

Advanced Topics: Usage of bwHPC and ForHLR clusters

Shamna Shamsudeen, SCC, KIT



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Hochschule
für Technik
Stuttgart



Hochschule Esslingen
University of Applied Sciences

Universität
Konstanz



UNIVERSITÄT
MANNHEIM



Universität Stuttgart

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



KIT
Karlsruher Institut für Technologie



ulm university universität
uulm



Outline

- Access and Data transfer topics
 - Access + rights, auto logout
 - Hardware accelerated visualisation @ bwUniCluster, ForHLR
 - Best practise: data sharing
- Architecture topics
 - Cluster topology, interconnect
 - Best practise: parallel file system
- Software topics
 - Best practise: installing own software
 - Best practise: compiling code
- Questions from participants

Reference: bwHPC-C5 Best Practices Repository

- Most information given by this talk can be found at <http://bwhpc.de/wiki>:

bwHPC Wiki

Search

page discussion view source

Main Page

Online
User and Best Practice Guides
of
Baden-Württemberg's HPC services

HPC Services

The federated HPC competence centers of tier 3 provide and maintain user guides and best practice guides for the compute clusters of **tier 3**:

- **bwUniCluster**
- **bwForCluster JUSTUS**
- **bwForCluster MLS&WISO**
- **bwForCluster NEMO**
- **bwForCluster BinAC**

User and best practice guides for compute cluster of higher HPC tiers in Baden-Württemberg can be found here:

- bwHPC tier 1: **Hazel Hen**
- bwHPC tier 2: **ForHLR**

HPC Data Storage Services

For user guides of the data storage services:

- **bwFileStorage**
- **SDS@hd**

bwHPC Wiki

- Home
- Best Practices Repository
- bwHPC News
- Wiki help

Best Practice Guides

- Overview
- Batch Jobs
- Software Modules
- Compiler
- Numerical Libraries
- Parallel Programming

bwHPC tier 3

- bwUniCluster
- bwForCluster JUSTUS
- bwForCluster MLS&WISO
- bwForCluster NEMO
- bwForCluster BinAC

bwHPC tier 1+2

- Hazel Hen
- ForHLR

bwHPC Support Services

- bwHPC courses
- Support/Ticket System
- Cluster Information System

bwHPC Data Storage

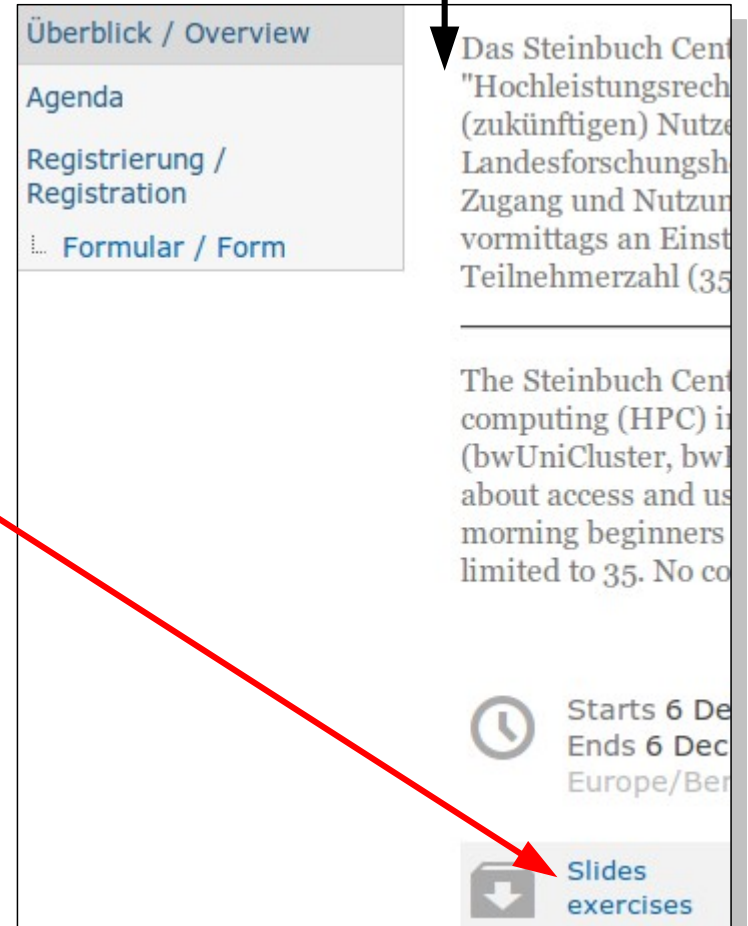
- bwFileStorage

Where to get the slides?

■ https://indico.scc.kit.edu/e/bwhpc_course_2018-10-10

uc1:/pfs/data1/software_uc1/bwhpc/kit/workshop/2018-10-10

■ Slides



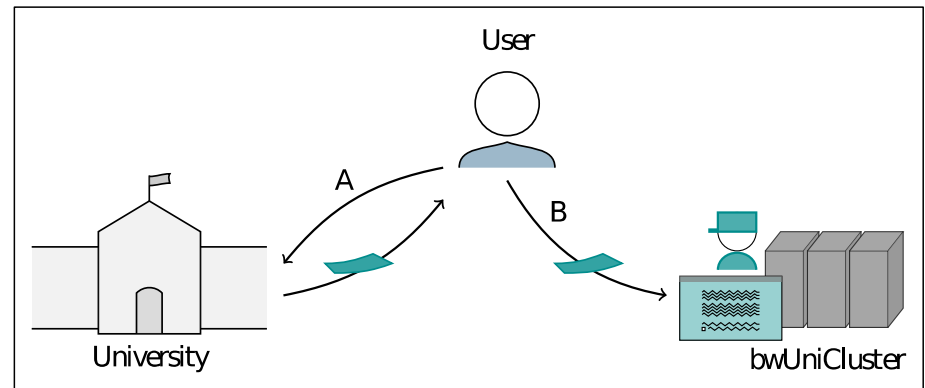
The screenshot shows a web interface for an event. On the left, a vertical menu contains the following items: 'Überblick / Overview' (highlighted), 'Agenda', 'Registrierung / Registration', and 'Formular / Form'. On the right, there is a main content area with text in German and English. A black arrow points from the top of the page down to the 'Überblick / Overview' menu item. A red arrow points from the 'Slides' text in the main text area down to a button at the bottom right of the page. The button is labeled 'Slides exercises' and features a downward-pointing arrow icon.

Access and Data: Adv. topics

Rev: Access – bwUniCluster & extension

Registration:

- 1. bwUniCluster entitlement
- 2. <https://bwidm.scc.kit.edu/>
- 3. questionnaire



Login:

@ „old partition“: `$ ssh <UserID>@uc1.scc.kit.edu`

@ „extension“: `$ ssh <UserID>@uc1e.scc.kit.edu`

- Any difference?
 - concerning compile code
 - but not concerning job submission

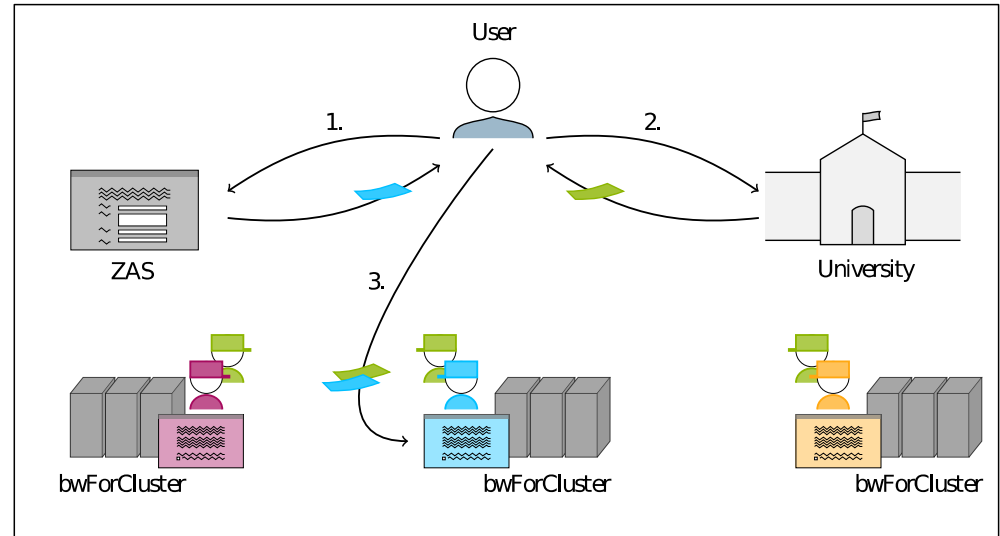
Auto logout

- Variable “TMOU” is set for 10 hours.
→ If the user is continuously 10 hours inactive then he/she will be automatically logged out

Rev: Access – bwForClusters

Registration:

1. Central Application Site (ZAS)
2. bwForCluster entitlement
3. bwForCluster webreg.



Login:

@ JUSTUS

```
$ ssh <UserID>@justus.uni-ulm.de
```

@ MLS&WISO

```
$ ssh <UserID>@bwforcluster.bwservices.uni-heidelberg.de
```

@ NEMO

```
$ ssh <UserID>@login.nemo.uni-freiburg.de
```

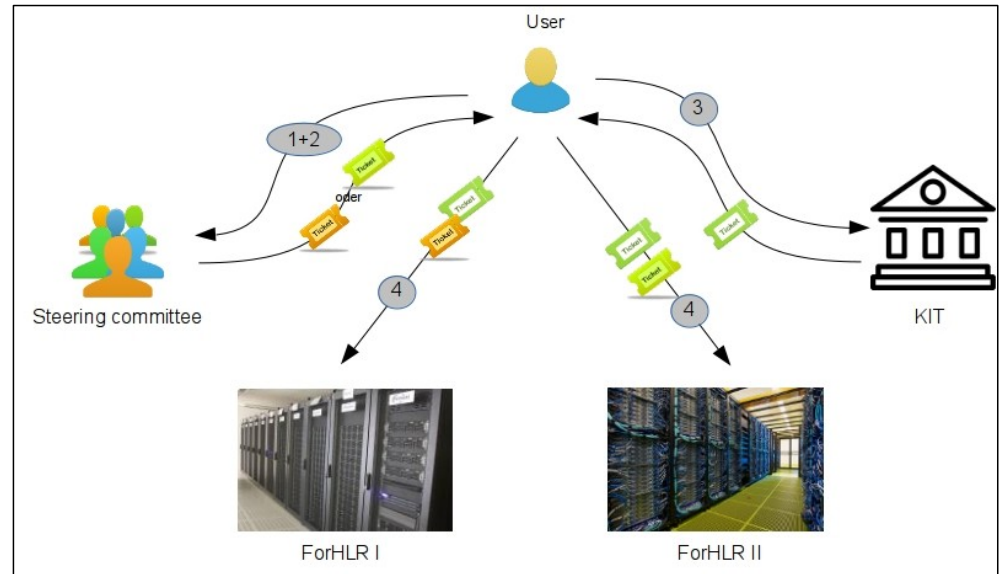
@ BinAC

```
$ ssh <UserID>@login0{1,2,3}.binac.uni-tuebingen.de
```

Rev: Access – ForHLR I and II

Registration:

- 1. Online Proposal Form
- 2. Peer reviewed proposal
- 3. ForHLR access form
- 4. <https://bwidm.scc.kit.edu/>



Login:

@ ForHLR I : `$ ssh <UserID>@fh1.scc.kit.edu`

@ ForHLR II: `$ ssh <UserID>@fh2.scc.kit.edu`

Auto logout

- Variable "TMOUT" is set for 10 hours.

Passwordless Login (linux + macOS)

- SSH private + public key pair

```
$ ssh-keygen -t rsa
```

→ Important: by all means **secure** private key **with passphrase**

- Transfer, e.g. bwUniCluster

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub <UserID>@uc1.scc.kit.edu
```

- Store passphrase in *keystore* (Ubuntu: keyring, Mac: keychain)

```
Ubuntu: $ ssh-add
```

- Login, e.g. bwUniCluster

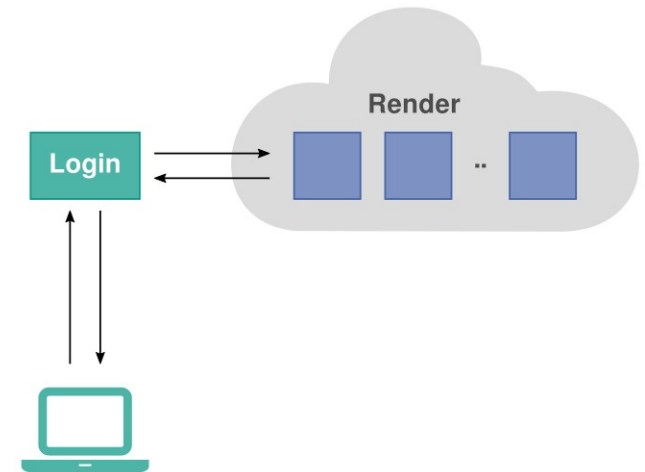
```
$ ssh <UserID>@uc1.scc.kit.edu
```

→ this requires now each time at actual login to ask for passphrase

→ for active sessions if passphrase in keystore, no login password anymore required

Remote Visualization (1)

- The Linux 3D graphics stack is based on X11 and OpenGL. This has some drawbacks in conjunction with remote visualization.
 - Rendering takes place on the client, not the cluster
 - Whole 3D model must be transferred via network to the client
 - Many round trips in the X11 protocol negatively influence interactivity
 - X11 is not available on non-Linux platforms
 - Compatibility problems between client and cluster can occur
- To avoid all those problems the module „**start_vnc_desktop**“ is provided on bwUniclusterc and ForHLR II for remote visualization. More details at [bwHPC wiki](#)



Remote Visualization (2)

```
uc1:~$ start_vnc_desktop --hw-rendering
```

Hint for TurboVNC Viewer users (command line):

```
vncviewer ExtSSH=1 Via=yc8563@uc1.scc.kit.edu Server=vc1n02:1 Password=AgGQmo8z
```

Hint for TurboVNC Viewer users (GUI)

Fill in the following entry field:
VNC server: **vc1n02:5901**

Click "Options" and choose tab "Security".
Fill in the following entry fields:

Gateway (SSH server or UltraVNC repeater)
SSH user: **yc8563**
Host: **uc1.scc.kit.edu**
Click "OK"

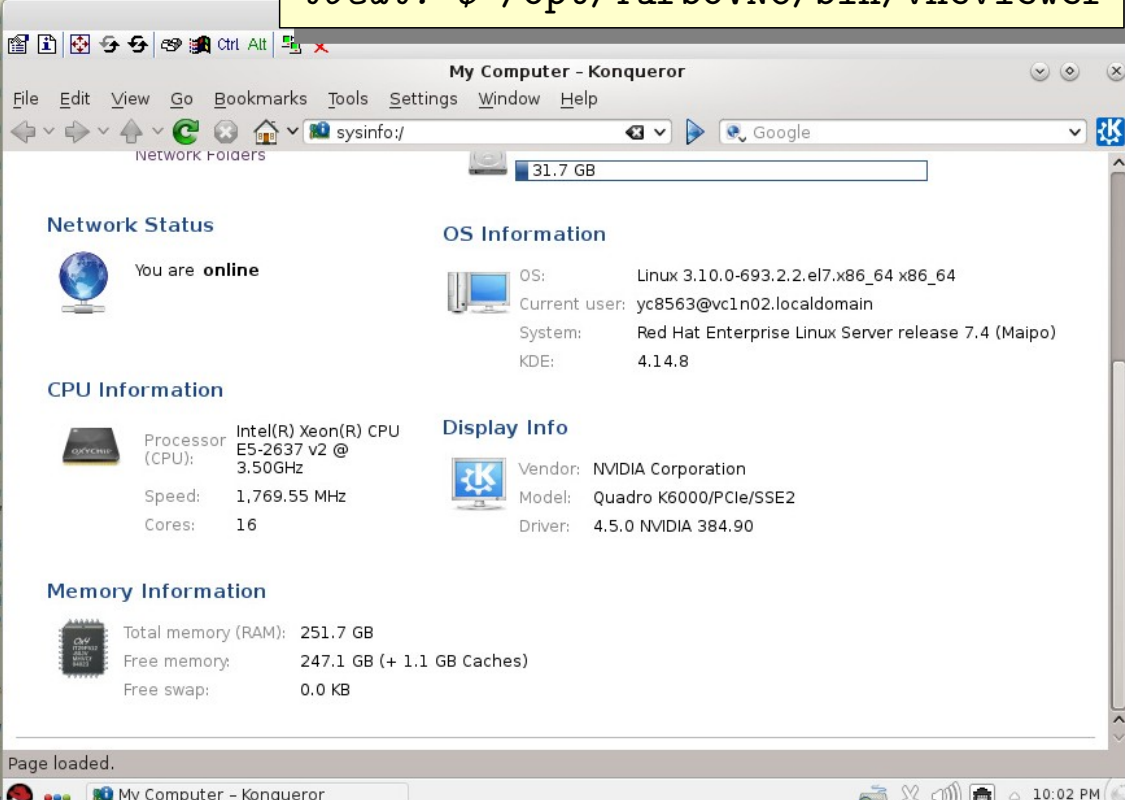
Click "Connect"

VNC Authentication:
Password: **AgGQmo8z**

Hint for installing VNC viewer:

```
/usr/bin/start_vnc_desktop --help-client
```

```
local:~$ /opt/TurboVNC/bin/vncviewer
```



The screenshot shows a Konqueror web browser window titled "My Computer - Konqueror". The address bar shows "sysinfo:/". The main content area displays system information in a grid layout:

- Network Status:** You are **online**.
- OS Information:** OS: Linux 3.10.0-693.2.2.el7.x86_64 x86_64; Current user: yc8563@vc1n02.localdomain; System: Red Hat Enterprise Linux Server release 7.4 (Maipo); KDE: 4.14.8.
- CPU Information:** Processor (CPU): Intel(R) Xeon(R) CPU E5-2637 v2 @ 3.50GHz; Speed: 1,769.55 MHz; Cores: 16.
- Display Info:** Vendor: NMDIA Corporation; Model: Quadro K6000/PCIe/SSE2; Driver: 4.5.0 NVIDIA 384.90.
- Memory Information:** Total memory (RAM): 251.7 GB; Free memory: 247.1 GB (+ 1.1 GB Caches); Free swap: 0.0 KB.

The status bar at the bottom shows "Page loaded." and the system tray includes icons for network, volume, and the time 10:02 PM.

Best practise: Data Sharing (1)

- How to share data with another person on the same cluster?
 1. Do not share folders in your \$HOME, use workspaces!

```
$ ws_allocate sharing 30
Workspace created. Duration is 720 hours.
Further extensions available: 3
/pfs/work2/workspace/scratch/ab1234-sharing-0
$ ls -ld $(ws_find sharing)
drwx----- 2 ab1234 xyz 4096 Oct  9 00:42 /pfs/work2/workspace/scratch/ab1234-sharing-0
```

→ workspace is private!

2. Adjust permissions to your needs:

- a.) Allow all users of your group to have access

```
$ chmod g+x $(ws_find sharing)
$ ls -ld $(ws_find sharing)
drwx--x--- 2 ab1234 xyz 4096 Oct  9 00:42 /pfs/.../ab1234-sharing-0
```

Best practise: Data Sharing (2)

- How to share data with another person on the same cluster?

2. Adjust permissions to your needs:

b.) Allow **another user** to have **full access** but **force group inheritance**

```
$ chmod g+s $(ws_find sharing)
$ ls -ld $(ws_find sharing)
drwx--S--- 2 ab1234 xyz 4096 Oct  9 00:42 /pfs/.../ab1234-sharing-0
```

→ use ACL (access control lists)

To add group uvw:
\$ setfacl -m g:uvw:rwx ...

```
$ setfacl -m u:cd5678:rwx $(ws_find sharing)
$ ls -ld $(ws_find sharing)
drwxrws---+ 2 ab1234 xyz 4096 Oct  9 00:42 /pfs/.../ab1234-sharing-0

$ getfacl $(ws_find sharing)
...
# owner: ab1234
# group: xyz
# flags: -s-
user::rwx
user:cd5678:rwx
group:---
...
```

Best practise: Data Sharing (3)

- How to share data with another person on the same cluster?

2. Adjust permissions to your needs:

c.) Revoke other user's access to workspace „sharing“

```
$ setfacl -x u:cd5678 $(ws_find sharing)
$ ls -ld $(ws_find sharing)
drwx--S---+ 2 ab1234 xyz 4096 Oct  9 00:42 /pfs/.../ab1234-sharing-0
$ getfacl $(ws_find sharing)
...
# owner: ab1234
# group: xyz
# flags: -s-
user::rwx
group:---
...
```

Loading custom environment

- Automatically customize your CLI session
 - If you always need particular global variables, aliases, bash functions
 - Add to `$HOME/.bashrc`
 - e.g., own variable to point your workspace „sharing“

```
export WS=$(ws_find sharing)
```

```
$ ssh ab1234@uc1
$ echo $WS
/pfs/work2/workspace/scratch/ab1234-sharing-0
```

- e.g. own function to jump to your workspace „sharing“

```
goto_ws(){ cd $(ws_find sharing) ; }
```

```
$ ssh ab1234@uc1
$ type goto_ws
goto_ws is a function
goto_ws ()
{
    cd $(ws_find sharing)
}
$ goto_ws
```

Do not omit the spaces

Architecture: Adv. topics

HPC Cluster Architecture: bwUniCluster

■ Node types:

■ Login nodes

- Thin and Broadwell

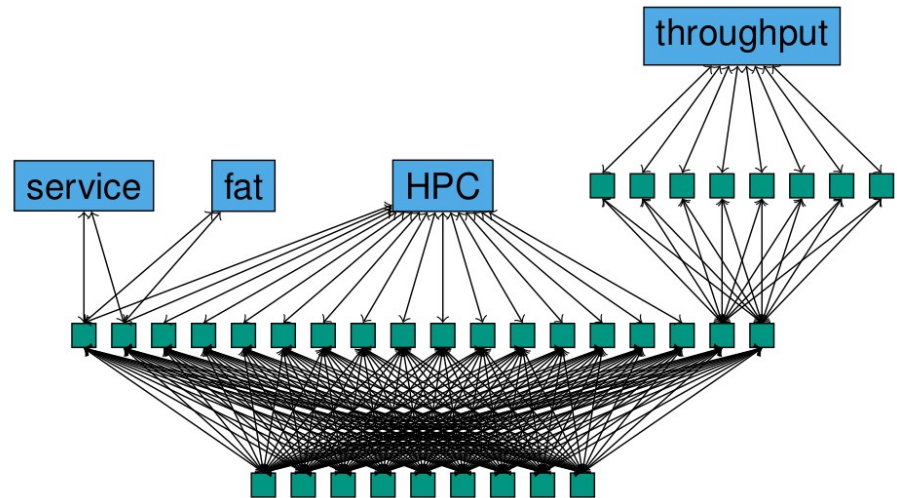
■ Compute Nodes

- Thin, Fat and Broadwell

■ File Server Nodes

■ Administrative Server Nodes

→ all connected by interconnect



■ More details at:

http://www.bwhpc-c5.de/wiki/index.php/BwUniCluster_Hardware_and_Architecture

Interconnect

- Channel-based, switched fabric architecture connecting nodes
 - Allows different network topologies with various degrees of channel sharing
 - Sharing types: non-blocking, 1:X-blocking
- Interconnects do have superior throughput and latency performance
 - high bandwidth (up to 100 GBit), low latency (600 ns Ping) network
 - full offloading until Layer 4, remote Direct Memory Access
- Implications
 - Makes MPI and global I/O (\$HOME, and workspaces) applicable
 - Setup of non-blocking or next-to-non-blocking compute islands
 - job placement aligned to compute islands
- bwUniCluster, ForHLR Phase 1
 - To get linkage info for each node, use:

```
#!/bin/bash
envar="$HOSTNAME:Addr=${SLURM_TOPOLOGY_ADDR}\n$HOSTNAME:Pat=${SLURM_TOPOLOGY_ADDR_PATTERN}"
module load mpi/openmpi
mpirun echo ${envar}
```

Hyper-threading

- Simultaneous multithreading (SMT)
- Physical cores -> doubling into logical cores
 - 2 logical cores share execution part (engine, cache, bus) of 1 physical core
- But: only max number of physical cores requestable by queueing system
 - e.g. bwUniCluster thin nodes:

```
msub -1 nodes=1:ppn=32  
msub -1 nodes=1:ppn=16
```
- Other half of logical cores reserved to OS background task
 - increases overall performance, OS jitter are better distributed

Job I/O statistics

- PFS I/O statistics of a batch job can be collected

- 1. Determine the PFS name of your job directory, e.g. your workspace

```
$ df --output=source <job_directory> | sed 1d | cut -d: -f3  
/pfs2wor2
```

- 2. Add PFS name during job submission:

```
$ msub -W lustrestats:<PFS_name> ...
```

- Optional: Get results by e-mail

```
$ msub -W lustrestats:<PFS_name> -M name@domain ...
```

Best Practise: Parallel file systems (1)

■ What not to do:

- Do not run jobs in \$HOME → use workspaces
- Do not generate +10000 files on workspaces → change application code
- Do not run any kind of data base on PFS
- Do not use PFS for your entire research data storage → clear out periodically
- → use bwFileStorage/LSDf-DIS or SDS@hd

■ Rev: How to check your quota:

```
$ lfs quota -u $(whoami) $HOME
```

- \$HOME @ bwUniC., ForHLR:

■ **New:** Send reminder for workspace renewal

```
$ ws_send_ical.sh <workspace> <email>
```

Best Practise: Parallel file systems (2)

■ Improving Performance of PFS:

- On PFS files striped over storage subsystems, i.e. large files are separated into stripes and distributed to different storage subsystems

- Stripe size = size of chunks (default = 1 MB)
- Stripe count = number of used storage subsystems per file/directory
 - New files and subdirs inherit stripe count from parent dir
 - default: stripe_count_my_\$HOME = 1; stripe_count_my_\$WORK = 2

- Get stripe count:

```
$ lfs getstripe <my_file>  
$ lfs getstripe -d <my_dir>
```

- Set stripe count:

```
$ lfs setstripe -c<num> <my_file/my_dir>
```

- num = -1 → use all available storage subsystems (\$HOME=20, \$WORK=40)
- New stripe count of parent dir → stripe count of existing files inside unchanged
 - To recursively change → copy all content to new directory
- New stripe count is not saved in PFS backup

Best Practise: Parallel file systems (3)

- Improving Performance of PFS:
 - When to change stripe count?
 - Many tasks use few huge files
→ stripe count = -1
 - To avoid overlapping issues:
→ setting tasks to use $N*1\text{MB}$ blocks
 - Enhancing throughput of a single file by ONE task
→ stripe count between 2 and 8
 - General rules:
 - Transfer data in large blocks and store in few large files
 - Make use of large caches, i.e., collect large blocks and write them sequentially at once
 - Avoid competitive file access

Software

Best Practise: Installing Own Software

■ Check list:

- Legal issues: do you have licence for your software?
- Disk space?
 - Check if software would exceed quota
 - **On ForHLR** never **use** \$HOME but **always \$PROJECT**
- Installation procedure?
 - If compilation exceeds 10 min
 - Install via interactive batch job on a compute node
 - Never use: *make -j* → but: *make -j <number>*
 - Never simply use binaries on different architecture
 - But: recompile or compile supporting multiple architecture
 - Use guides stored in software modulefiles

■ Help:

- Contact support ([bwTicketPortal](#)),
- apply for [Tiger Team Support](#)

Best Practise: Compiling code

- Details to compilation and makefile tutorial:

- See today's talk „Compile, Makefiles“

- Clusters with different architecture generations

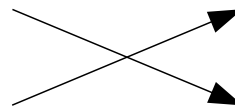
e.g. bwUniCluster

- thin+fat nodes (sandy bridge) vs. broadwell nodes

a) compile code on corresponding login node

```
uc1:~$ icc/ifort -xHost ...
```

```
uc1e:~$ icc/ifort -xHost ...
```



Run @uc1 → crash

Run @uc1e → slower

b) compile code including multiple, feature-specific code paths

```
uc1e:~$ icc/ifort -xCORE-AVX2 -axAVX...
```

Thank you for your attention!

Questions?