bw|HPC

KIT | NHR
National High-Performance Computing Center

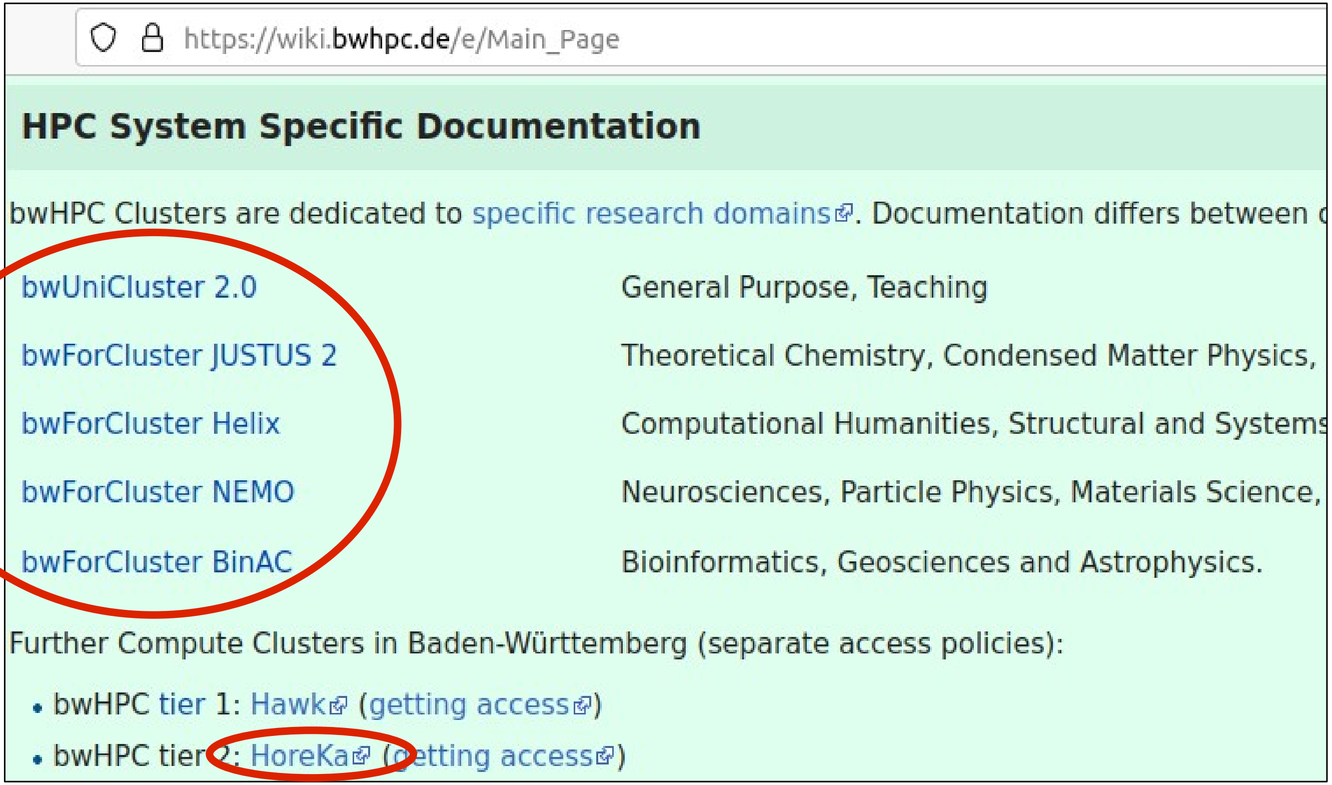
File Systems

Roland Laifer
KIT, SCC

Funding: Baden-Württemberg MINISTERIUM FÜR WISSENSCHAFT, FORSCHUNG UND KUNST
Bundesministerium für Bildung und Forschung

Reference: bwHPC Wiki

- Most information given by this talk can be found at https://wiki.bwhpc.de
  - select cluster
  - then select File Systems



https://wiki.bwhpc.de/e/Main_Page

HPC System Specific Documentation

bwHPC Clusters are dedicated to specific research domains. Documentation differs between c

bwUniCluster 2.0	General Purpose, Teaching
bwForCluster JUSTUS 2	Theoretical Chemistry, Condensed Matter Physics,
bwForCluster Helix	Computational Humanities, Structural and Systems
bwForCluster NEMO	Neurosciences, Particle Physics, Materials Science,
bwForCluster BinAC	Bioinformatics, Geosciences and Astrophysics.

Further Compute Clusters in Baden-Württemberg (separate access policies):

- bwHPC tier 1: Hawk (getting access)
- bwHPC tier 2: HoreKa (getting access)

Material: Slides & Scripts

- https://indico.scc.kit.edu/e/hpc_course_2024-04-09
- BwUniCluster 2.0: /opt/bwhpc/common/workshops/2024-04-09/
- HoreKa: /software/all/workshop/2024-04-09/

How to read the following slides

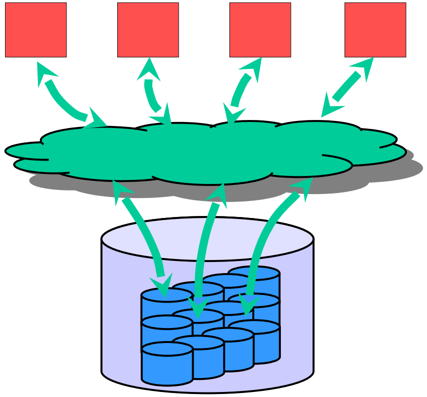
Abbreviation/Colour code	Full meaning
`$ command -option value`	$ = prompt of the interactive shell The full prompt may look like: `user@machine:path$` The command has been entered in the interactive shell session

`<integer>`
`<string>`
	<> = Placeholder for integer, string etc

`foo, bar`	Metasyntactic variables

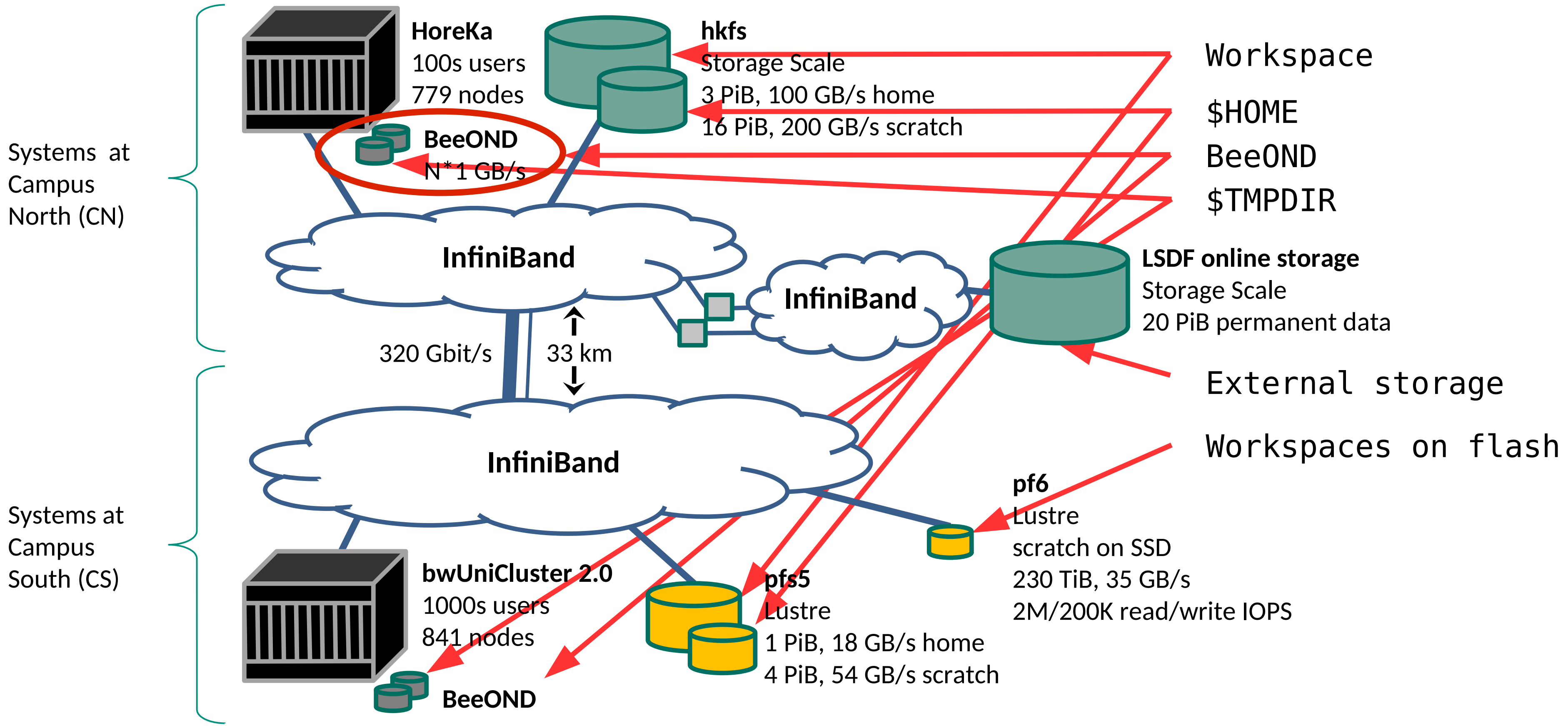
Parallel file systems

- **Most important parallel file systems**
  - Lustre
    - Used on most of the largest HPC systems
  - IBM Storage Scale (aka GPFS)
    - Used in industry and on many HPC systems
  - BeeGFS
    - Underlying file system for BeeGFS On Demand (BeeOND)



- **Lustre, GPFS, BeeGFS**
  - Follow POSIX standard, i.e. applications just work, and provide same view from all nodes
  - Offer large capacity and parallel access from many nodes
  - Good performance for huge files and access with large chunks
  - Dislike small files, random I/O, or many metadata (open, close, stat, create, remove) operations
    - Hence for some applications I/O on laptop with SSD might be faster
    - Reasons: communication over network, locking to guarantee consistency

HPC clusters and file systems @ KIT



How to use each File System (1)

- $HOME = Home directory

  → Software, configuration files, final results

  → Omit heavy I/O

- Workspaces = Working directories with lifetime

  → Intermediate results, huge input/output data sets

  → Scratch data which needs to be shared between nodes

  → Omit small files, tiny block sizes, lots of metadata operations

  If not possible to omit, KIT and HoreKa users can use Workspaces on flash storage

- $TMPDIR = Separate file system on each node using local disks

  → Data is only available during job runtime on the local node

  → Possibly transfer data here within a batch job

  → All sorts of I/O allowed

How to use each File System (2)

- BeeOND = Private file system for batch job
  - → Data is only available during job runtime on the batch job nodes
  - → Possibly transfer data here within a batch job
  - → All sorts of I/O allowed, only available on bwUniCluster 2.0 and HoreKa
- External storage
  - → Archive scientific data, move data here when data sets become too large
  - → Each organization has different solutions, examples are RDA or LSDF at KIT
  - → Use huge files or compressed archives
- Summary
  - → Use $HOME for permanent data
  - → Use workspaces for huge files and sequential I/O
  - → Use $TMPDIR or BeeOND with many (> 10000) small files or random I/O

File System properties overview

Property	$HOME	Workspace	$TMPDIR	BeeOND[1]	LSDF[1]	WS on flash[1]
Visibility	+++	+++	+	++	+++	+++
Lifetime	+++	++	+	+	+++	++
Capacity	+	+++	+	++	++	+
Seq. perf.	+	+++	+	+++	++	+++
Random perf.	+	+	+++	++	+	++
No impact on other users	+	++	+++	+++	+	++
Backup	+	–	–	–	+	–

[1] Only available on bwUniCluster 2.0 and HoreKa

File System properties of bw clusters and of HoreKa

Property	$HOME	Workspace	$TMPDIR	BeeOND[1]	LSDF[1]	WS on flash[1]
Visibility	all nodes	all nodes	local node	job nodes	login + job	all nodes
Lifetime	permanent	few weeks	job runtime	job runtime	permanent	few weeks
Usable capacity	40 GB - 10 TB	10 TB - 250 TB	128 GB - 7 TB	N * 750 GB	per project	1 TB
Usable inodes	2 mil - unlinited	1 mil - unlimited	unlimited	unlimited	per project	5 mil
Backup	yes, except Helix	no	no	no	yes	no
Total perf.	medium, 100s - 1000s MB/s	huge, 10s GB/s	100s MB/s per node	N * 100s MB/s	10s GB/s	huge, 10s GB/s

[1] Only available on bwUniCluster 2.0 and HoreKa

$HOME = Home directory

- $HOME is visible on all nodes of a cluster
- Properties of $HOME on different clusters

Cluster	Quota capacity limit	Quota file limit	Backup
JUSTUS 2	400 GB per user	2 mill. per user	Yes
Helix	200 GB per user	unlimited	No
NEMO	100 GB per user	unlimited	Yes
BinAC	40 GB per user	unlimited	Yes
BwUniCluster 2.0	1000 GB per user also limit per organization	10 mill. per user	Yes
HoreKa	10 TB per project	10 mill. per project	Yes

$HOME on bwUniCluster 2.0

- HowTo goto:

```
$ cd $HOME
```

```
$ cd
```

- User's quota usage and limits:

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

```
Disk quotas for usr ab1234 (uid 9999):
          Filesystem    used   quota   limit   grace   files   quota    limit   grace
/home/kit/scc/ab1234  9.904G      0k      1T       -  176849       0 10000000       -
```

- Organization quota usage and limits:

```
$ lfs quota -ph $(grep $(echo $HOME | sed -e "s|/[^/]*/[^/]*$||") \
    /pfs/data5/project_ids.txt | cut -f 1 -d\ ) $HOME
```

$HOME / $PROJECT on HoreKa

- $HOME and $PROJECT are identical if your account is member of one project
  - Otherwise to change to another project which will also modify $PROJECT:

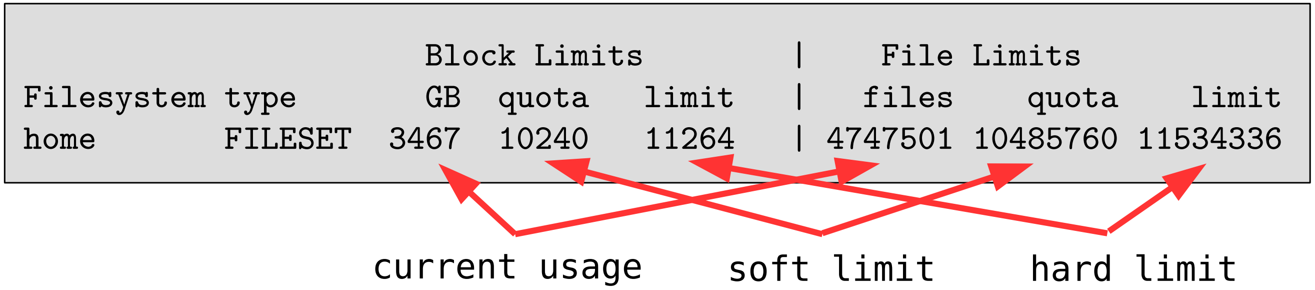
```
$ newgrp <another_project_group>
```

- Project group quota usage and limits:
  - First start an interactive job (sometimes this step is not needed):

```
$ salloc -p dev_cpuonly -n 1 -t 20 --mem=500
```

  - Show usage and limits of your project group on the $HOME file system:

```
$ /usr/lpp/mmfs/bin/mmlsquota -j $PROJECT_GROUP --block-size G -C hkn.scc.kit.edu hkfs-home
```



```
                          Block Limits          |    File Limits
Filesystem type          GB   quota    limit    |   files     quota     limit
home         FILESET   3467   10240    11264    | 4747501 10485760 11534336
```

Exercise 1: Show quotas

- Login to bwUnicluster 2.0 or HoreKa and show list of commands for exercises:
  - BwUniCluster:

    ```
    $ cat /opt/bwhpc/common/workshops/2024-04-09/pfs_commands.txt
    ```

  - HoreKa:

    ```
    $ cat /software/all/workshop/2024-04-09/pfs_commands.txt
    ```

- Use Cut & Paste to execute the first commands which show your quotas

Workspaces = Working directories with lifetime

- **Workspace**: lifetime on allocated folder
  - Available on all clusters, visible on all nodes of a cluster
  - HowTo:

    → https://wiki.bwhpc.de/e/Workspace

`$ ws_allocate` *foo* `10`	Allocate workspace *foo* for 10 days
`$ ws_list`	List your workspaces
`$ ws_find` *foo*	Get absolute path of workspace *foo*
`$ ws_extend` *foo* `5`	Extend lifetime of your workspace *foo* by 5 days from now. Number of extensions depends on cluster.
`$ ws_release` *foo*	Manually erase your workspace *foo*
`$ ws_… -F` *ffuc* `…`	Select non default workspace file system with -F (works for any command)

Properties of Workspaces on different clusters

Cluster	Capacity limit	File limit	Max lifetime	Max extensions
JUSTUS 2	20 TB per user	5 mill. per user	90 days	unlimited
Helix	10 TB per user	unlimited	30 days	10 times
NEMO	10 TB per user	1 mill. per user	100 days	99 times
BinAC	unlimited	unlimited	30 days	3 times
BwUniCluster 2.0	40 TB per user	30 mill. per user	60 days	3 times
HoreKa	250 TB per user	50 mill. per user	60 days	3 times

Exercise 2: Create workspace

- Allocate two workspaces

```
$ ws_allocate ws01 30
Info: could not read email from users config ~/.ws_user.conf.
Info: reminder email will be sent to local user account
Info: creating workspace.
/pfs/work7/workspace/scratch/myuser-ws01
remaining extensions  : 3
remaining time in days: 30

$ ws_allocate -F ffuc ws_ssd 50
Info: could not read email from users config ~/.ws_user.conf.
Info: reminder email will be sent to local user account
Info: creating workspace.
/pfs/work8/workspace/ffuc/scratch/myuser-ws_ssd
remaining extensions  : 3
remaining time in days: 50
```

Exercise 3: List workspace

- List workspaces

```
$ ws_list
id: ws_ssd
     workspace directory  : /pfs/work8/workspace/ffuc/scratch/myuser-ws_ssd
     remaining time       : 49 days 23 hours
     creation time        : Wed Oct  6 18:59:11 2021
     expiration date      : Thu Nov 25 17:59:11 2021
     filesystem name      : pfs6wor8
     available extensions : 3
id: ws01
     workspace directory  : /pfs/work7/workspace/scratch/myuser-ws01
     remaining time       : 29 days 23 hours
     creation time        : Wed Oct  6 18:55:17 2021
     expiration date      : Fri Nov  5 17:55:17 2021
     filesystem name      : pfs5wor7
     available extensions : 3
```

Exercise 4: Find workspace path

- Find workspace path and switch to it

```
$ ws_find ws01
/pfs/work7/workspace/scratch/myuser-ws01

$ ws_find -F ffuc ws_ssd
/pfs/work8/workspace/ffuc/scratch/myuser-ws_ssd

$ cd $(ws_find ws01)
$ pwd
/pfs/work7/workspace/scratch/myuser-ws01
```

Exercise 5: Extend workspace lifetime

- Extend the lifetime of a workspace

```
$ ws_extend ws01 60
Info: could not read email from users config ~/.ws_user.conf.
Info: reminder email will be sent to local user account
Info: extending workspace.
Info: changed mail address to myuser
Info: changed reminder setting.
/pfs/work7/workspace/scratch/myuser-ws01
remaining extensions  : 2
remaining time in days: 60
$ ws_list ws01
id: ws01
     workspace directory  : /pfs/work7/workspace/scratch/myuser-ws01
     remaining time       : 59 days 23 hours
     creation time        : Wed Oct  6 19:01:02 2021
     expiration date      : Sun Dec  5 18:01:02 2021
     filesystem name      : pfs5wor7
     available extensions : 2
```

File system on each node using local disks ($TMPDIR)

- **Node local storage on SSDs**
  - Usage with environment variable $TMPDIR
    - On JUSTUS 2 $TMPDIR is file system in main memory and $SCRATCH is on local SSD
  - Separate private directory on each node of a batch job, created at job start and destroyed at job end
    - Make sure you have copied your data back to a workspace or $HOME within your job
  - HowTo:

    → https://wiki.bwhpc.de/e/BwUniCluster2.0/Hardware_and_Architecture#.24TMPDIR
- **Usage example**
  - Outside batch job create archive with compressed input dataset on a workspace:

    ```
    $ tar -cvzf $(ws_find data-ssd)/dataset.tgz dataset/
    ```
  - In batch script extract compressed input dataset to local SSD:

    ```
    tar -C $TMPDIR/ -xvzf $(ws_find data-ssd)/dataset.tgz
    ```
  - In batch script application reads data from dataset on SSD and writes results to SSD:

    ```
    myapp -input $TMPDIR/dataset/myinput.csv -outputdir $TMPDIR/results
    ```
  - In batch script save results to a workspace:

    ```
    rsync -av $TMPDIR/results $(ws_find data-ssd)/results-${SLURM_JOB_ID}/
    ```

BeeOND = Private file system for batch job

- **BeeOND (BeeGFS On-Demand)**
  - Available only on bwUniCluster 2.0 and on HoreKa
  - Private file system for batch job, created at job start and destroyed at job end
    - Make sure you have copied your data back to a workspace or $HOME within your job
  - Parallel file system, visible on nodes allocated to a batch job
  - Uses local disks (SSDs) of each node to store the data
    - Capacity is limited: 750 GB * *number of nodes used in batch job*
  - Request creation in job script or on command line:

```
#SBATCH --constraint=BEEOND
```

```
$ sbatch -C BEEOND ...
```

  - Use path below */mnt/odfs/${SLURM_JOB_ID}* to access BeeOND, e.g.

```
$ cd /mnt/odfs/${SLURM_JOB_ID}/stripe_default
```

  - HowTo:
    → https://wiki.bwhpc.de/e/BwUniCluster_2.0_Hardware_and_Architecture#BeeOND_.28BeeGFS_On-Demand.29

LSDF Online Storage = External storage for special users

- **LSDF Online Storage**
  - Available only on bwUniCluster 2.0 and on HoreKa for special users
    - intended usage for scientific measurement data and data-intensive scientific simulation results

    → https://www.scc.kit.edu/en/services/11228.php
  - Visible on login nodes and on batch job nodes if access was requested
    - Access from external with different protocols is also possible
  - Request access in job script or on command line:

```
#SBATCH --constraint=LSDF
```

```
$ sbatch -C LSDF ...
```

  - Use environment variables $LSDF, $LSDFPROJECTS, $LSDFHOME to access, e.g.

```
$ cd ${LSDF}
```

  - HowTo:

    → https://wiki.bwhpc.de/e/BwUniCluster_2.0_Hardware_and_Architecture#LSDF_Online_Storage

Workspaces on flash storage

- **Workspaces on flash storage**
  - Available only on bwUniCluster 2.0 and on HoreKa for KIT users and HoreKa users
    - File system is visible on all nodes of both clusters
    - All storage devices are based on flash (no hard disks)

      → low access times and higher IOPS rates

      → mainly useful on bwUniCluster because of long network distance from HoreKa
  - Use via workspace commands
    - Add switch *-F ffuc* on bwUniCluster 2.0 and *-F ffhk* on HoreKa
    - Path to each workspace is visible and can be used on both clusters
  - Show quota usage and limits:

```
$ lfs quota -uh $(whoami) /pfs/work8
```

  - HowTo:

    → https://wiki.bwhpc.de/e/BwUniCluster2.0/Hardware_and_Architecture#Workspaces_on_flash_storage

Exercise 6: Run performance tests

- **Create interactive session**
  - BwUniCluster:

```
$ salloc -p single --reservation=ws -n 1 -t 20 --mem=1000
```

- **Sequential write throughput**
  - On workspace

```
$ dd if=/dev/zero of=$(ws_find ws01)/dd_file bs=1G count=2
```

  - On $TMPDIR

```
$ dd if=/dev/zero of=${TMPDIR}/$(whoami)_dd_file bs=1G count=2
```

- **Random I/O (IOPS) performance**
  - Define program path of fio

    BwUniCluster:

```
$ fio="/opt/bwhpc/common/workshops/2024-04-09/pfs_perf/fio"
```

  - On workspace

```
$ $fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=test \
--filename=$(ws_find ws01)/fio_file --bs=4k --iodepth=64 --size=300M --readwrite=randwrite
```

  - On $TMPDIR

```
$ $fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=test \
--filename=$TMPDIR/fio_file --bs=4k --iodepth=64 --size=300M --readwrite=randwrite
```

Exercise 7: Run performance tests

- Release workspaces

```
$ ws_release ws01

$ ws_release -F ffuc ws_ssd

$ ws_list
```

Remarks on energy efficiency

- Reducing I/O is always good
  - Speeds up your application and reduces power costs
  - For example omit debug output in normal job runs
- Optimizing I/O reduces power costs
  - Using local storage reduces power consumption on network and servers
  - Reduced job runtime typically means less power consumption
  - For example use the file system which fits best for your I/O