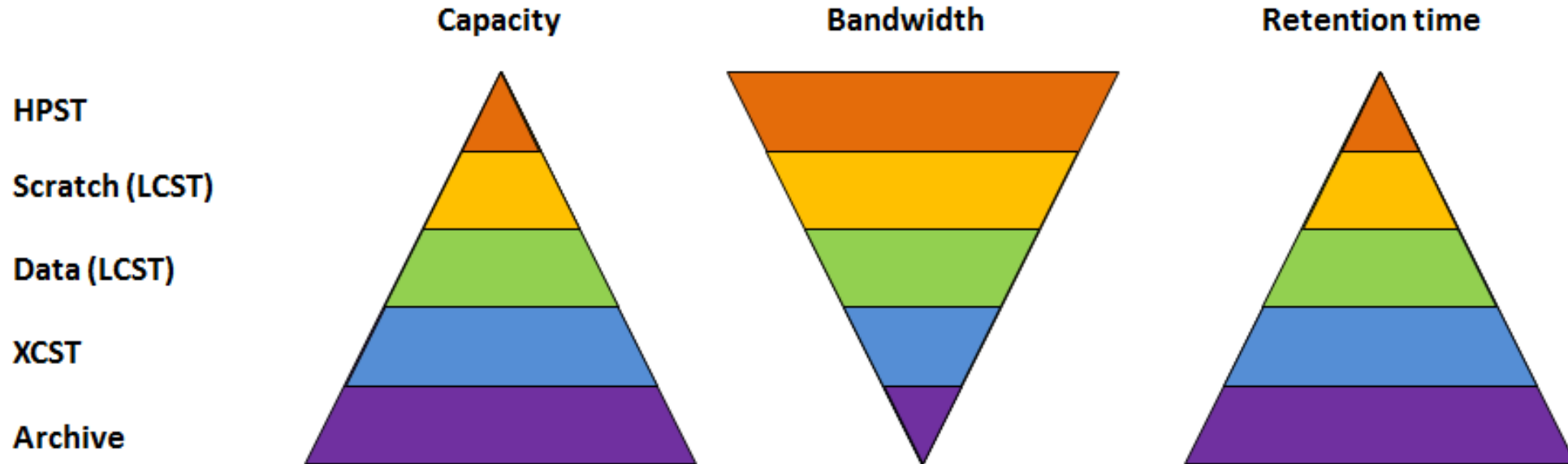


JUST JÜLICH STORAGE CLUSTER

30. MAY 2023 | STEPHAN GRAF (JSC)

TIERED STORAGE OFFERING



- High Performance Storage Tier (HPST): NVMe based Storage (low latency+high bandwidth)
- Large Capacity Storage Tier (LCST): Lenovo DSS Cluster (GNR, 5th Gen. of JUST, bandwidth optimized) → LCST
- Extended Capacity Storage Tier (XCST): GPFS Building Blocks (target: capacity) → XCST
- Archive: Tape storage (Backup + GPFS&TSM-HSM)

JUST CLUSTER(S)

Key Characteristics

- File system access: parallel, POSIX compliant
- No user login
- Cross mounted on HPC systems
- One global namespace

HPST	LCST	XCST	ARCHIVE
NVMe based Cache	Spinning Disc bandwidth optimized	Spinning Disc Capacity optimized	Tape
110 server	22 Building Blocks	8 Building Blocks	1 Building Block 4 Tape Libraries
1.100 NVMe drives (2TB)	7.500 x 10TB Discs	8.000 Discs (10TB, 12TB, 16TB)	>39.000 Tapes (8TB – 18TB)
2.2 PB (raw)	75 PB (raw)	~95 PB (raw)	>360 PB
Infinite Memory Engine	Spectrum Scale (GPFS)+ GPFS Native RAID	Spectrum Scale (GPFS)	Spectrum Scale + Spectrum Protect (GPFS + TSM-HSM)

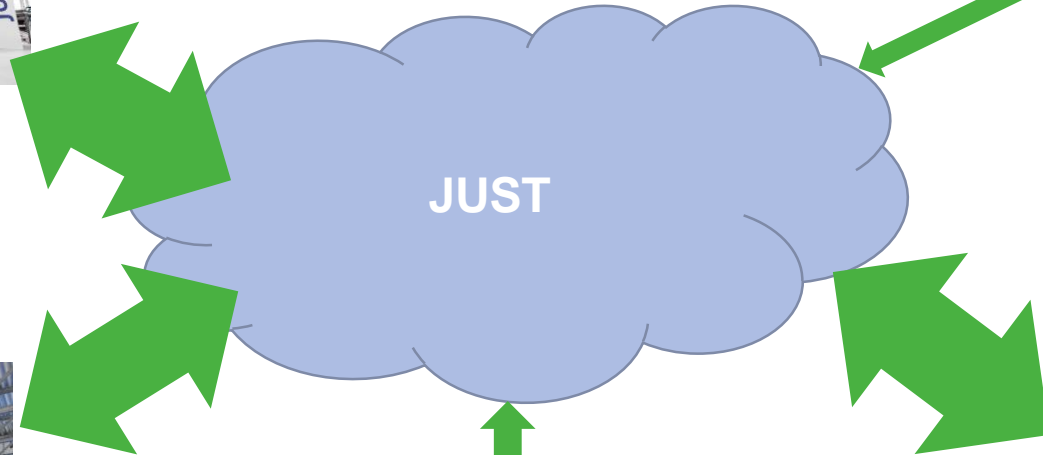
CENTRALIZED STORAGE



JUWELS + JUWELS Booster



JURECADC



DEEP



JUSUF



JUDAC

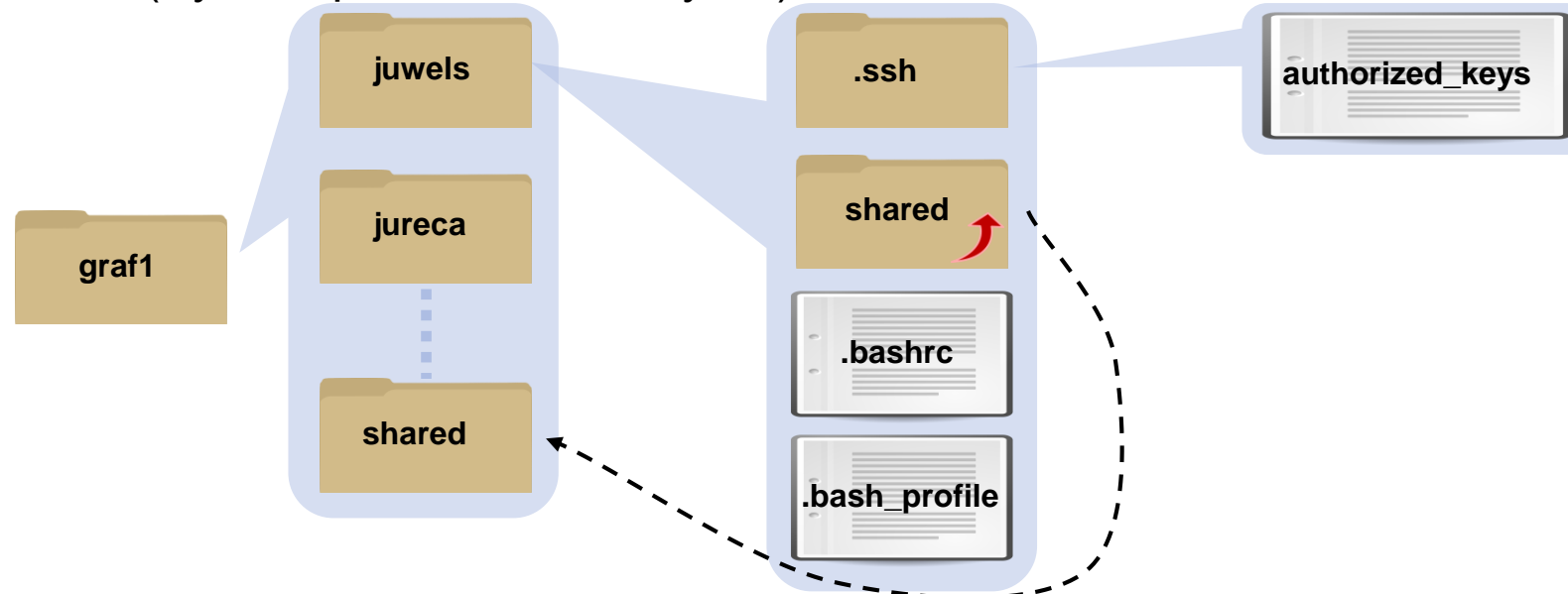
HPC USAGE MODEL @ JÜLICH

Key Characteristics

- One account per user: `surname#` (# is a consecutive number)
- Separation of user and project data - user must be joined to project to get access
- Data owner is the project
- Two project types: Compute + Data
- Project membership realized by UNIX groups
 - User's primary group: `jusers`
 - User's secondary groups: <list of project groups user is joined>
 - Files/directories created in project directory belongs to project group, realized by setGID bit: `drwxrws---`
 - !! **Owner can overrule it !!** (`chown`, `rsync`, `cp -pR`, ...)
- Project Quota accounted on directory base

USER DIRECTORY (HOME)

- Path: `/p/home/juser/<userid>`
- **Small quota per user: 20 GB + 80.000 files**
- Data is in **Backup**
- Store your personal data (System profiles, SSH Key, ...)
- For user `graf1`:



- Separate HOME on each system, e.g. on JUDAC: `$HOME = /p/home/jusers/graf1/judac`
- Link to **shared** folder

SCRATCH DIRECTORY

Compute Project


- Bandwidth optimized
 - JUST is capable of >300 GB/s
 - JURECA and JUWELS can achieve up to 200 GB/s by design
- Belongs to compute project
- Path: `/p/scratch/<group>`
\$SCRATCH = `/p/scratch/cjsc`
- Temporary files, checkpointing
- **Default Quota per group: 90 TB + 4 million files**
- **No Backup**
- **!!!Data deleted after 90 days without access!!!**
- Empty directories are deleted after 3 days

HPST: CACHE LAYER FOR \$SCRATCH

Compute Project

- High Bandwidth & Low Latency, optimal for “hard” IO pattern like small random access
- Available on JUWELS, JURECADC and JUSUF
- Compute project proposal must point out it’s requirements (depends on IO pattern)
- Path: `/p/cscratch/fs/<group>` → is a mapping to `/p/scratch/<group>`
\$CSCRATCH = `/p/cscratch/fs/cjsc`
- Access realized by additional secondary group
- **POSIX access (fuse) and native interface**
- **Quota per project: 20 TB (soft quota)**
- **Global namespace**
- **Data staging**

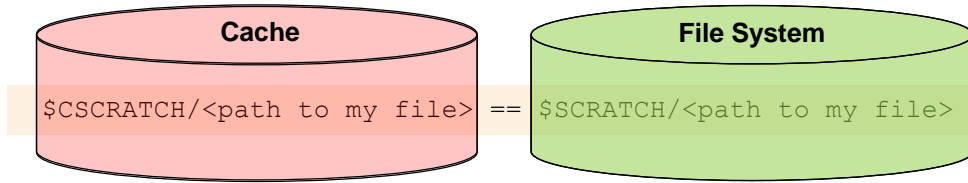
Slize	size	Bandwidth*
JUWELS	1036 TB	1024/600 GB/s
JURECADC	844 TB	800/600 GB/s
JUSUF	230 TB	240/220 GB/s
Total	2110 TB	2064/1420 GB/s



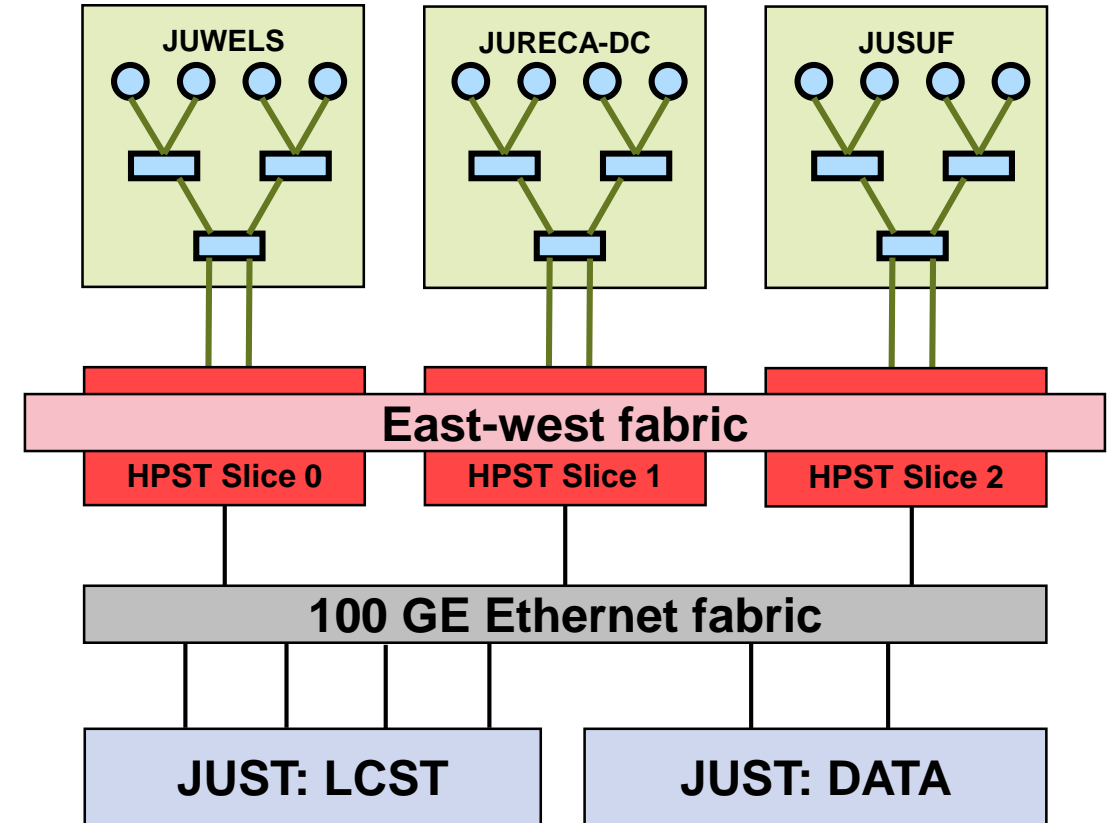
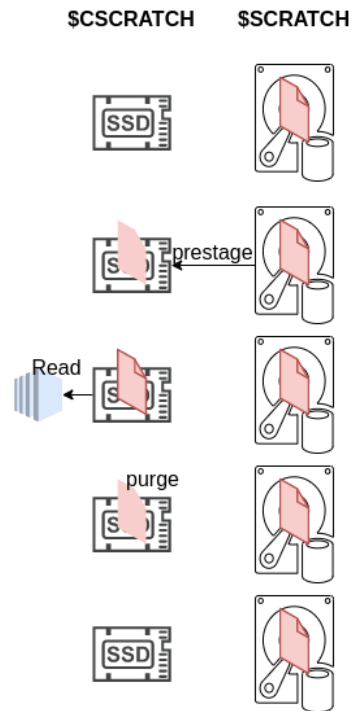
Contact
JSC application support
to get access

HIGH PERFORMANCE STORAGE TIER

Usage scenarios

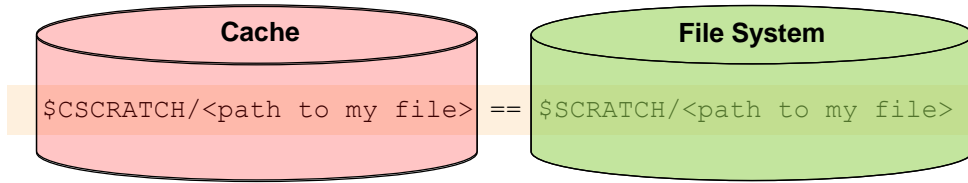


1. Read file using HPST

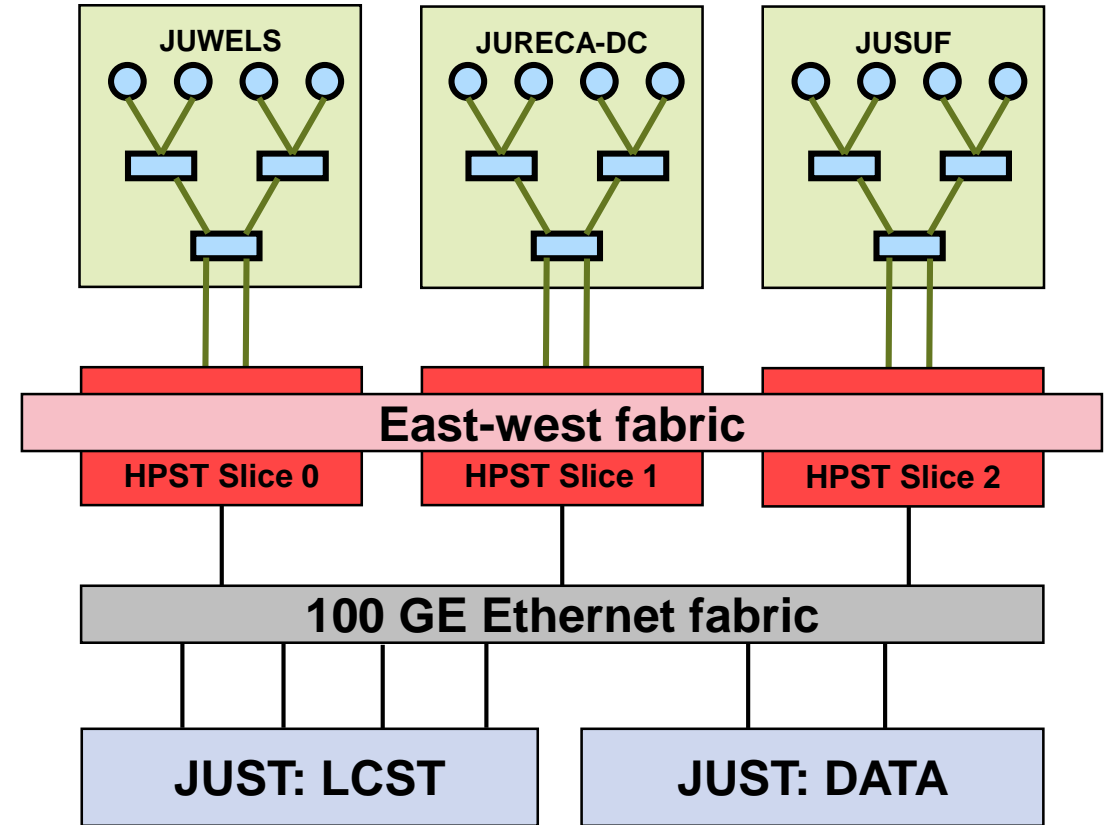
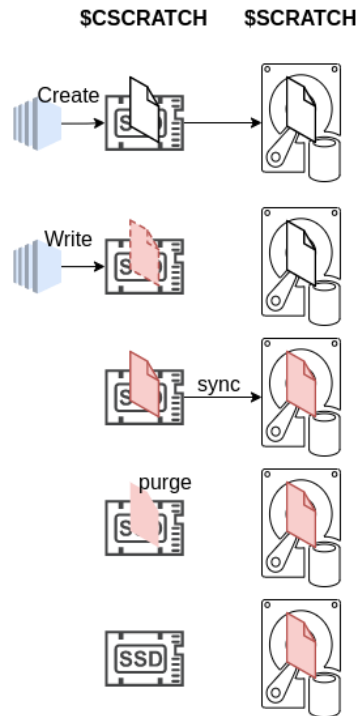


HIGH PERFORMANCE STORAGE TIER

Usage scenarios

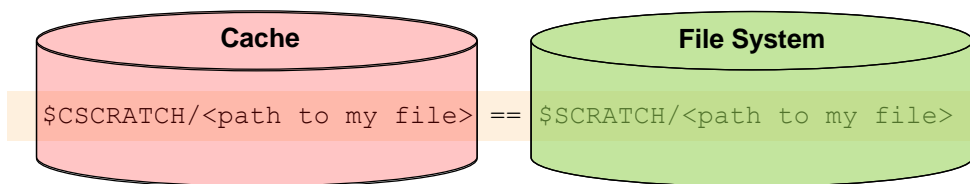


2. Write new file to HPST

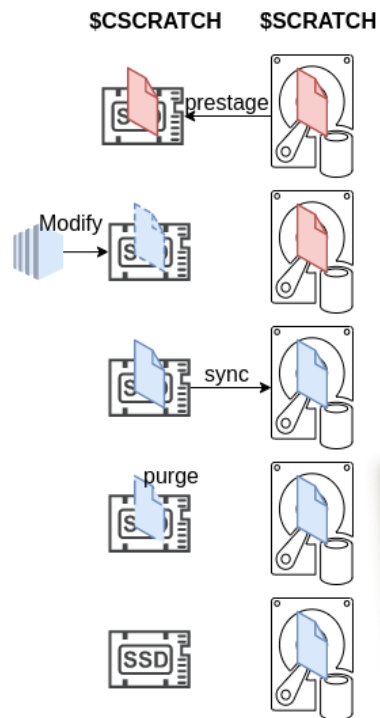


HIGH PERFORMANCE STORAGE TIER

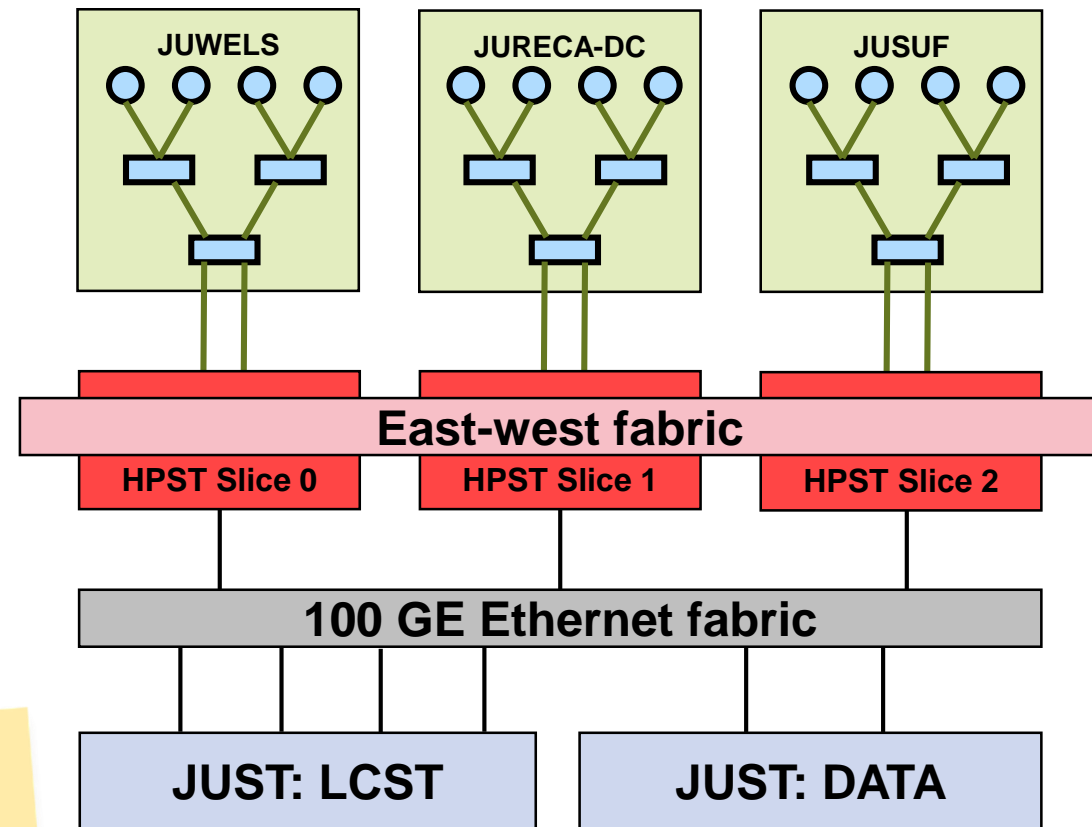
Usage scenarios



3. Read and modify file using HPST



In preparation:
IO statistics
included in job reporting



PROJECT REPOSITORY

Compute Project

- Data repository for the compute project
Path: `/p/project/<group>` e.g: `$PROJECT = /p/project/cjsc`
- Default Quota: 16 TB / 3 Mio inodes (files)
- Data is backed up
- Lifetime depends on project time span → longterm storage/archiving can be realized by a ***data project***

FASTDATA REPOSITORY

Data Project

- High Bandwidth (close to \$SCRATCH)
- Data project proposal must point out it's requirements for *FASTDATA*
- Path: `/p/fastdata/<group>` e.g: `$FASTDATA = /p/fastdata/zam`
- **Quota per group: as granted to project**
- Data is backed up

LARGEDATA REPOSITORY

Data Project

- Separate storage cluster (XCST)
- High Capacity (disk based)
- Data project proposal must point out it's requirements for *LARGEDATA*

Repository Type: File System

- Path: `/p/largedata[2]/<group>`
`$DATA = /p/largedata/zam`
- Quota per group: as granted to project
- Data is backed up
- Data sharing to Community/World by VM (on request)

Repository Type: Object Storage

- Supported protocols: **OpenStack Swift** and **S3**
- **Client environment on JUDAC available**
<https://apps.fz-juelich.de/jsc/hps/just/object-storage.html>
- **Backup: Disaster Recovery**



ARCHIVE REPOSITORY

Data Project

- Filesystem consist of 2 tiers: disks (cache) and tapes (long term)
- Path: /p/arch[2]/<group>
\$ARCHIVE = /p/arch/zam
- Archive your results
- Only available on login
- **Quota per group: as g**
- **Data are in Backup**
- Special rules:
 - Files > 7 days are migration candidate → moved to tape
 - Recall per file is expensive (1 minute mount time + 100 MB/s)
→ **use (zipped) tar balls > 1TB**
 - **Avoid renaming of directory structures** (may trigger huge recalls)

```
[zdv124@judac01:/p/arch/zam/zdv124> ls -lisah
total 320K
 407977 128K drwx-----  2 zdv124 zam  64K May 18 10:01 .
 407555 128K drwxr-xr-x 316 root  sys  64K May 24 15:00 ..
18062260  64K -rw-r--r--  1 zdv124 zam   5 Sep  2 2011 datum.txt
12920848   0 -rw-r--r--  1 zdv124 zam  12G Jun  3 2015 Vervet_s0050_tiff.tgz
```


FILESYSTEMS - SUMMARY

File System	Shell Variable	Description	Project Type	Characteristics
home	\$HOME	Users HOME File Systems		User Quota: 10GB/40.000 Files Files in Backup
project	\$PROJECT	Compute Project File System	Compute	Group Quota: 16TB/3Mio Files Files in Backup
scratch	\$SCRATCH	Compute Project Scratch File System	Compute	Group Quota: 90TB/4Mio Files Files deleted after 90 days
cscratch	\$CSCRATCH	Fast cache for scratch file system	Compute	Group Quota: 20TB Data staging required
fastdata	\$FASTDATA	High Bandwidth and large Capacity File System	Data	Group Quota: depends Files in Backup
largedata	\$DATA	Large Capacity (Disk based)	Data	Group Quota: depends Files in Backup
arch arch2	\$ARCHIVE	Archive File System (Tape)	Data	Group Quota: depends Files in Backup Migration to tape

JUDAC – JUELICH DATA ACCESS

Data access and transfer cluster

- All HPC user can login on judac: `ssh <userid>@judac.fz-juelich.de`
- Independent from HPC systems (e.g. in maintenance)
- Access to Jülich Object Storage: openStackClient
- Purpose: data transfer in & out the HPC filesystems
 - **scp, rsync**
 - Standard ssh setup can be used (connection must be initiated from external)
 - Use **screen** or **tmux** for long running data transfer
 - **jutil**
 - **Grid Tools**
 - **UNICORE FTP** (next slide)
- For more information go to [→ JUDAC Web Page](#)

DATA TRANSFER TO/FROM JÜLICH USING UNICORE

- Install client from [sourceforge](https://sourceforge.net/projects/unicore/) on your system (1x)
- Create client SSH key (1x)

```
[user@home ~]$ mkdir -p $HOME/.uftp  
[user@home ~]$ ssh-keygen -a 100 -t ed25519 -f $HOME/.uftp/id_uftp
```

- Prepare client environment (1x)

```
[user@home ~]$ export UFTP_USER=<your_remote_user_id>  
[user@home ~]$ export UFTP_AUTH_URL=https://uftp.fz-juelich.de:9112/UFTP_Auth/rest/auth/JUDAC:  
[user@home ~]$ export UFTP_KEY=$HOME/.uftp/id_uftp
```

- Copy public key to JUDAC (UNICORE server) (1x)

```
[user@home ~]$ ssh $UFTP_USER@judac.fz-juelich.de 'mkdir -p $HOME/.uftp'  
[user@home ~]$ scp $HOME/.uftp/id_uftp.pub $UFTP_USER@judac.fz-juelich.de:~/.uftp/authorized_keys
```

- Upload/download data

```
[user@home ~]$ uftp cp --user $UFTP_USER --identity $UFTP_KEY "file.tar" $UFTP_AUTH_URL/p/home/jusers/$UFTP_USER/jureca  
[user@home ~]$ uftp cp --user $UFTP_USER --identity $UFTP_KEY $UFTP_AUTH_URL/p/home/jusers/$UFTP_USER/juwels/file.tar .
```

DATA SHARING INSIDE JÜLICH HPC

Different use cases and solutions for sharing data between users:

1. Use compute project repository (\$PROJECT)

Any user can be joined to project without access to project's compute resources

2. Use data project

Members of different compute projects can join a common data project

3. Single files

All users can access common directory „**\$SCRATCH**/. . /share“. Remember the automatic file deletion after 90 days!

4. Software project

Special data project which is mounted on compute nodes

HINTS & TIPS

- Create **checksum** on data files
- Restore files from backup: **adsmback**
 - **available only on JUDAC**
 - Calls IBM TSM Backup/Restore GUI
 - Hard to guaranty daily backup → snapshots available, eg: `$PROJECT/.../.snapshots/daily-YYYYMMDD/<project>/`
- Quota usage information: **jutil**
 - Project group quota info:
`jutil project dataquota -p <project>`
 - User quota info: `jutil user dataquota -u <user>`
- SSH/SCP usage
 - Multiple external (scripted) access can be classified as an attack
→ Firewall will block external IP
- Take care of your files
 - No special characters in filenames (newline, tab, escape, ...)

AND FINALLY

- **Filesystem status:** <https://status.jsc.fz-juelich.de/>
- **JUST web pages (e.g. FAQ)** <https://go.fzj.de/JUST>
- **JUDAC web pages (e.g. data transfer, object store)** <https://go.fzj.de/JUDAC>
- Jülich HPC Usage Model: <http://www.fz-juelich.de/ias/jsc/usage-model>
- JuDoor – manage accounts/projects, overview of resources, ... <http://www.fz-juelich.de/ias/jsc/judoor>
- For any problem (accessing files, access rights, restore, quota, data transfer, ...) contact JSC application support (sc@fz-juelich.de)
- If you want to optimize your application IO:
 - What is the access pattern?
 - Use IO libraries/formats HDF5, MPI-IO, SIONLIB, ...
 - Contact JSC application support (sc@fz-juelich.de)



Questions?