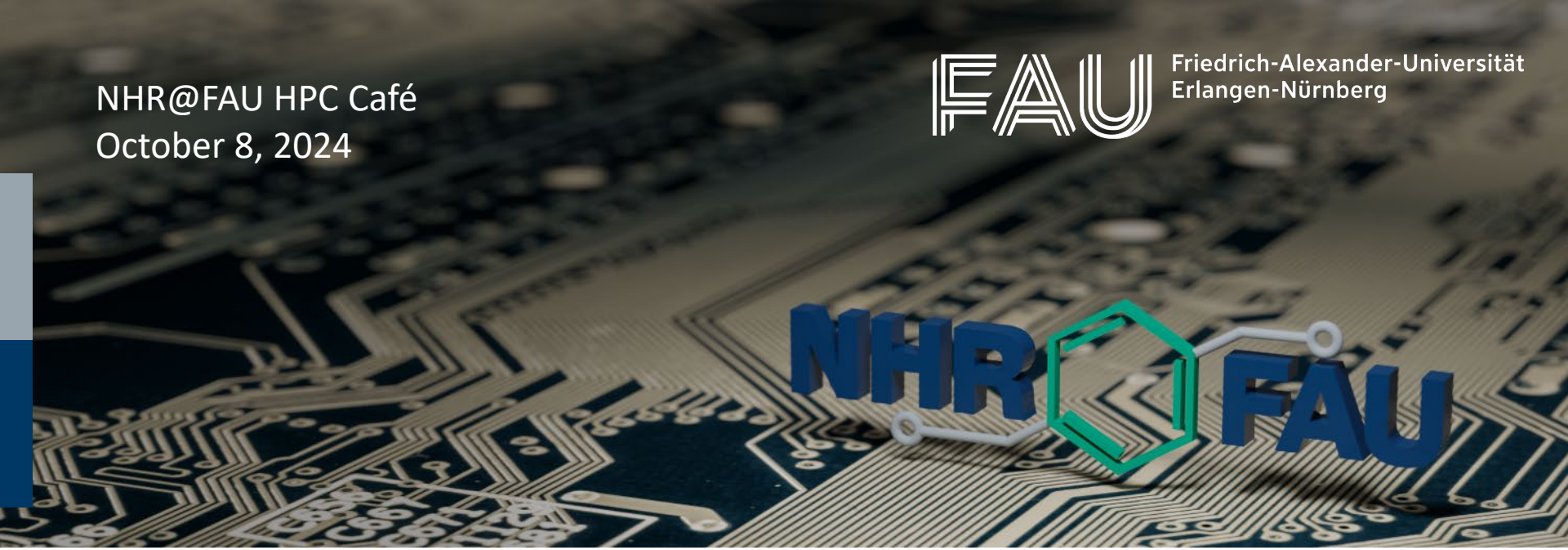


File systems and efficient data handling

Johannes Veh
Erlangen National High Performance Computing Center (NHR@FAU)

Hardware in a nutshell

HDD
Bandwidth: ~ 250 MB/s
Latency: 4+ ms
IOPS: ~ 200

SSD
Bandwidth: ~ 600 MB/s
Latency: 0,5 ms
IOPS: ~ 100.000

NVMe
Bandwidth: > 5.000 MB/s
Latency: 0,05 ms
IOPS: ~ 1.000.000

Picture
Author: Evan-Amos
License: Creative Commons Attribution-Share Alike 3.0 Unported

Picture
Author: D-Kuru/Wikimedia Commons
License: Creative Commons Attribution-Share Alike 3.0 Austria

3 racks with servers and disk arrays

928 HDDs, 20 SSDs, 2.400 kg

File systems

- File system == directory structure that can store files
- Several file systems can be “mounted” at a compute node
  - Similar to drive letters in Windows (C:, D:, …)
  - Mount points can be anywhere in the root file system

- Available file systems differ in size, redundancy and how they should be used

NHR@FAU file systems overview

Mount point	Access	Purpose	Technology	Backup	Snap-shots	Data lifetime	Quota
`/home/hpc`	`$HOME`	Source, input, important results	NFS on central servers, small	YES	YES @30 min	Account lifetime	50 GB
`/home/vault`	`$HPCVAULT`	Mid-/long-term storage	Central servers	YES	YES @24h	Account lifetime	500 GB

`/home/woody`
`/home/saturn`
`/home/titan`
`/home/atuin`
`/home/janus`
	`$WORK`	Short-/mid-term storage, General-purpose	Central NFS server	(NO)	NO	Account lifetime	Tier3: 1 TB, NHR: project quota

`/lustre`	`$FASTTMP` (only within Fritz+Alex)	High performance parallel I/O	Lustre parallel FS via InfiniBand	NO	NO	High watermark	Only inodes
`/anvme`	(only within Fritz+Alex)	High performance IOPS	Lustre parallel FS via InfiniBand	NO	NO	Workspace lifetime	Only inodes
`/???`	`$TMPDIR`	Node-local dir	SSD/NVMe/ ramdisk	NO	NO	Job runtime	NO

https://doc.nhr.fau.de/data/filesystems

Redundancy: snapshots vs backup

- Backup
  - Offline on tape to be recovered in case of system failure or data loss
  - Not recoverable by user

- Snapshots
  - Located on same file system as original data
  - In any directory:
    `$ cd .snapshots`
  - Kept for a specified amount of time
  - Data can be recovered by user

```
unrz55@sauron:~/programming/py/.snapshots $ ls -F
@GMT-2018.12.30-03.00.00/  @GMT-2019.01.23-11.00.00/  @GMT-2019.01.24-05.00.00/
@GMT-2019.01.06-03.00.00/  @GMT-2019.01.23-13.00.00/  @GMT-2019.01.24-07.00.00/
@GMT-2019.01.13-03.00.00/  @GMT-2019.01.23-15.00.00/  @GMT-2019.01.24-07.30.00/
@GMT-2019.01.18-03.00.00/  @GMT-2019.01.23-17.00.00/  @GMT-2019.01.24-08.00.00/
@GMT-2019.01.19-03.00.00/  @GMT-2019.01.23-19.00.00/  @GMT-2019.01.24-08.30.00/
@GMT-2019.01.20-03.00.00/  @GMT-2019.01.23-21.00.00/  @GMT-2019.01.24-09.00.00/
@GMT-2019.01.21-03.00.00/  @GMT-2019.01.23-23.00.00/  @GMT-2019.01.24-09.30.00/
@GMT-2019.01.22-03.00.00/  @GMT-2019.01.24-01.00.00/
@GMT-2019.01.23-03.00.00/  @GMT-2019.01.24-03.00.00/
```

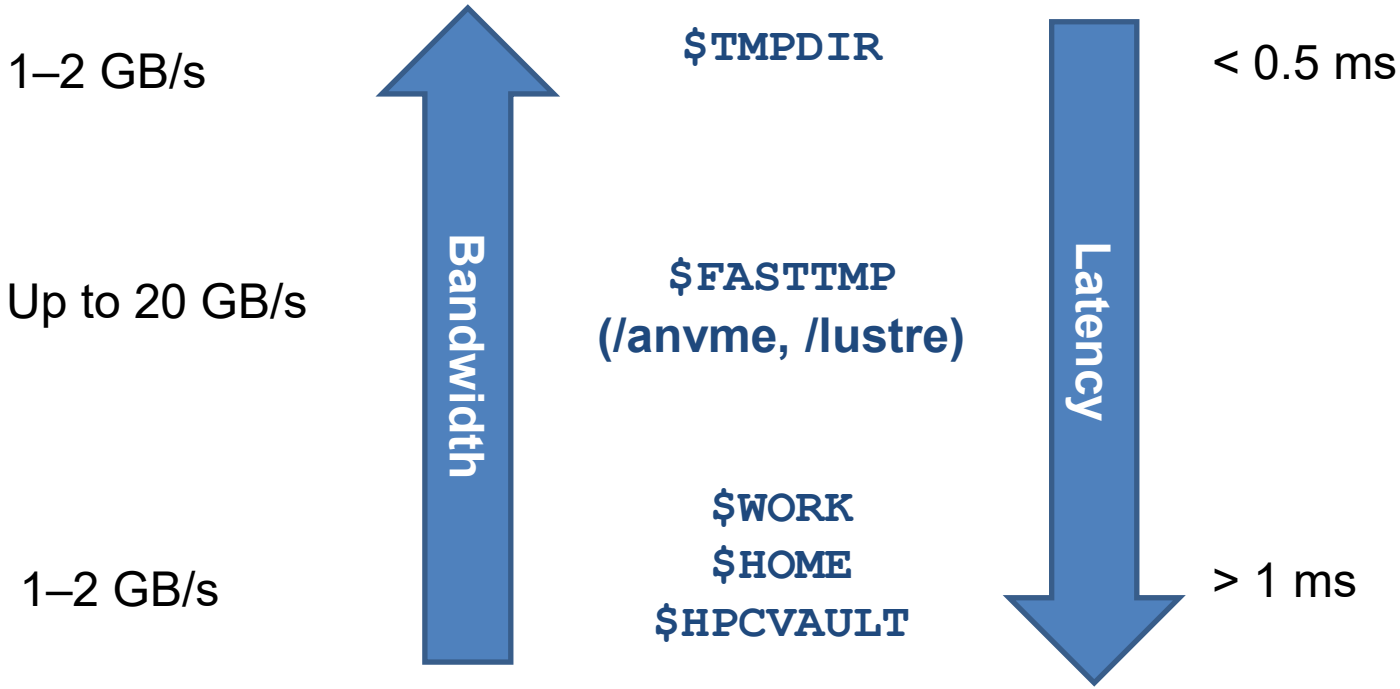
Workspaces (used for anvme)

- Currently only available on Fritz and Alex

- Script based manager for temporal directories
  - *ws_allocate <name> [<days>] # create a directory*
  - *ws_find <name> # return path to workspace*
  - *ws_list [<pattern>] # return information about workspaces*

  - https://doc.nhr.fau.de/data/workspaces

Bandwidth of storage systems



Types of data and where to store them

Source code, scripts,
Small input data,
configuration files, …

Important result files,
larger input data

Reproducible result files,
checkpoint-restart,
intermediate results

Checkpoint-restart,
intermediate results

Input data (escpecially for ML/AI),
Intermediate results

$HOME

$HPCVAULT

$WORK

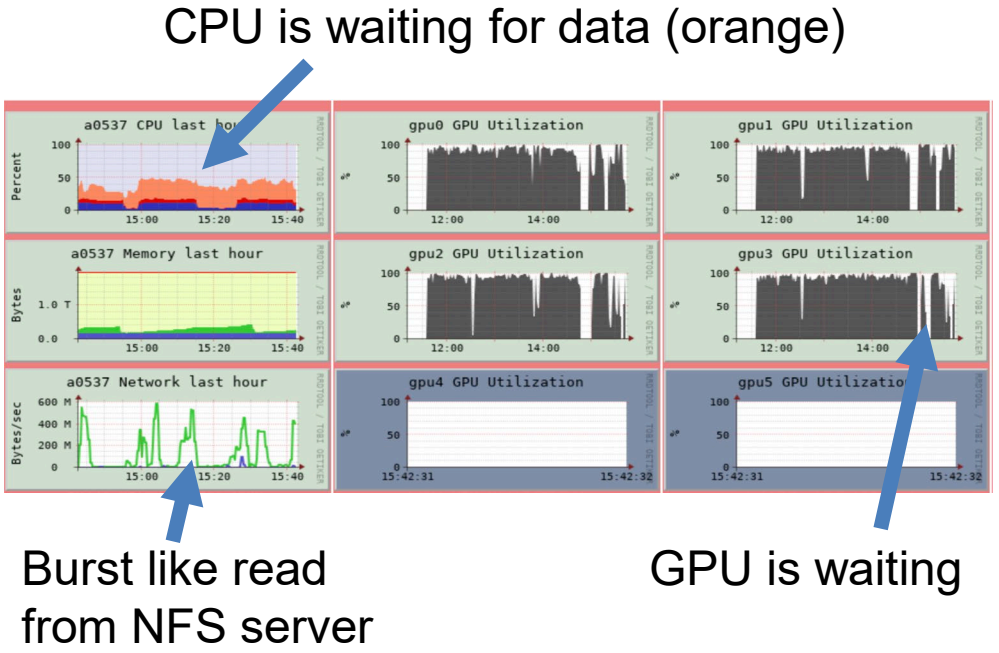
$FASTTMP
(/lustre, /anvme)

($TMPDIR)

The best I/O is I/O you do not do

- Clever data placement

- Improvement:
  - Copy data at jobstart to **`$TMPDIR`**



https://hpc.fau.de/about-us/success-stories/#Performance-gain-of-AI-application-with-data-stored-on-TMPDIR
https://hpc.fau.de/about-us/success-stories/#Speeding-up-machine-learning-on-GPU-accelerated-Cluster-nodes

Main Problem with NFS (and parallel FS)

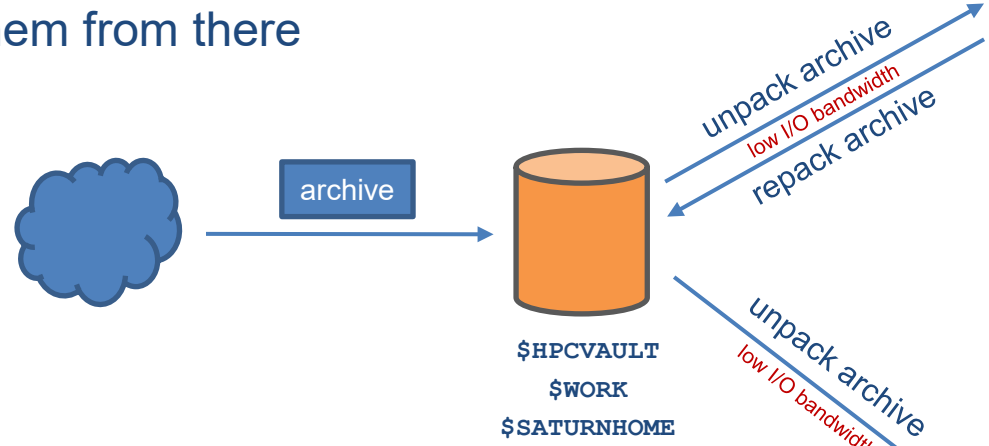
- In a job, avoid *accessing* large numbers of files located at **`$HOME, $HPCVAULT, $WORK`**
- Expensive operations on NFS (and also parallel file systems):
  - Access file stats like creation/modification time, permissions…
  - Opening/closing files
- These cause high load on servers
  - This slows down your job and impacts all other users
- Use instead
  - if supported by application: HDF5, file-based databases
  - pack files into an archive (e.g. tar + optional compression) and use node-local SSDs (huge amounts of file opens are no problem there)

Working with Archives and Node-Local SSDs

Do not unpack archive to:

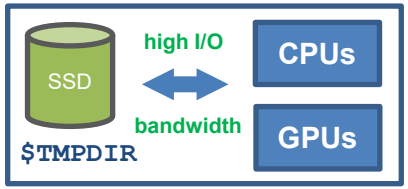
`$HOME/$HPCVAULT/$WORK`

Unpack files to node-local SSDs only and use them from there



Optionally: if original archive must be altered

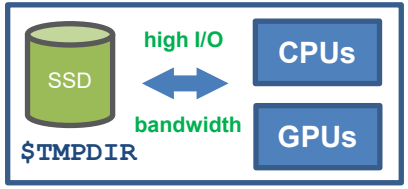
- unpack it to node local SSD (interactive job)
- optionally change files
- repack files and copy back to NFS



tinygpu / alex node

For simulation, training, …

- unpack archive to node local SSD
- perform simulation/training



tinygpu / alex node

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Questions? Suggestions?

Missed a talk?

https://hpc.fau.de/teaching/hpc-cafe/

Futher questions?

Send a mail to hpc-support@fau.de