

Table of Contents

File Systems	2
Available file systems	2
Stripe Pattern Details	3
Additional infos	3
Notes	3

File Systems

Available file systems

On the DEEP-EST system, three different groups of file systems are available:

- the [?JSC GPFS file systems](#), provided via [?JUST](#) and mounted on all JSC systems;
- the DEEP-EST (and SDV) parallel BeeGFS file systems, available on all the nodes of the DEEP-EST system;
- the file systems local to each node.

The users home folders are placed on the shared GPFS file systems. With the advent of the new user model at JSC ([?JUMO](#)), the shared file systems are structured as follows:

- **\$HOME:** each JSC user has a folder under `/p/home/jusers/`, in which different home folders are available, one per system he/she has access to. These home folders have a low space quota and are reserved for configuration files, ssh keys, etc.
- **\$PROJECT:** In JUMO, data and computational resources are assigned to projects: users can request access to a project and use the resources associated to it. As a consequence, each user has a folder within each of the projects he/she is part of. For the DEEP project, such folder is located under `/p/project/cdeep/`. Here is where the user should place data, and where the old files generated in the home folder before the JUMO transition can be found.

The DEEP-EST system doesn't mount the \$SCRATCH and \$ARCHIVE file systems, as it is expected to provide similar functionalities with its own parallel file systems.

The following table summarizes the characteristics of the file systems available in the DEEP-EST and DEEP-ER (SDV) systems:

Mount Point	User can write/read to/from	Cluster	Type	Global / Local	SW Version	Stripe Pattern Details	Maximum Measured Performance (see footnotes)	Description	Other
/ghome	/ghome/adv	SDV DEEP-EST	GPFS exported via NFS	Global				Home directory; used only for configuration files.	
/project	/project/adv	SDV DEEP-EST	GPFS exported via NFS	Global				Project directory; GPFS main storage file system; not suitable for performance-relevant applications or benchmarking	
/work	/work/adv	DEEP-EST*	BestGPFS	Global	BestGPFS 7.1.2			Work file system	*Not available in the SDV but only through a Log network connection
/scratch	/scratch	DEEP-EST	efs local partition	Local*				Scratch file system	*No comments to use instead of temp for storing temporary files
/home/localhost	/home/localhost	DAM partition	efs local SSD partition	Local*				Scratch file system	*1.5 TB total QnapESD Data Center (DC) P4000X (NVMe) PC4x 4x, 2.5", 3D XPoint)
/home/localhost	/home/localhost	DAM partition	efs4 local SSD partition	Local*				Scratch file system	*1.5 TB total QnapESD Data Center (DC) P4000X (NVMe) PC4x 4x, 2.5", 3D XPoint)
/adv-work	/adv-work/adv	SDV (deep-adv nodes via EXT4L, deep-adv n/wg via GSE only) (DEEP-EST (G SE only))	BestGPFS	Global	BestGPFS 7.1.2	Type: RAID0, Controller: 512K, Number of storage targets: element 4	2037.05 MB/s write, 1300.82 MB/s read 16MB read 15202 ops/sec read, 5111 ops/sec write*	Work file system	*Test results and parameters used stored in JURE. user@deep-er \$ cd /root/localhost; user@deep-er \$ jiohelp result benchmarks user@deep-er \$ cd /root/localhost; user@deep-er \$ jiohelp result benchmarks
/home	/home/adv	SDV	NVMe device	Local	BestGPFS 7.1.2	Block size: 4K	1145 MB/s write, 3108 MB/s read 1393.68 ops/sec read, 18257 ops/sec write*	1 NVMe device available at each SDV compute node	*Test results and parameters used stored in JURE. user@deep-er \$ cd /root/localhost; user@deep-er \$ jiohelp result benchmarks user@deep-er \$ cd /root/localhost; user@deep-er \$ jiohelp result benchmarks
/home/adv	/home/adv	SDV	BestGPFS On Demand running on the NVMe	Local	BestGPFS 7.1.2	Block size: 512K	1130 MB/s write, 2447 MB/s read 1251.4 ops/sec read, 16424 ops/sec write*	1 BestOND instance at each SDV device	*Test results and parameters used stored in JURE. user@deep-er \$ cd /root/localhost; user@deep-er \$ jiohelp result benchmarks user@deep-er \$ cd /root/localhost; user@deep-er \$ jiohelp result benchmarks

Stripe Pattern Details

It is possible to query this information from the deep login node, for instance:

```
manzano@deep $ fhgfs-ctl --getentryinfo /work/manzano
Path: /manzano
Mount: /work
EntryID: 1D-53BA4FF8-3BD3
Metadata node: deep-fs02 [ID: 15315]
Stripe pattern details:
+ Type: RAID0
+ Chunksize: 512K
+ Number of storage targets: desired: 4

manzano@deep $ beegfs-ctl --getentryinfo /sdv-work/manzano
Path: /manzano
Mount: /sdv-work
EntryID: 0-565C499C-1
Metadata node: deeper-fs01 [ID: 1]
Stripe pattern details:
+ Type: RAID0
+ Chunksize: 512K
+ Number of storage targets: desired: 4
```

Or like this:

```
manzano@deep $ stat -f /work/manzano
File: "/work/manzano"
ID: 0      Namelen: 255      Type: fhgfs
Block size: 524288      Fundamental block size: 524288
Blocks: Total: 120178676 Free: 65045470 Available: 65045470
Inodes: Total: 0        Free: 0

manzano@deep $ stat -f /sdv-work/manzano
File: "/sdv-work/manzano"
ID: 0      Namelen: 255      Type: fhgfs
Block size: 524288      Fundamental block size: 524288
Blocks: Total: 120154793 Free: 110378947 Available: 110378947
Inodes: Total: 0        Free: 0
```

See <http://www.beegfs.com/wiki/Striping> for more information.

Additional infos

Detailed information on the **BeeGFS Configuration** can be found [?here](#).

Detailed information on the **BeeOND Configuration** can be found [?here](#).

Detailed information on the **Storage Configuration** can be found [?here](#).

Detailed information on the **Storage Performance** can be found [?here](#).

Notes

- The /work file system which is available in the DEEP-EST prototype, is as well reachable from the nodes in the SDV (including KNLs and ml-gpu nodes) but through a slower connection of 1 Gig. The file system is therefore not suitable for benchmarking or I/O task intensive jobs from those nodes
- Performance tests (IOR and mdtest) reports are available in the BSCW under DEEP-ER → Work Packages (WPs) → WP4 → T4.5 - Performance measurement and evaluation of I/O software → Jülich DEEP Cluster → Benchmarking reports:
<https://bscw.zam.kfa-juelich.de/bscw/bscw.cgi/1382059>

- Test results and parameters used are stored in JUBE:

```
user@deep $ cd /usr/local/deep-er/sdv-benchmarks/synthetic/ior
user@deep $ jube2 result benchmarks

user@deep $ cd /usr/local/deep-er/sdv-benchmarks/synthetic/mdtest
user@deep $ jube2 result benchmarks
```