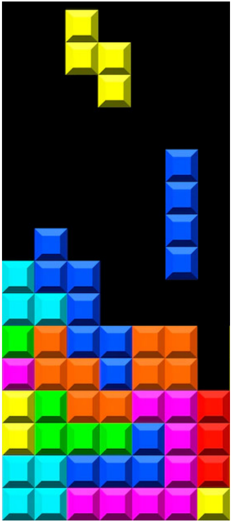
Agreed, we need a batch system.



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 → heuristics allowed.

Useful functions of a batch system

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://doc.../jobs_and_resources/slurm` .

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

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

- General
- Slurm examples

Software Development at ZIH's HPC systems

HPC Projects

HPC Support

Slurm examples

Slurm interactive example:

```
ssh login1.barnard.hpc.tu-dresden.de
srun --ntasks=1 --cpus-per-task=1 --time=1:00:00 \
     --mem-per-cpu=1000 --pty bash -l
```

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
- QoS for interactive jobs will be set automatically. Highest prio for these.

Slurm examples

Normal MPI parallel job `sbatch <myjobfile>` (submitted from a login node)

```
#SBATCH --time=8:00:00
#SBATCH --ntasks=64
#SBATCH --mem-per-cpu=780
#SBATCH --mail-type=end
#SBATCH --mail-user=marie@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.

Slurm examples

Requesting multiple GPU cards (e.g. on the `alpha` system)

```
#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=marie@tu-dresden.de

#SBATCH -o stdout
#SBATCH -e stderr

echo 'Running program...'
```

Slurm: Job monitoring

Basic question: Why does my job not start? Try: `whypending <jobid>` (or PIKA, see later)

```
> 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:

The following nodes are available:
    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

```
> show_resources
Workspaces
-----------------
Sum in projects   HORSE: size     files  WALRUS: size    files
   s_zih-sw          7.28 GiB   111,099      0.00 B          0
   hpcsupport      435.07 GiB   357,259     69.63 GiB       27
   p_vendortest      2.29 TiB   711,500     16.10 MiB      284

Own data in workspaces             size   files    last access
   /data/horse/ws/mark-test3   4.00 kiB       1    11 days ago
   /data/horse/ws/mark-test4   4.00 kiB       1    11 days ago
   /data/horse/ws/mark-test5  10.94 kiB       3    42 days ago
   ==> Total:                 18.94 kiB       5

Permanant Storage
-----------------
Own data in /home: 500.38 MiB in 15,511 files  ( 1.0% of 50.00 GiB
     limit)

Projects                        files       size        limit       %
   hpcsupport                 142,108  200.39 GiB  300.00 GiB    66.8
   p_vendortest                     1    0.00 B    100.00 GiB     0.0
```

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* → *NormShares*
- used resources "during the last few days" *RawUsage* → *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

Use sinfo (-T to show reservations).

```
> sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
barnard*     up   infinite      1  down* n1609
barnard*     up   infinite      3   resv n[1509,1512,1604]
barnard*     up   infinite    239    mix n[1003-1004, ...]
barnard*     up   infinite    387  alloc n[1001-1002, ...]
```

```
> sinfo -T
RESV_NAME           STATE           START_TIME             END_TIME     DURATION  NODELIST
hpc_sw_barnard     ACTIVE  2024-08-22T16:00:00  2025-08-22T16:00:00  365-00:00:00  n1604
p_darwin_324       ACTIVE  2024-09-30T09:00:00  2024-10-07T09:00:00    7-00:00:00  n[1002,1006..]
```

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=romeo
   JobState=PENDING Reason=Resources Dependency=(null)
   Priority=49533 QOS=normal
   StartTime=Unknown SubmitTime=2020-09-18T14:16:06
```

Detailed 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=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/marie
   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 [--me]` – show [my] current job queue
- `sprio` – show priorities of current job queue
- `sbcast`, `sgather` – efficiently distribute/collect data files to/from compute nodes
- `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!

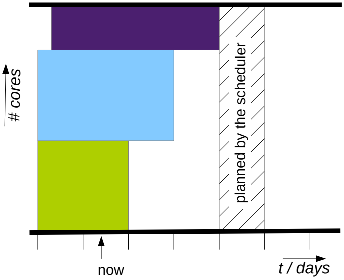
- Maybe a reservation prevents my job from starting (`sinfo -T`)
- Maybe an older large job is scheduled and waits for resources:

```
∼ > 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.

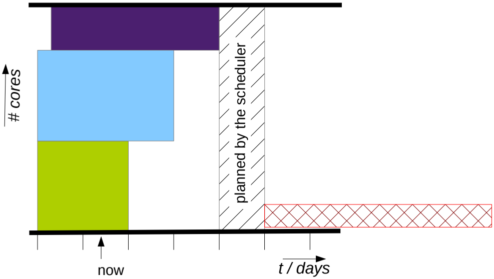
- Maybe fragmentation would be too high.

Backfilling

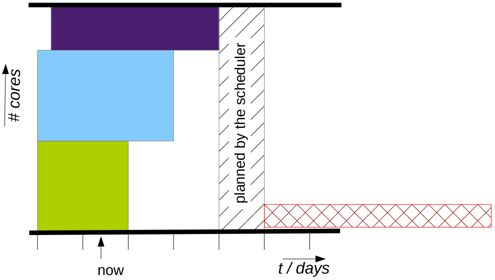


My job to be placed:

Backfilling

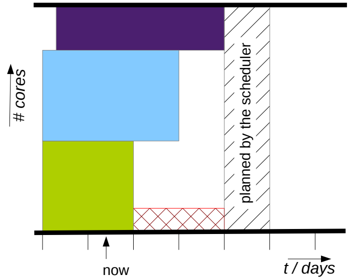


Backfilling



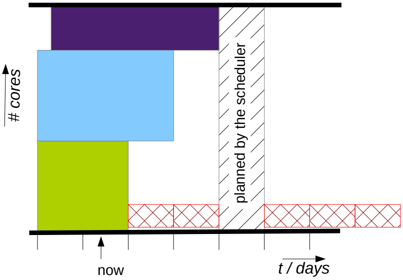
I know my job better:

Backfilling



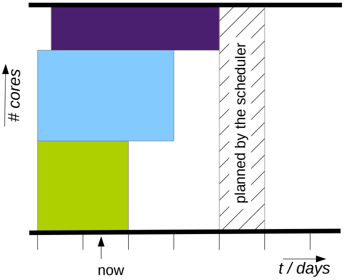
Estimate the maximum run-time of your job!

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)

Checkpointing during training

Example pytorch:

```
 #checkpointing either after each epoch, or after a defined number of epochs
torch.save({
'epoch':epoch,
'model_state_dict': model.state_dict(),
'optimizer_state_dict': optimizer.state_dict(),
'loss':loss,
...
},PATH)
#call data
checkpoint=torch.load(PATH,weights_only=True)
model.load_state_dict(checkpoint['model_state_dict'])
optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
epoch = checkpoint['epoch']
loss=checkpoint['loss']
```

Example using tensorflow:

```
cp_callback= tf.keras.callbacks.ModelCheckpoing(filepath=checkpoint_path, save_weights_only=
      True,verbose=1)

 #include the callback with model.fit
model.fit(train_images,
            train_labels,...,
            callbacks=[cp_callback])
```

Let Slurm do its work!

The batch system (Slurm) manages resources 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 --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 --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
- each cluster comes with separate login nodes (same hardware!) that can be used for interactive work
- can use terminal, JupyterHub or GUI via virtual Desktops

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

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

Remember

- JupyterHub (`https://doc.../access/jupyterhub`)
- Virtual Desktops (`https://doc.../software/virtual_desktops`)
- Nice DCV (`https://doc.../access/desktop_cloud_visualization`)

Availability

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

- short jobs lead to higher fluctuation (limits 1/2/7 days)
- plan resources in advance (publication deadline) - reserve nodes

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

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

HPC Projects

HPC Support

Software development

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

- compilers
- GPU programming
- mathematical libraries
- debugging
- performance tuning

Available compilers

Which compilers are installed?

- Starting point: `https://doc.../software/compilers`
- Up-to-date information: `module spider ...`

Available compilers

Which compilers are installed?

- Starting point: `https://doc.../software/compilers`
- 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://doc.../software/compilers`
- 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)
- parallel programs: synchronization
- file input/output

Expensive operations

Time consuming operations in scientific computing:

- division, power, trigonometric and exponential functions
- un-cached memory operations (bandwidth, latency)
- parallel programs: synchronization
- file input/output

How to find performance bottlenecks?

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

Low hanging fruits - Compiling

What is the needed floating point precision?
32 bit vs. 64 bit memory for each floating number impacts on

- memory footprint
- computing speed

→ e.g. `-freal-M-real-N` for Fortran code

Low hanging fruits - Compiling

What is the needed floating point precision?
32 bit vs. 64 bit memory for each floating number impacts on

- memory footprint
- computing speed

→ e.g. `-freal-M-real-N` for Fortran code

What is the needed floating point accuracy?

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

→ see man pages!

Low hanging fruits - Compiling

What is the needed floating point precision?
32 bit vs. 64 bit memory for each floating number impacts on

- memory footprint
- computing speed

→ e.g. `-freal-M-real-N` for Fortran code

What is the needed floating point accuracy?

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

→ see man pages!

Compile on the particular target system (login node) to use automatic hardware optimizations.

- e.g. compile for Barnard+Romeo compatibility with Intel compiler:
  `-Ofast -mavx -axCORE-AVX2,CORE-AVX512`

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

Software Development at ZIH's HPC systems

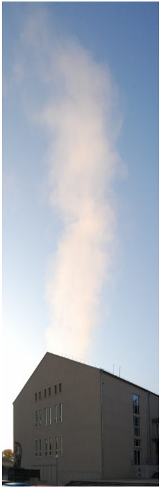
- Compiling
- Tools

HPC Projects

HPC Support

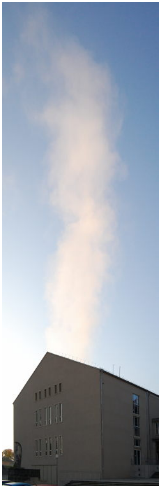
On HPC systems: Efficient code is essential!

- the same code runs on several 1000 CPUs, multiplying the energy and runtime to thousands or millions of CPU core-h
- just using more CPUs does not produce a speedup (wrong parallelization or job placement)
- speedup strongly depends on program capabilities (called parallel scalability)



On HPC systems: Efficient code is essential!

- the same code runs on several 1000 CPUs, multiplying the energy and runtime to thousands or millions of CPU core-h
- just using more CPUs does not produce a speedup (wrong parallelization or job placement)
- speedup strongly depends on program capabilities (called parallel scalability)



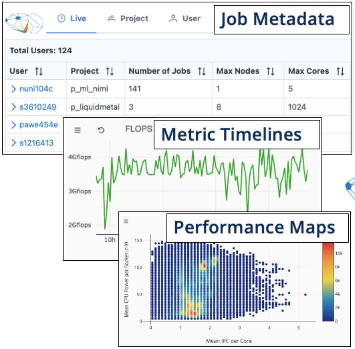
How do I tell?

- Identify inefficient jobs with monitoring
  → PIKA
- Identify slow code regions with profiling
  → e.g. Score-P + Cube
- Identify slow code in detail with tracing
  → e.g. Score-P + Vampir

PIKA - Analyzing Job Performance

A hardware performance monitoring stack to identify inefficient HPC jobs

- performance data is collected with every job run
- a web portal allows easy access to own performance data
- automatic detection of pathological jobs via issue tagging
- issue jobs can be discussed with ZIH performance experts



Job Metadata

Live | Project | User

Total Users: 124

User	Project	Number of Jobs	Max Nodes	Max Cores
nuni104c	p_ml_nimi	141	1	5
s3610249	p_liquidmetal	3	8	1024
pawe454e				
s1216413				

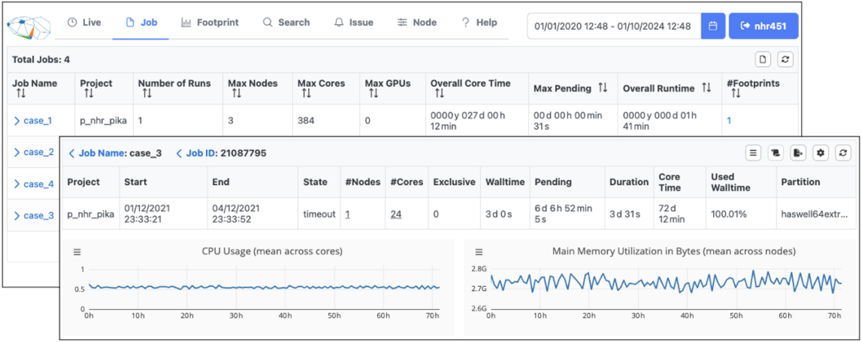
Metric Timelines

Performance Maps

https://doc.../software/pika

PIKA - Job Overview

Top down approach to detailed job metadata and timeline performance metrics



Total Jobs: 4

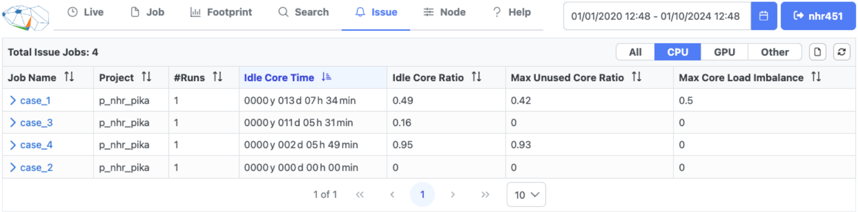
Job Name ⇅	Project ⇅	Number of Runs ⇅	Max Nodes ⇅	Max Cores ⇅	Max GPUs ⇅	Overall Core Time ⇅	Max Pending ⇅	Overall Runtime ⇅	#Footprints ⇅
› case_1	p_nhr_pika	1	3	384	0	0000 y 027 d 00 h 12 min	00 d 00 h 00 min 31 s	0000 y 000 d 01 h 41 min	1
› case_2									
› case_4									
› case_3									

‹ 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	01/12/2021 23:33:21	04/12/2021 23:33:52	timeout	1	24	0	3 d 0 s	6 d 6 h 52 min 5 s	3 d 31 s	72 d 12 min	100.01%	haswell64extr...

PIKA - Analyzing Pathological Jobs (Issue)

Jobs with performance issues sorted by idle core time



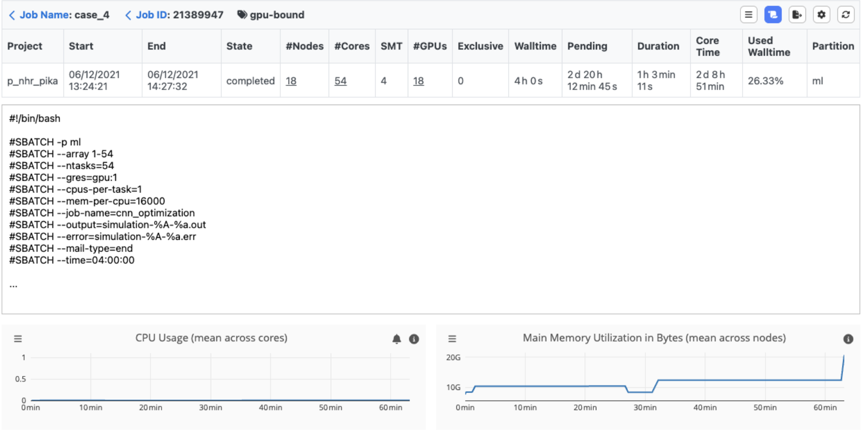
Total Issue Jobs: 4

Job Name	Project	#Runs	Idle Core Time	Idle Core Ratio	Max Unused Core Ratio	Max Core Load Imbalance
case_1	p_nhr_pika	1	0000 y 013 d 07 h 34 min	0.49	0.42	0.5
case_3	p_nhr_pika	1	0000 y 011 d 05 h 31 min	0.16	0	0
case_4	p_nhr_pika	1	0000 y 002 d 05 h 49 min	0.95	0.93	0
case_2	p_nhr_pika	1	0000 y 000 d 00 h 00 min	0	0	0

Jobs are examined for performance issues daily and may be tagged accordingly. They will be available in the **Issue** view no later than 24 hours after the completion of the job. (applies only to jobs that have run for at least one minute and have used more than one CPU core)

PIKA - Analyzing Pathological Jobs (Issue)

Very low CPU usage may result from incorrect resource allocation (see job script view)



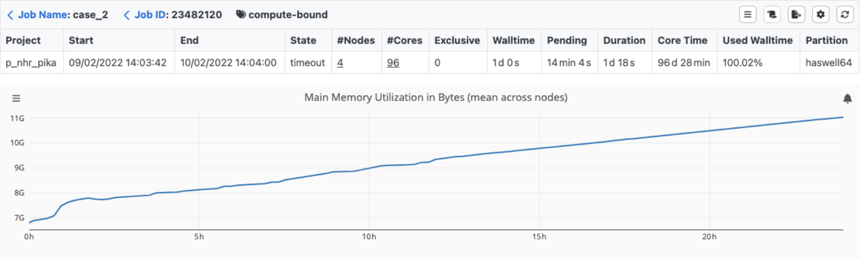
< **Job Name**: **case_4** < **Job ID**: **21389947** **gpu-bound**

Project	Start	End	State	#Nodes	#Cores	SMT	#GPUs	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
p_nhr_pika	06/12/2021 13:24:21	06/12/2021 14:27:32	completed	18	54	4	18	0	4 h 0 s	2 d 20 h 12 min 45 s	1 h 3 min 11 s	2 d 8 h 51 min	26.33%	ml

```
#!/bin/bash

#SBATCH -p ml
#SBATCH --array 1-54
#SBATCH --ntasks=54
#SBATCH --gres=gpu:1
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=16000
#SBATCH --job-name=cnn_optimization
#SBATCH --output=simulation-%A-%a.out
#SBATCH --error=simulation-%A-%a.err
#SBATCH --mail-type=end
#SBATCH --time=04:00:00

...
```

PIKA - Analyzing Pathological Jobs (Issue)

Linear increases in memory usage may indicate a potential memory leak



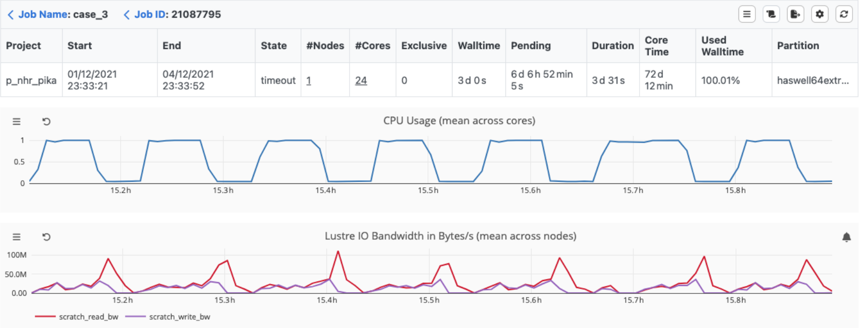
Job Name: case_2   Job ID: 23482120   compute-bound

Project	Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
p_nhr_pika	09/02/2022 14:03:42	10/02/2022 14:04:00	timeout	4	96	0	1d 0s	14min 4s	1d 18s	96d 28min	100.02%	haswell64

Main Memory Utilization in Bytes (mean across nodes)

PIKA - Analyzing Pathological Jobs (Issue)

An inverse correlation between CPU usage and I/O metrics may indicate the presence of blocking I/O operations

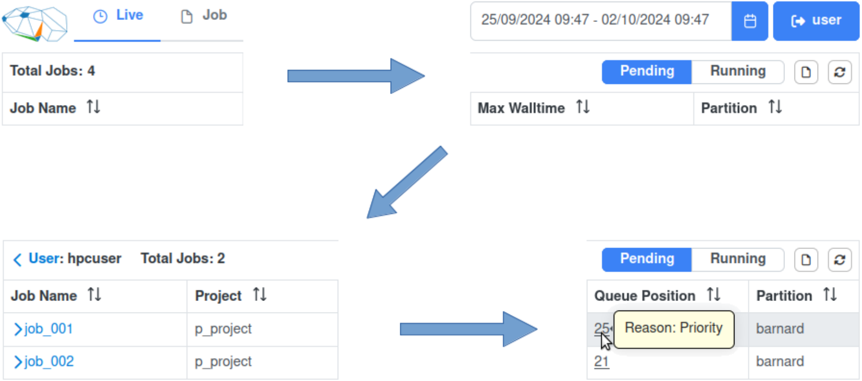


Job Name: case_3 **Job ID**: 21087795

Project	Start	End	State	#Nodes	#Cores	Exclusive	Walltime	Pending	Duration	Core Time	Used Walltime	Partition
p_nhr_pika	01/12/2021 23:33:21	04/12/2021 23:33:52	timeout	1	24	0	3 d 0 s	6 d 6 h 52 min 5 s	3 d 31 s	72 d 12 min	100.01%	haswell64extr...

PIKA - whypending

Basic question: Why does my job not start?
Via web-GUI: PIKA → Live → Pending → Job → hover 'Queue Position'



Total Jobs: 4

Job Name ⇅

Max Walltime ⇅	Partition ⇅

User: hpcuser   Total Jobs: 2

Job Name ⇅	Project ⇅
job_001	p_project
job_002	p_project

Queue Position ⇅	Partition ⇅
25	barnard
21	barnard

Reason: Priority

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.

→ Identify slow code regions – 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
  ```

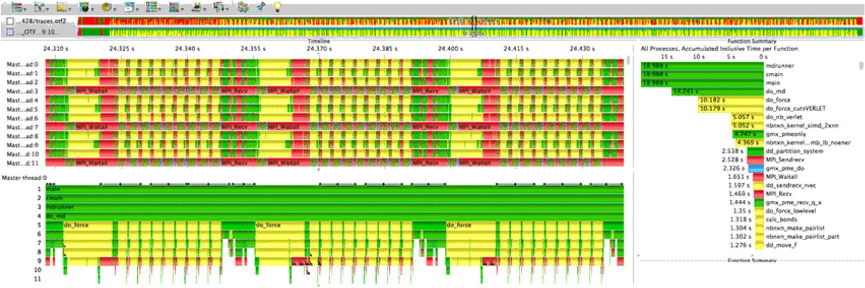
- view: `less analysis.txt`

```
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
```

Tracing

More details please!

- Tracing: Recording a program run with timestamps for each function call, memory access, process communication, etc.
- Score-P (instrumentation) + Vampir (visualization) → Starting point: `https://doc.../software/vampir`
- Please ask our experts!



All Processes, Accumulated Inclusive Time per Function

18.988 s	mdrunner
18.988 s	cmain
18.988 s	main
14.241 s	do_md
10.182 s	do_force
10.179 s	do_force_cutsVERLET
5.057 s	do_nb_verlet
5.052 s	nbnxn_kernel_simd_2xnn
4.747 s	gmx_pmeonly
4.369 s	nbnxn_kernel...mb_lb_noener
2.538 s	dd_partition_system
2.528 s	MPI_Sendrecv
2.326 s	gmx_pme_do
1.651 s	MPI_Waitall
1.597 s	dd_sendrecv_rvec
1.469 s	MPI_Recv
1.444 s	gmx_pme_recv_q_x
1.35 s	do_force_lowlevel
1.318 s	calc_bonds
1.304 s	nbnxn_make_pairlist
1.302 s	nbnxn_make_pairlist_part
1.276 s	dd_move_f

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

Software Development at ZIH's HPC systems

HPC Projects

- Project application
- Management of HPC projects

HPC Support

HPC project

High Performance Computing is organized in projects. A project starts with an application via the online application portal „Jards".

Project life cycle

NHR Starter / NHR Test ⟶ **NHR Normal / NHR Large**

- Application testing
- Code development
- Benchmarking

Tier3 (TUD local)

- TUD & research institutions from Saxony without sufficient HPC resources
- courses & teaching resources
- limited resources

ScaDS.AI

- Associated members

More information on project types and the application process:
`https://tu-dresden.de/zih/hochleistungsrechnen/zugang/projektantrag`

HPC project application

All relevant information for a project application:
`https://tu-dresden.de/zih/hochleistungsrechnen/zugang/projektantrag`

	Starter	Test	Normal	Large	Tier3
max. CPU/GPU Resources	fixed	$1\,M/10\,k$	$20\,M/150\,k$		$400\,k/4\,k$
Resource estimation		●	●	●	●
Detailed Description			●	●	
Technical Review		●	●	●	●
Scientific Review			●*	●●	

* quick processing when funded by DFG, EU, or a Bundesministerium

- Starter: Quickest & easiest option, fixed amount, only once per person
- Test: For development & scalability testing only
- Normal: Fair amount of resources and effort
- Large: Largest amount and preparation time (starts quarterly)

Caution

- finalized project applications cannot be modified
- applications cannot be transferred (copied) between NHR ↔ Tier3

HPC project application

All relevant information for a project application:
`https://tu-dresden.de/zih/hochleistungsrechnen/zugang/projektantrag`

	Starter	Test	Normal	Large	Tier3
max. CPU/GPU Resources	fixed	$1\,M/10\,k$	$20\,M/150\,k$		$400\,k/4\,k$
Resource estimation		●	●	●	●
Detailed Description			●	●	
Technical Review		●	●	●	●
Scientific Review			●*	●●	

* quick processing when funded by DFG, EU, or a Bundesministerium

Detailed project description:

- scientific description of the project
- preliminary work, state of the art...
- objectives, used methods, software
- estimation and justification of needed resources
- proposal templates are provided on the above mentioned website

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

Software Development at ZIH's HPC systems

HPC Projects

- Project application
- Management of HPC projects

HPC Support

Management of HPC projects

Who...

- project leader ("Principal Investigator", PhD or higher) → accountable
- project administrator ("Person of Contact") → responsible

What...

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

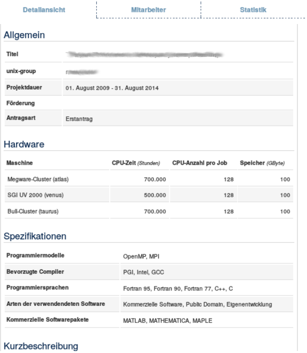
. . . agreed upon by the users with the ‚Nutzungsbedingungen' (terms of use)

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

- view
  technical details of the project
- add and remove users
- define
  a technical administrator
- view statistics
  (resource consumption)



Allgemein

Titel	[redacted]
unix-group	[redacted]
Projektdauer	01. August 2009 - 31. August 2014
Förderung	
Antragsart	Erstantrag

Hardware

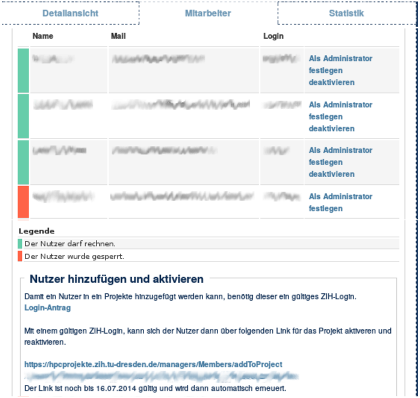
Maschine	CPU-Zeit *(Stunden)*	CPU-Anzahl pro Job	Speicher *(GByte)*
Megware-Cluster (atlas)	700.000	128	100
SGI UV 2000 (venus)	500.000	128	100
Bull-Cluster (taurus)	700.000	128	100

Spezifikationen

Programmiermodelle	OpenMP, MPI
Bevorzugte Compiler	PGI, Intel, GCC
Programmiersprachen	Fortran 95, Fortran 90, Fortran 77, C++, C
Arten der verwendendeten Software	Kommerzielle Software, Public Domain, Eigenentwicklung
Kommerzielle Softwarepakete	MATLAB, MATHEMATICA, MAPLE

Kurzbeschreibung

Online project management



	Name	Mail	Login	
🟩	[redacted]	[redacted]	[redacted]	Als Administrator festlegen deaktivieren
🟩	[redacted]	[redacted]	[redacted]	Als Administrator festlegen deaktivieren
🟩	[redacted]	[redacted]	[redacted]	Als Administrator festlegen deaktivieren
🟥	[redacted]	[redacted]	[redacted]	Als Administrator festlegen

Legende

🟩	Der Nutzer darf rechnen.
🟥	Der Nutzer wurde gesperrt.

Nutzer hinzufügen und aktivieren

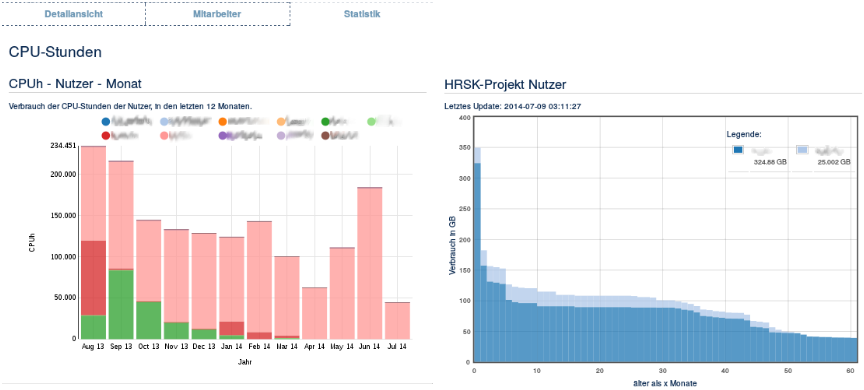
Damit ein Nutzer in ein Projekte hinzugefügt werden kann, benötig dieser ein gültiges ZIH-Login.
Login-Antrag

Mit einem gültigen ZIH-Login, kann sich der Nutzer dann über folgenden Link für das Projekt aktiveren und reaktivieren.

https://hpcprojekte.zih.tu-dresden.de/managers/Members/addToProject
[redacted]

Der Link ist noch bis 16.07.2014 gültig und wird dann automatisch erneuert.

Online project management



CPU-Stunden

CPUh - Nutzer - Monat

Verbrauch der CPU-Stunden der Nutzer, in den letzten 12 Monaten.

HRSK-Projekt Nutzer

Letztes Update: 2014-07-09 03:11:27

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

Software Development at ZIH's HPC systems

HPC Projects

HPC Support

- Channels of communication
- Support types and topics

Channels of communication

ZIH → users:

- training course “Introduction to HPC at ZIH”
- HPC wiki: `https://doc.zih.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

HPC SUPPORT

● Operation Status

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
      `hpc-support@tu-dresden.de`
      please: use your `...@tu-dresden` address as sender and
      include information as shown in “Create a ticket”
      `https://doc.../support/support`
    - phone: service desk (0351) 463 40000
- personal contact
  - phone call, email, talk at the Mensa
  - socializing is fine... but: risk of forgetting

Channels of communication

Users ↔ ZIH:

- HPC Q & A session
  - consulting hour, every Monday 1:30 pm (except holidays)
  - `https://tu-dresden.de/zih/die-einrichtung/termine/termine/nhr-trainings/qa-session-nhr-at-tud`
- NHR AI Q & A session ”From laptop to supercomputer“
  - every Thursday at 2 pm
  - `https://www.nhr-verein.de/ki-auf-hochleistungsrechnern`

TECHNISCHE UNIVERSITÄT DRESDEN
Supercomputing
Q&A session @ZIH
NHR
CIDS
↗ tu-dresden.de/zih/qa-sessions-nhr-at-tud

Agenda

Linux command line - Introduction

Access to HPC systems at ZIH

Compute hardware

Batch System

Software Development at ZIH's HPC systems

HPC Projects

HPC Support

- Channels of communication
- Support types and topics

Support types and topics

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 ...

Support types and topics

HPC Software questions:

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

→ restrictions of this support:

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

Support types and topics

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

Support types and topics

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

- Hartmut Mix, Noah Löwer (project management)
- Matthias Kräußlein (project infrastructure and accounting)
- Guilherme Calandrini, Etienne Keller (technical support)
- Steffen Reichert (technical support software)
- Danny Rotscher (Slurm, senior technical support)
- Ulf Markwardt (Slurm, senior technical support... head of the group)

Domain specific support

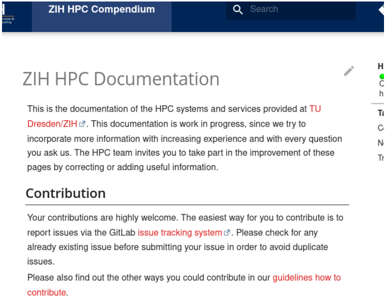
- Natalie Breidenbach (Life sciences)
- Sebastian Döbel (AI)
- Yan Ge (Life sciences)
- Noah Löwer (Earth system sciences)

(We are hiring.)

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



ZIH HPC Documentation

This is the documentation of the HPC systems and services provided at TU Dresden/ZIH. This documentation is work in progress, since we try to incorporate more information with increasing experience and with every question you ask us. The HPC team invites you to take part in the improvement of these pages by correcting or adding useful information.

Contribution

Your contributions are highly welcome. The easiest way for you to contribute is to report issues via the GitLab issue tracking system. Please check for any already existing issue before submitting your issue in order to avoid duplicate issues.

Please also find out the other ways you could contribute in our guidelines how to contribute.

Or open a ticket via `hpc-support@tu-dresden.de`.

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

Your help is required

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

The authors gratefully acknowledge the computing time made available to them on the high-performance computer at the NHR Center of TU Dresden.

The authors are grateful to the NHR Center 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 a hint about the publication

Thank you!

This presentation - and much more - can be found at

`https://doc.zih.tu-dresden.de`