bw|HPC

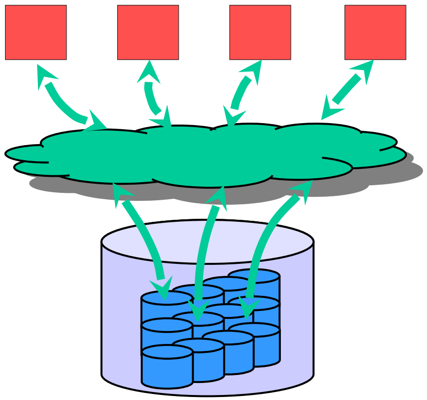
KIT | NHR
National High-Performance Computing Center

File Systems

Begatim Bytyqi
KIT, SCC

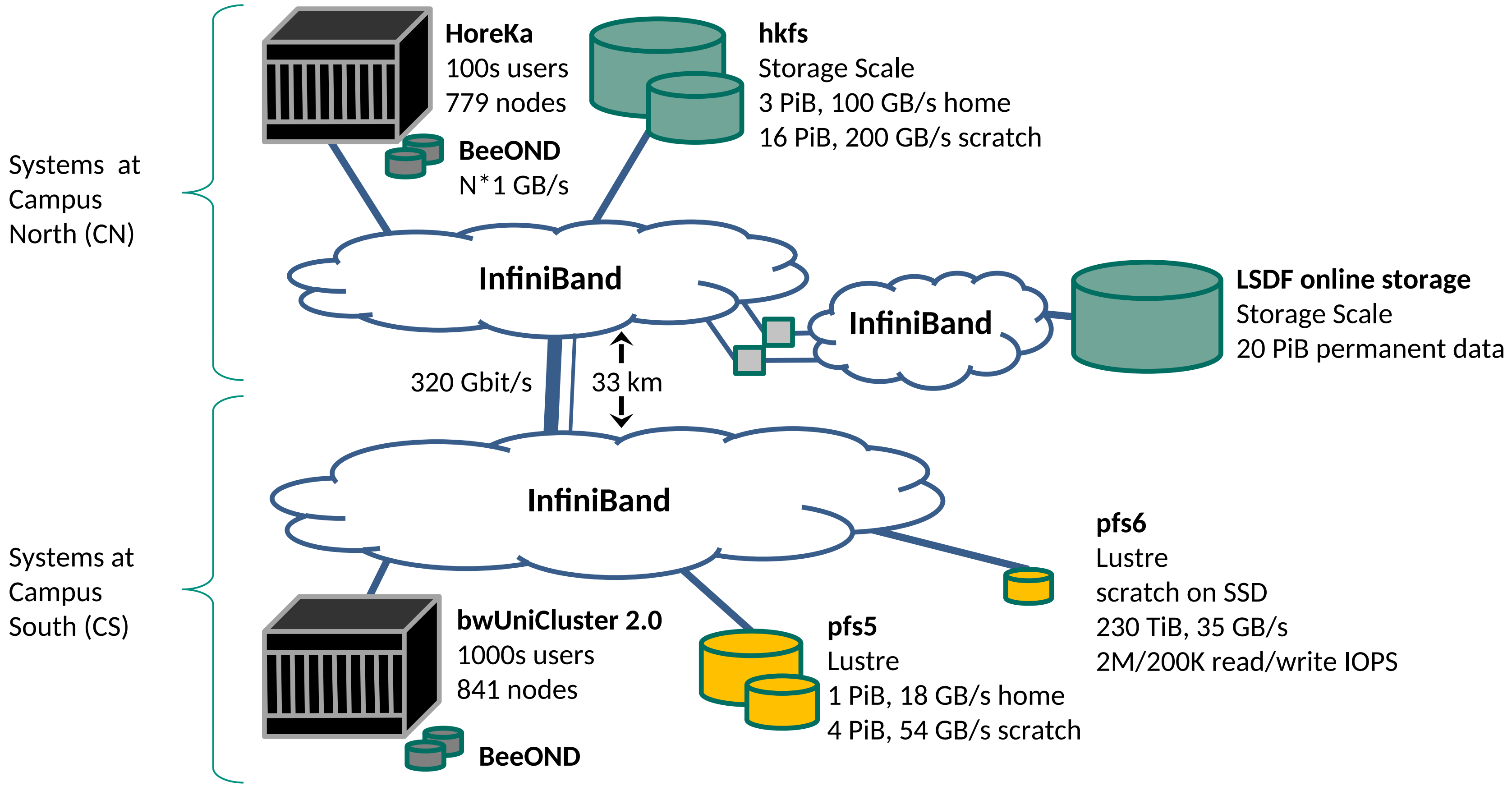
Funding: Baden-Württemberg MINISTERIUM FÜR WISSENSCHAFT, FORSCHUNG UND KUNST
Bundesministerium für Bildung und Forschung

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 with bwUniCluster 2.0



Systems at Campus North (CN)

HoreKa
100s users
779 nodes

BeeOND
N*1 GB/s

hkfs
Storage Scale
3 PiB, 100 GB/s home
16 PiB, 200 GB/s scratch

LSDF online storage
Storage Scale
20 PiB permanent data

Systems at Campus South (CS)

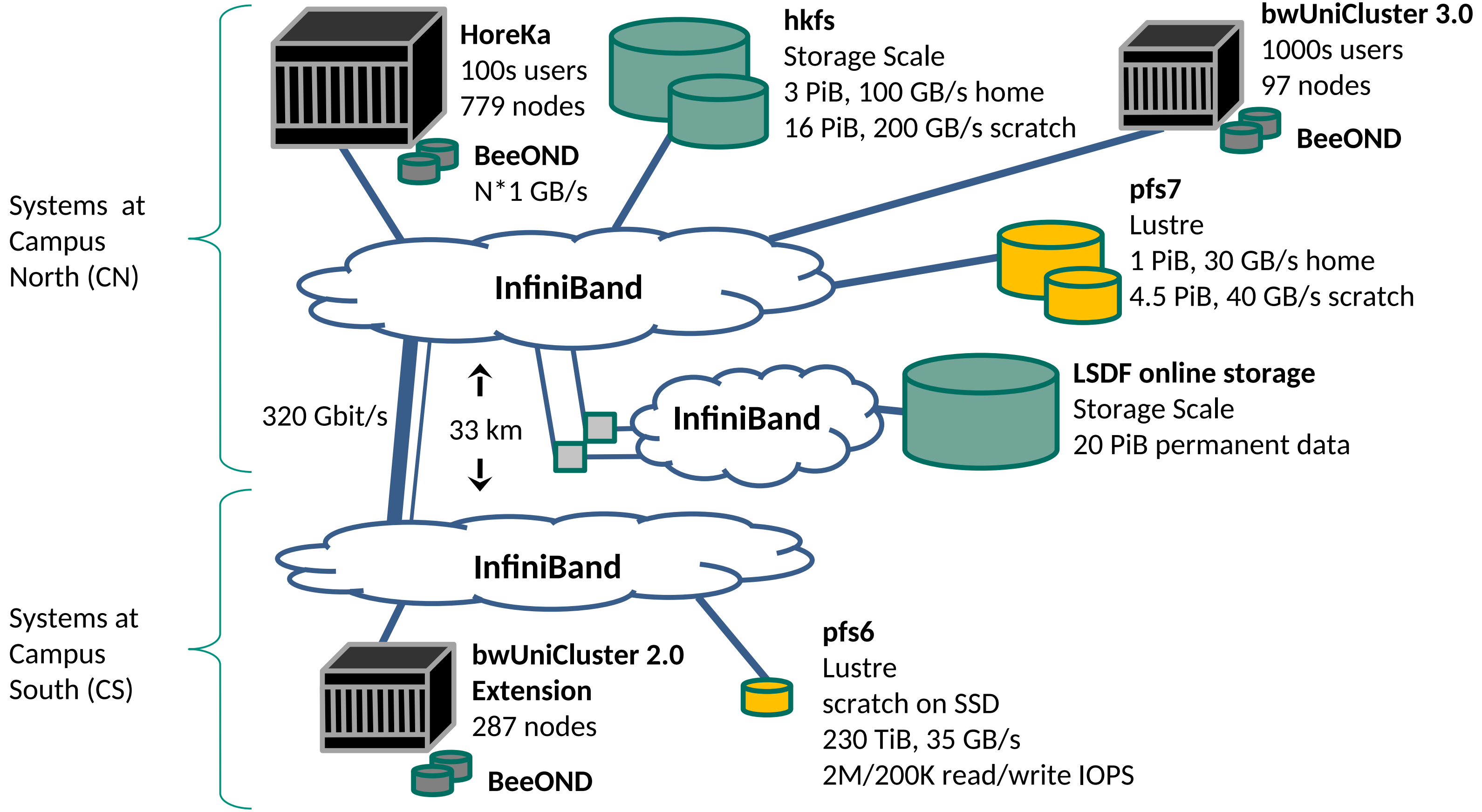
bwUniCluster 2.0
1000s users
841 nodes

BeeOND

pfs5
Lustre
1 PiB, 18 GB/s home
4 PiB, 54 GB/s scratch

pfs6
Lustre
scratch on SSD
230 TiB, 35 GB/s
2M/200K read/write IOPS

HPC clusters and file systems @ KIT with bwUniCluster 3.0



Systems at Campus North (CN)

HoreKa
100s users
779 nodes

BeeOND
N*1 GB/s

hkfs
Storage Scale
3 PiB, 100 GB/s home
16 PiB, 200 GB/s scratch

bwUniCluster 3.0
1000s users
97 nodes

BeeOND

InfiniBand

pfs7
Lustre
1 PiB, 30 GB/s home
4.5 PiB, 40 GB/s scratch

320 Gbit/s

33 km

InfiniBand

LSDF online storage
Storage Scale
20 PiB permanent data

InfiniBand

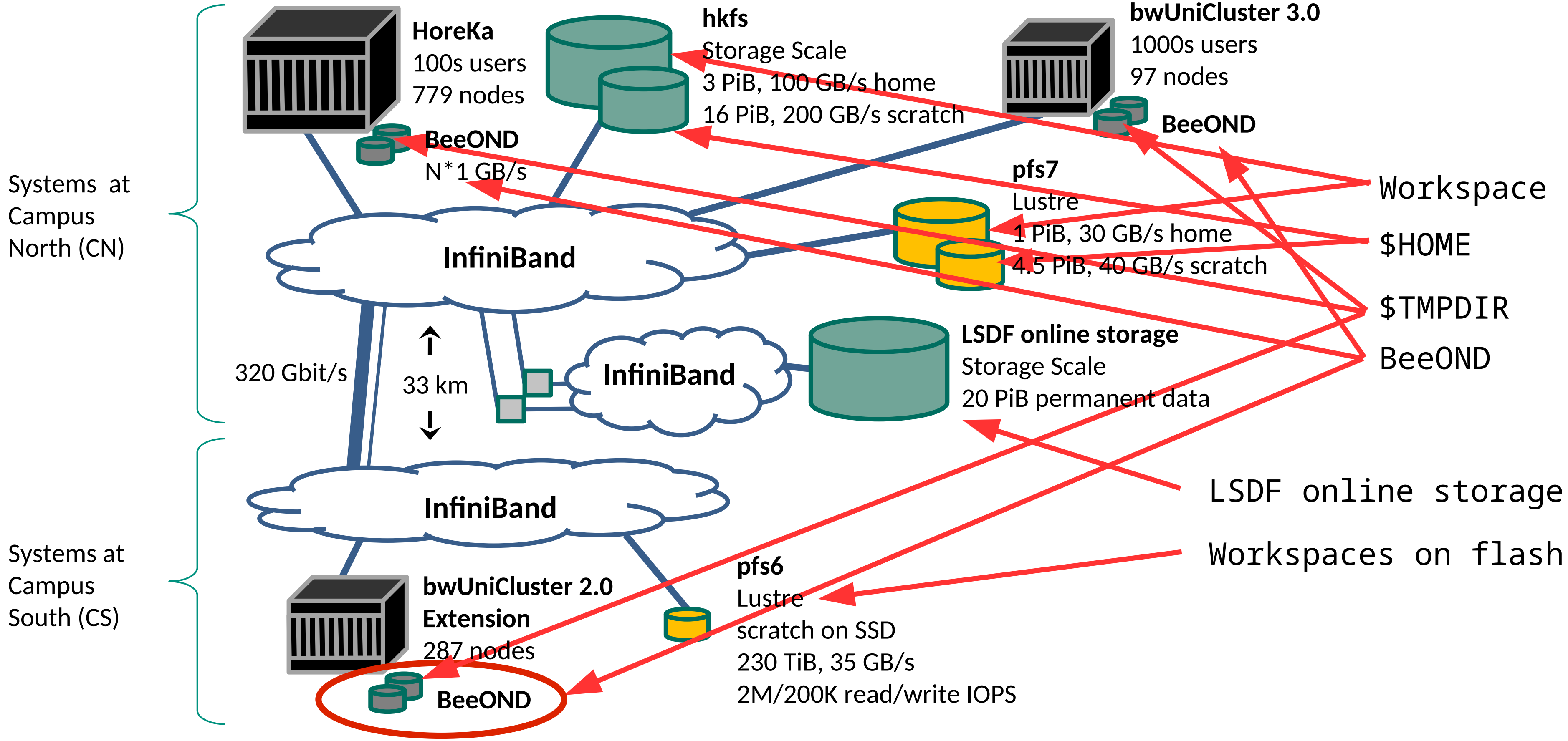
Systems at Campus South (CS)

bwUniCluster 2.0 Extension
287 nodes

BeeOND

pfs6
Lustre
scratch on SSD
230 TiB, 35 GB/s
2M/200K read/write IOPS

Names and locations of HPC clusters file systems @ KIT



File System properties overview

Property	$HOME	Workspace	$TMPDIR	BeeOND[1]	LSDF[1]	WS on flash[1]
Visibility	+++	+++	+	++	+++	+++
Lifetime	+++	++	+	+	+++	++
Capacity	+	+++	+	++	++	+
Seq. perf.	++	+++	+	+++	++	+++
Random perf.	+	+	+++	++	+	++
Impact on other users	+++	++	+	+	+++	++
Backup	+	–	–	–	+	–

[1] Only available on bwUniCluster 3.0 and HoreKa

File System detailed 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 - unlimited	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 3.0 and HoreKa

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/BwUniCluster3.0/Hardware_and_Architecture#$TMPDIR
- **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 3.0 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/BwUniCluster3.0/Hardware_and_Architecture#BeeOND_(BeeGFS_On-Demand)

LSDF Online Storage = External storage for special users

- **LSDF Online Storage**
  - Available only on bwUniCluster 3.0 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/BwUniCluster3.0/Hardware_and_Architecture#LSDF_Online_Storage

Workspaces on flash storage

- **Workspaces on flash storage**
  - Available only on bwUniCluster 3.0 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

    → use this file system with queue cpu_il (Ice Lake nodes) on bwUniCluster 3.0

    Note: Long network distance and high latency from these nodes to normal workspace file system
  - Use via workspace commands
    - Add switch *-F ffuc* on bwUniCluster 3.0 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/BwUniCluster3.0/Hardware_and_Architecture/Filesystem_Details#Workspaces_on_flash_storage

Remarks for exercise

- Login to bwUnicluster 3.0 or HoreKa and show list of commands for exercises:
  - BwUniCluster:

    ```
    $ cat /opt/bwhpc/common/workshops/2025-04-10/pfs_commands.txt
    ```

  - HoreKa:

    ```
    $ cat /software/all/workshop/2025-04-10/pfs_commands.txt
    ```

- Use Cut & Paste to execute the commands
  - Start with the first command to create workspace *ws01*

Exercise 1: 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/2025-04-10/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
```