

Wikiprint Book

Title: File Systems

Subject: DEEP - Public/User_Guide/Filesystems

Version: 36

Date: 04.05.2024 22:08:27

Table of Contents

File Systems	3
Available file systems	3
Stripe Pattern Details	4
Additional infos	4
Notes	4

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 can create folders within each of the projects he/she is part of (with either personal or permissions to share with other project members). For the DEEP project, the project 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 from GPFS, 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 select host software	Cluster	Type	Global / Local	BW Variant	Stripe Pattern Details	Maximum Measured Performance (see footnotes)	Description	Other
/home	/home/user	DEP-EST	GFS exported via NFS	Global				Home directory used only for configuration files.	
/opt/mgmt	/opt/mgmt/cluster	GV, DEEP-EST	GFS exported via NFS	Global				Project directory. GFS main storage file system; not suitable for performance relevant applications or benchmarks.	
/work	/work/bep	DEEP-EST	SeaWFS	Global	SeaGFS 7.1.2			Holds the system	*Also available in pre-GV but only through 1 Gb network
/scratch	/scratch	DEEP-EST	All local partition	Local*				Scratch file system for temporary data. Will be cleaned up after job finished	*Recommendation instead of using temporary file
/home/cluster	/home/cluster	DAM partition	local SSD (RAID 0)	Local*				Scratch file system for temporary data. Will be cleaned up after job finished	*1.5 TB total Optimized SSD Data Center (DC) PEXIDEH (PCIe x8, 2.0T, SD, xParity)
/home/cluster/home	/home/cluster/home	DAM partition	local SSD (RAID 0)	Local*				Scratch file system for temporary data. Will be cleaned up after job finished	*1.5 TB total Optimized SSD Data Center (DC) PEXIDEH (PCIe x8, 2.0T, SD, xParity)
/green/red	/green/red	DAM partition (all nodes)	OCF/M in replicated mode	Local*				2 Btu- dp-dmcc1_2 TB in dp-shg02.05	27 Jun 11, 13, 15]
/work-node	/work-node	GV (deep-edv nodes via EXTOLL, deep-edv-gpu via GV only), DEEP-EST (V OE only)	SeaGFS	Global	SeaGFS 7.1.2	Type: RADOS, Checksum: SIPM, Number of stripe groups: default = 4	1831 GB 1860 vmem, 1388 GB 1202 spinx mem, 5111 spinx mem	Work file system	See page 90: radw-benchmarks-report.html for details
/home	/home/myp	GV	N/A/no device	Local	SeaGFS 7.1.2	Block size: 4K	1145 MB/s write, 1108 MB/s read 12018 spinx mem, ACGF spinx mem	N/A/no device available in each GV node	See page 90: radw-benchmarks-report.html for details
/home/back	/home/back	GV	SeaWFS On Command mirroring the N/A's	Local	SeaGFS 7.1.2	Block size: 512K	1130 MB/s write, 1047 MB/s read 12011 spinx mem, each N/A device	Redundance mirroring on each N/A device	See page 90: radw-benchmarks-report.html for details

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
```