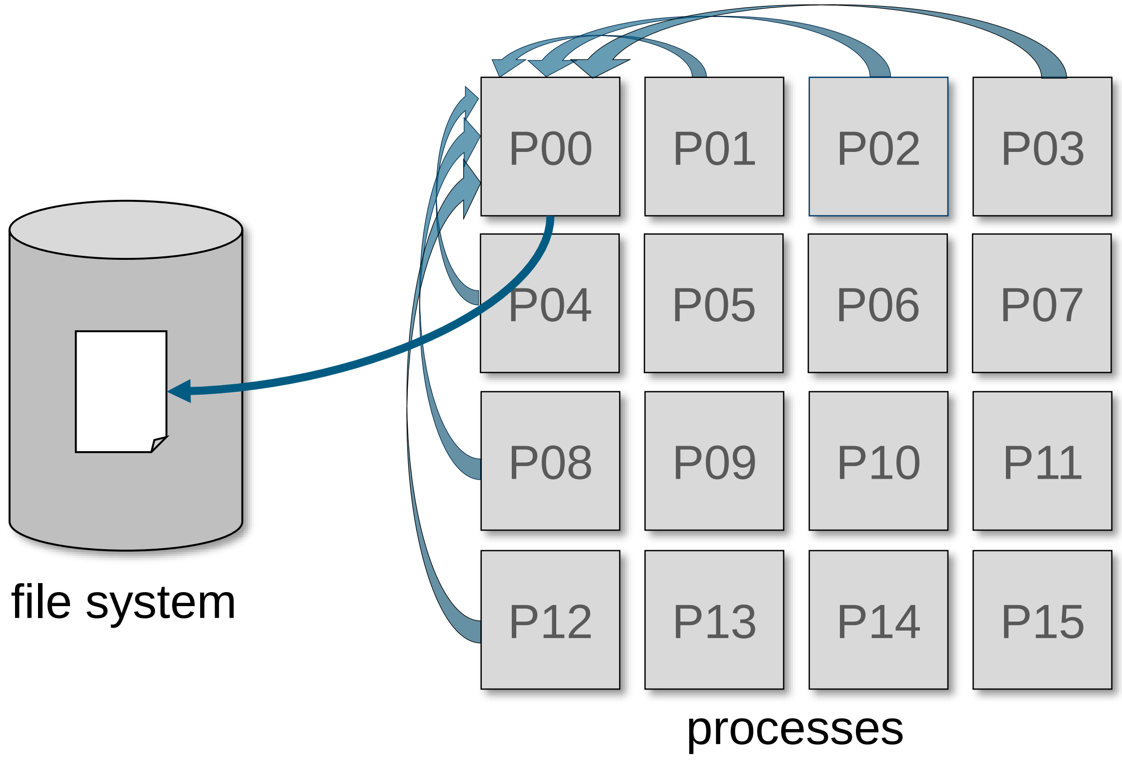
PARALLEL I/O AND PORTABLE DATA FORMATS

INTRODUCTION AND PARALLEL I/O STRATEGIES

22.02.2022 I SEBASTIAN LÜHRS (S.LUEHRS@FZ-JUELICH.DE)

Parallel I/O Strategies

One process performs I/O



Parallel I/O Strategies

One process performs I/O

+ Simple to implement

- I/O bandwidth is limited to the rate of this single process
- Additional communication might be necessary
- Other processes may idle and waste computing resources during I/O time

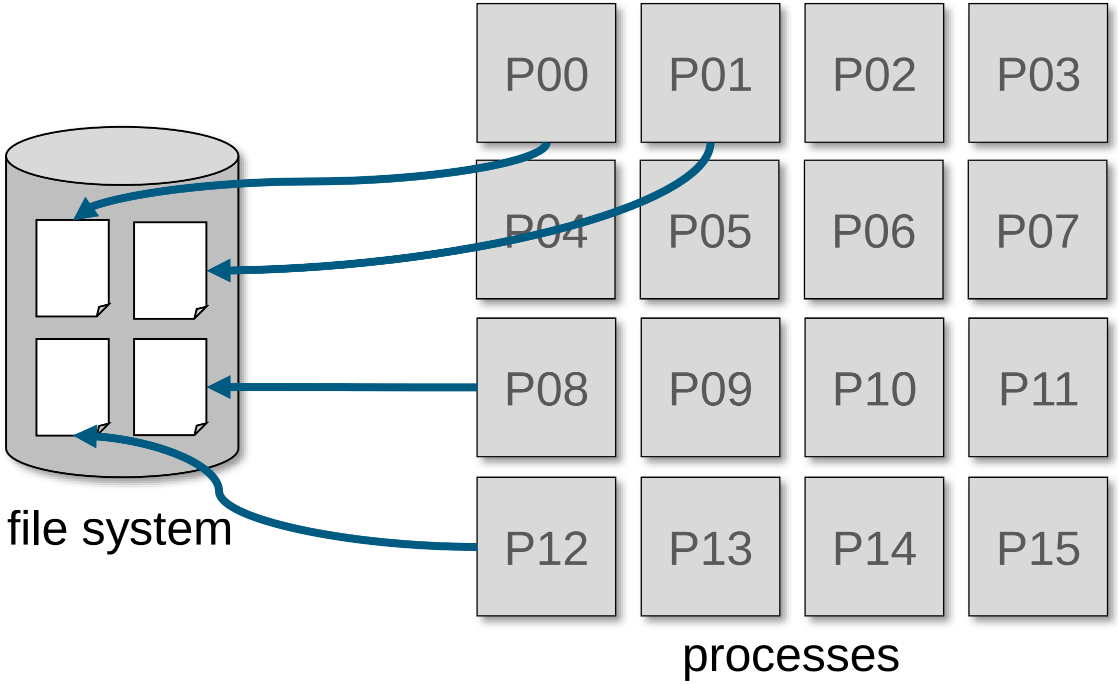
Parallel I/O Pitfalls

Frequent flushing on small blocks

- Modern file systems in HPC have **large file system blocks** (e.g. 16MB)
- A flush on a file handle forces the file system to perform all pending write operations
- If application writes in small data blocks, the same file system block it has to be **read and written multiple times**
- Performance degradation due to the inability to combine several write calls

Parallel I/O Strategies

Task-local files



Parallel I/O Strategies

Task-local files

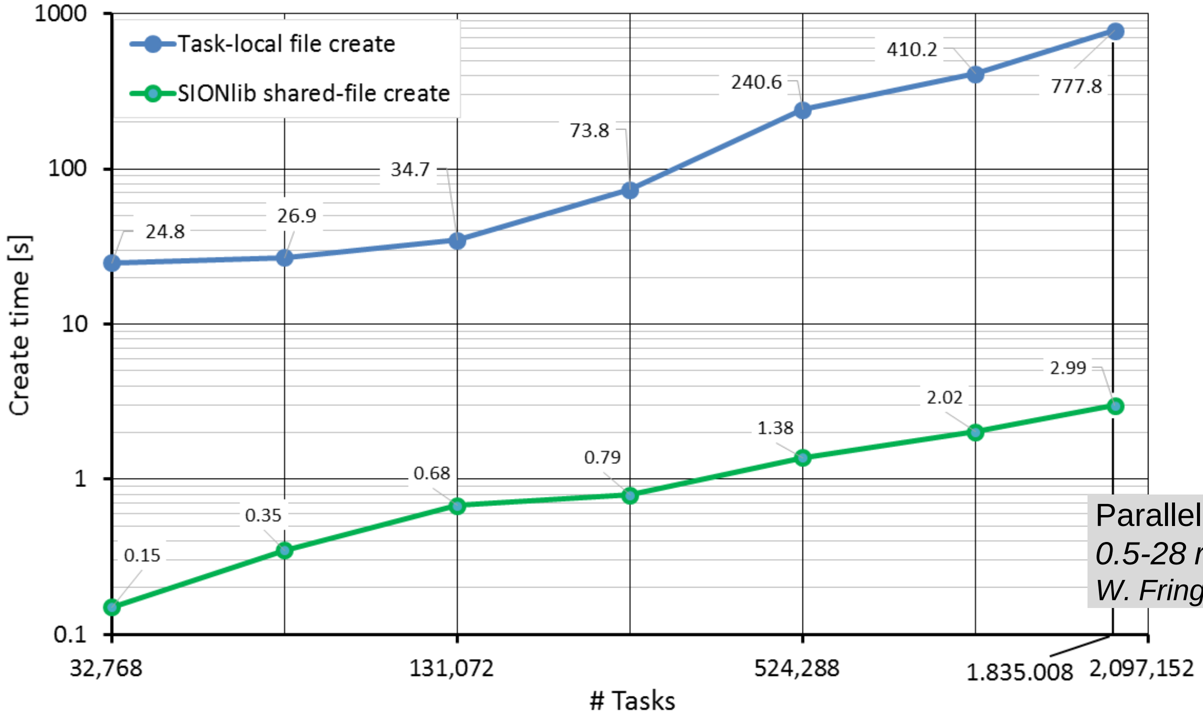
+ Simple to implement
+ No coordination between processes needed
+ No false sharing of file system blocks

- Number of files quickly becomes unmanageable
- Files often need to be merged to create a canonical dataset
- File system might serialize meta data modification

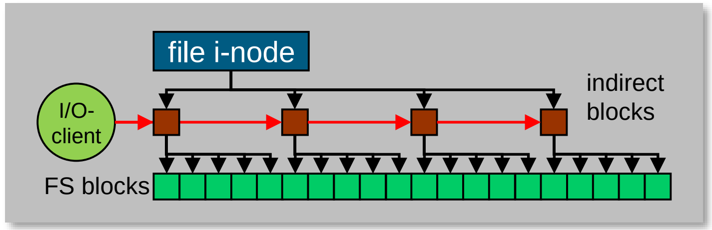
Parallel I/O Pitfalls

Serialization of meta data modification

Example: Creating files in parallel in the same directory



Parallel file creation on *JUQUEEN*
0.5-28 racks, 64 tasks/node
W. Frings

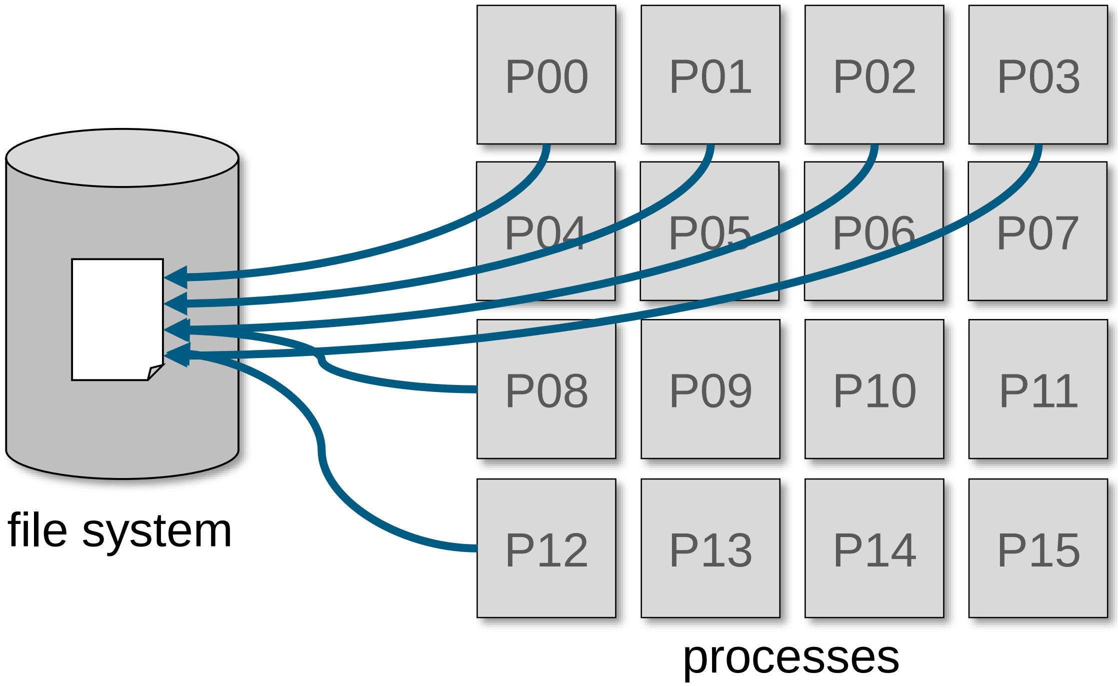


- Meta-data wall on file level
  - File changes by multiple processes can cause serialization
  - File meta-data management
  - Locking

The creation of 2.097.152 files costs 113.595 core hours on JUQUEEN!

Parallel I/O Strategies

Shared files



Parallel I/O Strategies

Shared files

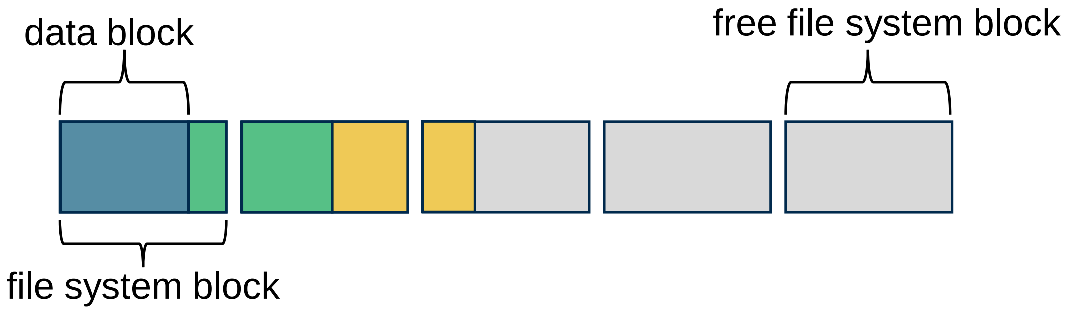
+ Number of files is independent of number of processes
+ File can be in canonical representation (no post-processing)

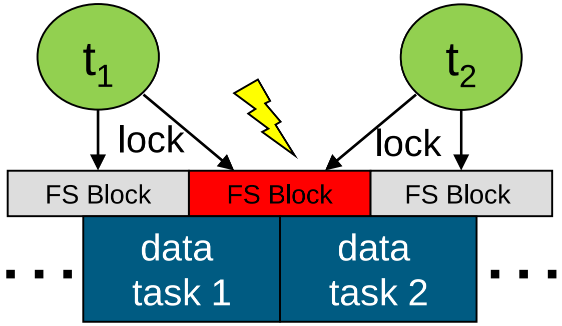
- Uncoordinated client requests might induce time penalties
- File layout may induce false sharing of file system blocks

Parallel I/O Pitfalls

False sharing of file system blocks

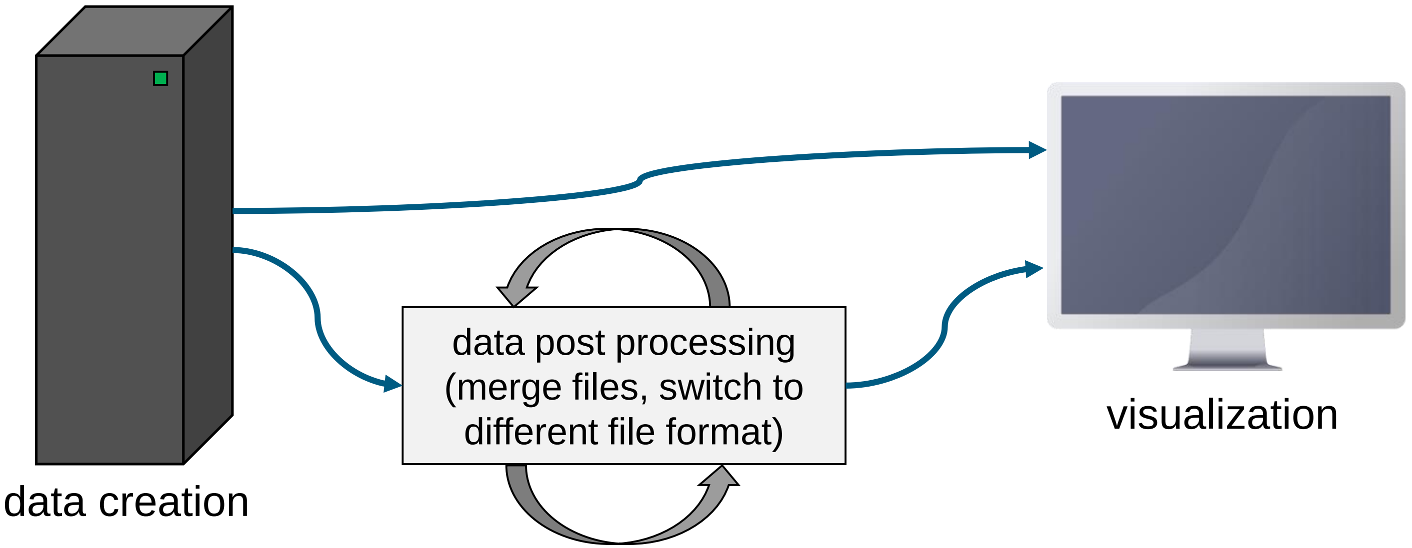
- Data blocks of individual processes do not fill up a complete file system block
- Several processes share a file system block
- Exclusive access (e.g. write) must be serialized
- The more processes have to synchronize the more waiting time will propagate





I/O Workflow

- Post processing can be very time-consuming (> data creation)
  - Widely used portable data formats avoid post processing
- Data transportation time can be long:
  - Use shared file system for file access, avoid raw data transport
  - Avoid renaming/moving of big files (can block backup)



Parallel I/O Pitfalls

Portability

- Endianness (byte order) of binary data
- Conversion of files might be necessary and expensive

2,712,847,316

=

10100001 10110010 11000011 11010100

Address	Little Endian	Big Endian
1000	11010100	10100001
1001	11000011	10110010
1002	10110010	11000011
1003	10100001	11010100

Parallel I/O Pitfalls

Portability

- Memory order depends on programming language
- Transpose of array might be necessary when using different programming languages in the same workflow
- Solution: Choosing a portable data format (HDF5, NetCDF)

1	2	3
4	5	6
7	8	9

→

Address	row-major order (e.g. C/C++)	column-major order (e.g. Fortran)
1000	1	1
1001	2	4
1002	3	7
1003	4	2
1004	5	5
…	…	…

Storage Tiers

Different storage tiers with different optimization targets

Data staging at JSC

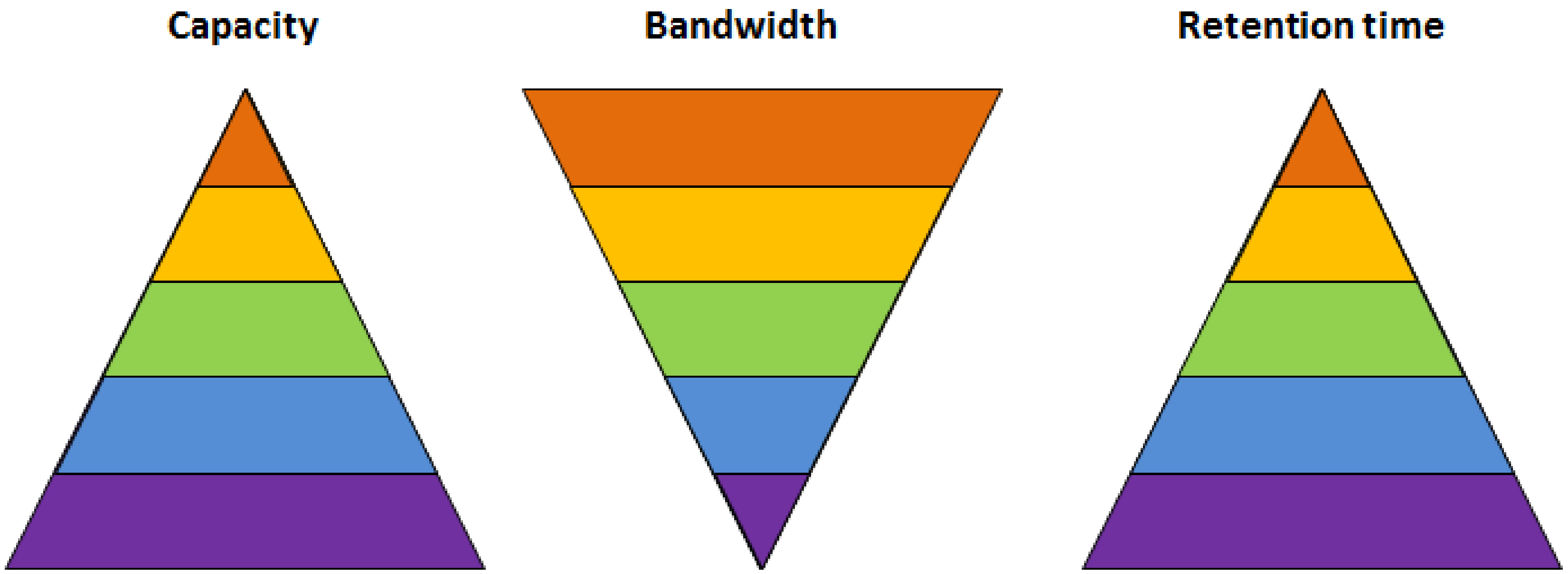
HPST

$SCRATCH

$FASTDATA

$DATA

$ARCHIVE

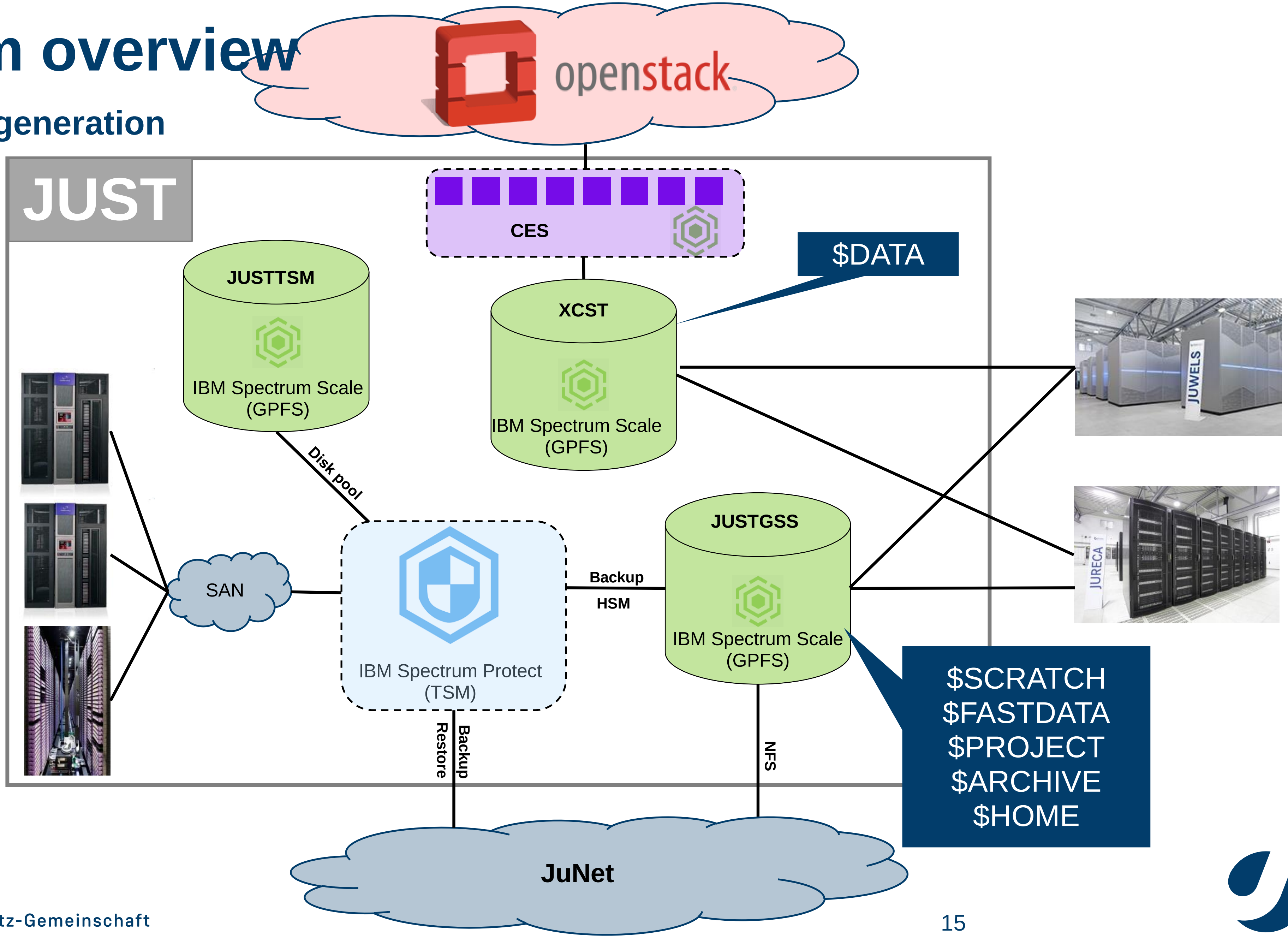


Tape Library

JUST 5

System overview

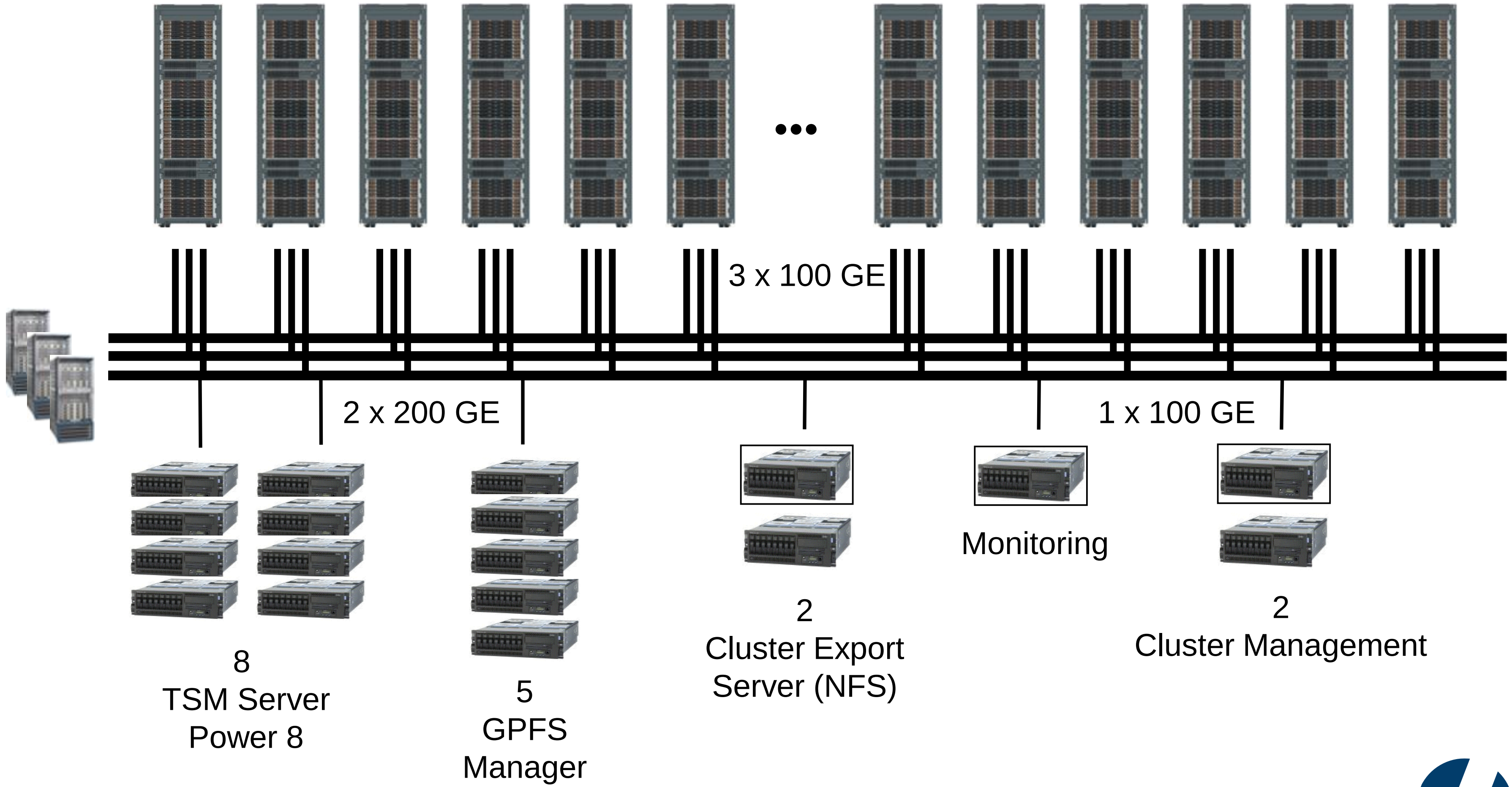
JUST – 5th generation



System overview

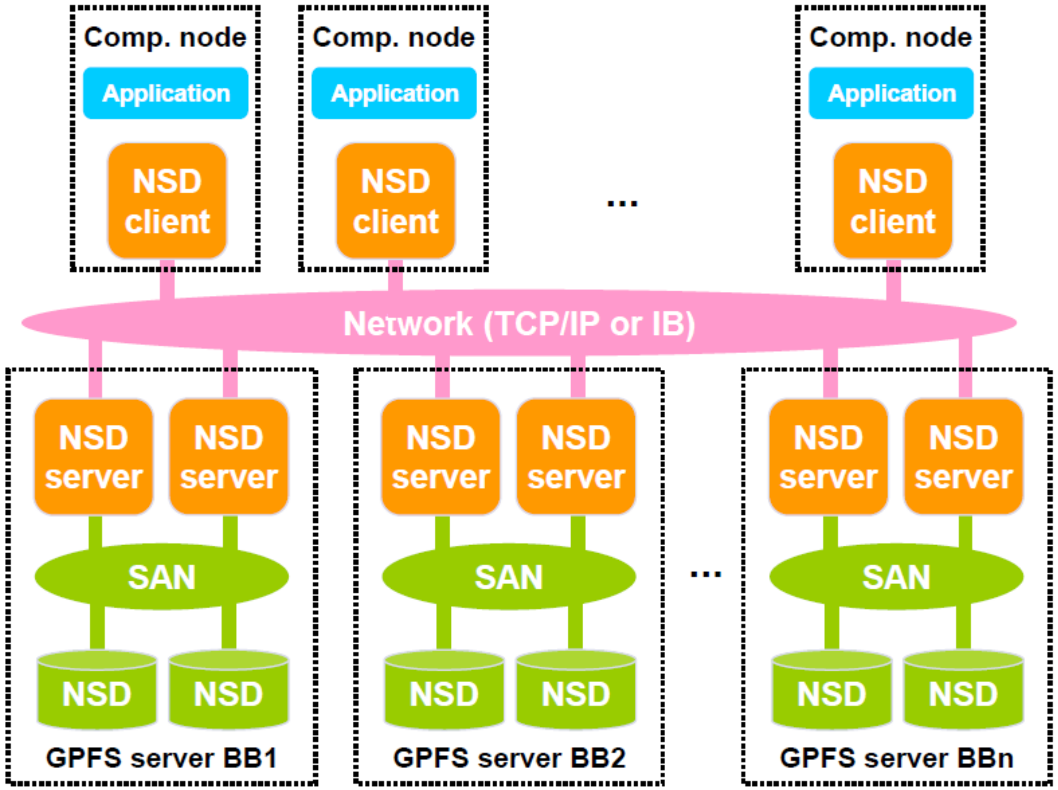
JUST – 5th generation

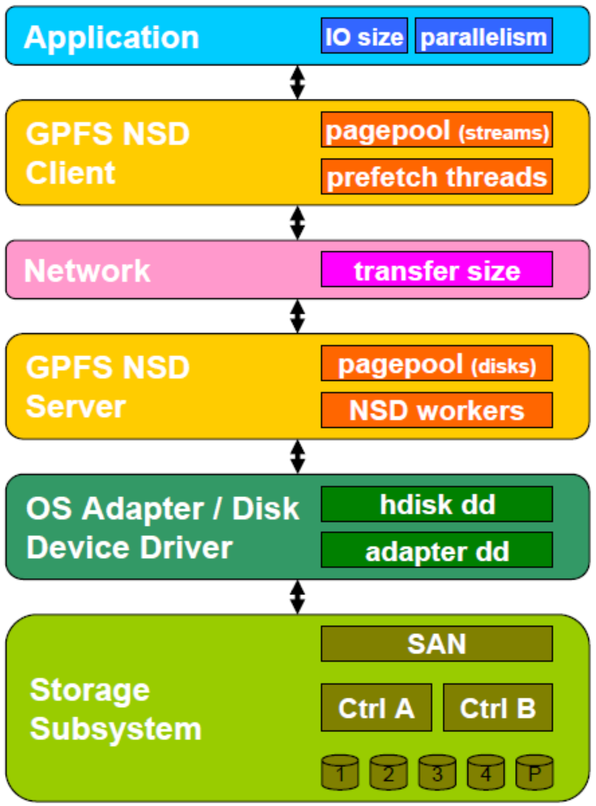
21 x DSS240 + 1 x DSS260 → 44 x NSD Server, 90 x Enclosure → +7.500 10TB disks



System overview

File I/O to GPFS





System overview

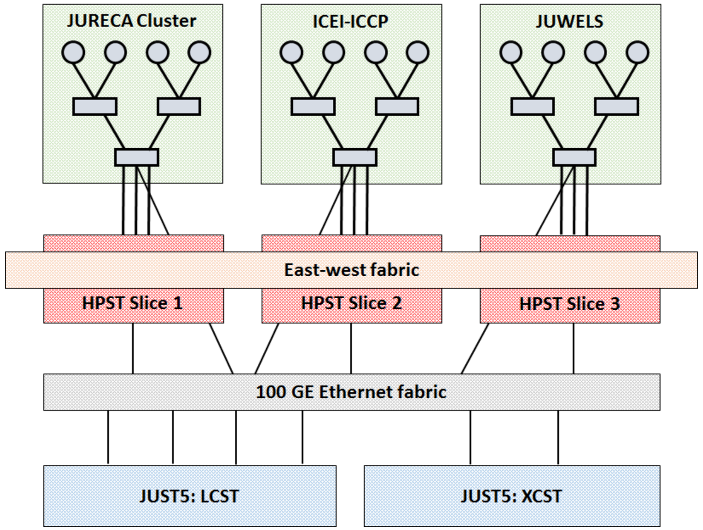
HPST

- Low latency - high bandwidth storage layer
- Funded partly by the ICEI project and being part of the Fenix Research Infrastructure

- Based on DataDirect Network (DDN) storage appliances
- Consists of a total of **110 servers** with an accumulated capacity of ~ **2 PBytes** and a nominal bandwidth of more than **2 TBytes/s**

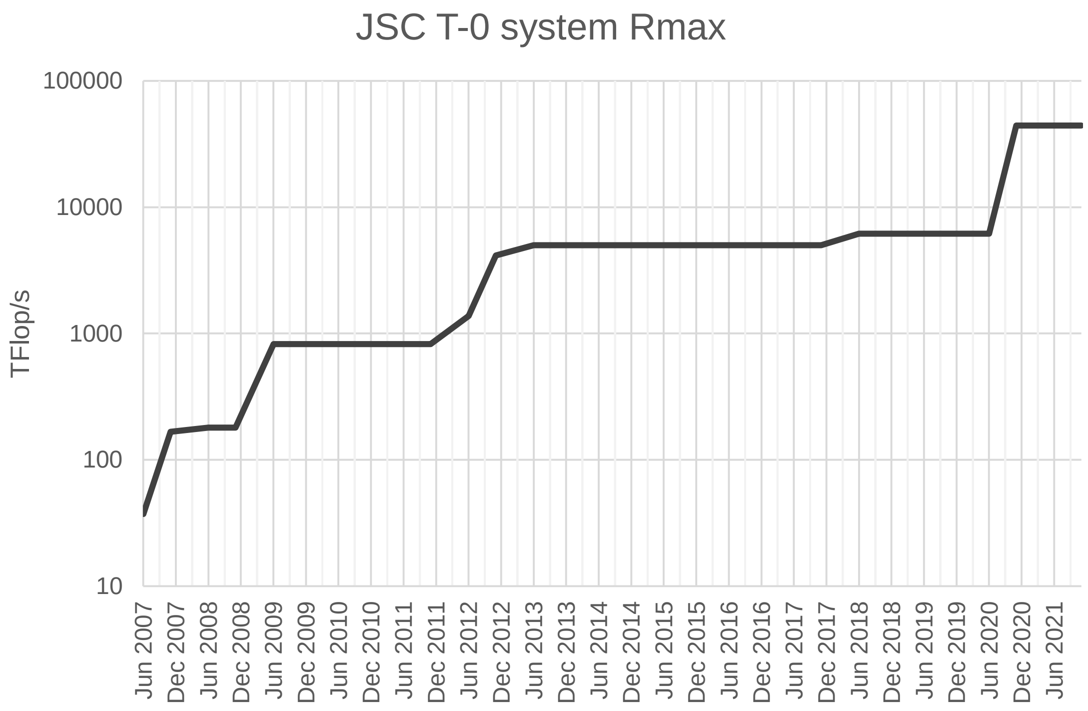
- Directly integrated into the high-speed InfiniBand-based interconnects of the client systems
- Each cluster has it's own "slice" of the HPST, but **one global namespace** (each cluster has access to data on "foreign slice")

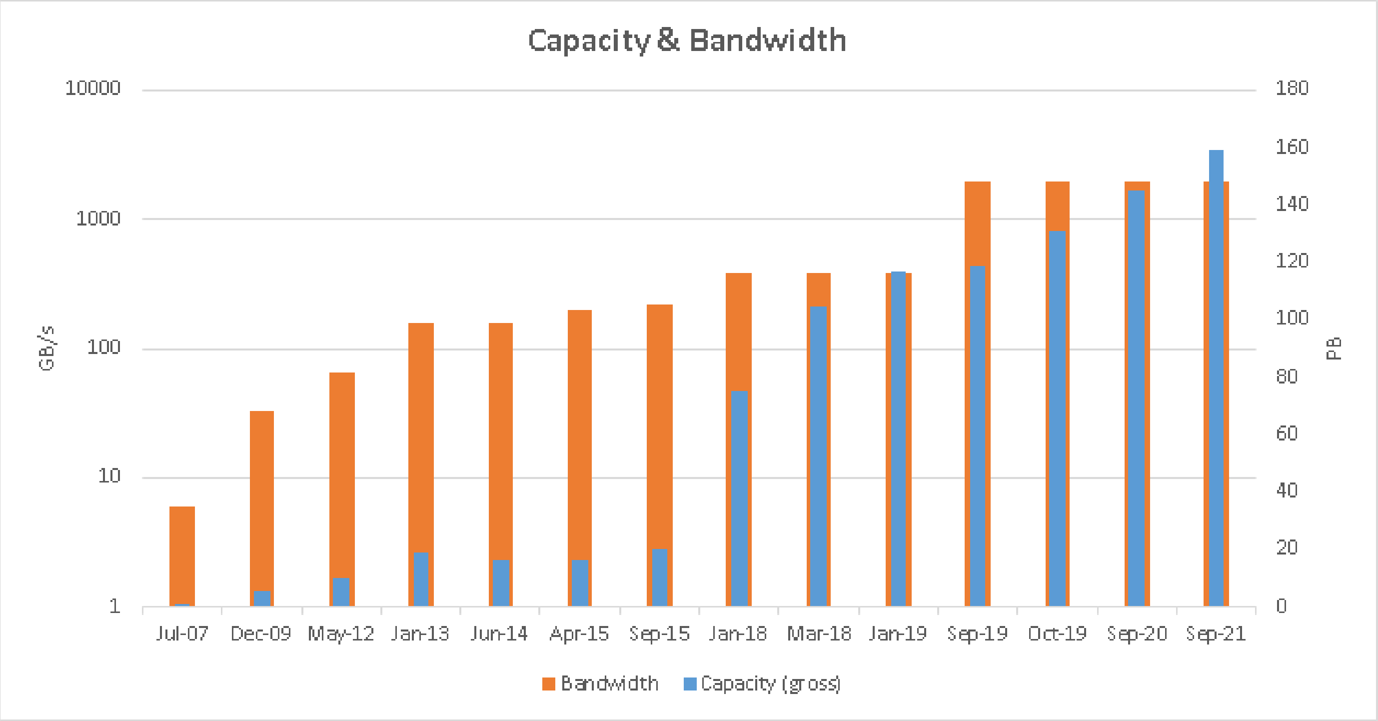
FENIX RESEARCH INFRASTRUCTURE
https://fenix-ri.eu/



System overview

Computational vs I/O performance





JÜLICH Forschungszentrum

Parallel I/O Software Stack