

**REGIONALES RECHENZENTRUM
ERLANGEN [RRZE]**



High Performance Computing in a Nutshell – Part 2

HPC Services, RRZE

Agenda

- Accessing HPC systems
 - Getting an account
 - SSH
 - NoMachine NX
- Systems at RRZE
- Working with data
 - File systems and their properties
 - Data transfer
- Software
 - Nodes vs. frontends
 - The module system
- Running jobs
 - Interactive work
 - Batch jobs & batch scripts
- Some dos and don'ts



GETTING AN ACCOUNT



Accounts

- You need a **separate account** (not your IdM account)
- HPC account application form
- **Account can access all HPC systems at RRZE**
- Ask your local RRZE contact person for help
- Expiration date!
- If you change your affiliation, you need a new HPC account
 - › Data migration may be required

Antrag auf Nutzung von HPC-Ressourcen am RRZE High Performance Computing

Antrag per FAX an 09131-85-29966 oder als Scan an rzze-central@fau.de oder per E-Mail an rzze@fau.de (bestehender HPC-Account)
Name und Vorname: _____

☐ Neuantrag ☐ Änderung/Verlängerung

IDM-Account (bestehender) HPC-Account

Antragsteller: ☐ Frau ☐ Herr ☐ Titel

Vorname: _____
Nachname: _____
E-Mail: _____
Telefon: _____
Nationalität(en): _____
HPC-Abgabedatum bis (DD.MM.JJJJ): _____

Bei Unklarheiten support-hpc@fau.de kontaktieren

HPC-Zielsysteme: _____
typische Jobgröße: _____
erwart. benötigte Rechenzeit: _____
benötigter Speicherplatz: _____

Auftraggeber: FAU-Ordnr. _____
Lehrstuhl- oder Instituts- stempel und -unterschrift: _____
RRZE-Kontaktperson: _____
Art der Anwendung / Name der Applikation: _____

Art/Finanzierung des Projekts: _____

Abrechnung der Rechenzeit über ☐ bestehende Kundennummer ☐ neue Kundennummer für folgendes Rechenzeitprojekt

Kulu: _____

Titel des Forschungsvorhabens: _____
für das Projekt insgesamt benötigte Rechenzeit: _____
Bewilligungszeitraum / Projektlaufzeit: _____
fördernde Institution und Förderkennzeichen: _____
Kurzbeschreibung der HPC-Aktivitäten im Forschungsvorhaben: _____
Wurde der Rechenzeitbedarf mit dem RRZE abgestimmt und im Antrag dargestellt?
Haben die Gutachter der Förderinstitution dazu Stellung genommen?

Personenbezogene Daten im Sinne der geltenden Datenschutzgesetze dürfen unter dieser Benutzerkennung nicht ohne Sondergenehmigung seitens des RRZE und des Datenschutzbeauftragten verarbeitet werden!
Dem Antragsteller ist bekannt, dass er sich durch eine missbräuchliche Benutzung der Informationsverarbeitungssysteme strafbar machen kann und dass beim Vorliegen eines Missbrauchs grundsätzlich Strafantrag gestellt wird. Des Weiteren bemüht sich der Antragsteller, die HPC-Systeme effizient zu nutzen und gängige HPC-Praktiken zu beachten.

Benutzerrichtlinien:
<https://www.rzze.fau.de/infocenter/rahmenbedingungen/richtlinien/benutzerrichtlinien/>

Antragsteller und Auftraggeber erklärt hiermit, von den Benutzungsrichtlinien sowie den ergänzenden Hinweisen auf der Rückseite dieses Antrags Kenntnis genommen zu haben.

Ort, Datum: _____ Unterschrift Antragsteller: _____
Unterschrift Auftraggeber/ Kontaktperson: _____
IDM-Kennung Auftraggeber/Kontaktpers.: _____

The HPC account application form (I)

Antrag auf Nutzung von HPC-Ressourcen am RRZE  High Performance Computing 

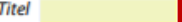
Stand 07/2017

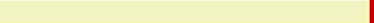
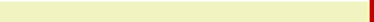
Antrag per FAX an 09131-85-29966
oder als Scan an rrze-zentrales@fau.de
oder per (Haus)post an RRZE/Serviceheke,
Martensstr. 1, 91058 Erlangen

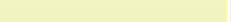
☒ Neuantrag ☐ Änderung/Verlängerung

IDM-Account  Your IdM account

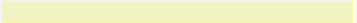
(bestehender) HPC-Account 

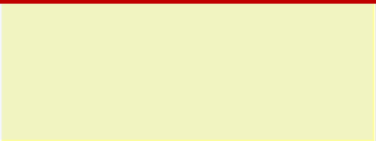
Antragsteller: ☐ Frau ☐ Herr Titel 

Vorname 
Nachname 
E-Mail 
Telefon 
Nationalität(en) 
HPC-Ablaufdatum bis DD.MM.JJJJ 
Bei Unklarheiten vorab support-hpc@fau.de kontaktieren

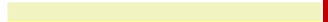
Auftraggeber: FAU-OrgNr. 

Lehrstuhl- oder
Instituts-
stempel
und
-anschrift  Chair data & seal

RRZE-Kontaktperson 

Art der
Anwendung /
Name der
Applikation  Very brief
description of
what you want to
do

Systems, requirements (brief!)

HPC-Zielsysteme 
typische Jobgröße 
insges. benötigte Rechenzeit 
benötigter Speicherplatz 

The HPC account application form (II)

Type of project:

- Education (master/bachelor)
- Standard (FAU employee)
- Research grant (BMBF, DFG, EU)
- Industry

The screenshot shows a form titled 'Art/Finanzierung des Projekts'. It includes a section for 'Abrechnung der Rechenzeit über' with radio buttons for 'bestehende Kundennummer' and 'neue Kundennummer für folgendes Rechenzeitprojekt'. Below this is a section for 'Titel des Forschungsvorhabens' and 'Bewilligungszeitraum / Projektklaufzeit'. A large red box highlights a section for 'Kurzbeschreibung der HPC-Aktivitäten im Forschungsvorhaben' and 'Wurde der Rechenzeitbedarf mit dem RRZE abgestimmt und im Antrag dargestellt?'. To the right of the form, there are yellow boxes with text: 'RRZE customer ID (ask contact person)' and 'More detailed info on third-party funded projects'.

RRZE customer ID
(ask contact
person)

More detailed
info on third-party
funded projects

Personenbezogene Daten im Sinne der geltenden Datenschutzgesetze dürfen unter dieser Benutzerkennung nicht ohne Sondergenehmigung seitens des RRZE und des Datenschutzbeauftragten verarbeitet werden!

Dem Antragsteller ist bekannt, dass er sich durch eine missbräuchliche Benutzung der Informationsverarbeitungssysteme strafbar machen kann und dass beim Vorliegen eines Missbrauchs grundsätzlich Strafantrag gestellt wird. Des weiteren bemüht sich der Antragsteller, die HPC-Systeme effizient zu nutzen und gängige HPC-Praktiken zu beachten.

Benutzerrichtlinien:

<https://www.rrze.fau.de/infocenter/rahmenbedingungen/richtlinien/benutzungsrichtlinien/>

Antragsteller und Auftraggeber erklärt hiermit, von den Benutzungsrichtlinien sowie den ergänzenden Hinweisen auf der Rückseite dieses Antrags Kenntnis genommen zu haben.

This section contains fields for 'Ort, Datum', 'Unterschrift Antragsteller', 'Unterschrift Auftraggeber/Kontaktperson', and 'IdM-Kennung Auftraggeber/Kontaktpers.'. A red box highlights the signature fields. To the right of the form, there is a yellow box with text: 'Signatures: you and your boss or contact person'.

Date & place

Signatures: you
and your boss or
contact person

Is there a fee for compute cycles?

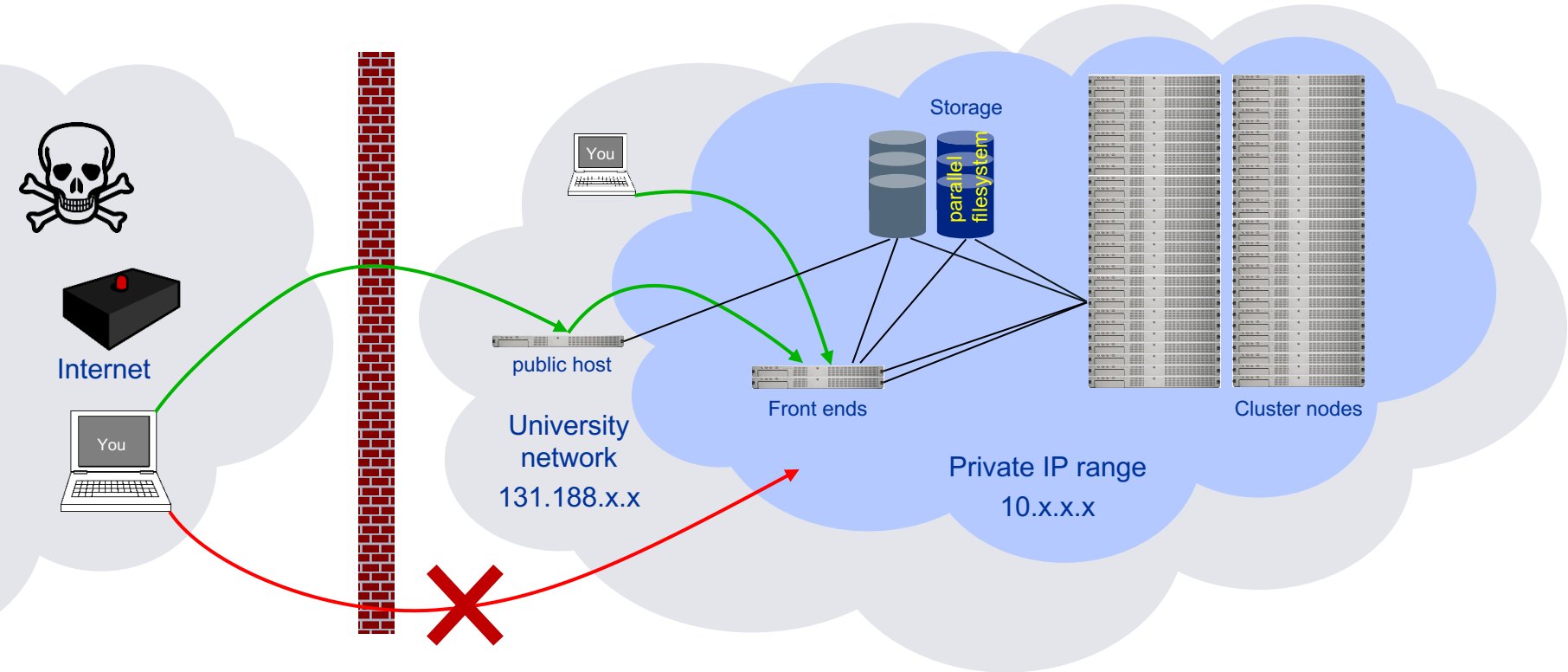
- CPU cycles are free for FAU-funded research and education
 - No special permissions, priorities, quotas,...
- DFG/BMBF projects etc.
 - Consult with HPC@RRZE before submitting the DFG proposal
- Industry
 - Set up contract with RRZE
 - Case by case basis
 - There is an official price list:
<https://www.rrze.fau.de/infocenter/preise-kosten/#hpc>



LOGGING INTO HPC SYSTEMS AT RRZE



Accessing a cluster system



Cluster access

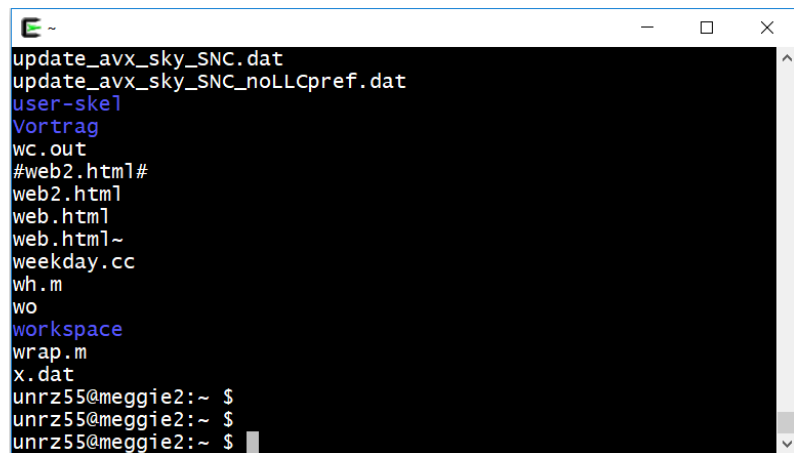
- Primary point of contact: cluster front-ends
 - `woody.rrze.uni-erlangen.de`
 - `emmy.rrze.uni-erlangen.de`
 - `meggie.rrze.uni-erlangen.de`
 - ...
 - Only available from within FAU (private IP addresses)
- Access from outside FAU: dialog server
 - `cshpc.rrze.uni-erlangen.de`
 - The only machine with a public IP address

Secure Shell

- By default: text mode only

```
$ ssh ihpc02h@meggie.rrze.uni-erlangen.de
```

- Basic knowledge of file handling, scripting, editing, etc. under Linux is required
- X11 forwarding with option **-X** or **-Y**
 - Requires local X server

A terminal window with a black background and white text. The window title is "E ~". The text in the terminal lists several files: update_avx_sky_SNC.dat, update_avx_sky_SNC_noLLCpref.dat, user-skel, Vortrag, wc.out, #web2.html#, web2.html, web.html, web.html~, weekday.cc, wh.m, wo, workspace, wrap.m, x.dat, and then three lines of the prompt "unrz55@meggie2:~ \$".

```
E ~
update_avx_sky_SNC.dat
update_avx_sky_SNC_noLLCpref.dat
user-skel
Vortrag
wc.out
#web2.html#
web2.html
web.html
web.html~
weekday.cc
wh.m
wo
workspace
wrap.m
x.dat
unrz55@meggie2:~ $
unrz55@meggie2:~ $
unrz55@meggie2:~ $
```

Secure Shell client programs

- Linux: OpenSSH available in any distribution
- Mac: ditto
- Windows
 - Putty (<https://putty.org/>)
 - OpenSSH under Cygwin (<http://cygwin.com/>)
 - › Cygwin == GNU/Linux-like environment under Windows
 - WinSCP (data transfer only) (<https://winscp.net>)
 - MobaXterm (<https://mobaxterm.mobatek.net/>)
 - › includes an embedded X server

NoMachine NX

- Similar to Windows Terminal Server
- User can attach/detach to/from running GUI session (Linux desktop) via network
- Local client installation required (available for Win/Linux/Mac)
- Uses SSH connection “under the hood”
- NX server running on cshpc

<https://www.anleitungen.rrze.fau.de/hpc/dialogserver/>



HPC SYSTEMS AT RRZE



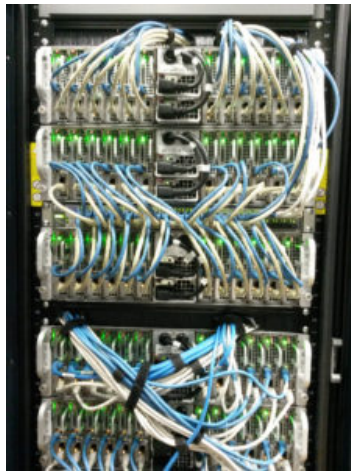
<https://www.anleitungen.rrze.fau.de/hpc/>

RRZE “Woody” cluster + TinyEth

main workhorse for *throughput*

■ Woody:

- all 176 nodes with 4 cores and high clock frequency (3.4/3.5 GHz) Intel Xeon E3-12xx v? processors
- at least 400 GB local HDD
- and Gbit only
 - › 40x Intel Sandybridge, 8 GB
 - › 72x Intel Haswell, 8 GB
 - › 8x Intel Skylake, 16 GB
 - › 56x Intel Skylake, 32 GB



■ TinyEth:

- 20 nodes (480 cores)
 - › 12 cores @ 2.66 GHz
 - › 48 GB
 - › 30/190/420 GB local HDD
- Single cores can be requested

RRZE “Emmy” cluster

main *parallel* workhorse

- 543 compute nodes (10.880 cores)
 - 2 Intel Xeon E5-2660v2 (Ivy Bridge) 2.2 GHz (10 cores)
 - 20 cores/node + SMT cores
- 64 GB main memory
- No local disks
- 16 accelerator nodes same CPUs
 - 6 nodes with 2 x NVIDIA K20 GPGPUs, 4 nodes with 1x K20
- Power consumption: ~160 KW (backdoor heat exchanger)
- Full QDR Infiniband fat tree
 - 40 GBit/s and < 2 μ s latency
- #210@Top500 (Nov 2013)

$$R_{\max} = 191 \text{ TF/s}$$



RRZE “Meggie” cluster

for scalable parallel jobs

- 728 Compute nodes (14.560 cores)
 - 2 Intel Xeon E5-2630 v4 (Broadwell)
2.2 GHz (10 cores)
 - 20 cores/node
 - 64 GB main memory
- No local disks
- Peak Performance:
 $R_{\text{peak}} = 0.5 \text{ PF/s}$
- #346@Top500 (Nov. 2016)
 $R_{\text{max}} = 0.48 \text{ PF/s}$
- Intel OmniPath network:
Up to 100 Gbit/s
- Price: € 2,5 Mio.
- Power:
120-200 kW
(depending on workload)



RRZE “TinyGPU” cluster

most nodes paid by individual user groups

- 7 nodes with 2x “Nehalem” @2.66 GHz, 24 GB, HDD, 2x GTX980
- 7 nodes with 2x “Broadwell” @2.2 GHz, 64 GB, 980 GB SSD, 4x GTX1080
- 10 nodes with 2x “Broadwell” @2.2 GHz, 64 GB, 980 GB SSD, 4x GTX1080Ti
- 3 nodes with 2x “Skylake” @3.2 GHz, 96 GB, 2.9 TB SSD, 4x Tesla V100



What is each system good for?

Cluster	#nodes	Appl.	Parallel FS	Local HDD	Description
Meggie	728	massively parallel	Yes	No	Newest RRZE cluster, highly parallel workloads. Access restricted.
Emmy	560	massively parallel	Yes	No	Current main cluster for parallel jobs
Woody	176	serial, single-node, throughput	No	Yes	High clock speed single-socket nodes for serial throughput
TinyEth	20	serial, single-node, throughput	No	Yes	Throughput workloads
TinyGPU	27	GPGPU	No	Yes, some w/ SSD	Different types of Nvidia GPGPUs; Access restrictions and throttling policies may apply
TinyFat	25	Large memory	No	Yes, some w/ SSD	128-512 GB memory per node. Access restrictions may apply.



WORKING WITH DATA



<https://www.anleitungen.rrze.fau.de/hpc/hpc-storage/>

File systems

- **File system == directory structure** that can store files
- Several file systems can be “**mounted**” at a compute node
 - Similar to drive letters in Windows (C:, D:, ...)
 - Mount points can be anywhere in the root file system
- Examples
 - › `/home/hpc` home directories
 - › `/scratch` some local disk for temporary data
 - › `/mnt/cdrom` disk in the CD-ROM drive

File system quotas

- File system may impose quotas on
 - Stored data **volume**
 - Number of **files and directories** (inodes, actually)
- Quotas may be set **per user** or **per group** (or both)
- **Hard quota**
 - Absolute upper limit, cannot be exceeded
- **Soft quota**
 - May be exceeded temporarily (e.g., for 7 days – **grace period**)
 - Turns into hard quota at end of grace period

Displaying the quota limits

```
$ quota -v          # generic command
Disk quotas for user unrz55 (uid 12050):
    Filesystem blocks  quota  limit  grace  files  quota  limit  grace
10.188.4.72:/gpfs/home/shared
           5092912  10485760 20971520          199854  500000 1500000
wnfs1.rrze.uni-erlangen.de:/srv/home
           131485528      0      0          666135      0      0

$ shownicerquota.pl # only on RRZE systems
Path          Used      SoftQ    HardQ    Gracetime  Filec    FileQ    FiHaQ    FileGrace
/home/hpc      5.1G    10.5G    21.0G      N/A      199K     500K    1,500K      N/A
/home/vault    69.4G   104.9G   209.7G      N/A      63K     100K     200K      N/A
/elxfs         4.0K     0.0K     0.0K      N/A       1     500K     600K      N/A
```

RRZE file systems overview

Mount point	Access	Purpose	Technology	Backup	Snapshots	Data lifetime	Quota
/home/hpc	\$HOME	Source, input, important results	NFS on central servers, small	YES	YES	Account lifetime	YES (small)
/home/vault	\$HPCVAULT	Mid-/long-term storage	Central servers, HSM	YES	YES	Account lifetime	YES
<i>diverse</i>	\$WORK	General purpose work directory	Central NFS server	NO	NO	Account lifetime	YES
/*lxf	\$FASTTMP (only within cluster)	High performance parallel I/O	Lustre parallel FS via InfiniBand	NO	NO	High watermark	Only inodes
/???	\$TMPDIR	Node-local job-specific dir	HDD/SDD/ramdisk	NO	NO	Job runtime	NO

File systems at RRZE: \$HOME

- Most important file system: your **\$HOME**
 - When you log in on any HPC system, you end up here
 - At RRZE, this is at **/home/hpc/<your_group>/<your_account>**
 - **Unchangeable 10 GB soft / 20 GB hard quota**
 - High-quality storage with backup and snapshots
- Snapshots
 - In any directory:
\$ cd .snapshots

```
unrz55@sauron:~/programming/py/.snapshots $ ls -F
@GMT-2018.12.30-03.00.00/ @GMT-2019.01.23-11.00.00/ @GMT-2019.01.24-05.00.00/
@GMT-2019.01.06-03.00.00/ @GMT-2019.01.23-13.00.00/ @GMT-2019.01.24-07.00.00/
@GMT-2019.01.13-03.00.00/ @GMT-2019.01.23-15.00.00/ @GMT-2019.01.24-07.30.00/
@GMT-2019.01.18-03.00.00/ @GMT-2019.01.23-17.00.00/ @GMT-2019.01.24-08.00.00/
@GMT-2019.01.19-03.00.00/ @GMT-2019.01.23-19.00.00/ @GMT-2019.01.24-08.30.00/
@GMT-2019.01.20-03.00.00/ @GMT-2019.01.23-21.00.00/ @GMT-2019.01.24-09.00.00/
@GMT-2019.01.21-03.00.00/ @GMT-2019.01.23-23.00.00/ @GMT-2019.01.24-09.30.00/
@GMT-2019.01.22-03.00.00/ @GMT-2019.01.24-01.00.00/
@GMT-2019.01.23-03.00.00/ @GMT-2019.01.24-03.00.00/
```

File systems at RRZE: \$HPCVAULT

- Long-term safe hierarchical data storage
 - `/home/vault/<your_group>/<your_account>`
 - Disk cache + tape storage
 - Old data is migrated automatically (and transparently) to tape
 - High-quality storage with backups and snapshots
 - Use for important stuff that you do not need often
 - 100 GB soft / 200 GB hard quota
- Only very large files (>>10 GB) can be migrated to tape
 - Only migrated data does not count against your quota

File systems at RRZE: \$WORK

- General purpose work directory
- Mid-term data storage with relaxed quota
 - May point to `/home/woody/<your_group>/<your_account>` or `/home/saturn/<your_group>/<your_account>`
 - Standard 200 GB quota, can be increased upon request
 - Standard NFS volume, no backup, no snapshots
- For stuff that can be re-generated if necessary
 - › Output files, pre-/post-processing, storage for small files
 - › Do NOT store the only copy of your PhD thesis there!

File systems at RRZE: \$FASTTMP

- `/<whatever>/<your_group>/<your_account>`
- Only accessible within a specific cluster
- High-performance **short-term** parallel storage
 - MPI-I/O, checkpointing,...
 - Bandwidth of several Gbytes per second (with parallel reads or writes)
 - Quota on inodes (number of files and folders)
 - **High watermark deletion**
- This is only for large binary files (>>100 MB)
- **This is only for large binary files (>>100 MB)**
- **This is only for large binary files (>>100 MB)**

Data transfer

- Most RRZE file systems are mounted at all HPC systems
 - Exception: parallel FS
- No NFS mounting from or to systems outside of RRZE
 - Because... reasons.
- → **scp / rsync** is the preferred file transfer tool from and to the outside

Recurse into subdirectories

Preserve time stamps and access modes

```
$ scp -r -p code unrz55@emmy.rrze.fau.de:/home/woody/unrz/unrz55  
$ scp unrz55@emmy.rrze.fau.de:results/output.dat .
```

- Windows: <https://winscp.net/>



SOFTWARE



<https://www.anleitungen.rrze.fau.de/hpc/environment>

Pre-installed software packages

- **Linux** standard distro **packages**
 - “Full” installation available on cluster front-ends
 - › Easy to add additional packages
 - Node installation: usually stripped down
 - › Not easy to add new software

The modules system

- Software provided locally by RRZE
 - Compilers, libraries, commercial and open software
 - Installed on central server and available on all cluster nodes
- A package must be made available in the user's environment to become usable
 - Command: **module**
 - All module commands **affect the current shell only!**

The module command

Show available modules: **module avail**

```
$ module avail
----- /apps/modules/data/applications -----
amber-gpu/14p13-at15p06-gnu-intelmpi5.1-cuda7.5 gromacs/4.6.6-mkl-IVB
amber-gpu/16p04-at16p10-gnu-intelmpi5.1-cuda7.5 gromacs/5.0.4-mkl-IVB
amber/12p21-at12p38-intel16.0-intelmpi5.1      gromacs/5.1.1-mkl-IVB_d
----- /apps/modules/data/development -----
cuda/7.5                intel64/16.0up04                intelmpi/5.1up03-intel
cuda/8.0                intel64/17.0up05 (default)          llvm-clang/3.8.1
cuda/9.0                intel64/18.0up02                opencl/intel-cpuonly-5.2.0.10002
cuda/9.1                intel64/18.0up03                openmpi/1.08.8-gcc
$
```

The module command

Load a module: **module load <modulename>**

```
$ module load intel64
$ icc -V
Intel(R) C Intel(R) 64 Compiler for applications running on Intel(R) 64, Version 17.0.5.239 Build
20170817
Copyright (C) 1985-2017 Intel Corporation. All rights reserved.
```

Display loaded modules: **module list**

```
$ module list
Currently Loaded Modulefiles:
  1) python/3.5-anaconda      2) intelmpi/2017up04-intel  3) mkl/2017up05             4) intel64/17.0up05
```

Module command summary

Command	What it does
<code>module avail</code>	List available modules
<code>module whatis</code>	Shows over-verbose listing of all modules
<code>module list</code>	Shows which modules are currently loaded
<code>module load <pkg></code>	Loads module <i>pkg</i> , i.e., adjusts environment
<code>module load <pkg>/<version></code>	Loads specific version of <i>pkg</i> instead of default
<code>module unload <pkg></code>	Undoes what the load command did
<code>module help <pkg></code>	Shows a detailed description of <i>pkg</i>
<code>module show <pkg></code>	Shows what environment variables <i>pkg</i> modifies and how



RUNNING JOBS



<https://www.anleitungen.rrze.fau.de/hpc/batch-processing>

Interactive runs on the front-ends

- The cluster front-ends are for interactive work
 - Editing, compiling, preparing input,...
 - Amount of compute time per binary is limited by system limits
 - › E.g., after 1 hour of CPU time your run will be killed
 - **MPI jobs are not allowed on front ends at RRZE**
- Front-ends are shared among all users, so **be considerate!**

```
iww042@meggie1$ emacs Makefile
iww042@meggie1$ make all
iww042@meggie1$ ./scripts/preprocess.py < inputfile
iww042@meggie1$ ./bin/a.out arg1 arg2 arg3
```


Batch jobs

- All big clusters have **resource manager software**
 - “**Batch system**”
 - Users can request resources for their jobs
 - › Number of nodes (optionally: type of nodes, memory, ...)
 - › Job runtime
 - › What to run (normally a shell script)
 - Popular batch systems: PBS Pro, **Torque**, **SLURM**, LSF, GridEngine
 - Some setups (e.g., at RRZE) allow **interactive batch jobs**
- What you do with your node allocation is entirely up to you
- **Most queues** at RRZE have a **24 hour wall time limit**



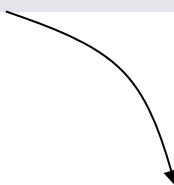
Example: Simple Torque batch script (Emmy)

- Most simple batch script (job1.sh):

```
#!/bin/bash -l  
~/bin/a.out arg1 arg2 arg3
```

- Submission:

```
iww042@emmy1$ qsub -l nodes=1:ppn=40,walltime=01:00:00 job1.sh  
1051341.eadm
```



Resource specification

Example: Complex Torque batch script (Emmy)

```
#!/bin/bash -l
```

```
#PBS -l nodes=4:ppn=40,walltime=06:00:00
```

```
#PBS -N Sparsejob_33
```

Job submission options:

Nodes, cores p/ node, time, name,...

Job option
sentinel

```
# jobs always start in $HOME: change to a temporary job dir on $WORK
```

```
mkdir ${WORK}/${PBS_JOBID}
```

```
cd ${WORK}/${PBS_JOBID}
```

`$PBS_*` variables contain
job-relevant data

```
# copy input file from location where job was submitted, and run
```

```
cp ${PBS_O_WORKDIR}/inputfile .
```

Actual run of
your binary

```
mpirun -npnode 20 ${HOME}/bin/a.out -i inputfile -o outputfile
```

Example: Managing a Torque job (Emmy)

```
iww042@emmy1$ qsub job2.sh  
1051342.eadm
```

- Job ID can be used to check and control the job later

```
iww042@emmy1$ qstat -a  
eadm:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	S	Elap Time
1051342.eadm	iww042	devel	test.sh	--	1	40	--	00:10:00	R	00:00:02

```
iww042@emmy1$ qalter -l walltime=02:00:00 1051342  
iww042@emmy1$ qdel 1051342
```

- stdout/stderr will be in <JobName>.[o|e]<JobID>

Torque user commands (non-exhaustive)

Command	Purpose	Options
<code>qsub [<options>] <job_script></code>	Submit batch job	<code>-l <resource_spec></code> <code>-N <JobName></code> <code>-q <queue_name></code> <code>-M your@email.de -m abe</code> <code>-X X11 forwarding</code> <code>-I interactive</code>
<code>qstat [<options>] [<JobID> <queue>]</code>	Check job status	<code>-a</code> friendly formatting <code>-f</code> verbose job info <code>-r</code> only running jobs <code>-n</code> show nodes of each job
<code>qdel <JobID></code>	Delete batch job	—

Interactive batch job with Torque (Emmy)

```
iww042@emmy1$ qsub -l nodes=2:ppn=40,walltime=01:00:00 -I
```

```
qsub: waiting for job 1051378.eadm to start
```

```
qsub: job 1051378.eadm ready
```

```
Starting prologue... Mon Jan 28 15:55:44 CET 2019
```

```
Master node: e0104
```

```
Kill all process from other users
```

```
Adjust oom killer config
```

```
Clearing buffers and caches on the nodes.
```

```
Power management available, enabling ondemand governor
```

```
End of prologue: Mon Jan 28 15:55:51 CET 2019
```

```
iww042@e0104$
```

Some resources reserved for small jobs during working hours

Mostly harmless ☺

Type stuff here

Example: SLURM batch script (Meggie)

```
#!/bin/bash -l
#SBATCH --nodes=4 --ntasks-per-node=20 --time=06:00:00
#SBATCH --job-name=Sparsejob_33
#SBATCH --export=NONE
unset SLURM_EXPORT_ENV                # avoid evil login shell settings

# jobs always start in $HOME: change to a temporary job dir on $WORK
mkdir ${WORK}/${SLURM_JOB_ID}
cd ${WORK}/${SLURM_JOB_ID}

# copy input file from location where job was submitted, and run
cp ${SLURM_SUBMIT_DIR}/inputfile .

srun --mpi=pmi2 --ntasks=80 --ntasks-per-node=20 ${HOME}/bin/a.out \
-i inputfile -o outputfile
```

SLURM batch job submission (Meggie)

```
iww042@meggie1$ sbatch job3.sh
```

```
Submitted batch job 357074
```

```
iww042@meggie1:~ $ squeue -l
```

```
Mon Jan 28 17:38:52 2019
```

JOBID	PARTITION	NAME	USER	STATE	TIME	TIME_LIMI	NODES	NODELIST (REASON)
357074	work	Sparsejo	iww042	RUNNING	0:35	1:00:00	4	m[0101-0104]

```
iww042@meggie1:~ $
```


Interactive batch job with SLURM (Meggie)

- `srun` command can queue/run a job directly

```
iww042@meggie1$ srun --nodes=2 --ntasks-per-node=20 \  
                  --time=01:00:00 --pty /bin/bash -l  
  
iww042@m0101$ echo $SLURM_SUBMIT_DIR  
/home/hpc/iww0/iww042  
iww042@m0101$
```

- X11 forwarding currently not supported
 - Workaround
 - › submit job with `sleep` command in batch script
 - › use `ssh -X` to log in to node

SLURM user commands (non-exhaustive)

Command	Purpose	Options
<code>sbatch [<options>] <job_script></code>	Submit batch job	<code>--time=HH:MM:SS</code> <code>--nodes=#</code> <code>--ntasks-per-node=#</code> <code>--mail-user=<address></code> <code>--mail-type=ALL BEGIN END ...</code> <code>--partition=work devel</code>
<code>squeue [<options>]</code>	Check job status	<code>-j <JobID></code> show job <code>-t RUNNING</code> show only running jobs <code>-u <USER></code> this user only
<code>scancel <JobID></code>	Delete batch job	—
<code>srun <options></code>	Run program	Many options; see man page



SOME DOS AND DON'TS



<https://www.anleitungen.rrze.fau.de/hpc/getting-started/>

Good practices

- **Be considerate.** Clusters are valuable shared resources that have been paid by the taxpayer.
- **Check your jobs** regularly
 - Are the results OK?
 - Does the job actually use the allocated nodes in the intended way? Does it run with the expected performance?
 - Memory consumption? Disk quota exceeded?

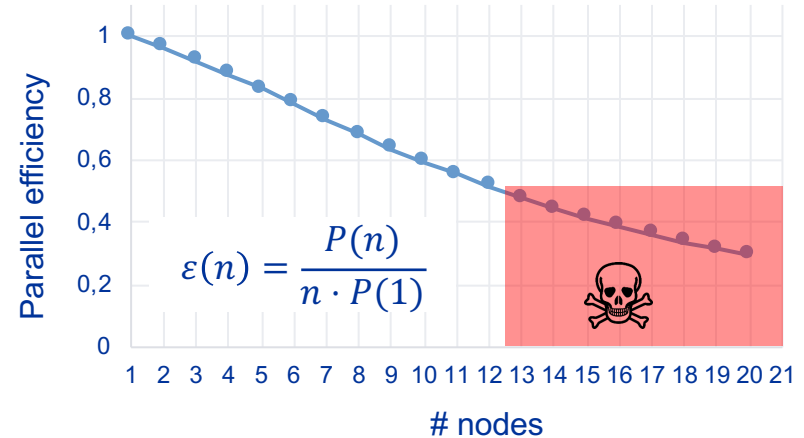
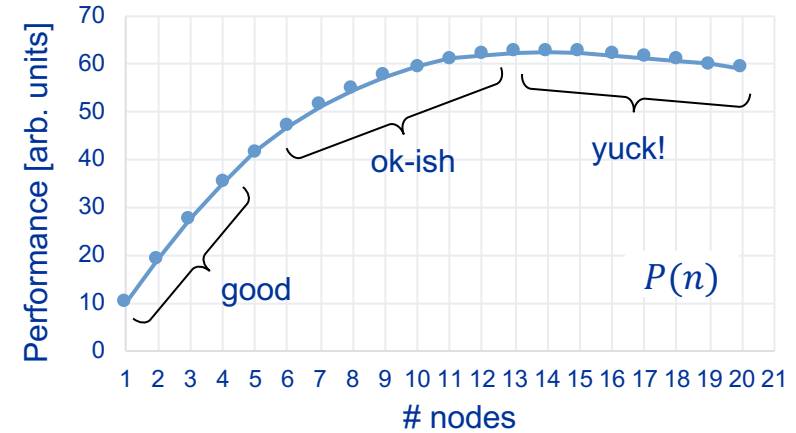
Why should I care?

- Minimum total cost (hardware, power, infrastructure, people) for one node-hour on modern HPC cluster: 0.5 €
- 1 node for 1 year: 4000 €
- 300 nodes for 1 day: 3600 €

This is money. Money that you burn cannot be used by others.

Good practices

- Use the **appropriate** amount of **parallelism**
 - Most workloads are not highly scalable
 - Best to run scaling experiments to figure out the “sweet spot”
 - Parallel efficiency < 50% is generally a waste of resources!



Good practices

- Use the **appropriate file system(s)**
 - #1 mistake: Overload metadata servers by doing tiny-size, high-frequency I/O to parallel FS
 - Delete obsolete data, move data to be archived to \$HPCVAULT
- **Do not re-use** other people's job scripts **if you don't understand** them completely
 - Things to look out for: file systems, number of nodes, cores per node, modules

Good practices

- Learn a **scripting language** to automate daunting, repetitive tasks
 - Bash, Python, Perl,...
- Talk to co-workers who are more experienced cluster users; let them educate you
- When **reporting a problem** to RRZE:
 - Use the official contact hpc-support@fau.de – this will immediately open a helpdesk ticket
 - Provide **as much detail as possible** so we know where to look
 - › “My job always crash please fix” will not do
 - › Username, Cluster, JobID, file system, time of event, ...





HOW TO LEARN MORE



Courses, tutorials, lectures, web sites

Upcoming courses

- **Node-Level Performance Engineering**
 - June 27–28, 2019, HLRS Stuttgart
 - Performance modeling and model-guided optimization
 - Not a beginner's programming course
- **Parallel Programming of High Performance Systems**
 - 5 days, February/March 2020, RRZE
 - Beginner's intro to cluster computing with OpenMP and MPI
 - Basic performance tools, basic code optimizations
 - Annual course; alternates between LRZ and RRZE

Upcoming courses

- **Advanced Topics in High Performance Computing**
 - March 2020, LRZ Garching
 - Code optimization, I/O, advanced OpenMP, PGAS languages
- **Programming Techniques for Supercomputers**
 - Summer term lecture (4V+2Ü) by Prof. Wellein
 - Computer architecture, performance modeling, parallel programming

Info on the web

- General info on HPC workflow and tools: HPC Wiki

[https://hpc-wiki.itc.rwth-aachen.de/hpc/HPC Wiki](https://hpc-wiki.itc.rwth-aachen.de/hpc/HPC_Wiki)

- RRZE docs:

<https://www.anleitungen.rrze.fau.de/hpc/>

REGIONALES RECHENZENTRUM ERLANGEN [RRZE]



THANK YOU.

HPC@RRZE

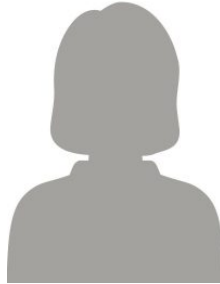
<https://hpc.fau.de>

Who we are



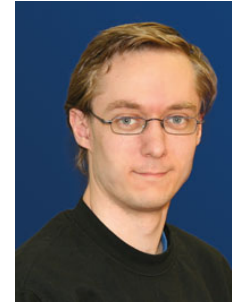
**Prof.
Gerhard Wellein**

Group lead



**Katrin Nusser,
M. Sc.**

System
administration,
user support



**Dipl.-Inf.
Michael Meier**

System
administration,
user support



**Dr.
Thomas Zeiser**

User support, system
administration, cross-
sectional activities



**Dr.
Georg Hager**

Cross-sectional
activities