Intermediate Filesystems Course

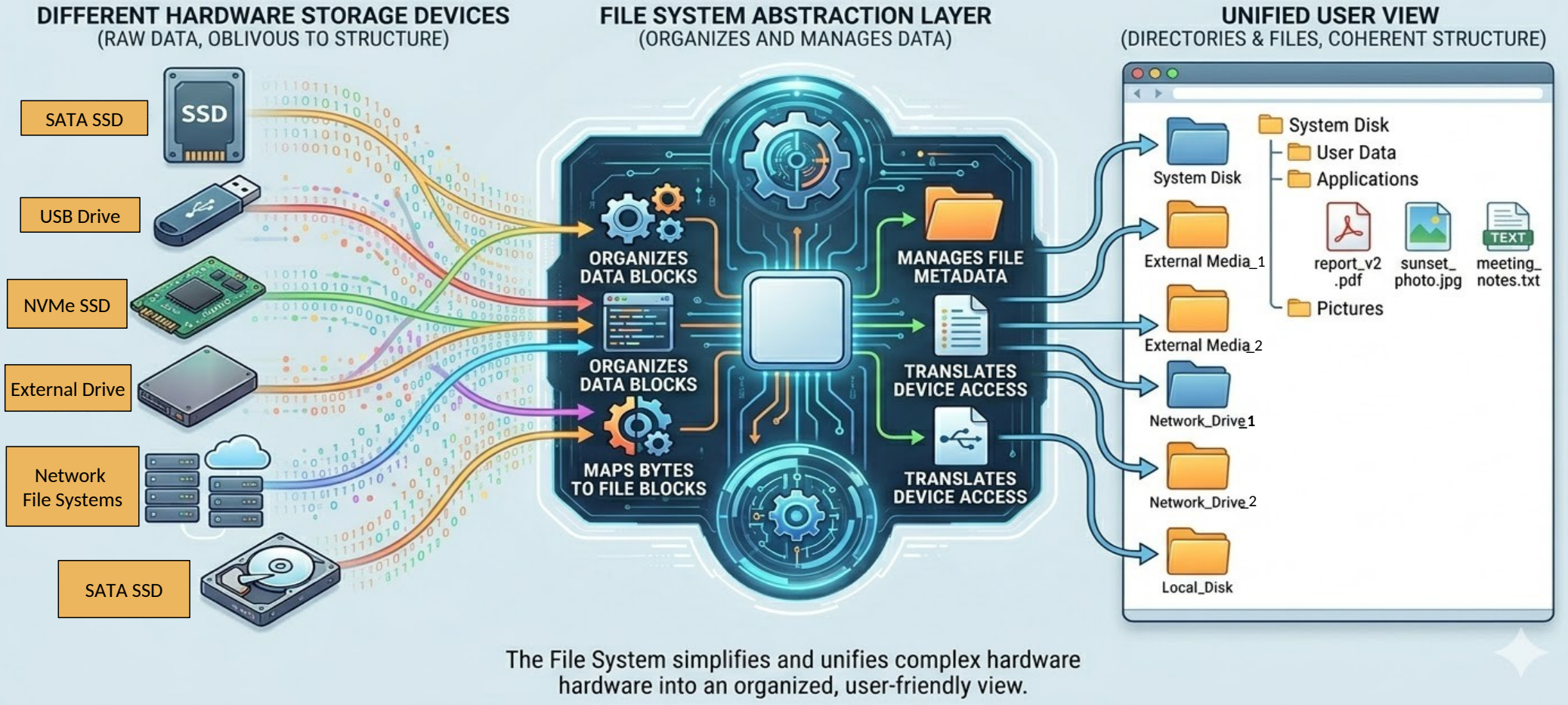
Begatim Bytyqi (KIT/SCC)

Baden-Württemberg
Ministerium für Wissenschaft,
Forschung und Kunst

Motivation & Goals

- **In this lesson**
  - Understand the core mechanics of file systems, including key concepts like:
    - **File, file system, virtual file system, inodes ...**
  - Compare different file systems by evaluating their pros & cons
  - Select the most appropriate file system for your specific use case
  - Exercises ....

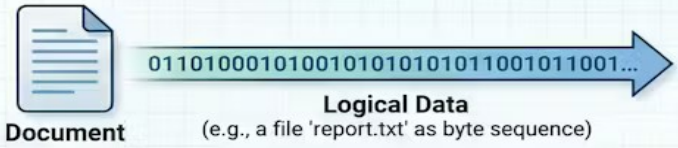
HOW A FILE SYSTEM ABSTRACTS HARDWARE COMPLEXITY



The File System simplifies and unifies complex hardware hardware into an organized, user-friendly view.

CONCEPT OF A FILE

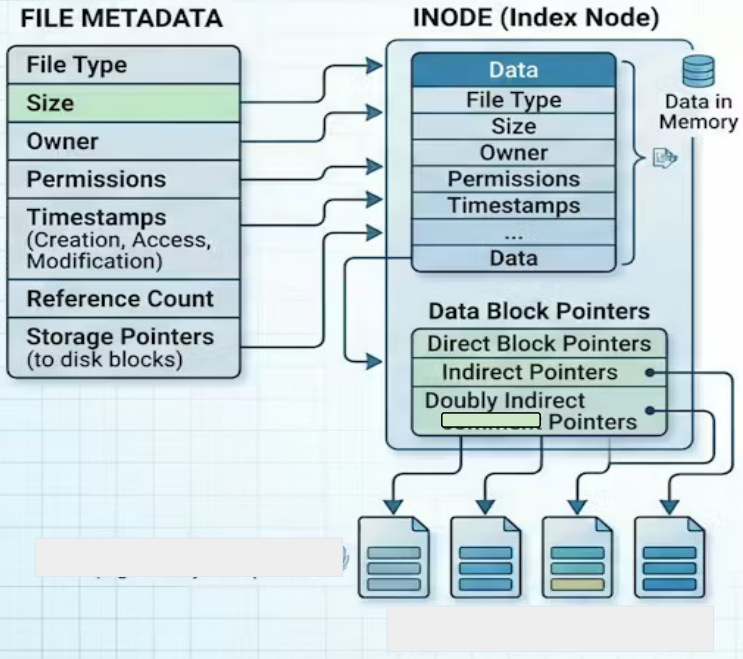
USER PERSPECTIVE



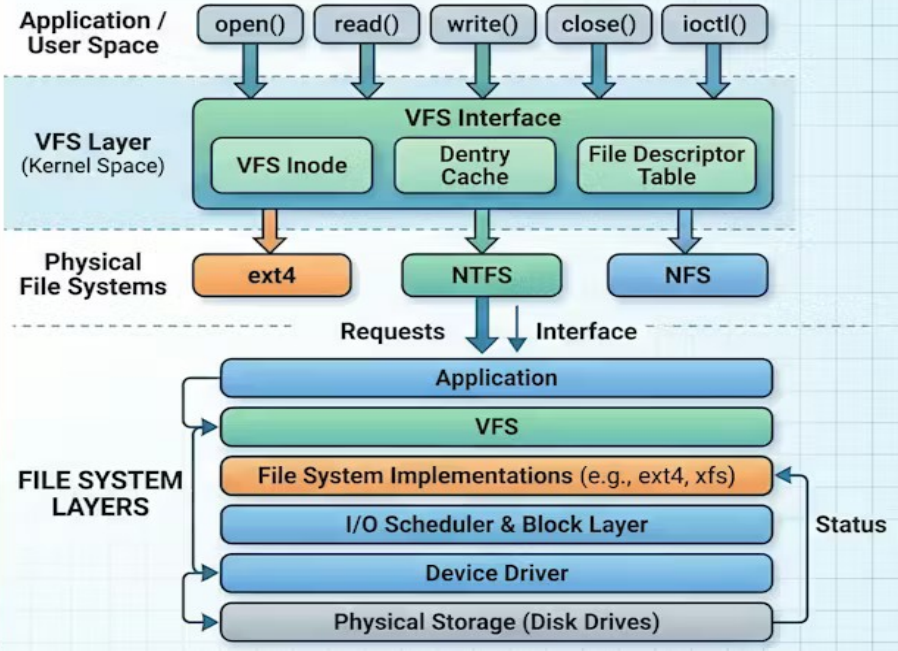
OS PERSPECTIVE

- **Everything is a file (Unix Operating Systems)**
- **File Metadata stored in Inodes**
- **File Data as a chain of data-blocks**
- **Directories are files as well.**

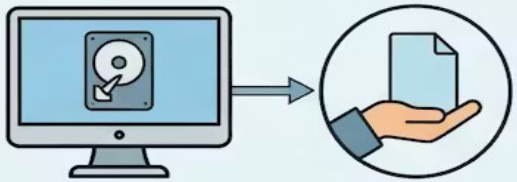
METADATA & INODES



VFS (VIRTUAL FILE SYSTEM)

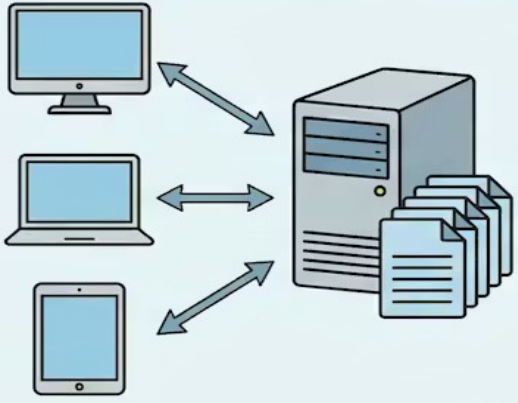


1. LOCAL FILE SYSTEM



Data stored locally, accessed by one computer only.

2. NETWORK FILE SYSTEM (NFS)

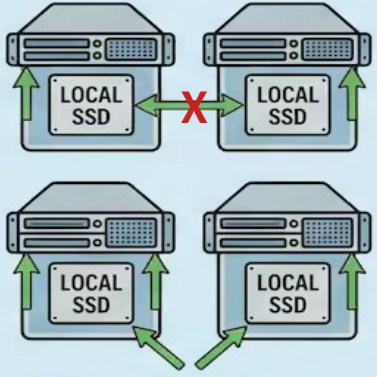


Data shared across a network, accessed by multiple computers.

THE DIFFERENCE: LOCAL NODE vs. PARALLEL FILE SYSTEM (In HPC)

LOCAL NODE FILE SYSTEM

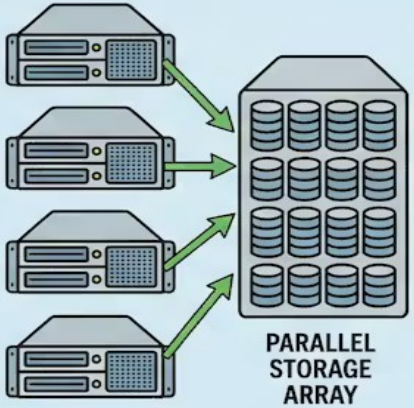
Each node uses its own local SSD for high-speed but isolated storage.



* Limited capacity
* Data is private to each node

PARALLEL FILE SYSTEM

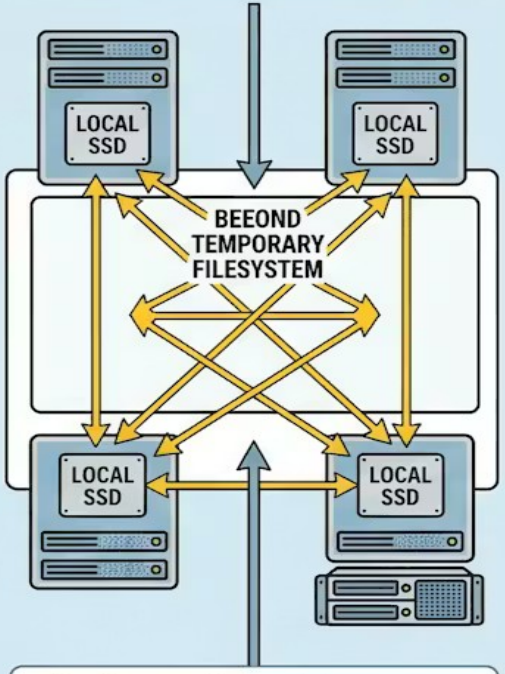
Single, shared namespace across all nodes, data striped across many storage targets for high performance.



* Large capacity
* Concurrent access
* Ideal for large datasets

SOMETHING IN-BETWEEN: BEEOND (BeeGFS On Demand)

A network filesystem but uses local SSDs for data storage.

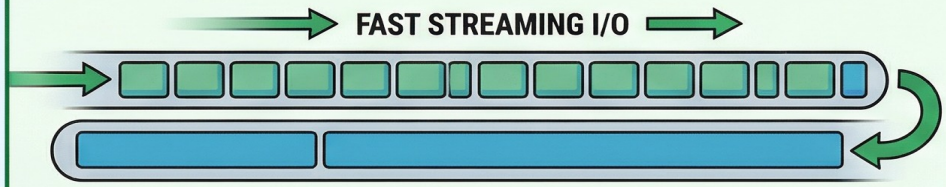


Creates a performant, temporary parallel filesystem across local node resources for specific compute jobs.

PARALLEL FILE SYSTEM CONSIDERATIONS

1. SEQUENTIAL DATA IS IDEAL

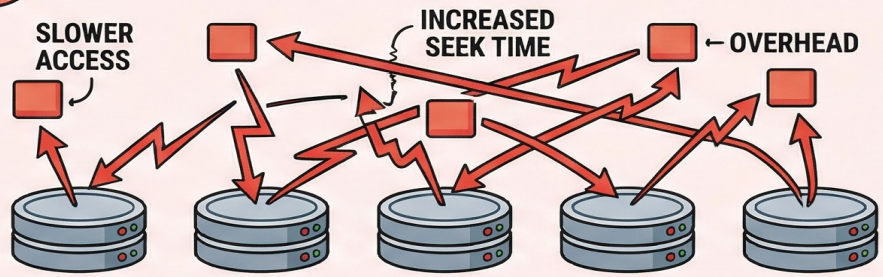
OPTIMIZED PERFORMANCE



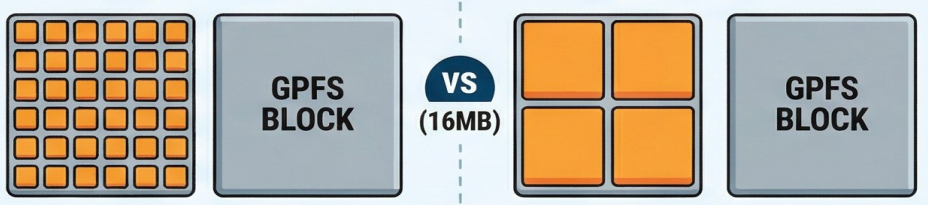
EX. SINGLE LARGE 1GB FILE

2. WHAT IS RANDOM I/O?

POOR PERFORMANCE

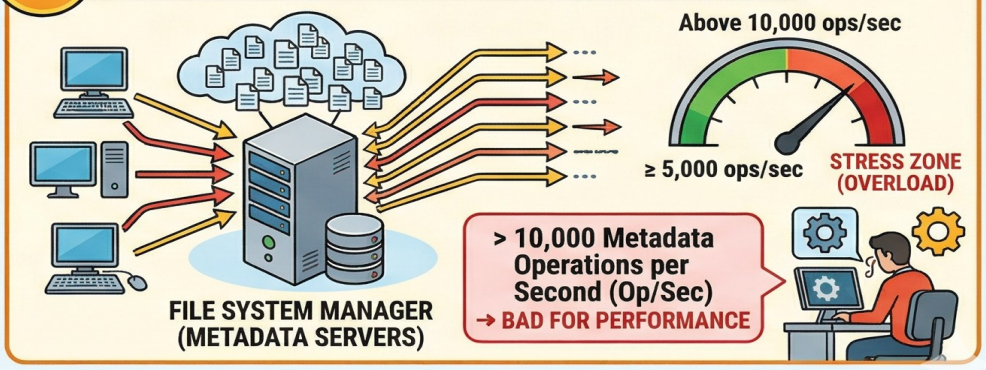


3. SMALL FILES vs. LARGE FILES (GPFS 16MB BLOCK)

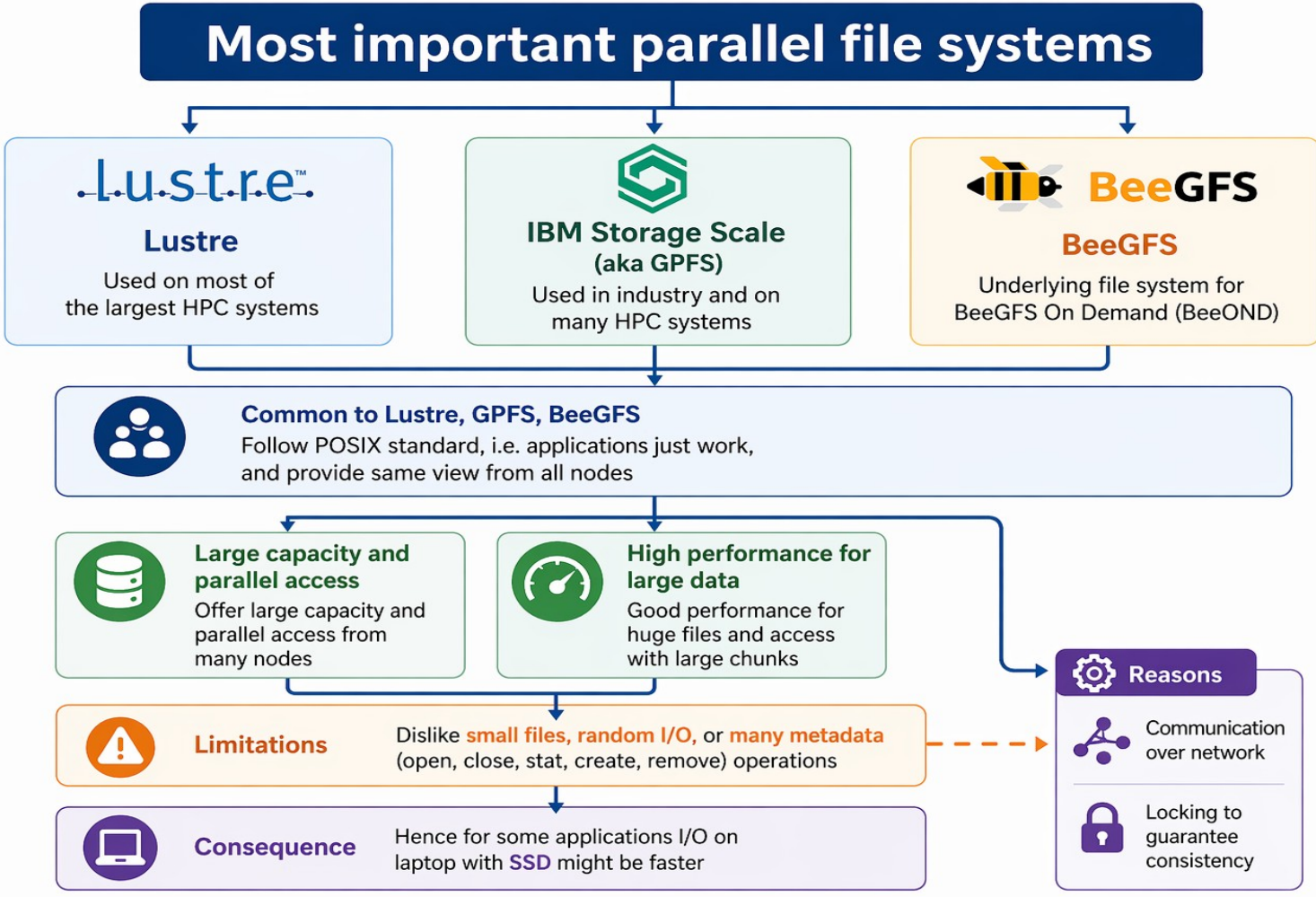


MINIMUM RECOMMENDED FILE SIZE >= 16 MB

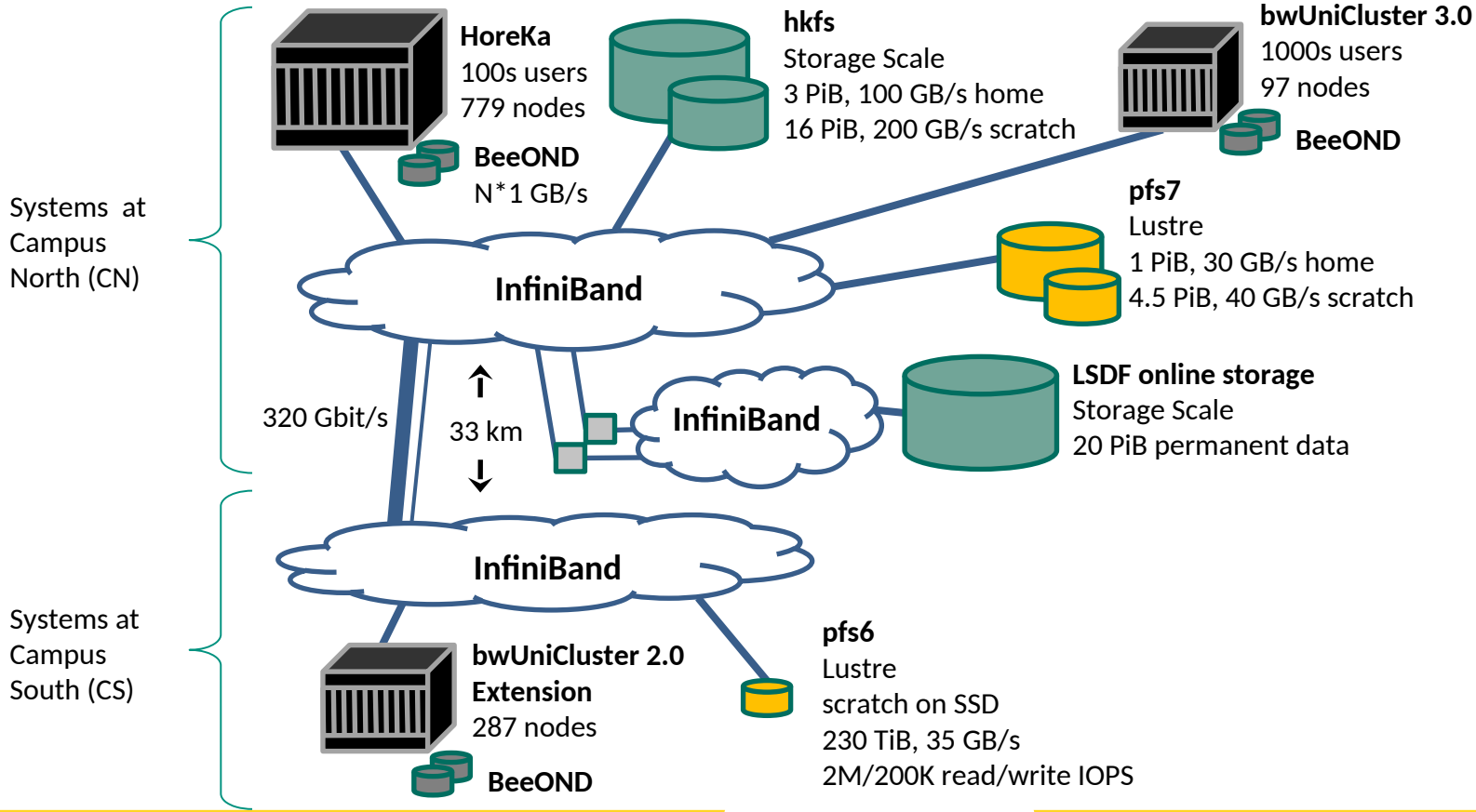
4. HOW MANY FILES? (METADATA STRESS)



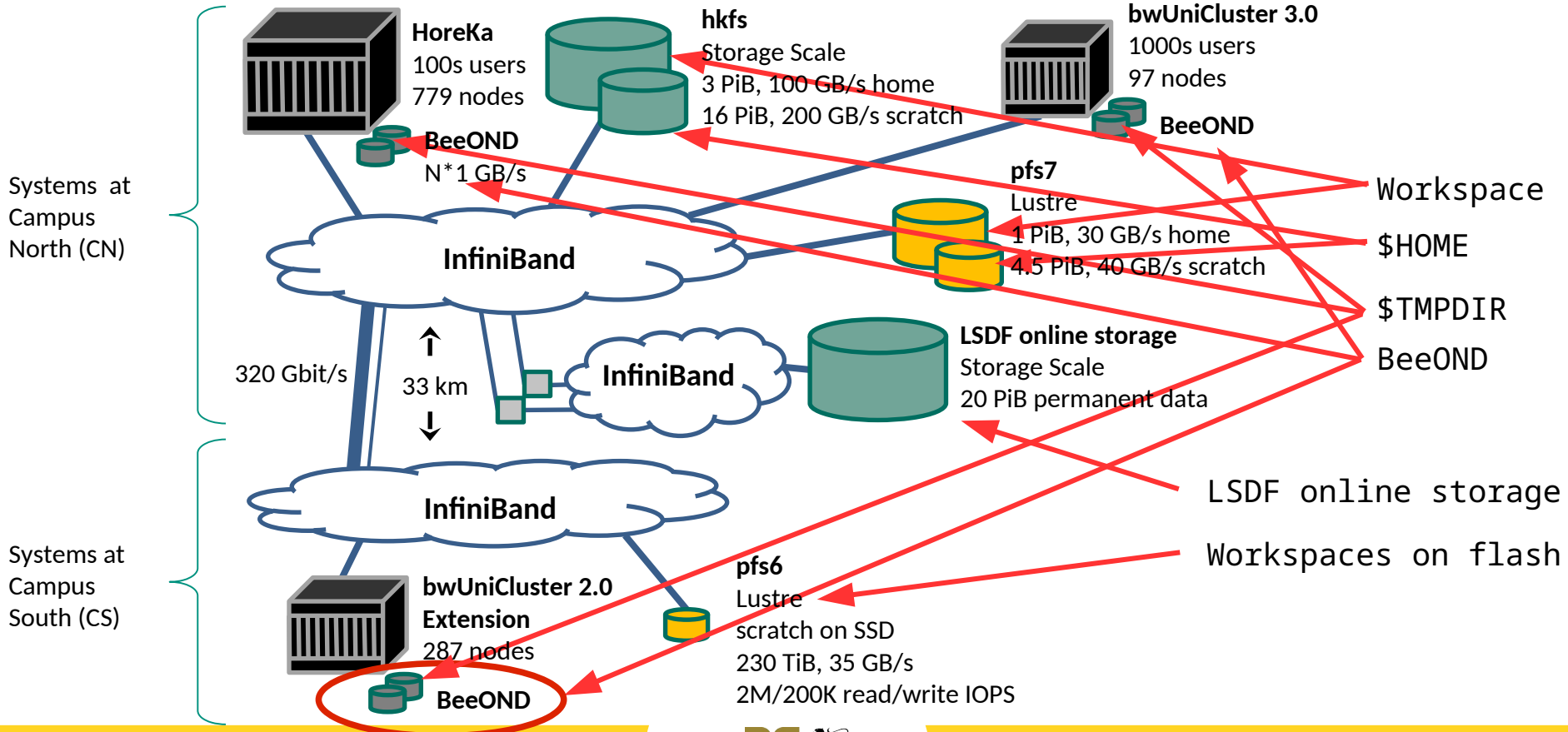
Most important parallel file systems

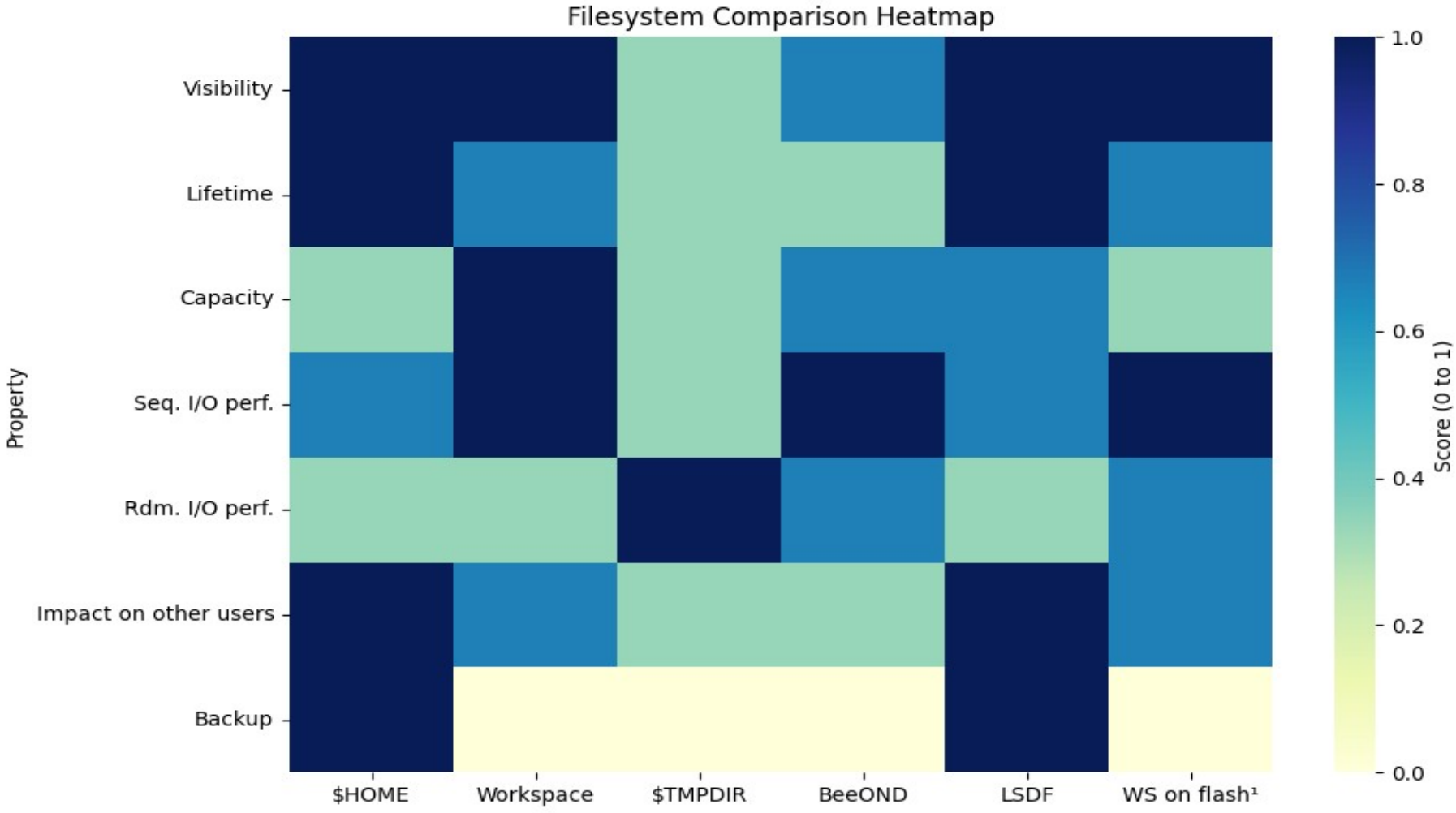


HPC clusters and file systems @ KIT



Names and locations of HPC clusters file systems @ KIT




Filesystem Comparison Heatmap
Property
Visibility
Lifetime
Capacity
Seq. I/O perf.
Rdm. I/O perf.
Impact on other users
Backup
$HOME
Workspace
$TMPDIR
BeeOND
LSDF
WS on flash¹
Score (0 to 1)
1.0
0.8
0.6
0.4
0.2
0.0

File System detailed properties of bwHPC clusters and of HoreKa

Property	$HOME	Workspace	$TMPDIR	BeeOND[1]	LSDF1	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 file system stores data in main memory and $SCRATCH is on local SSD
  - Separate temporary private directory on each node of a batch job
    - 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
    - Data staging is **required**
  - Request creation in job script or on command line:
  - Exclusive allocation is **required.**
- Request creation on job script or using the command line

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

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

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

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

→ 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
  - External 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

- 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 (similar to normal Workspace filesystems)
  - 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


1 PLAN & FUND
DATA MANAGEMENT PLAN (DMP)
Research data management begins with systematic planning: What data are needed, collected, processed and stored?
• Project funders usually require structured data management as early as the application stage.
• Funds for research data management can be applied for.
Tools
RDMO
Plan data management with RDMO
DMPTool
Create DMPs for funders and project management
2 COLLECT & ANALYZE
• The generation and collection of data, e.g. with sensors or simulations on high-performance computers, is essentially the responsibility of researchers.
• Analysis, e.g. by algorithms & software.
• Collect the necessary metadata to describe the research data: scientific, administrative, and IT metadata.
Tool: Electronic Laboratory Notebook (ELN)
Supports researchers in data collection and analysis.
3 PRESERVE & STORE
• Store research data in appropriate, subject-specific or institutional repositories.
• Store together with good documentation to enable subsequent use, especially by third parties.
• Important criterion: long-term archiving, e.g. via tape storage.
Repositories
Safe. Reliable. Sustainable.
Long-term archiving
e.g. via tape storage
4 PUBLISH & SHARE
• Define access conditions: access and usage rights, including patents and licenses.
• Assign persistent identifiers (PIDs) to uniquely identify and reference files.
• Integrate existing networks and specialist communities via trusted and, if possible, certified infrastructure.
PID
Access & Usage Rights
Licenses & Patents
doi
Persistent Identifiers (PIDs)
Certified Infrastructure
5 DISCOVER & REUSE
• Good research data management enables other researchers to search for and reuse results.
• Build on the current state of knowledge without generating data again at great expense.
• Find data via repositories and aggregating services such as re3data or DataCite.
• Observe the legal framework and good scientific practice.
re3data.org
DataCite
Legal framework
Good scientific practice
RESEARCH DATA MANAGEMENT
Good data. Trusted research. Greater impact.
Quality & Integrity
Ensure data quality and integrity at every step.
Collaboration
Work together. Share knowledge.
Sustainability
Plan for long-term value and preservation.
Impact
Well-managed data enable better research and society.

Remarks for exercise

- Login to bwUnicluster 3.0 or HoreKa and show list of commands for exercises:
  - BwUniCluster:
    ```
    $ cat /opt/bwhpc/common/workshops/2026-04-15/pfs_commands.txt
    ```
  - HoreKa:
    ```
    $ cat /software/all/workshop/2026-04-15/pfs_commands.txt
    ```
- Use Cut & Paste to execute the commands
  - Start with the first command to create workspace *ws01*

Exercise 1: Run performance tests on $TMPDIR and $Workspace

- **Create interactive session**
  - BwUniCluster:
```
$ salloc -p single --reservation=ws -n 1 -t 20 --mem=1000
```
- **Sequential write throughput**
  - On workspace
```
$ fio --filename=$(ws_find ws01)/seq_file --rw=write --bs=1G --size=2G
--ioengine=sync
```
  - On $TMPDIR
```
$ fio --filename=${TMPDIR}/$(whoami)_seq_file --rw=write --bs=1G --size=2G
--ioengine=sync –zero_buffers
```
- **Random I/O (IOPS) performance**
  - Define program path of fio

    BwUniCluster:
```
$ fio="/opt/bwhpc/common/workshops/2026-04-15/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 2: Run performance tests on BeeOND and $Workspace on Flash(WSF)

- **Create interactive session**
  - BwUniCluster:
```
$ salloc -p single --reservation=ws -n 1 -t 20 --mem=1000
```
- **Sequential write throughput**
  - On WSF
```
$ fio --filename=$(ws_find ws02)/seq_file --rw=write --bs=1G --size=2G
--ioengine=sync –zero_buffers
```
  - On BeeOND
```
$ fio --filename=/mnt/odfs/${SLURM_JOBID}/stripe_deafult/seq_file --rw=write --bs=1G --size=2G
--ioengine=sync –zero_buffers
```
- **Random I/O (IOPS) performance**
  - **Define program path of fio**
```
$ fio="/opt/bwhpc/common/workshops/2026-04-15/pfs_perf/fio"
```
  - On WSF
```
$ $fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=test \
--filename=$(ws_find ws02)/fio_file --bs=4k --iodepth=64 --size=300M
--readwrite=randwrite
```
  - On BeeOND
```
$ $fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=test
--bs=4k --filename=/mnt/odfs/${SLURM_JOBID}/stripe_deafult/fio_file
--bs=4k --iodepth=64 --size=300M --readwrite=randwrite
```