TECHNISCHE
UNIVERSITÄT
DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Introduction to HPC at ZIH

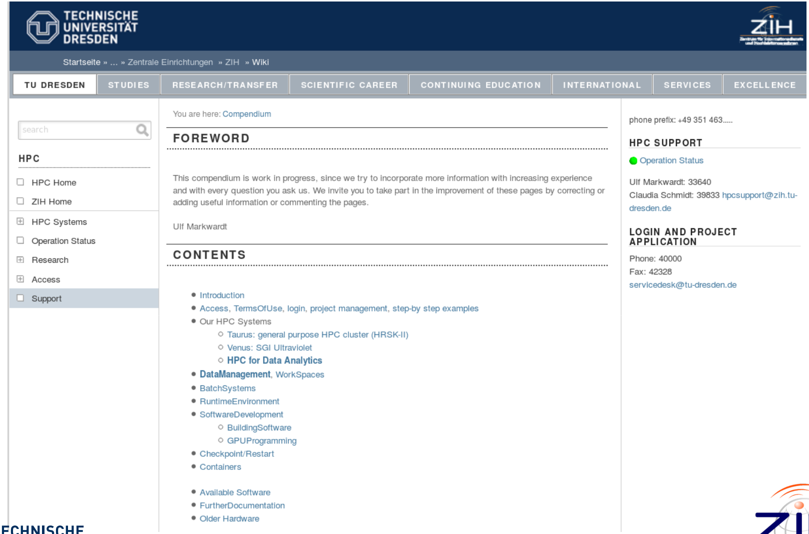
March 2021

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

ZIH
Center for Information Services &
High Performance Computing

HPC wiki has the answer

Please check our HPC wiki at `https://doc.zih.tu-dresden.de`



You are here: Compendium

FOREWORD

This compendium is work in progress, since we try to incorporate more information with increasing experience and with every question you ask us. We invite you to take part in the improvement of these pages by correcting or adding useful information or commenting the pages.

Ulf Markwardt

CONTENTS

- Introduction
- Access, TermsOfUse, login, project management, step-by step examples
- Our HPC Systems
  - Taurus: general purpose HPC cluster (HRSK-II)
  - Venus: SGI Ultraviolet
  - **HPC for Data Analytics**
- **DataManagement**, WorkSpaces
- BatchSystems
- RuntimeEnvironment
- SoftwareDevelopment
  - BuildingSoftware
  - GPUProgramming
- Checkpoint/Restart
- Containers

- Available Software
- FurtherDocumentation
- Older Hardware

phone prefix: +49 351 463.....

HPC SUPPORT

● Operation Status

Ulf Markwardt: 33640
Claudia Schmidt: 39833 hpcsupport@zih.tu-dresden.de

LOGIN AND PROJECT APPLICATION

Phone: 40000
Fax: 42328
servicedesk@tu-dresden.de

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
   - Access to HPC systems at ZIH
   - Compute hardware
   - HPC file systems
   - Software environment at ZIH
3. Batch System
   - General
   - Slurm examples
4. Software Development at ZIH's HPC systems
   - Compiling
   - Tools
5. HPC Support
   - Management of HPC projects
   - Channels of communication
   - Kinds of support
   - Beyond support

General

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

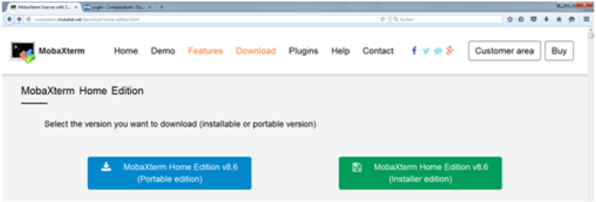
SSH access using MobaXterm

Step-by-step procedure at `https://doc.zih..../MobaXterm`

MOBAXTERM

Installation

- Follow this link to download MobaXTerm: http://mobaxterm.mobatek.net/download.html
- Choose the „Free Version" by clicking „Download now"
- Then choose the green "Installer Edition"-button


MobaXterm
Home
Demo
Features
Download
Plugins
Help
Contact
Customer area
Buy
MobaXterm Home Edition
Select the version you want to download (installable or portable version)
MobaXterm Home Edition v8.6 (Portable edition)
MobaXterm Home Edition v8.6 (Installer edition)

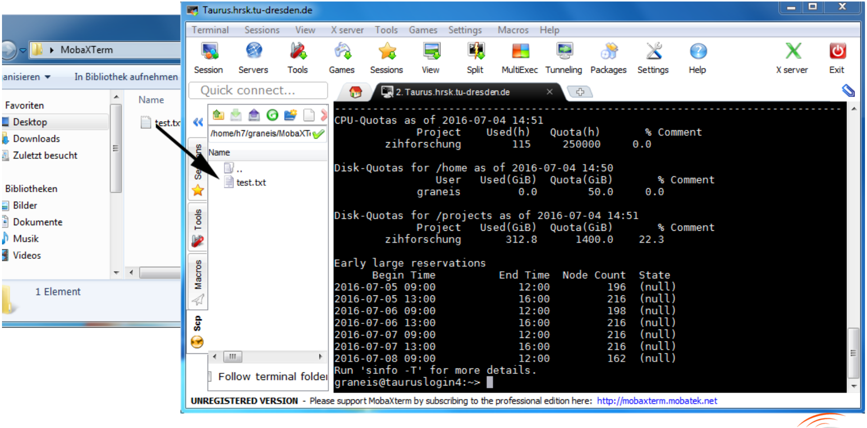
- After download you only have to choose the buttons for going on with the installation

Configuration

- If you have an icon of MobaXTerm on your desktop now open the program
- If you don't have this icon, search the program on your PC and open it

SSH access using 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:~>
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

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.

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.

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.

Basic commands

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

Basic commands (cont'd)

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

Basic commands (cont'd)

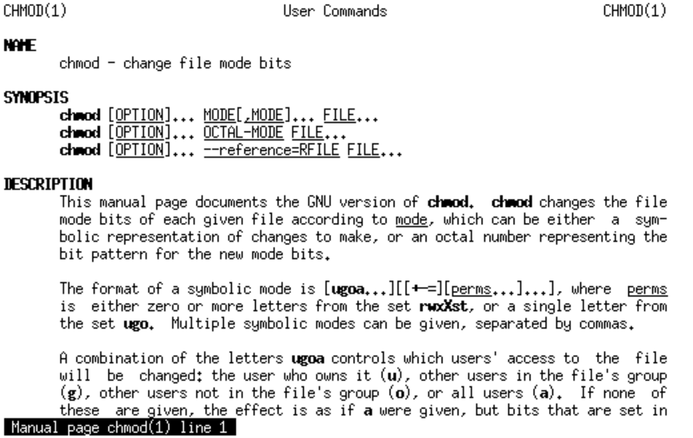
`echo`	display text to stdout `echo $PATH`
`cat`	display contents of a file `cat > newfile.txt`
`less`, `more`	pagewise display (`less README`)
`grep`	search for words/text (`grep result out.res`)
`file`	determine type of a file
`ps`	display running processes (`ps -axuf`)
`kill`	kill a process (`kill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

Editors:

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

Help at the command line

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



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

NAME
       chmod - change file mode bits

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

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

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

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

Linux file systems

- mounted remote file systems can be accessed like local resources.
- names are **case sensitiv**
- 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 `/sw`
- every user has her own home directory
  - `/home/<login>`
  - e.g. `/home/mark`

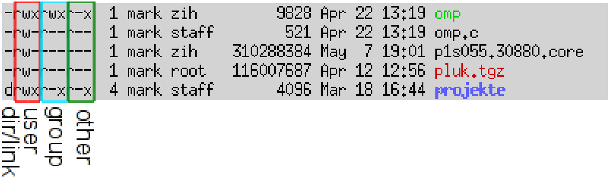
Special directories:

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

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

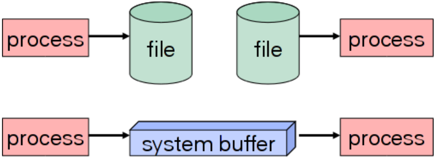
Linux is a text-oriented operating system. Input and output is 'streamable'.

- 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)

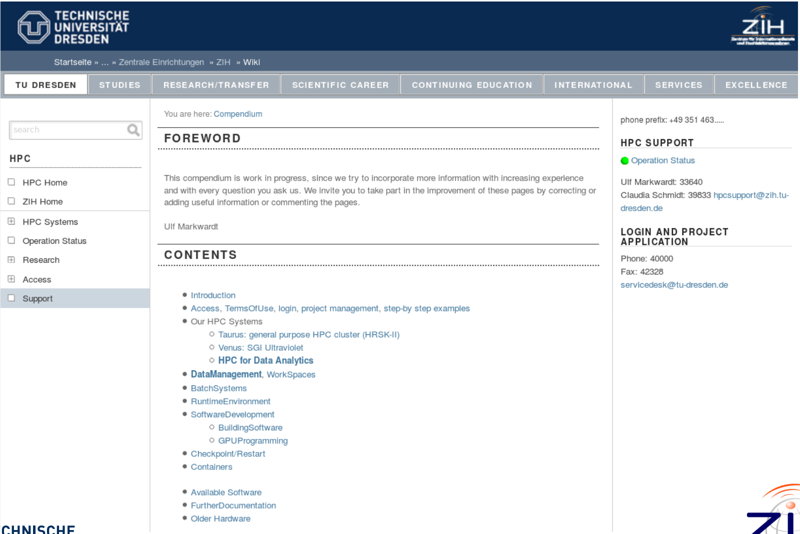
Hands-on training

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?

HPC wiki has the answer

Please check our HPC wiki at `https://doc.zih.tu-dresden.de`



You are here: Compendium

FOREWORD

This compendium is work in progress, since we try to incorporate more information with increasing experience and with every question you ask us. We invite you to take part in the improvement of these pages by correcting or adding useful information or commenting the pages.

Ulf Markwardt

CONTENTS

- Introduction
- Access, TermsOfUse, login, project management, step-by step examples
- Our HPC Systems
  - Taurus: general purpose HPC cluster (HRSK-II)
  - Venus: SGI Ultraviolet
  - **HPC for Data Analytics**
- **DataManagement**, WorkSpaces
- BatchSystems
- RuntimeEnvironment
- SoftwareDevelopment
  - BuildingSoftware
  - GPUProgramming
- Checkpoint/Restart
- Containers

- Available Software
- FurtherDocumentation
- Older Hardware

phone prefix: +49 351 463.....

HPC SUPPORT

● Operation Status

Ulf Markwardt: 33640
Claudia Schmidt: 39833 hpcsupport@zih.tu-dresden.de

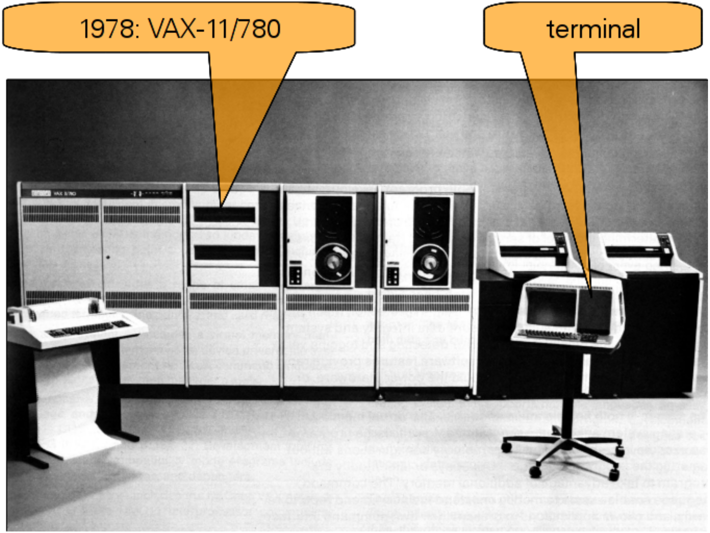
LOGIN AND PROJECT APPLICATION

Phone: 40000
Fax: 42328
servicedesk@tu-dresden.de

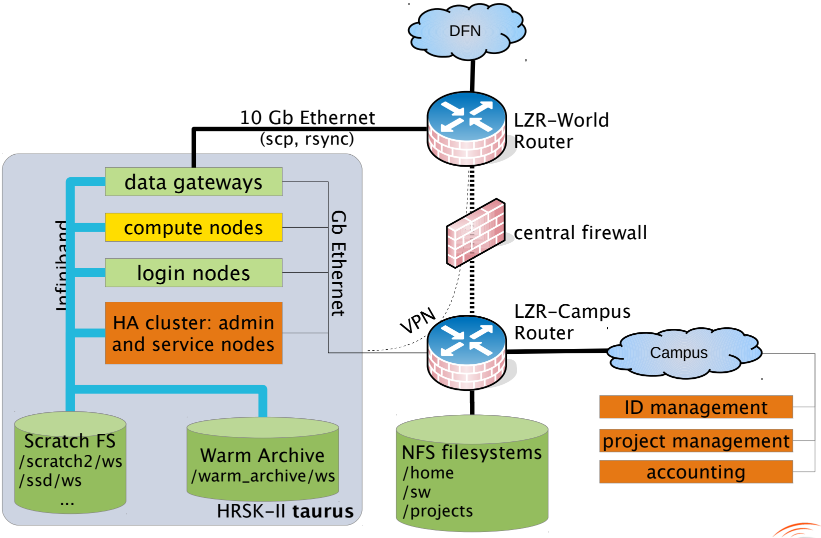
Agenda

1. Linux from the command line
2. HPC Environment at ZIH
   - Access to HPC systems at ZIH
   - Compute hardware
   - HPC file systems
   - Software environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
5. HPC Support

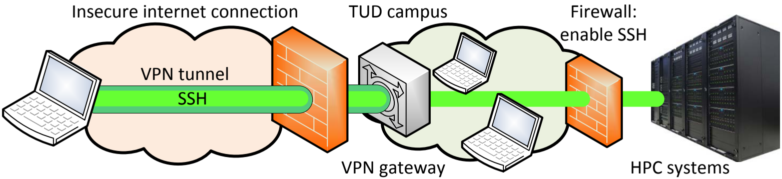
Computer and terminal



Access to the HPC systems



Firewall around HPC systems



The only 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

Data transfer (!) from acknowledged IP ranges, eg:

TU Freiberg	139.20.0.0/16
TU Chemnitz	134.109.0.0/16
Uni Leipzig	139.18.2.0/24

VPN for external users

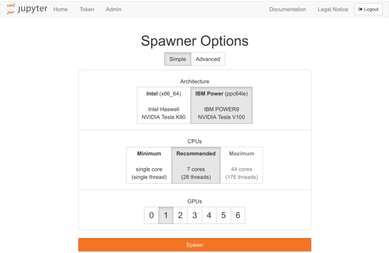
How-To for Linux, Windows, Mac can be found here: https://tu-dresden.de/zih/dienste/service-katalog/arbeitsumgebung/zugang_datennetz/vpn

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

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

Access to HPC

Unleash the HPC power with `ssh -X taurus.hrsk.tu-dresden.de` !
Or use a GUI from your Web browser → JupyterHub.



Spawner Options

Simple | Advanced

Architecture

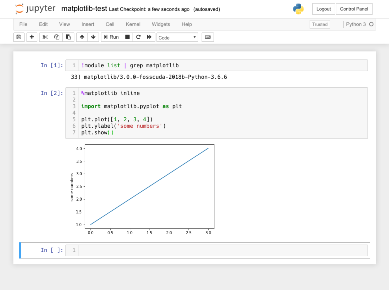
Intel (x86_64) Intel Haswell NVIDIA Tesla K80	IBM Power (ppc64le) IBM POWER9 NVIDIA Tesla V100

CPUs

Minimum single core (single thread)	Recommended 7 cores (28 threads)	Maximum 44 cores (176 threads)

GPUs

0 | 1 | 2 | 3 | 4 | 5 | 6



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()
```

In [ ]:

Detailled documentation can be found at `https://doc.zih..../JupyterHub` .

Agenda

1. Linux from the command line

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

3. Batch System

4. Software Development at ZIH's HPC systems

5. 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 e.g. MPI-CBG, HZDR, IFW, Uni Leipzig, TUBAF

HPC Infrastructure for Data Analytics

National competence center for data analytics “ScaDS” and its extension “ScaDS.AI” (with Universität Leipzig)

- hardware extensions
  - NVMe nodes (block storage over Infiniband),
  - nodes for machine learning (`ml`)
  - “warm archive” for research data, VM images...
  - compute (sub-) cluster (`romeo`)
  - large SMP system (`julia`)
  - GPU (sub-) cluster (`gpu3`)
- new methods to access systems complementary to “classical” HPC mode

Please check our landing page for Data Analytics on Taurus:
`https://doc.zih..../HPCDA`

Taurus

Overview

- General purpose cluster from Bull/Atos for highly parallel HPC applications (2013/2015)
- extended with hardware from NEC, IBM, HPE
- running with RHEL/Centos 7
- 1,029.9 TFlop/s total peak performance (rank 66 in top500, 06/2015) - now: 2.6 PFlop/s
- GPU partition with 128 dual GPUs
- all nodes have local SSD

Taurus

Heterogenous compute resources

- Normal compute nodes
    - 1456 nodes Intel Haswell, (2 x 12 cores), 64,128,256 GB
    - 32 nodes Intel Broadwell, (2 x 14 cores), 64 GB
    - 192 nodes AMD Rome (2 x 64 cores), 512 GB
- Large SMP nodes
    - 5 nodes with 2 TB RAM, Intel Haswell (4 x 14 cores)
    - 1 node with 48 TB RAM, Intel Cascade Lake (896 cores)
- Accelerator nodes
    - 64 nodes with 2 x NVidia K80, Intel Haswell (2 x 12 cores)
    - 32 nodes with 6 x NVidia V100-SXM2, IBM Power9 (2 x 22 cores)
    - 14 nodes with 3 x NVidia GTX1080, Intel Sandy Bridge (2 x 6 cores)
    - 34 nodes with 8 x NVidia A100-SXM4, AMD Rome (2 x 24 cores), 1 TB

Storage for data analytics

- 90 nodes with 8 NVMe cards (2 PB in total)
- warm archive powered by Quobyte
    - total of 13 PB
    - accessible as filesystem inside Taurus
    - S3 object store

AMD Rome nodes

Sub-cluster for data analytics

- 192 nodes, 512 GB RAM, 2x64 cores AMD Rome EPYC 7702
- Centos 7
- batch partition `romeo`
- for Intel compiler use `intel/2019b` toolchain with `-mavx2 -fma`
- use Intel MKL with environment `export MKL_DEBUG_CPU_TYPE=5`

More information on `https://doc.zih..../RomeNodes`

Large SMP system - taurussmp8

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

- 48 TB shared memory
- 10,6 TFlop/s peak performance
- 896 cores Intel 8276M CPU (Cascade Lake) 2.20GHz
- 370 TB local NVMe (64 devices)
  - 87 TB volume mounted for testing and smaller projects at `/nvme/1/<projectname>`, quota 100GB per project
  - propose for larger quota or dedicated storage up to the full capacity - temporarily
- RHEL 7
- batch partition `julia`

Attention: Software based on OpenMPI should not run here.
More information on `https://doc.zih..../SDFlex`

IBM Power nodes

Sub-cluster for machine learning

- 2 x IBM Power9 CPU (2.80 GHz, 22 cores)
- 256 GB RAM DDR4 2666MHz
- 6x NVIDIA VOLTA V100 with 32GB HBM2
- NVLINK bandwidth 150 GB/s between GPUs and host

Attention:This is not an x86 architecture!

- New software has to be built on these nodes.
- Simple copy-and-paste of python environments from other systems does not work.
- A virtual machine can be used to build singularity containers for the IBM Power nodes, see `https://doc.zih..../VMTools`

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
   - Access to HPC systems at ZIH
   - Compute hardware
   - HPC file systems
   - Software environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
5. HPC Support

Overview

Hierarchy of file systems: **speed** vs. **size** vs. **duration**:

- local SSD `/tmp`
- BeeGFS for data analytics `/beegfs`
- HPC global `/ssd`
- HPC global `/scratch`
- HPC global `/projects`, `/home`
- warm archive `/warm_archive`
- TUD global intermediate archive
- TUD global long term storage

The **number of files** (billions) is critical for all file systems.
Filenames should be encoded using UTF8.

Local disk

Recommended at Taurus:

- SSD: best option for lots of small I/O operations, limited size ($\sim$ 100GB),
- ephemeral: data will be deleted automatically after finishing the job,
- Each node has its own local disk. Attention: Multiple processes on the same node share their local disk,
- path to the local disk is `/tmp`

High-IOPS file system

Fastest (very stable!) parallel file system (IOPS) at each x86-system:

- large parallel file system for high number of I/O operations,
- management via workspaces,
- All HPC nodes share this file system.

Attention: Data might get lost.

BeeGFS file system(s)

Fast parallel file systems for partition

- large parallel file system for high number of I/O operations,
- based on NVMe,
- management via workspaces,
- haswell nodes within “island 6“ and newer nodes (accessible with additions Slurm option, see below) share this.
- only fast shared filesystem available on IBM Power Nodes!

Project specific BeeGFS file systems may be granted upon request.

Attention: Data might get lost.

Lustre Scratch file system

Fastest parallel file systems (streaming) at each x86-system:

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

Attention: Data might get lost. Probably not.

Permanent file systems

Common file system for all ZIH's HPC machines:

- Very slow and small, but with multiple backups.
- Deleted files are accessible via the logical `.snapshot` directory. This directory contains weekly, daily, and hourly snapshots. Copy your file to where you need it.
- Paths to permanent storage are
  - `/home/<login>` (20 GB !) and
  - `/projects/<projectname>` - mounted read-only on compute nodes!

  with different access rights (cf. Terms of Use).
- All HPC systems of ZIH share these file systems.

Warm archive

Large storage at each x86-system:

- large parallel file system for moderately high bandwidth,
- management via workspaces,
- all HPC nodes share this file system,
- mounted read-only on compute nodes

Archive

Common tape based file system:

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

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.

Avalable file systems for job selection

Not all global filesystems are always available. Let the batchsystem decide!

- A cron job automatically checks the availability of each mounted system on every node and sets Slurm features accordingly.
- This feature can be selected at job submission with the additional option `--constraint` or `-C`
- Example: `srun -C fs_lustre_scratch2 ...`

Available file system features (`https://doc.zih..../Slurmfeatures`) are:

feature	description
`fs_lustre_scratch2`	`/lustre/scratch2 mounted` read-write (`/scratch`)
`fs_lustre_ssd`	`/lustre/ssd` mounted read-write
`fs_warm_archive_ws`	`/warm_archive/ws mounted` read-only
`fs_beegfs_global0`	`/beegfs/global0` mounted read-write

Project specific BeeGFS file systems may be granted upon request.

Data management

Automated workflows vs. manual control

- 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?

- 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.

Workspaces

Tool for users to manage their storage demands:
`https://doc.zih..../WorkSpaces`

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

Storage system	**Lifetime**	**Remarks**
beegfs	30 days	High-IOPS file system on NVMes
ssd	30 days	High-IOPS file system on SSDs
scratch	100 days	High streaming bandwidth on disks.
warm_archive	1 year	Capacity file system on disks.

Workspace - examples

Available workspaces:

```
mark@tauruslogin3:~> ws_find -l
available filesystems:
warm_archive
scratch
ssd
```

Allocation:

```
mark@tauruslogin3:~> ws_allocate -F ssd SPECint
Info: creating workspace.
/lustre/ssd/ws/mark-SPECint
remaining extensions  : 2
remaining time in days: 5
```

Notification:

```
mark@tauruslogin3:~> ws_send_ical -m ulf.markwardt@tu-dresden.de \
-F ssd SPECint
Sent reminder for workspace SPECint to ulf.markwardt@tu-dresden.de
please do not forget to accept invitation
```

→Calender invitation: “Workspace SPECint will be deleted on host Taurus”

Workspace - examples

List all allocated workspaces

```
mark@tauruslogin3:~> ws_list
id: SPECint
     workspace directory  : /lustre/ssd/ws/mark-SPECint
     remaining time       : 4 days 23 hours
     creation time        : Wed Sep 18 09:41:08 2019
     expiration date      : Mon Sep 23 09:41:08 2019
     filesystem name      : ssd
     available extensions : 2
```

Extend the life time of a workspace

```
mark@tauruslogin3:~> ws_extend -F ssd SPECint 10
Info: extending workspace.
/lustre/ssd/ws/mark-SPECint
remaining extensions  : 1
remaining time in days: 10
```

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

```
mark@tauruslogin3:~> ws_list -F ssd
id: SPECint
     workspace directory  : /lustre/ssd/ws/mark-SPECint
     remaining time       : 9 days 23 hours
     creation time        : Wed Sep 18 09:43:01 2019
     expiration date      : Sat Sep 28 09:43:01 2019
     filesystem name      : ssd
     available extensions : 1
```

Workspace - examples

Manually delete your workspace with

`ws_release -F <file system> <workspace name>`

Workspace within a job

```
#!/bin/bash
#SBATCH --partition=haswell
...
COMPUTE_DIR=gaussian_$SLURM_JOB_ID
ws_allocate -F ssd $COMPUTE_DIR 7
export GAUSS_SCRDIR=/ssd/ws/$USER-$COMPUTE_DIR
srun g16 inputfile.gjf logfile.log

#Delete ASAP
test -d $GAUSS_SCRDIR && rm -rf $GAUSS_SCRDIR/*
ws_release -F ssd $COMPUTE_DIR
```

For “small” number of files: Delete as soon as possible using “rm”

Workspace

Expiration of workspaces

- Expired workspaces are moved automatically to another location.
- After a grace period (30...60d) they are marked for final deletion.
- During this time workspaces can be restored by the user using `ws_restore`.
- Deletion is final - pay attention to expiration date.

Data transfer

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

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

Data transfer

Mounted file systems:

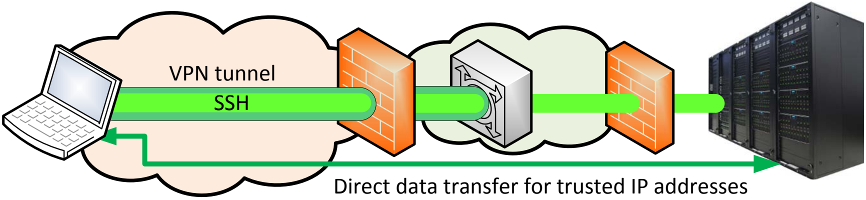
- all global HPC file systems (/home, /projects, /beegfs/global0,...)
- /warm_archive
- /archiv gateway to the tape archive
- below /grp selected NFS shares aka “Gruppenlaufwerke”

Very simple usage like

```
dttar -czf /warm_archive/ws/jurenz-sim/results_20190820.tgz \
           /scratch/ws/jurenz-sim21/results
dtls /archiv/mark
```

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

Remark: Do not use rsync to sync a large number of files to the warm archive.

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
    - Access to HPC systems at ZIH
    - Compute hardware
    - HPC file systems
    - Software environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
5. HPC Support

Modules

Installed software is organized in modules.

A module is a user interface, that:

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

Private modules files are possible (e.g. group-wide installed software).

Installed Software

Installed software (appr. 2000 modules) can be found at...
`https://doc.zih..../SoftwareModulesList` (daily updated)

abaqus, abinit, ace, adolc, afni, amber, ansys, ansysem, asm, autoconf, automake, autotools, AVS, bazel, bison, boost, bullxde, bullxmpi, casita, ceph, cereal, cg, clFFT, cmake, collectl, comsol, conn, cp2k, ctool, cube, cuda, cusp, cython, dalton, darshan, dash, dataheap, ddt, dftb+, dmtcp, doxygen, dune, dyninst, eigen, eman2, ensight, extrae, fftw, flex, fme, freecad, freeglut, freesurfer, fsl, ga, gamess, gams, gaussian, gautomatch, gcc, gcl, gcovr, gctf, gdb, gdk, geany, ghc, git, glib, gmock, gnuplot, gpaw, gperftools, gpi2, gpudevkit, grid, gromacs, gsl, gulp, gurobi, h5utils, haskell, hdeem, hdf5, hoomd, hyperdex, imagemagick, intel, intelmpi, iotop, iotrack, java, julia, knime, lammps, lbfgsb, libnbc, liggghts, llvm, lo2s, ls, ls-dyna, lumerical, m4, map, mathematica, matlab, maxima, med, meep, mercurial, metis, mkl, modenv, motioncor2, mpb, mpi4py, mpirt, mumps, must, mvapich2, mxm, mysql, namd, nedit, netcdf, netlogo, numeca, nwchem, octave, octopus, opencl, openems, openfoam, openmpi, opentelemac, orca, oscar, otf2, papi, paraview, parmetis, pathscale, pdt, petsc, pgi, pigz, protobuf, pycuda, pyslurm, python, q, qt, quantumespresso, r, redis, relion, ripgrep, root, ruby, samrai, scala, scalasca, scons, scorep, sftp, shifter, siesta, singularity, sionlib, siox, spm, spm12, spparks, sqlite3, stack, star, suitesparse, superlu, svn, swig, swipl, tcl, tcltk, tecplot360, tesseract, texinfo, theodore, tiff, tinker, tmux, totalview, trace, trilinos, turbomole, valgrind, vampir, vampirtrace, vasp, visit, vmd, vtk, wannier90, wget, wxwidgets, zlib

Module environments

Different module environments:

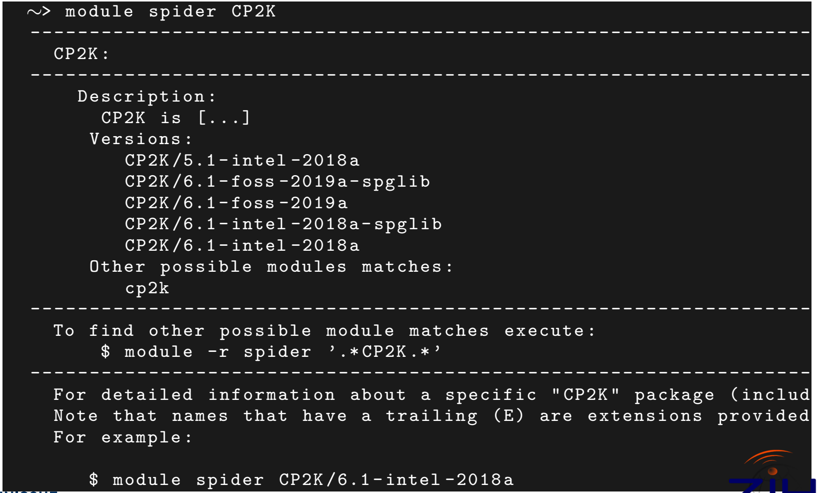
- scs5 - for software built from “recipes” with EasyBuild (default), x86 nodes
- ml - software for machine learning nodes **only for IBM Power nodes**
- hiera - hierachical module environment (more details later)

```
 ~ > module load modenv/scs5
  The following have been reloaded with a version change:
    1) modenv/classic => modenv/scs5

 ~ > module load modenv/ml
  The following have been reloaded with a version change:
    1) modenv/scs5 => modenv/ml
```

Module usage

- Check `https://doc.zih..../SoftwareModulesList`
- Identify module and version (case-sensitive!)

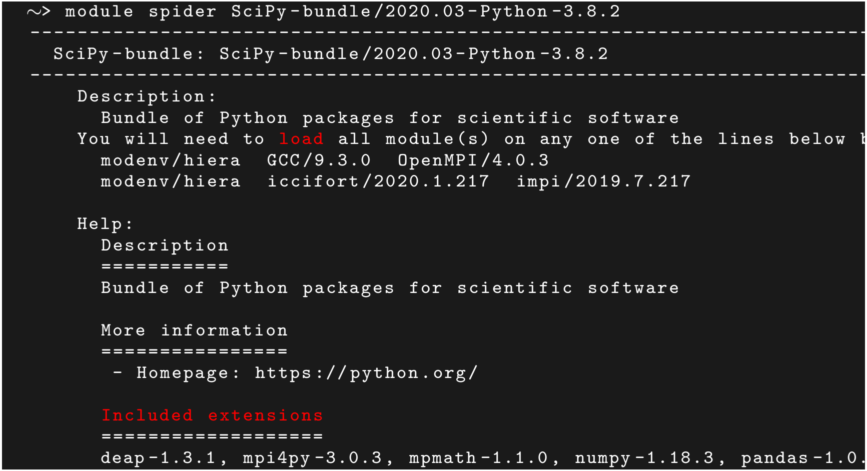


```
~> module spider CP2K
-----------------------------------------------------------------
  CP2K:
-----------------------------------------------------------------
    Description:
      CP2K is [...]
     Versions:
        CP2K/5.1-intel-2018a
        CP2K/6.1-foss-2019a-spglib
        CP2K/6.1-foss-2019a
        CP2K/6.1-intel-2018a-spglib
        CP2K/6.1-intel-2018a
     Other possible modules matches:
        cp2k
-----------------------------------------------------------------
  To find other possible module matches execute:
      $ module -r spider '.*CP2K.*'
-----------------------------------------------------------------
  For detailed information about a specific "CP2K" package (includ
  Note that names that have a trailing (E) are extensions provided
  For example:

     $ module spider CP2K/6.1-intel-2018a
```

Module usage

Information from `module spider`



```
 ~> module spider SciPy-bundle/2020.03-Python-3.8.2
 -----------------------------------------------------------------------
   SciPy-bundle: SciPy-bundle/2020.03-Python-3.8.2
 -----------------------------------------------------------------------
     Description:
       Bundle of Python packages for scientific software
     You will need to load all module(s) on any one of the lines below b
       modenv/hiera  GCC/9.3.0  OpenMPI/4.0.3
       modenv/hiera  iccifort/2020.1.217  impi/2019.7.217

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

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

       Included extensions
       ===================
       deap-1.3.1, mpi4py-3.0.3, mpmath-1.1.0, numpy-1.18.3, pandas-1.0.
```

Modules for different architectures

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

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

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

Module commands

`module avail` - lists all available modules (in the current module environment)

`module spider` - lists all available modules (across all module environments)

`module list` - lists all currently loaded modules

`module show <modname>` - display informations about `<modname>`

`module load <modname>` - loads module `modname`

`module save` - saves the current modules, to be reloaded at the next login

`module rm <modname>` - unloads module `modname`

`module purge` - unloads all modules

Module hiera

Consists of three levels

- **Core** - Base modules without dependencies + compilers and toolchain modules. Always visible.
- **Compiler** - Software depending on a certain compiler or toolchain becomes visible after loading the module.
- **MPI** - Loading an MPI library makes software available that was built with this MPI.

Modules for HPC applications

Loading compiler, MPI, and numeric library (MKL)

```
~> module --force purge
~> module load modenv/hiera
~> module load intel
Module intel/2020a and 8 dependencies loaded.

~> module li
Currently Loaded Modules:
  1) modenv/hiera  (S)   5) iccifort/2020.1.217   9) imkl/2020.1.217
  2) GCCcore/9.3.0       6) numactl/2.0.13       10) intel/2020a
  3) zlib/1.2.11         7) UCX/1.8.0
  4) binutils/2.34       8) impi/2019.7.217

  Where:
   S:  Module is Sticky, requires --force to unload or purge
```

```
~> mpicc -show
icc -I/sw/installed/impi/2019.7.217-iccifort-2020.1.217/intel64/include
-L/sw/installed/impi/2019.7.217-iccifort-2020.1.217/intel64/lib/release
-L/sw/installed/impi/2019.7.217-iccifort-2020.1.217/intel64/lib
-Xlinker --enable-new-dtags -Xlinker -rpath
-Xlinker /sw/installed/impi/2019.7.217-iccifort-2020.1.217/intel64/lib/
-Xlinker -rpath -Xlinker /sw/installed/impi/2019.7.217-iccifort-2020.1.
```

```
~> mpicc hello.c
~> srun -n 4 -t 1 -N 1 --mem-per-cpu=500 ./a.out
```

Remarks

Commercial codes requiring licenses (Matlab, Ansys)

- basic principle: do not use them extensively, we have only a limited number of licenses!
- Matlab: use the Matlab compiler (`https://doc.zih....//Mathematics`)

Containers

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

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
   - General
   - Slurm examples
4. Software Development at ZIH's HPC systems
5. HPC Support

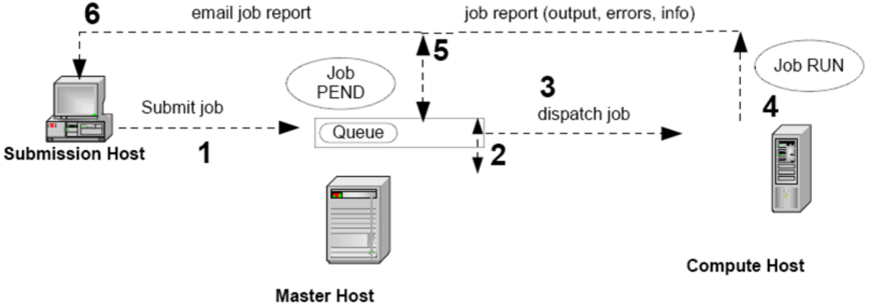
Overview

Why do we need a batchsystem?

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

Workflow of a batch system

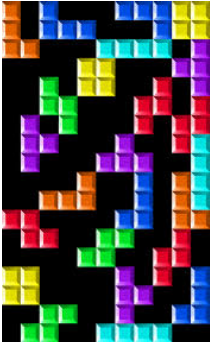
Agreed, we need a batchsystem.



Multi-dimensional optimizations

Optimization goals:

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



Optimization is NP-hard → heuristics allowed.

Useful functions of a batchsystem

Basic user functions:

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

Additional functions:

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

Job submission: required information

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

- requirements: cores, memory per core, (nodes), additional resources (GPU)
- 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 frequently!
kill a job	`scancel`
cluster status	`sinfo` - Not frequently!
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.zih..../Slurm`

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

Slurm partitions

- haswell – largest compute partition, Intel x86_64 based, most software runs here. Differenz sizes of RAM managed by job submit plugin.
- broadwell – 32 nodes comparable to haswell. Intel x86_64 based. Most software runs here.
- romeo – powerful compute partition, AMD x86_64 based, most software should run here.
- julia – largest SMP node, Intel x86_64 based. For memory-consuming software. Don't use OpenMPI.
- gpu2 – GPU partition, Intel x86_64 based. Most GPU software runs here.
- ml – powerful GPU partition for Machine Learning. IBM Power based. Only special software runs here.
- gpu3 – GPU partition, Intel x86_64 based. Coming soon.
- hpdlf – GPU partition for deep learning project, Intel x86_64 based. Most GPU software runs here.

- interactive – haswell nodes for interactive jobs
- gpu2-interactive – gpu2 nodes for interactive jobs
- romeo-interactive – romeo nodes for interactive jobs
- haswell64ht – haswell nodes with activated HyperThreads

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
   - General
   - Slurm examples
4. Software Development at ZIH's HPC systems
5. HPC Support

Slurm examples

Slurm interactive example:

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

Slurm X11 example:

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

Remarks:

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

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

Slurm examples

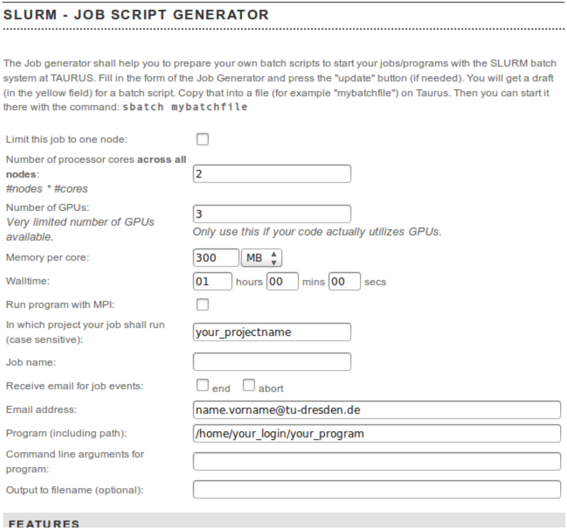
Requesting multiple GPU cards

```
#SBATCH --partition=gpu2
#SBATCH --time=4:00:00
#SBATCH --job-name=MyGPUJob
#SBATCH --nodes=16
#SBATCH --ntasks-per-node=2
#SBATCH --cpus-per-task=8
#SBATCH --gres=gpu:2
#SBATCH --mem-per-cpu=3014
#SBATCH --mail-type=END
#SBATCH --mail-user=ulf.markwardt@tu-dresden.de

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

Slurm generator

Good starting point: `https://doc.zih..../Slurmgenerator`



SLURM - JOB SCRIPT GENERATOR

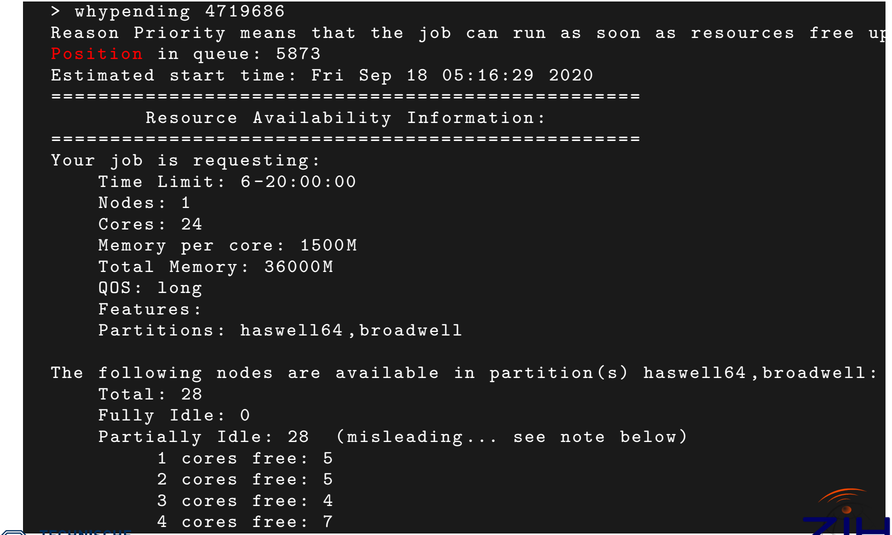
The Job generator shall help you to prepare your own batch scripts to start your jobs/programs with the SLURM batch system at TAURUS. Fill in the form of the Job Generator and press the "update" button (if needed). You will get a draft (in the yellow field) for a batch script. Copy that into a file (for example "mybatchfile") on Taurus. Then you can start it there with the command: `sbatch mybatchfile`

Limit this job to one node:	□
Number of processor cores **across all nodes**: *#nodes * #cores*	2
Number of GPUs: *Very limited number of GPUs available.*	3 *Only use this if your code actually utilizes GPUs.*
Memory per core:	300 MB
Walltime:	01 hours 00 mins 00 secs
Run program with MPI:	□
In which project your job shall run (case sensitive):	your_projectname
Job name:	
Receive email for job events:	□ end □ abort
Email address:	name.vorname@tu-dresden.de
Program (including path):	/home/your_login/your_program
Command line arguments for program:	
Output to filename (optional):	

FEATURES

Slurm: Job monitoring

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



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

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

Slurm: Fair share monitoring

Is my fair share really so low???

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

Project information

Look at the login screen. Or `showquota`

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

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

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

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

What is fair...?

Fair share of a project is based on

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

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

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

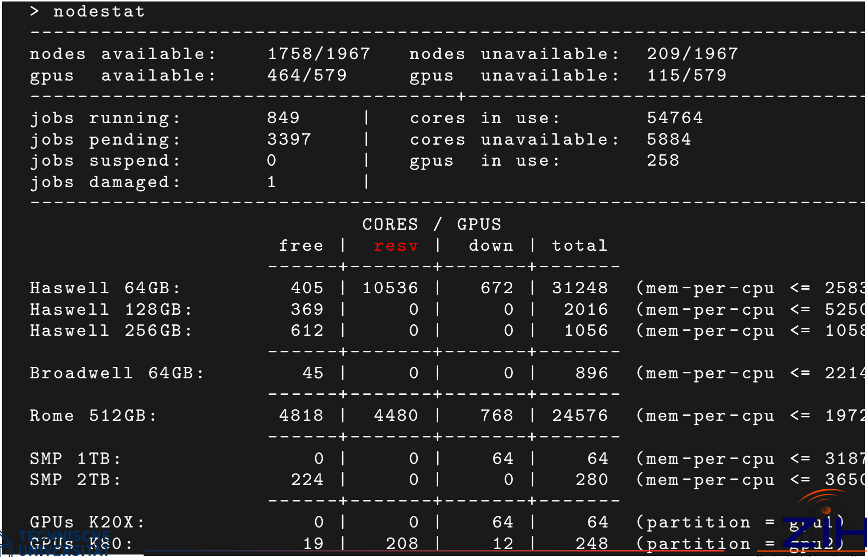
See: `https://slurm.schedmd.com/priority_multifactor.html`

TECHNISCHE UNIVERSITÄT DRESDEN

ZIH Center for Information Services & High Performance Computing

System information

Look at the login screen. Or `nodestat`



```
> nodestat
----------------------------------------------------------------------
nodes available:    1758/1967   nodes unavailable:  209/1967
gpus  available:    464/579     gpus  unavailable:  115/579
-----------------------------------+----------------------------------
jobs running:       849      |   cores in use:       54764
jobs pending:       3397     |   cores unavailable:  5884
jobs suspend:       0        |   gpus  in use:       258
jobs damaged:       1        |
----------------------------------------------------------------------
                              CORES / GPUS
                    free |  resv |  down | total
                  -------+-------+-------+-------
Haswell 64GB:        405 | 10536 |   672 | 31248  (mem-per-cpu <= 2583
Haswell 128GB:       369 |     0 |     0 |  2016  (mem-per-cpu <= 5250
Haswell 256GB:       612 |     0 |     0 |  1056  (mem-per-cpu <= 1058
                  -------+-------+-------+-------
Broadwell 64GB:       45 |     0 |     0 |   896  (mem-per-cpu <= 2214
                  -------+-------+-------+-------
Rome 512GB:         4818 |  4480 |   768 | 24576  (mem-per-cpu <= 1972
                  -------+-------+-------+-------
SMP 1TB:               0 |     0 |    64 |    64  (mem-per-cpu <= 3187
SMP 2TB:             224 |     0 |     0 |   280  (mem-per-cpu <= 3650
                  -------+-------+-------+-------
GPUs K20X:             0 |     0 |    64 |    64  (partition = gpu1)
GPUs K80:             19 |   208 |    12 |   248  (partition = gpu2)
```

Simple job monitoring

Job information

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

Detailled job monitoring

Detailled job information

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

Slurm tools

`scontrol show ...`

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

More tools

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

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

Still... not starting

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

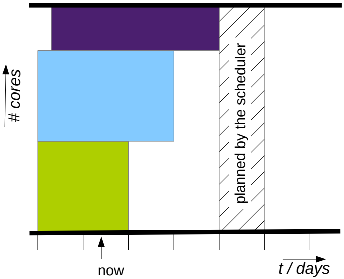
- Maybe a reservation prevents my job from starting (`sinfo -T`)
- Maybe an older large job is scheduled and waits for resources:

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

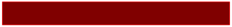
  Here is job 4832990 with a very high priority, scheduled for a certain time (see `scontrol show job 4832990`) . If my job would finish before that one it could be backfilled.

- Maybe fragmentation would be too high.

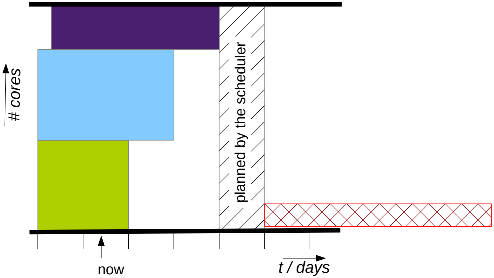
Backfilling



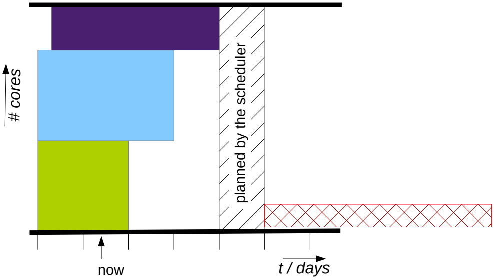
My job to be placed:



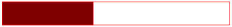
Backfilling



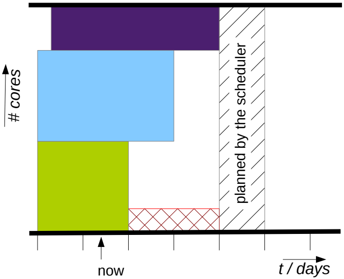
Backfilling



I know my job better:

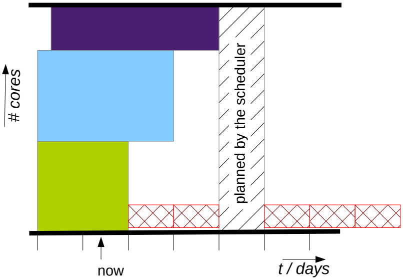


Backfilling



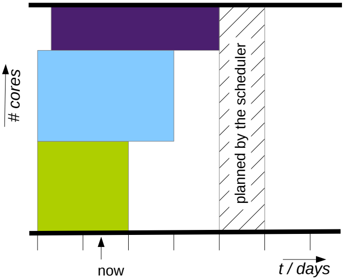
Estimate the maximum run-time of your job!

Backfilling

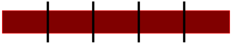


Try to use shorter jobs!

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)
- If application does not support checkpointing:
  1. `module load dmtcp`
  2. modify your batch script like this:
     `srun dmtcp_launch --ib --rm ./my-mpi-application`
  3. run the modified script like `dmtcp_sbatch -i 28000,800 mybatch.sh`
     This creates chain jobs of length 28000 s, planning 800 s for I/O
- more details at `https://doc.zih..../CheckpointRestart`

Efficient use of resources

Make use of heterogeneity of the system

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

Let Taurus work!

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

Normal jobs

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

Run-time considerations

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

Nelle's Pipeline III

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

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

Nelle's Pipeline III

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

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

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

Nelle's Pipeline III

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

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

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

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

Working on Taurus

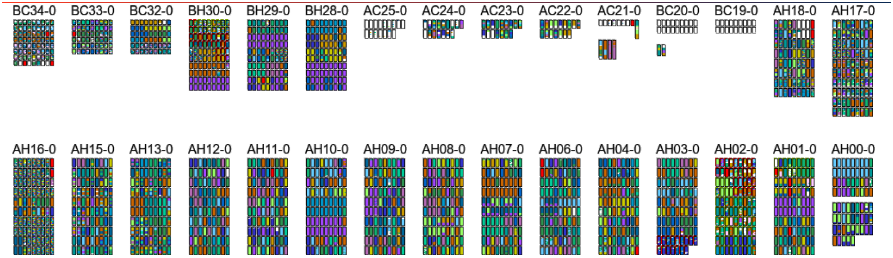
Interactive jobs

- for pre- or post- processing, compiling and testing / developmemt
- can use terminal or GUI via X11
- partitions `interactive`, `romeo-interactive` and `gpu2-interactive` are reserved for these jobs.
- check options for “HPC in a Browser” (`https://doc.zih..../VirtualDesktops`)

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

- licensed product installed on Taurus
- documentation in (`https://doc.zih....//DesktopCloudVisualization`)
- e.g. rendering with ParaView using GPUs

Availability



High utilization - good for "us" - bad for the users?

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

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
   - Compiling
   - Tools
5. HPC Support

Questionnaire

Are you already an HPC user...?

- Ⓐ yes
- Ⓑ no

Questionnaire

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

- Ⓐ chemistry and materials science
- Ⓑ life sciences
- Ⓒ physics
- Ⓓ mechanical engineering
- Ⓔ earth sciences

If none of the above matches: abstain.

Questionnaire

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

- Ⓐ commercial software
- Ⓑ community software
- Ⓒ “self” developed codes

Available compilers

Which compilers are installed?

- Starting point: `https://doc.zih..../Compilers`
- Up-to-date information: `https://doc.zih..../SoftwareModulesList`

Available compilers

Which compilers are installed?

- Starting point: `https://doc.zih..../Compilers`
- Up-to-date information: `https://doc.zih..../SoftwareModulesList`

Which one is "the best"?

- Newer versions are better adapted to modern hardware.
- Newer versions implement more features (e.g. OpenMP 4.0, C++11, Fortran 2010).
- GNU compilers are most portable.
- Listen to hardware vendors. (But not always.)

Available compilers

Which compilers are installed?

- Starting point: `https://doc.zih..../Compilers`
- Up-to-date information: `https://doc.zih..../SoftwareModulesList`

Which one is "the best"?

- Newer versions are better adapted to modern hardware.
- Newer versions implement more features (e.g. OpenMP 4.0, C++11, Fortran 2010).
- GNU compilers are most portable.
- Listen to hardware vendors. (But not always.)

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

Expensive operations

Time consuming operations in scientific computing:

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

Expensive operations

Time consuming operations in scientific computing:

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

How to find performance bottlenecks?

- Tools available at ZIH systems (perf, hpctoolkit, Vampir, PAPI counters),
- `https://doc.zih..../PerformanceTools`
- experience...
- Ask ZIH staff about your performance issues!

Low hanging fruits

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

- memory footprint,
- computing speed.

Low hanging fruits

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

- memory footprint,
- computing speed.

What is the needed floating point accuracy?

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

Low hanging fruits

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

- memory footprint,
- computing speed.

What is the needed floating point accuracy?

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

→ see man pages!

Options for Intel compiler: “-axavx” for Haswell and “-mavx2 -fma” for AMD ROME.
Or compile on the target system in an interactive job (SD Flex/AMD Rome/IBM Power)

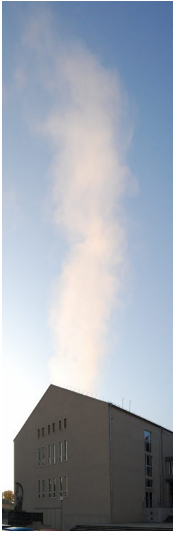
Intel training course: `https://doc.zih..../SoftwareDevelopment`

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
   - Compiling
   - Tools
5. HPC Support

On HPC systems: Efficient code is essential!

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



Profiling

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

Profiling allows you to learn

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

while it was executing.

→ Identify slow code – redesign it!

Profiling

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

Profiling allows you to learn

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

while it was executing.

→ Identify slow code – redesign it!

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

Using GNU's gprof

part of GCC available on most unix systems

- compiling and linking (`-pg`):

  ```
  g++ -pg my_prog.cpp -o my_prog
  ```

- execute to produce profiling information:

  ```
  ./my_prog
  ```

- get human readable information:

  ```
  gprof my_prog gmon.out > analysis.txt
  ```

- analysis: `vi analysis.txt`

```
Flat profile:

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

Comment: see also Intel slides.

SLURM profiling with HDF5 (on Taurus)

SLURM offers the option to gather profiling data from every task/node of the job.

- task data, i.e. CPU frequency, CPU utilization, memory consumption, I/O
- energy consumption of the nodes - subject of HDEEM research project
- Infiniband data (currently deactivated)
- Lustre filesystem data (currently deactivated)

The aggregated data is stored in an HDF5 file in `/scratch/profiling/${USER}`.

SLURM profiling with HDF5 (on Taurus)

SLURM offers the option to gather profiling data from every task/node of the job.

- task data, i.e. CPU frequency, CPU utilization, memory consumption, I/O
- energy consumption of the nodes - subject of HDEEM research project
- Infiniband data (currently deactivated)
- Lustre filesystem data (currently deactivated)

The aggregated data is stored in an HDF5 file in `/scratch/profiling/${USER}`.

Caution:

- Profiling data may be quite large. Please use `/scratch` or `/tmp`, not HOME.
- Don't forget to remove the `--profile` option for production runs!

SLURM profiling with HDF5

Example

- Create task profiling data:

  ```
  srun -t 20 --profile=Task --mem-per-cpu=2001 \
    --acctg-freq=5,task=5 \
    ./memco-sleep --min 100 --max 2000 --threads 1 --steps 2
  ```

- Merge the node local files (in `/scratch/profiling/${USER}`) to a single file (maybe time-consuming):
  - login node: `sh5util -j <JOBID> -o profile.h5`
  - in jobscripts:

    ```
    sh5util -j ${SLURM_JOBID} -o /scratch/ws/mark-prof/profile.h5
    ```

External information:

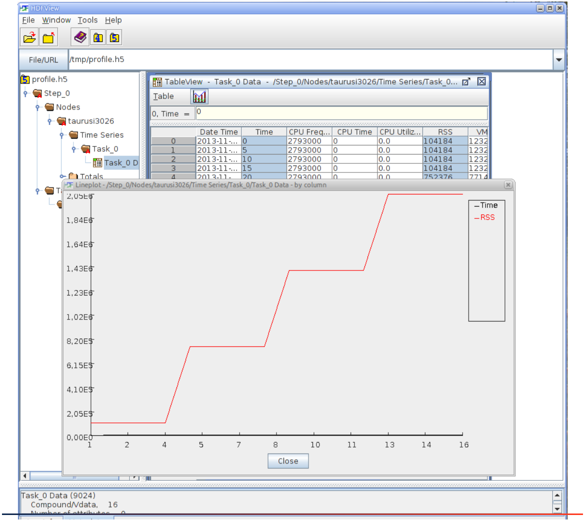
http://slurm.schedmd.com/hdf5_profile_user_guide.html

http://slurm.schedmd.com/sh5util.html

SLURM profiling with HDF5

View example data

```
module load hdf5/hdfview; hdfview.sh /scratch/ws/mark-prof/profile.h5
```



0, Time = 0

	Date Time	Time	CPU Freq...	CPU Time	CPU Utiliz...	RSS	VM
0	2013-11-...	0	2793000	0	0.0	104184	1232
1	2013-11-...	5	2793000	0	0.0	104184	1232
2	2013-11-...	10	2793000	0	0.0	104184	1232
3	2013-11-...	15	2793000	0	0.0	104184	1232
4	2013-11-	20	2793000	0	0.0	752376	7714

```
Task_0 Data (9024)
   Compound/Vdata,   16
   Number of attributes = 0
```

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
5. HPC Support
   - Management of HPC projects
   - Channels of communication
   - Kinds of support
   - Beyond support

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 3500 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

Management of HPC projects

Who...

- project leader (normally chair of institute) → accountable
- project administrator (needs HPC login) → responsible

What...

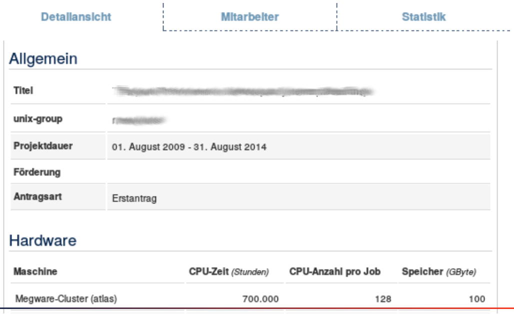
- manage members of the project (add + remove)
  (remark: external users need login..)
- check storage consumption within the project,
- retrieve data of retiring members
- contact for ZIH

Online project management

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

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

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



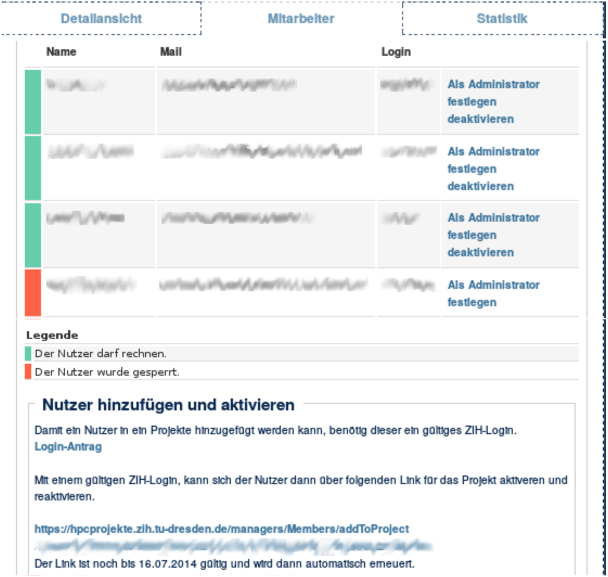
Allgemein

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

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.

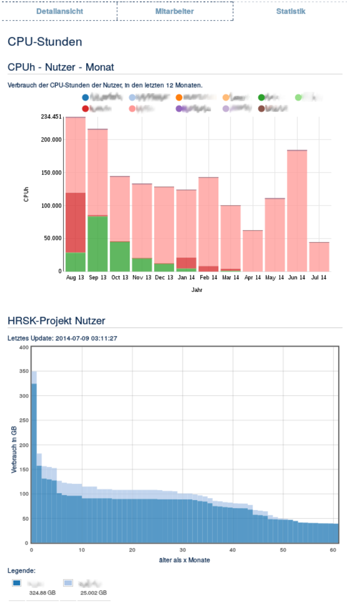
Nutzer hinzufügen und aktivieren

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

Legende:

🟦 [redacted]	🟦 [redacted]
324.88 GB	25.002 GB

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
5. HPC Support
   - Management of HPC projects
   - Channels of communication
   - Kinds of support
   - Beyond support

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
      `hpcsupport@zih.tu-dresden.de`

      please: use your `...@tu-dresden` address as sender and voluntarily include: name of HPC system, job ID...
    - phone: service desk (0351) 463 40000
    - planned: self service portal
- personal contact
  - phone call, email, talk at the Mensa
  - socializing is fine... but: risk of forgetting

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
5. HPC Support
   - Management of HPC projects
   - Channels of communication
   - Kinds of support
   - Beyond support

Kinds of support

HPC management topics:

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

HPC usage requests:

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

Kinds of support

HPC Software questions:

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

→ restrictions of this support:

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

Kinds of support

Performance issues

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

Storage and workflow issues

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

Kinds of support

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

ScaDS.AI
DRESDEN LEIPZIG

- https://www.scads.de/services
- services@scads.de

HPC Support Team for Taurus

HPC support group

- Claudia Schmidt (project management)
- Matthias Kräußlein (accounting and project infrastructure)
- Lars Jitschin,Loc Nguyen Dang Duc, Etienne Keller
- Danny Rotscher (Slurm, technical support)
- Ulf Markwardt (Slurm, technical support... head of the group)

Agenda

1. Linux from the command line
2. HPC Environment at ZIH
3. Batch System
4. Software Development at ZIH's HPC systems
5. HPC Support
   - Management of HPC projects
   - Channels of communication
   - Kinds of support
   - Beyond support

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 Dresden Leipzig
- Nvidia CCoE (GPU), IPCC (Xeon Phi)

Research topics

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

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

You can help

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

The computations were performed on an HPC system at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden.

We thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of compute resources.

Recapitulation

Most important topics:

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

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