

hpc-support@tu-dresden.de

Introduction to HPC@ZIH

Dresden, Februar 2024

HPC wiki has the answer

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

The screenshot shows the top navigation bar of the ZIH HPC Compendium website. On the left is the TU Dresden logo with the text 'TECHNISCHE UNIVERSITÄT DRESDEN'. Next to it is the ZIH logo with the text 'ZIH High Performance Computing'. The central part of the bar has the text 'ZIH HPC Compendium'. To the right is a search bar with a magnifying glass icon and the word 'Search'. Further right is a link to 'GitLab hpc-compendium' with a star icon and a counter of 7 stars.

ZIH HPC Compendium

Home

Quick Start

- Application for Login and Resources >
- Access to ZIH Systems >
- Data Transfer >
- Data Life Cycle Management >
- User Environment >
- Applications and Software >
- Software Development and Tools >
- HPC Resources and Jobs >
- User Support >
- Archive >
- Contribute to Documentation >

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-dresden.de

HPC Support

Operation Status
hpcsupport@zih.tu-dresden.de

Table of contents

- Contribution
- News
- Training and Courses

Agenda

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

Batch System

- General

- Slurm examples

HPC Support

- Management of HPC projects

Agenda

Compute hardware

HPC file systems

Software environment at ZIH

Access to HPC systems at ZIH

Batch System

HPC Support

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)

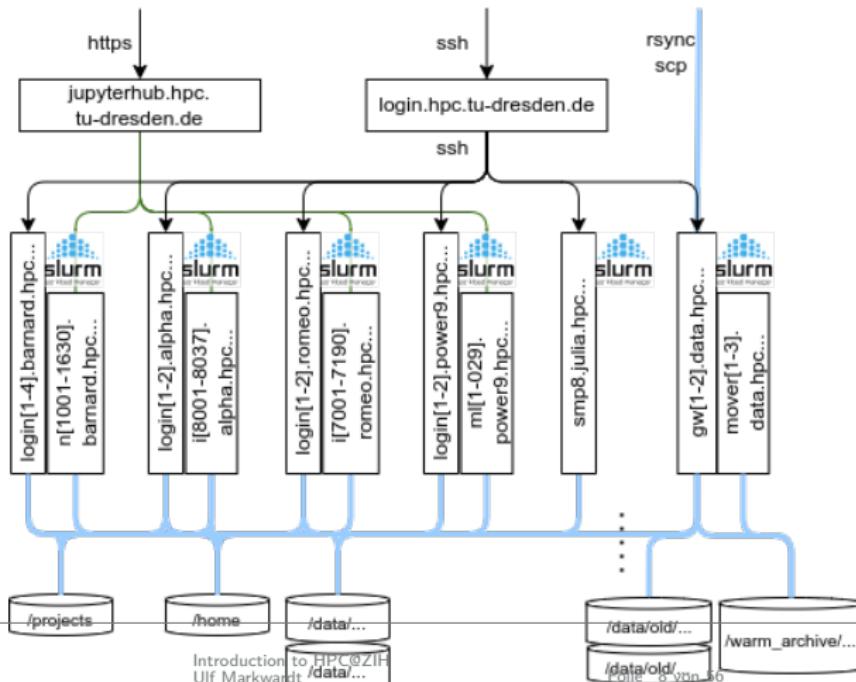
- new methods to access systems complementary to “classical” HPC mode
- large team for AI related research and support



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

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`

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 Compute 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

Compute hardware

HPC file systems

Software environment at ZIH

Access to HPC systems at ZIH

Batch System

HPC Support

Overview

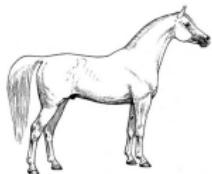
Properties of file systems:

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

Overview

- local SSD `/tmp`
- HPC global `/projects`, `/home`
- HPC global `/data/horse`
- 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 ($\sim 100\text{GB}$),
- 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`

Scratch file system

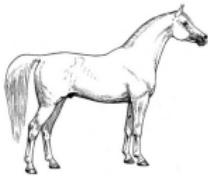
Workhorse: powered by Lustre

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 calendar 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
horse	100 days	High streaming bandwidth, disks.
walrus	1 year	Capacity file system, disks.

Workspace - examples

Available filesystems

```
marie@login1 > ws_find -l
available filesystems:
horse
walrus
```

Allocation

```
marie@login1 > ws_allocate -F walrus specimen 20
Info: creating workspace.
```

Notification:

```
marie@login1 > ws_send_ical -m <your-mail>@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@login1 > ws_list -F walrus
id: specimen
  workspace directory : /data/walrus/ws/marie-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

```
marie@login1 > ws_extend -F walrus specimen 10
Info: extending workspace.
/data/walrus/ws/marie-specimen
remaining extensions : 2
remaining time in days: 10
```

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

```
marie@login1 > ws_list -F walrus
id: specimen
  workspace directory : /data/walrus/ws/marie-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, dtrsnc, dttar.
- The transfer job is then created, queued, and processed automatically.
- User gets an email after completion of the job.
- Additional commands: dtinfo, dtqueue.

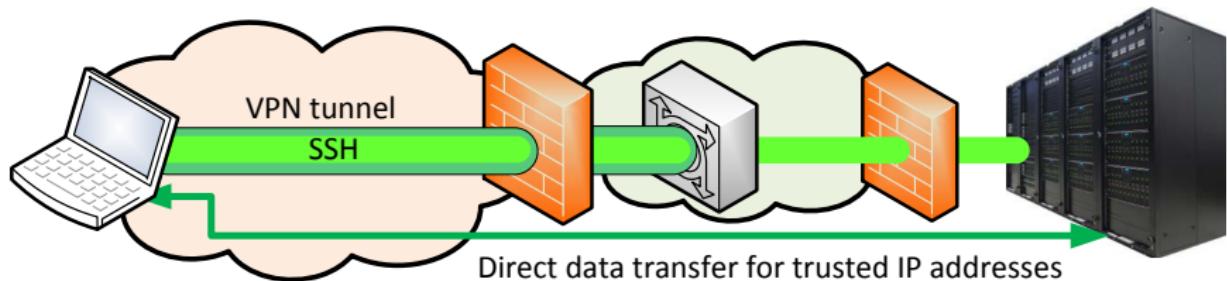
Very simple usage like

```
dttar -czf /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

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

Batch System

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 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>

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

```
marie@login1 > 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):

```
marie@login1 > 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`

```
marie@login1 > 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/2022.05" 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
```

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

```
marie@login1 ~ module load foss/2022a
Module foss/2022a and 21 dependencies loaded.

marie@login1 ~ 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@login1 ~ mpicc hello.c

marie@login1 ~ 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 use these extensively, we have only a limited number of licenses!
- Matlab: use the Matlab compiler <https://compendium.../software/mathematics/#matlab>

Containers

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

Agenda

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

Batch System

HPC Support

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: **Virtual Private Network**

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-LGIN>@tu-dresden.de
password     <ZIH-PASSWORD>
```

Access to HPC

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

The screenshot shows two main windows. On the left is a "Spawner Options" dialog box with tabs for "Simple" and "Advanced". It lists available architectures: Intel (x86_64), IBM Power (ppc64le), Intel Haswell, NVIDIA Tesla K80, and NVIDIA Tesla V100. Below this are sections for "CPUs" (Minimum: single core (single thread), Recommended: 7 cores (28 threads), Maximum: 44 cores (176 threads)) and "GPUs" (0, 1, 2, 3, 4, 5, 6). A large orange "Spawn" button is at the bottom. On the right is a Jupyter notebook interface titled "jupyter matplotlib-test Last Checkpoint a few seconds ago (autosaved)". The notebook contains two code cells and a plot. Cell [1] shows a command to list modules and check for matplotlib. Cell [2] contains Python code to import matplotlib, create a plot with points [1, 2, 3, 4], and show it. The resulting plot is a line graph with four points at (1, 1), (2, 2), (3, 3), and (4, 4).

Detailed documentation can be found at <https://compendium.../access/jupyterhub>.

Agenda

Batch System

General

Slurm examples

HPC Support

Overview Slurm

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

Overview Slurm

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

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

- job arrays,
- job dependencies,
- multi-threaded jobs

Agenda

Batch System

General

Slurm examples

HPC Support

Slurm examples

Slurm interactive example:

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

Slurm examples

Normal MPI parallel job `sbatch <myjobfile>`

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

```
#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=<your-mail>@tu-dresden.de

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

System information

Look at the login screen. Or `nodestat`

```
> nodestat
-----
# nodes: 630          (537 available / 93 unavailable)

jobs running:      3266 | cores in use:      97444
jobs pending:       852  | cores unavailable:  9672
jobs suspend:        0   |
jobs damaged:       3   |

-----
          CORES
free | resv | down | total
-----+-----+-----+
 6710 |   624 |  9672 | 65520
-----+-----+-----+
```

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=
    haswell164,romeo
  JobState=PENDING Reason=Resources Dependency=(null)
  Priority=49533 QoS=normal
  StartTime=Unknown SubmitTime=2020-09-18T14:16:06
```

Detailed job monitoring

Detailed 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=haswell164,romeo AllocNode:Sid=tauruslogin3:5741
ReqNodeList=(null) ExcNodeList=(null)
NodeList=(null)
NumNodes=171 NumCPUs=4096 NumTasks=4096 CPUs/Task=1 ReqB:S:C:T=0:0:0:1
TRES=cpu=4096,mem=1200G,node=1,billing=4096
Socks/Node=** NtasksPerN:B:S:C=0:0:0:1 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>

Agenda

Batch System

HPC Support
Management of HPC projects

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
- preliminary work, state of the art...
- objectives, used methods
- software, estimation of needed resources and scalability

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.

Detallansicht	Mitarbeiter	Statistik
Allgemein		
Title [REDACTED]		
unix-group	[REDACTED]	
Projektdauer	01. August 2009 - 31. August 2014	
Förderung		
Antragsart	Erstantrag	
Hardware		
Maschine	CPU-Zeit (Stunden)	CPU-Anzahl pro Job
Megware-Cluster (atlas)	700.000	128
SGLUV 2000 (venus)	500.000	128
Bull-Cluster (taurus)	Introduction to HPC@ZIH Ulf Markwardt Dresden, Februar 2024	700.000 128 100
Spezifikationen		

Thank you!

The full presentation - and much more - can be found at

<https://compendium.hpc.tu-dresden.de>