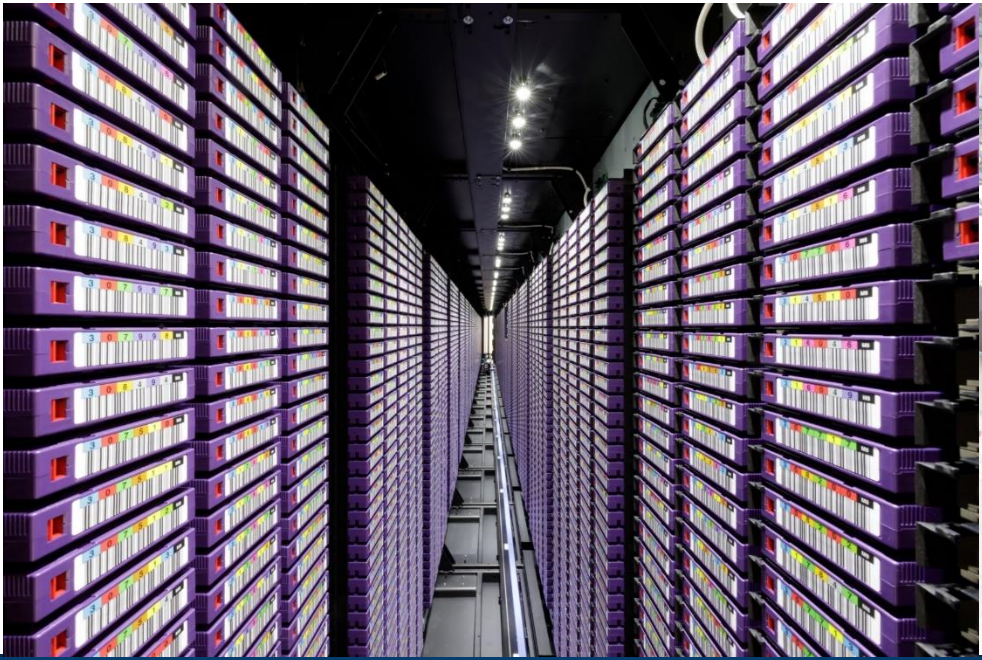


FROM XCST TO HPST

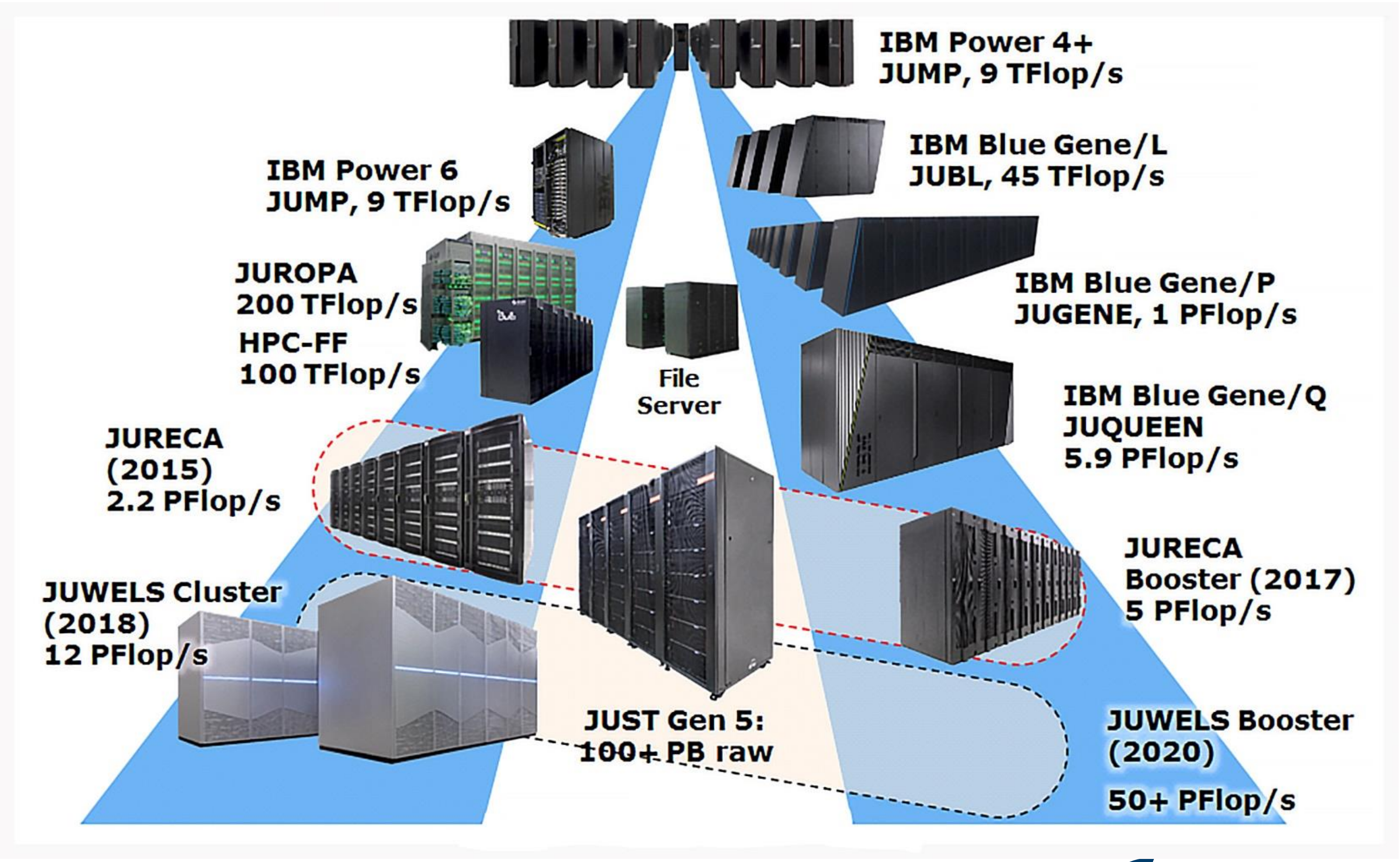
JUELICH STORAGE CLUSTER JUST

25. MARCH 2021 I STEPHAN GRAF (JSC)

Mitglied der Helmholtz-Gemeinschaft

JÜLICH Forschungszentrum

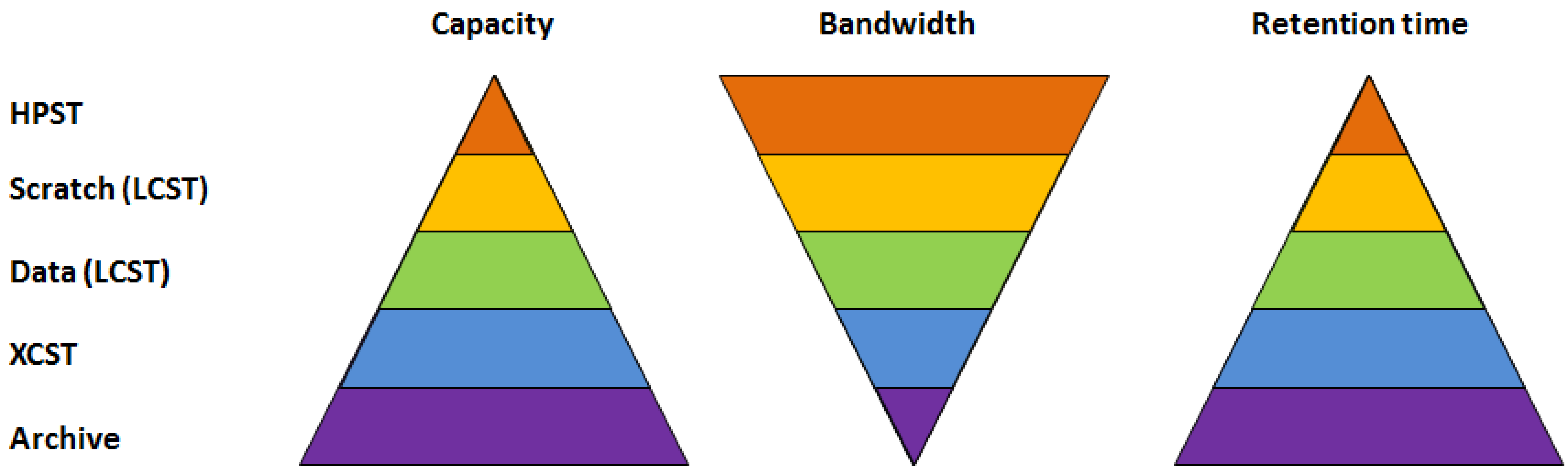
DUAL ARCHITECTURE STRATEGY



IO USE CASES

Scenario	Example	Access Requirement	Lifetime	Data Reliability
Small random IO	Deep Learning - AI	Low Latency (+ High Bandwidth)	Up to a couple of weeks	Can be recreated easily
Streaming IO	Checkpointing	High Bandwidth	Up to a couple of weeks	Can be recreated easily
Store data for the project	Input/Output data	Moderate Bandwidth	Up to 3 years?	Backup
Share data	- Colleagues/partners - Other project - community	Moderate latency Moderate bandwidth Special protocols	Multiple years	Backup
Archive data	Keep data after project has ended for review	Write once – read never	10 years +	Backup

TIERED STORAGE OFFERING

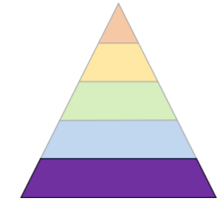


- High Performance Storage Tier (HPST):
  NVMe based Storage (low latency+high bandwidth) → pilot phase
- Large Capacity Storage Tier (LCST): disk based, bandwidth optimized
- Extended Capacity Storage Tier (XCST): disk based, capacity optimized
- Archive: Tape storage + disk based cache

IBM SPECTRUM SCALE (GPFS)

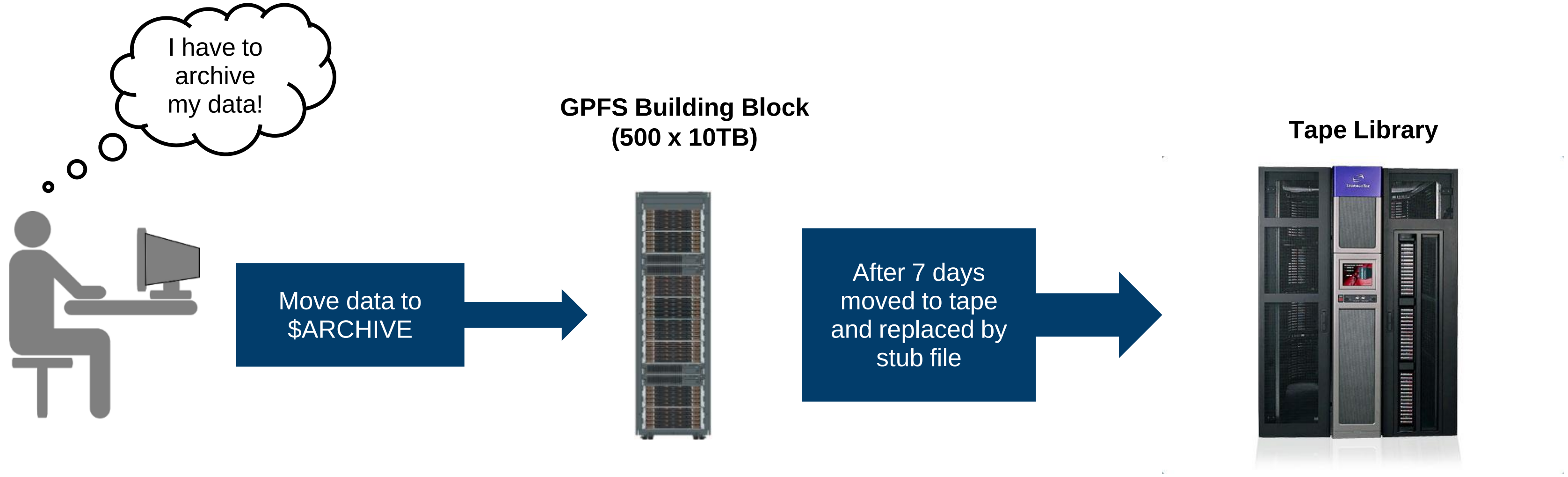
Cluster File System

- General Parallel File System
- Optimized for HPC parallel file access
- POSIX compliant
- Client-Server architecture
- Special features included
  - RDMA
  - Hierachical Storage Managemant (**HSM**, Spectrum Protect)
  - Optimized backup (mmBackup, Spectrum Protect)
  - Cluster Export Services – Protocol support (**NFS**, SMB, **S3/Swift**, HDFS)
  - AFM (Active File Management)
  - Encryption …



ARCHIVE

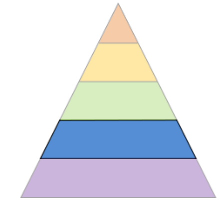
GPFS combined with HSM




```
[zdv124@judac01:/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 date.txt
12920848     0 -rw-r--r--   1 zdv124 zam   12G Jun  3  2015 Vervet_s0050_tiff.tgz
```

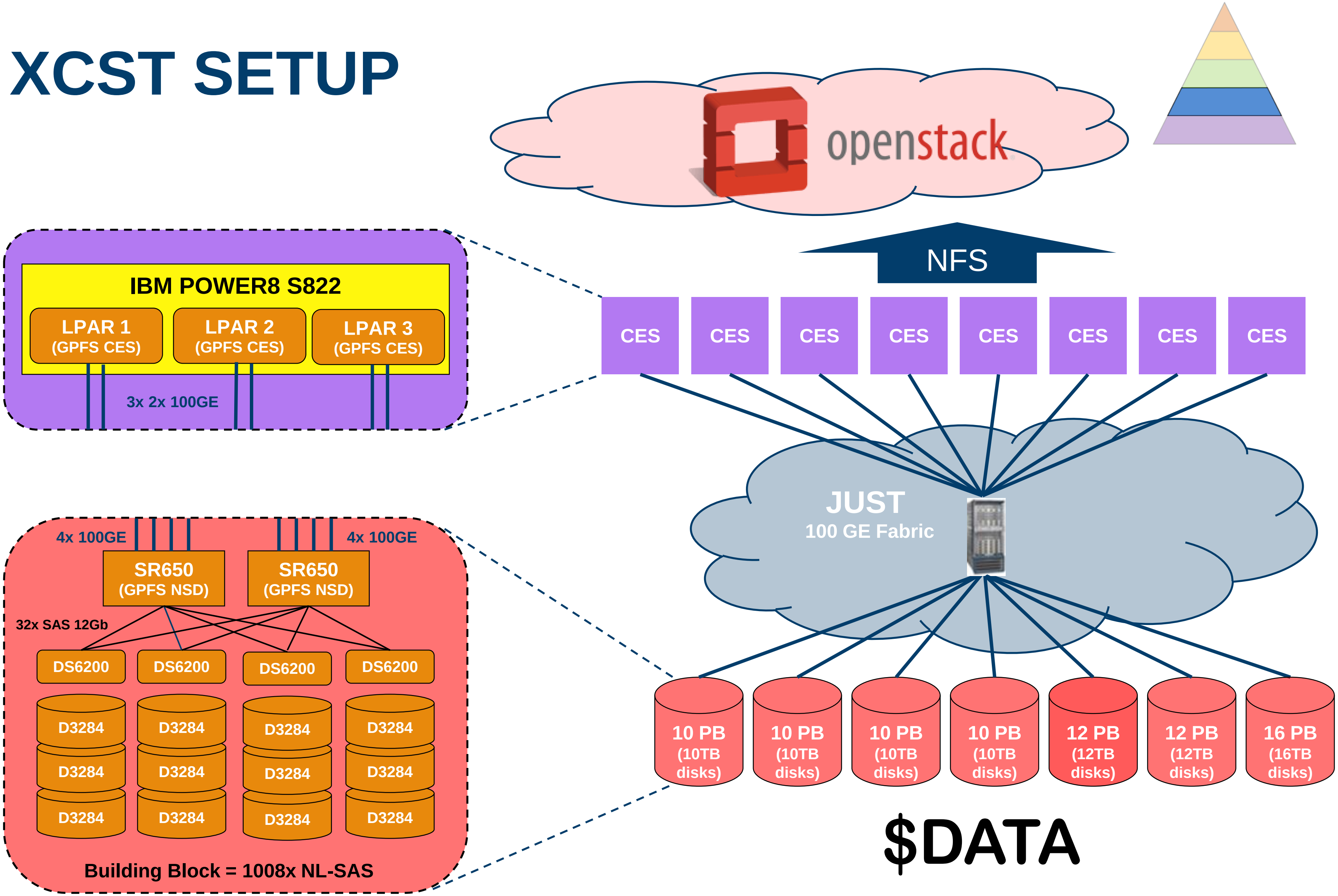
XCST

Extendet Capacity Storage Tier



- Goal: „online“ data (disk), maximum capacity
- Campaign storage: storing and sharing your data
- Available on
  - HPC login nodes (not on the compute nodes)
  - JUDAC system (JUelich Data Access)
  - HDF Cloud (openstack)
- Phased installation: new hardware every year, but different disks (10/12/16TB) this year final phase 5 will be installed
- Two different repository types
  - File system access
  - Object store (in preparation)
- One global namespace

XCST SETUP

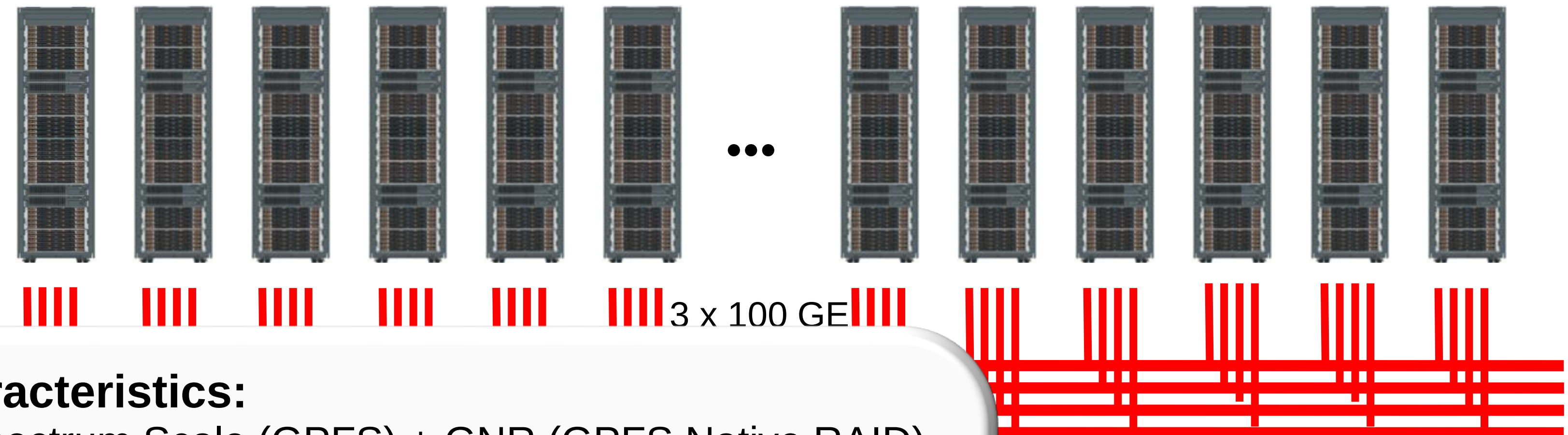


openstack
NFS
IBM POWER8 S822
LPAR 1 (GPFS CES)
LPAR 2 (GPFS CES)
LPAR 3 (GPFS CES)
3x 2x 100GE
CES CES CES CES CES CES CES CES
JUST
100 GE Fabric
4x 100GE
4x 100GE
SR650 (GPFS NSD)
SR650 (GPFS NSD)
32x SAS 12Gb
DS6200 DS6200 DS6200 DS6200
D3284
Building Block = 1008x NL-SAS
10 PB (10TB disks)
10 PB (10TB disks)
10 PB (10TB disks)
10 PB (10TB disks)
12 PB (12TB disks)
12 PB (12TB disks)
16 PB (16TB disks)
$DATA

LCST

Large Capacity Storage Tier

21 x GPFS Building Blocks → 42 x NSD Server, 84 x Enclosure → +7.000 x 10TB disks

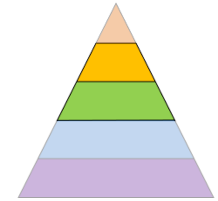


Characteristics:

- Spectrum Scale (GPFS) + GNR (GPFS Native RAID)
  - Declustered RAID technology
  - End-to-End data integrity
- Spectrum Protect (TSM) for Backup
- Hardware:
  - x86 based server + RHEL 7
  - IBM Power 8 + AIX 7.2
  - 100GE network fabric
- 70 PB gross capacity
- Bandwidth: **400 GB/s**

2 Cluster Export Server (NFS)

LCST

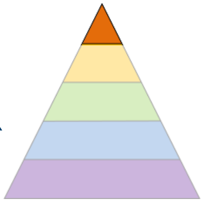


Multiple file systems in one cluster

- **`/p/scratch`**
  → temporary file system for compute jobs, high bandwidth
  - No backup, data are deleted after 90 days
- **`/p/fastdata`**
  → high bandwidth file system for persistent data to compute (e.g. training data)
  - Data are in backup
- 3 building blocks for the other file systems
  - **`/p/project`** → compute project repository
  - **`/p/software`** → data repository for sharing software between projects
  - …

The truth
Both uses the same disks (18 building blocks), so performance is the same

HIGH PERFORMANCE STORAGE TIER



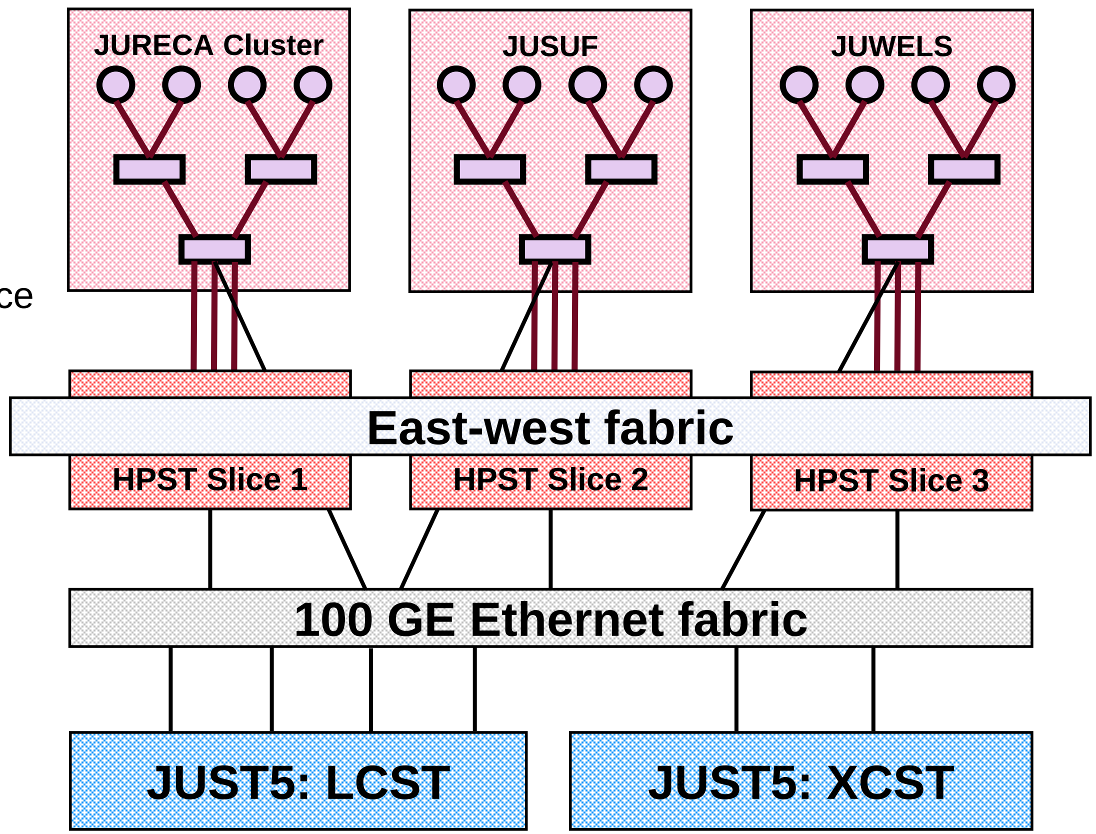
NVMe based cache to underlying file system

- Goal: High Bandwidth + low Latency for compute jobs
- Solution: NVMe Storage cluster + Infinite Memory Engine (IME) from DDN
- Design decision:
  - Each cluster has it’s own “slice” of the HPST, but
  - One global namespace (each cluster has access to data on “foreign slice”)
  - Direct integrated in cluster fabric (IB)
- Client access by mount point (FUSE) or IME native interface
- Cache of the $SCRATCH file system
- CLI + API for data migration & data staging (e.g. flush, sync, pre-stage, pin)
- Path: **`/p/ime-scratch/fs/<group>`** → is a mapping to **`/p/scratch/<group>`**
- Access is managed by special HPST group ID (GID)

HIGH PERFORMANCE STORAGE TIER

Infinite Memory Engine (IME) from DDN

- 110 Server IME-140, each
  - 2 x HDR 100 (North connection)
  - 1 x HDR 100 (East-West connection)
  - 1 x 100 GE (South connection)
  - 10 x NVMe drives (2TB) for HPST data
  - 1 x internal NVMe IME commit log device



Slice	Size	Bandwidth*
JUWELS	1036 TB	1024/600 GB/s
JURECA	844 TB	800/600 GB/s
JUSUF	230 TB	240/220 GB/s
Total	**2110 TB**	**2064/1420 GB/s**

*committed values

TIERED STORAGE OFFERING

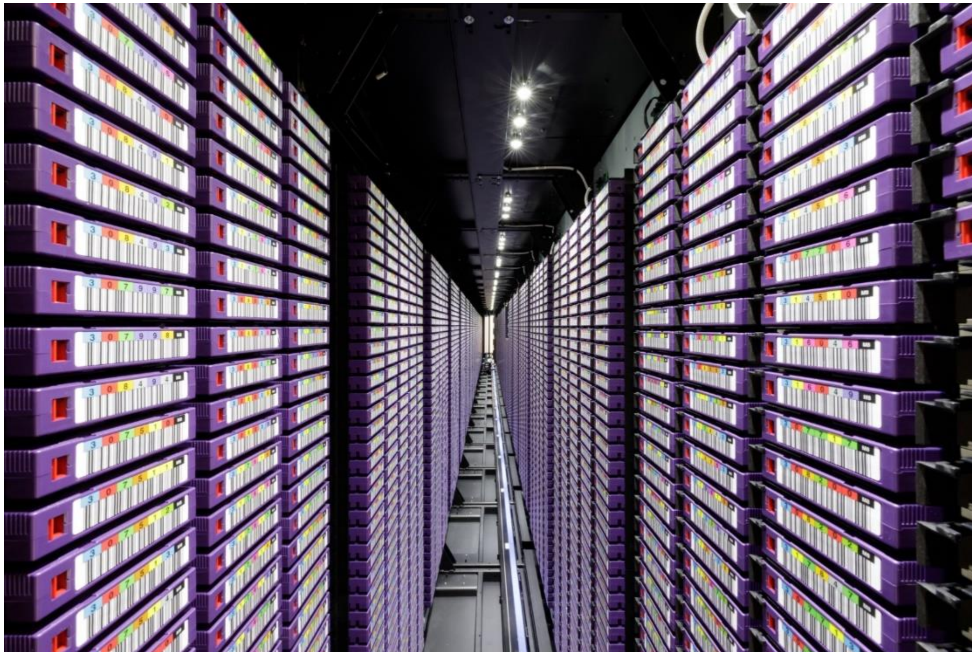
Summary

Layer	ENV Name	characteristics	IO type	Write	Read
HPST	**-**	- NVMe - Cache - Compute with hard IO pattern	Single client Multiple client Multiple client (small IO)	3,3 GB/s 600 GB/s 550 GB/s	7,4 GB/s 730 GB/s 320 GB/s
Scratch (LCST)	**$SCRATCH**	- Temporary files of compute jobs - streaming	Single client Multiple client	8GB/s >300 GB/s	9 GB/S 400 GB/s
Data (LCST)	**$FASTDATA**	- Persistent files for compute jobs - streaming	Single client Multiple client	8GB/s >300 GB/s	9 GB/S 400 GB/s
XCST	**$DATA**	- Persistent files - Share data with external users (Web services)	Single client Multiple client	8 GB/s >50 GB/s	9 GB/s > 70 GB/s
Archive	**$ARCHIVE**	- Archive data - Read data seldom	Single client	~8 GB/s	~200 MB/s

DATA SHARING INSIDE JÜLICH HPC

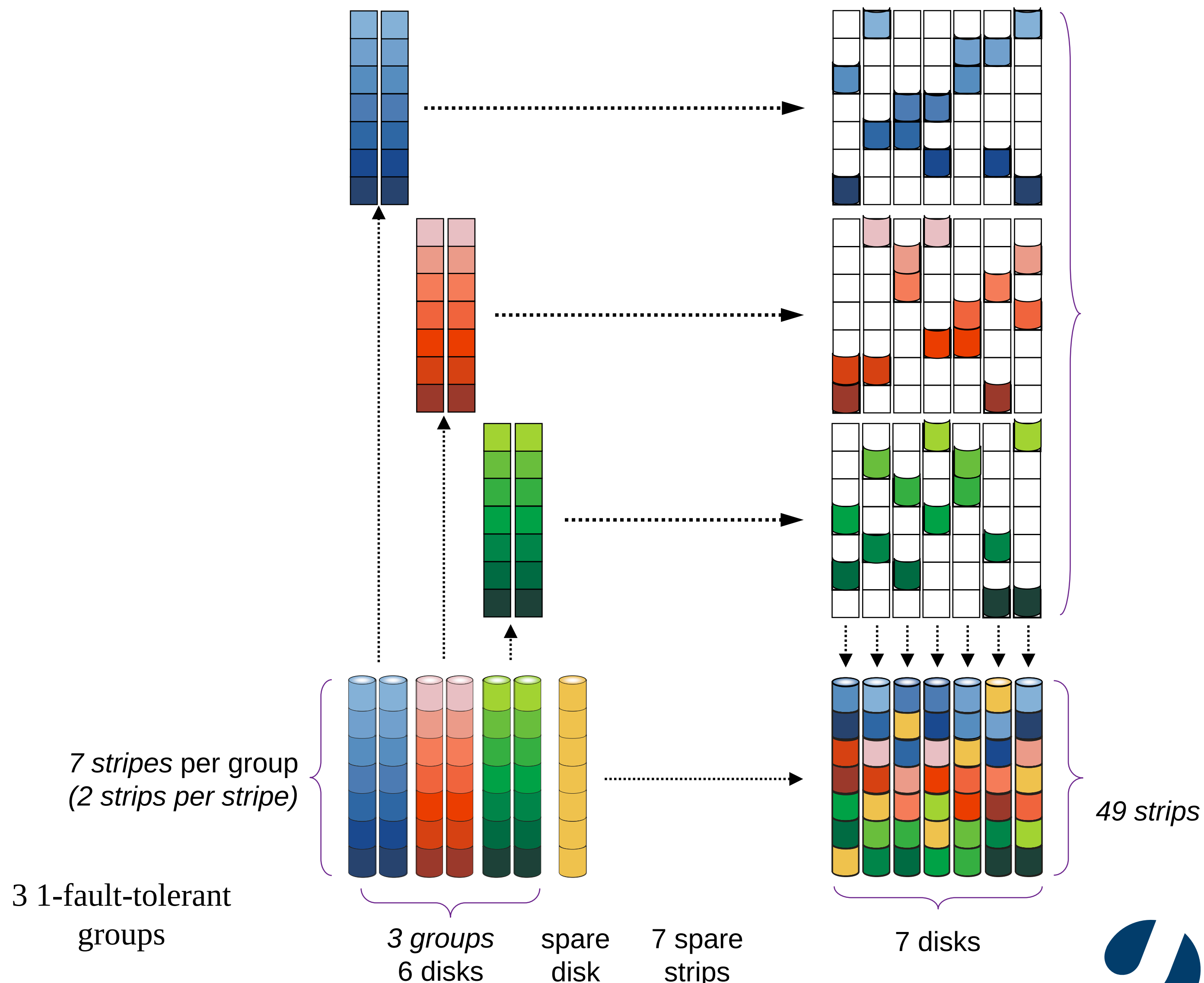
Different use cases and solutions for sharing data between users:

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

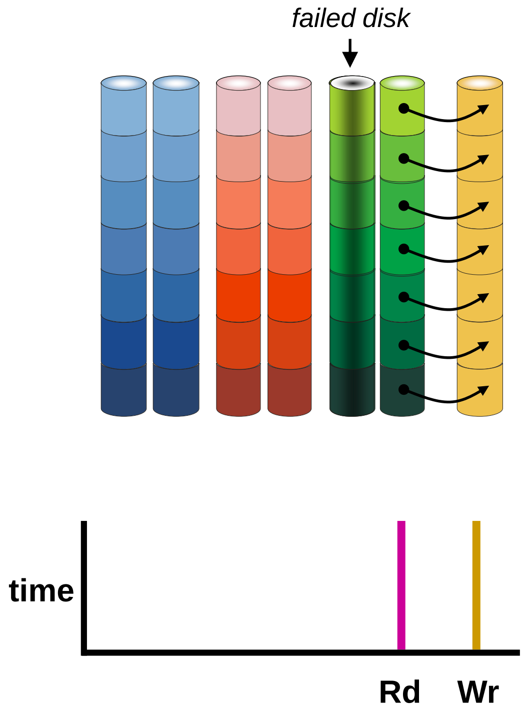


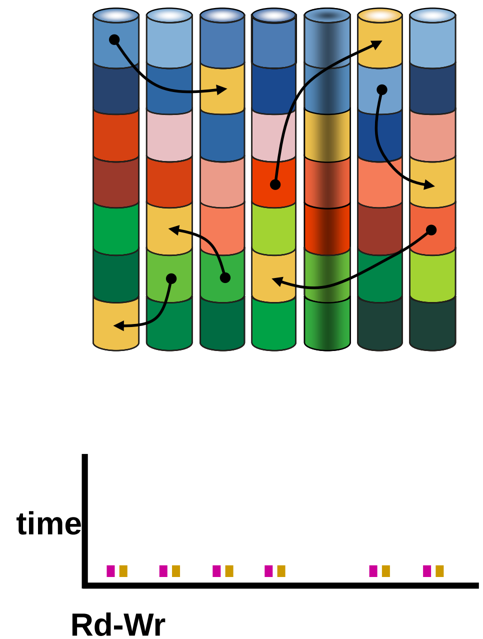
THANK YOU FOR YOUR ATTENTION

DECLUSTERED RAID



DECLUSTERED RAID REBUILD EXAMPLE





Disk failure causes disk rebuild
➔ Volume degraded for a long time
➔ performance impact for file system

Disk failure causes strips rebuild
➔ all disc involved
➔ Volume degraded for a short time
➔ minimized performance impact