DIY Tiering and Online FS Migration mit AFM

Ben Langenberg
19.10.2022

Helmholtz-Zentrum für Umweltforschung - UFZ

Top500 #1 06/2022 Frontier @ORNL
Anzahl Cores 8.730.112
Speicherkapazität 700 PB Orion Storage

[1]

Gliederung

I. Vorstellung Person & Zentrum - *2min*
II. EVE Linux Cluster - *1min*
III. GPFS & Storage Setup - *2min*
IV. DIY Tiering mit Policy Engine - *7min*
V. Aus der Not geboren - AFM - *8min*

Vorstellung
Ben Langenberg, 35 Jahre alt

- Seit **2005** am Helmholtz-Zentrum für Umweltforschung Leipzig
- Seit **2006** im HPC Bereich Linux tätig
- Seit **2010** GPFS Administrator
- **2012-2017** Dozent Berufsakademie Paralleles Rechnen
- Erste GPFS Version: **3.4.3-3**
- Einige **ISC** Besuche Hamburg, Frankfurt, Leipzig
- 5 Tage Schulung **GPFS Basic Administration** 2012 in Düsseldorf durch *b2 systeme*
- **2017-2020** Teamleiter EndPointService (HPC 50%)
- Seit **2021** Teamleiter HPC (kleines Team mit 5 FTE Windows & Linux HPC)
- Heute das erste mal **Spectrum Scale User Days**

Zentrum
Helmholtz-Zentrum für Umweltforschung - UFZ

- **Eins** von 18 Helmholtz Zentren in Deutschland
- **1994** gegründet
- **~ 1200** Mitarbeitende +- 100 Gäste
- **42** Nationalitäten
- **3** Standorte Leipzig, Magdeburg und Halle
- **5** wissenschaftliche Themenbereiche mit insgesamt 36 Departments

[2]

- **IT** Abteilung mit ca. 35 FTE (incl. Studenten und Auszubildende 60 Personen)
- Gegliedert in **5** Teams darunter ein Team **HPC**

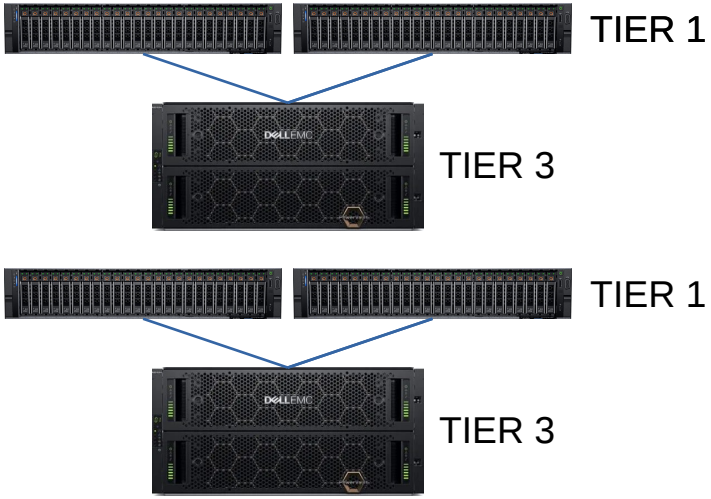
Linux Cluster
EVE

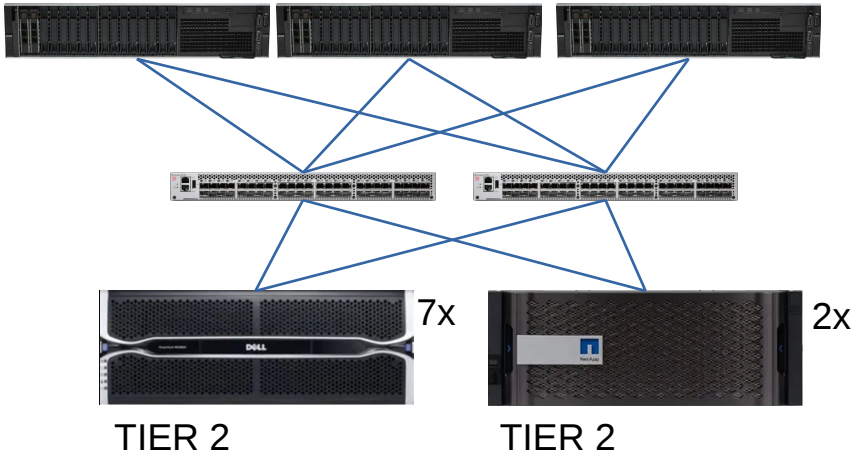
- **42x** Dell R750 56 Cores / 512 GB RAM = **2352**
- **46x** Dell R640 40 Cores / 384|1536 GB RAM = **1840**
- **4x** Login Knoten (2x für alle und 2x Fat Node BIOINF + DATA SCIENCE)
- **2x** Head Knoten
- **7x** Fileserver (Details folgen)
- **4192** Cores
- NVIDIA GPU Zoo: **7x** A100 80GB, **2x** A100 40GB, **2x** V100, **2x** K80
- Cornelis Networks (ehem. Intel/Dell OEM) Omnipath Interconnect **100GBit/s**
- **4.5 PB** GPFS Storage
- CentOS7 & RHEL8 (Wie gehts weiter Rocky oder ALMA :)?)
- **SLURM** Workload Manager
- Module System Lmod & Easybuild Build System
- **Ansible** Systemmanagement in IT Public Gitlab Projekt

Linux Cluster
aktuelles GPFS Setup

- Spectrum Scale **5.2.1-1**
- **2** Filesysteme
  - Home & Software **15TB**
  - Scratch & Daten **4.5PB**
- **7** Fileserver
  - 4x Dell R7525 *1. Filesystem*
    - je **16x 6,4TB** NVMe SSD
    - *je im paar SAS direct Attached Dell ME4084 mit je* **84x 16TB** SATA 7k
  - 3x Dell R740 *1. Filesystem / 2. Filesystem*
    - 2x 16Gbit/s SAN Anbindung
- 2 Brocade 6510 48 Port SAN Switchen
- **1x** MD3820f mit **12x 1.84TB** SSD (Home + Software) *2. Filesystem*
- **7x** DellMD3860f mit je **60x 6TB** SATA 7k *1. Filesystem*
- **2x** Netapp E2860 mit je **60x 8TB** SATA 7k *1. Filesystem*

Linux Cluster
aktuelles GPFS Setup schematisch



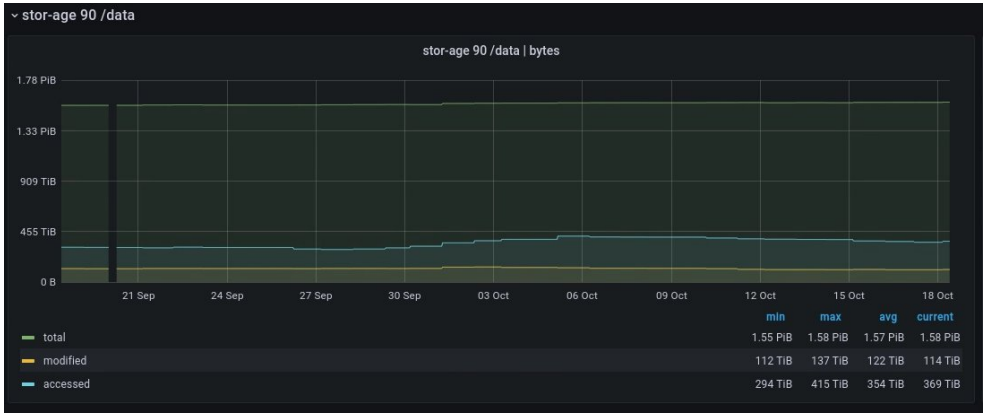


Linux Cluster
Tiering

- Datenanalyse **2020** durch optimierte Tools *[stor-age]*
- Täglicher Timer auf alle Filesets mit **unused** und **unmodified**
- Im Monat werden im Durchschnitt ca. **50-100TB** Modified (Stand 2020)

```
stor-age: analyzing /data/msb

Directory Age Bytes     Accessed  Percent Modified Percent Files    Accessed Percent Modified Percent
/data/msb 160 294.1 TiB 174.1 TiB  59.19% 31.2 TiB  10.61% 44448221 3674020   8.27%  2223037      5%


real    12m25.005s
user    8m29.941s
sys     5m51.497s
filer10 ~ # time stor-age --progress --format table 160 -- /data/msb
```

Linux Cluster
Tiering



bad boys

top unaccessed bytes bad boys

Metric	Current
/data/satellite	185.8 TiB
/data/msb	137.7 TiB
/data/bioinf	115.5 TiB
/data/edge	92.2 TiB
/data/hicam	92.2 TiB

1 2 3 4 5 6 7 8 9

Linux Cluster
Tiering

- Preis Leistung bei SAS NL HDD im **16TB** Bereich am besten
- Durch Masse mit **6TB/8TB** SAS NL HDD SAN Notwendig dessen Komplexität wir zurück bauen wollen
- Direct Attached SAS bringt mehr Durchsatz

—> Kauft euch doch ne **ESS**!?

- Nicht möglich durch Neuinvestition in den OmniPath Interconnect 100 im Jahr **2019**
- Mit der Hilfe von **Jochen Zeller (SVA)** konzipierten wir ein Selbst gebautes Tiering

- Mit Hilfe von GPFS Replizierter NVMe Speicher als ***TIER1 + Metadaten***
- Vorhandene **6TB/8TB** SAS NL Disks als ***TIER2***
- Neue **16TB** SAS NL Disks als ***TIER3***

```
filer11 ~ # mmlspool gpfs1 --block-size auto
Storage pools in file system at '/gpfs1':
Name                    Id   BlkSize Data Meta    Total Data        Free Data             Total Meta            Free Meta
system                   0      8 MB   no  yes             0        0 (  0%)              46.58T        33.05T ( 71%)
nvme                 65537      8 MB  yes   no          326T      199.3T ( 61%)                0             0 (  0%)
nlsasme              65538      8 MB  yes   no        1.817P      732.4T ( 39%)                0             0 (  0%)
nlsasmd              65539      8 MB  yes   no        2.459P        1.481P ( 60%)              0             0 (  0%)
```

Linux Cluster
Tiering Migration

- Wird Stündlich durch einen der systemd timer gestartet
- $1 je nach Füllstand des **nvme** Pools/TIERs

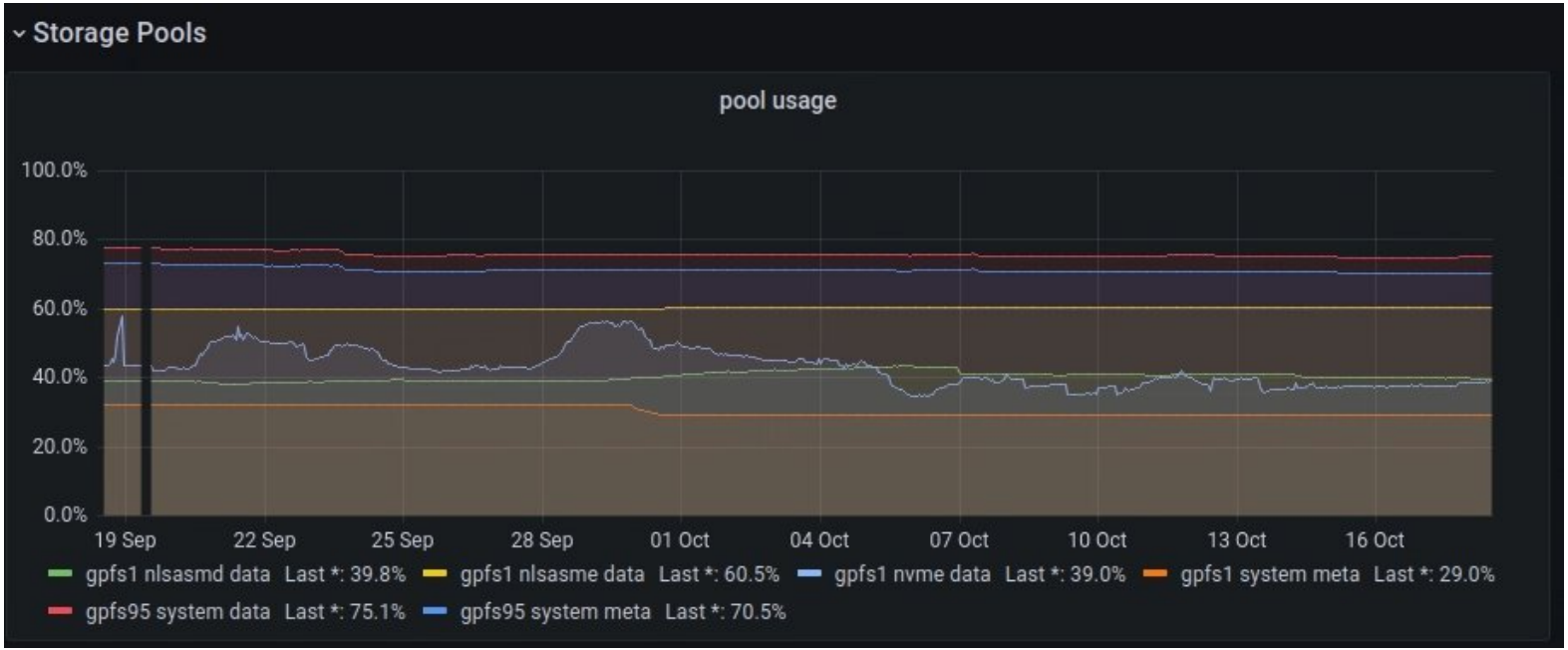
```
RULE 'migration'
  MIGRATE
    FROM POOL 'nvme' TO POOL 'nlsas' REPLICATE (1)
    WEIGHT(0)
    WHERE (
          NOT is_offline
      AND NOT is_empty
      AND NOT pattern_excluded
      AND     inactive
      AND     (KB_ALLOCATED > $1)
    )
EOF
}
```

```
policy=$tmp/policy

nvme_usage="$(mmoxi pool-percent gpfs1 nvme)"

high_watermark='80'
low_watermark='60'

if [[ $nvme_usage -lt 42 ]]
then
  kb_threshold=1048576 # 1 GiB in KiB
elif [[ $nvme_usage -lt $low_watermark ]]
then
  kb_threshold=131072 # 128 MiB in KiB
elif [[ $nvme_usage -lt $high_watermark ]]
then
  kb_threshold=16384 # 16 MiB in KiB
else
  kb_threshold=1024 # 1 MiB in KiB
fi

eve_log_info "nvme: ${nvme_usage}%: threshold: $kb_threshold KB_ALLOCATED"

policy.out $kb_threshold > "$policy"
```

- Workflows von **TIER 2** und **TIER3** nur lesend
- Migration nach **TIER3** händisch, individuell auf Fileset Basis

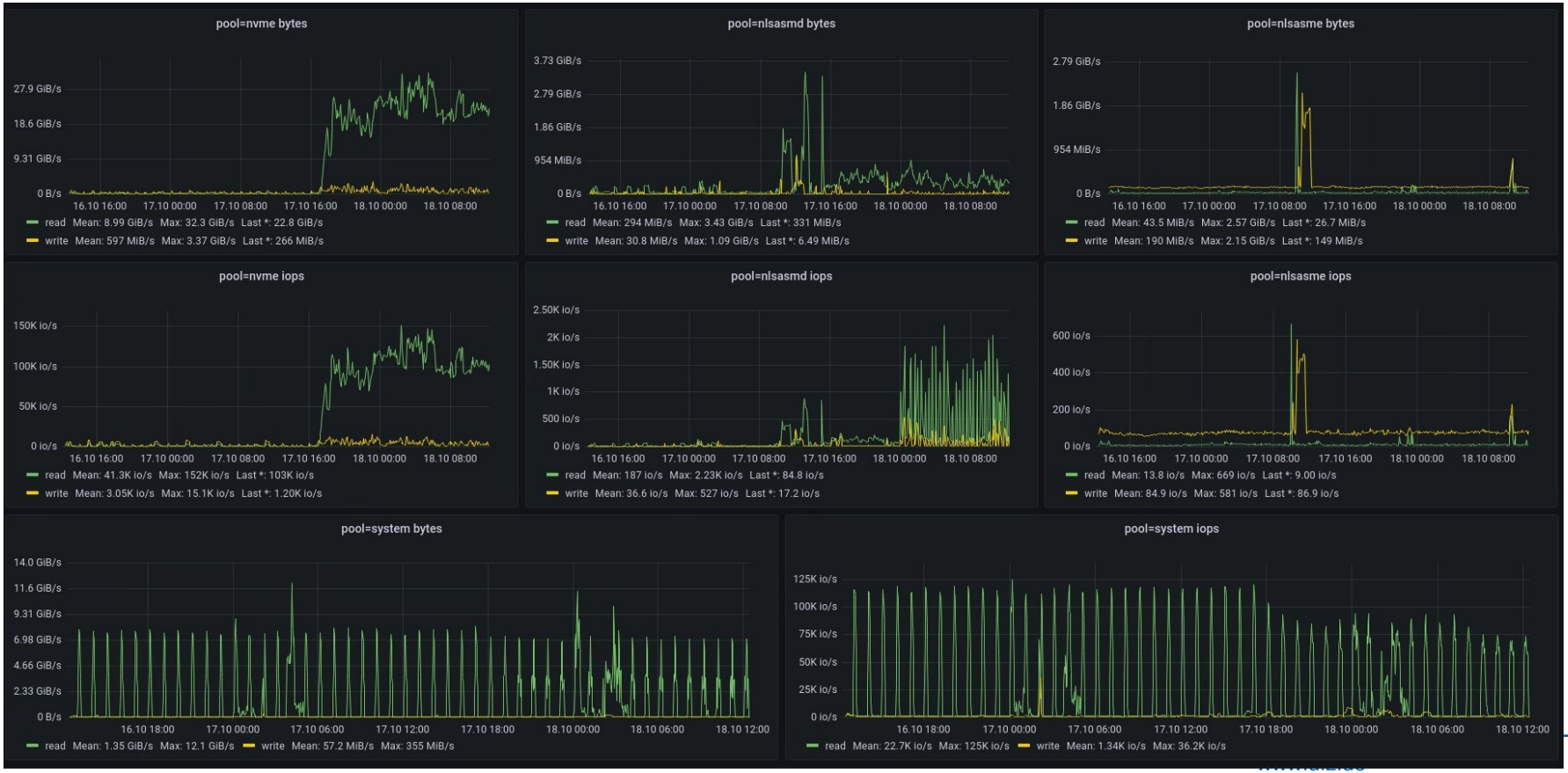
[mmoxi]

Linux Cluster
Tiering Migration



[mmoxi]

Linux Cluster
Tiering Migration



[mmoxi]

Linux Cluster
Tiering Migration

Feintuning über einen Zeitraum von 3-4 Monaten

- Monitoring der Auslastung und Belegung je Pool [mmoxi]
- Anpassen der Migrationspolicyschwellwerte (Welche Dateien sollen migriert werden)
- Beobachtung der Workflows der Einzelnen Nutzer:innengruppen

Lessons Learned

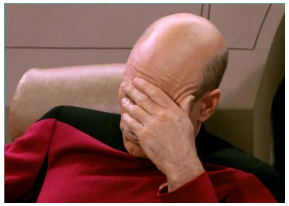
- Neue IO Patterns / Nutzerinnen:gruppen können durch hohe Schreibgeschwindigkeit bei relativ geringer Kapazität des **nvme** Pools das Pool innerhalb weniger Stunden füllen
- Anfänglicher Anpassungsbedarf der Migrationen
- Wiederherstellung des Replikationssyncs zwischen **nvme** Fileserverpaaren zwischen **1-2h**

AFM
Ursprünglicher Plan

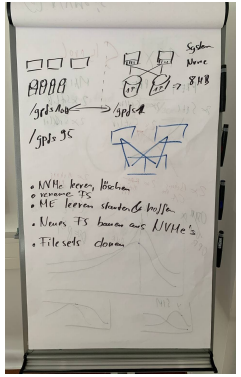
Plan für Ende Oktober / Anfang November 2021

- Integration 2x Dell R7525 AMD Fileserver mit je **16x 6,4TB** NVMe SSD Speicher
- Aufbau Tiering mit Initialplacement auf die TIERs
- Integration 1x Dell ME4084 mit **84x 16TB** SAS NL HDD
- Integration neuer NSDs ins vorhandende GPFS Filesystem (Existiert seit **2014**) und online Migration durch löschen alter NSDs
- Dabei sollte je Zwei NVMe je Server als Metadaten NSD dienen -> **4x 6.4 TB NSDs**

AFM
Aber



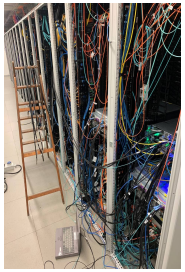
- Metadaten des Filesystems ursprünglich auf 200GB NSDs (SSD FC Storage) welches über die Jahre mit 400GB NSDs und schließlich 800GB NSDs erweitert wurde
- Bei der Initialisierung des *system* Pools **2014** durch **200GB NSDs** wurde die maximale NSD Größe für dieses auf **2.5TB** festgelegt *(eine GPFS Allocation Rules die man nie wahrnimmt und sich denkt, so groß werden Metadaten NSDs bei uns sowieso nie)*
- **6.4 TB != 2.5TB**



<u>Abwägung der Möglichkeiten:</u>

- **3 GPT** Partitionen **2.2TB** auf einer NVMe und jeweils 3 NSDs ans GPFS übergeben *(technisch möglich aber nicht von IBM supported)*
- Erstellung neues Filesystem und Migration aller Daten über traditionelle Methoden im POSIX *(Wartungsfenster von **3-7 Tagen** würde zeitlich gesprengt werden)*

AFM
Wir könnten es doch trotzdem Live machen :)



- Neues Filesystem mit optimierten Einstellungen (größere Blockgröße)
- Jedes im alten Filesystem vorhandene GPFS Fileset wird im neuen Filesystem angelegt und es wird eine Beziehung zueinander hergestellt

- Weitere **ME4084 (84x16TB NVMe)** während des Wartungfensters geliefert wobei eine Anbindung erst später geplant war diente als weitere Hauptkapazitätsherberge für das neue Filesystem so dass insgesamt **2PB** in **TIER3** zur Verfügung standen
- Nach dem Prefetchen der **Metadaten aller Filesets** System konnte die **Userschaft** mit Performance Einschränkungen wieder das System nutzen
- Im POSIX Userspace der Filesysteme alle Dateien und Ordner vorhanden

```
mmafmconfig enable /$OLDFS/data/cathyd
mmcrfileset $NEWFS data_eve_cathyd -p afmMode=lu,afmTarget=gpfs:///$OLDFS/data/cathyd
mmlinkfileset $NEWFS data_eve_cathyd -J /$NEWFS/data/cathyd
mmchfileset $NEWFS data_eve_cathyd -p afmEnableAutoEviction=no
```

- ➢ Sehr hohe Schreiblast auf den ME Systemen mit 16TB HDD
- ➢ Stück für Stückes leeren der TIER2 Storages und manuelles entfernen aus dem alten / hinzufügen ins neue FS

AFM
8 Wochen Online Migration

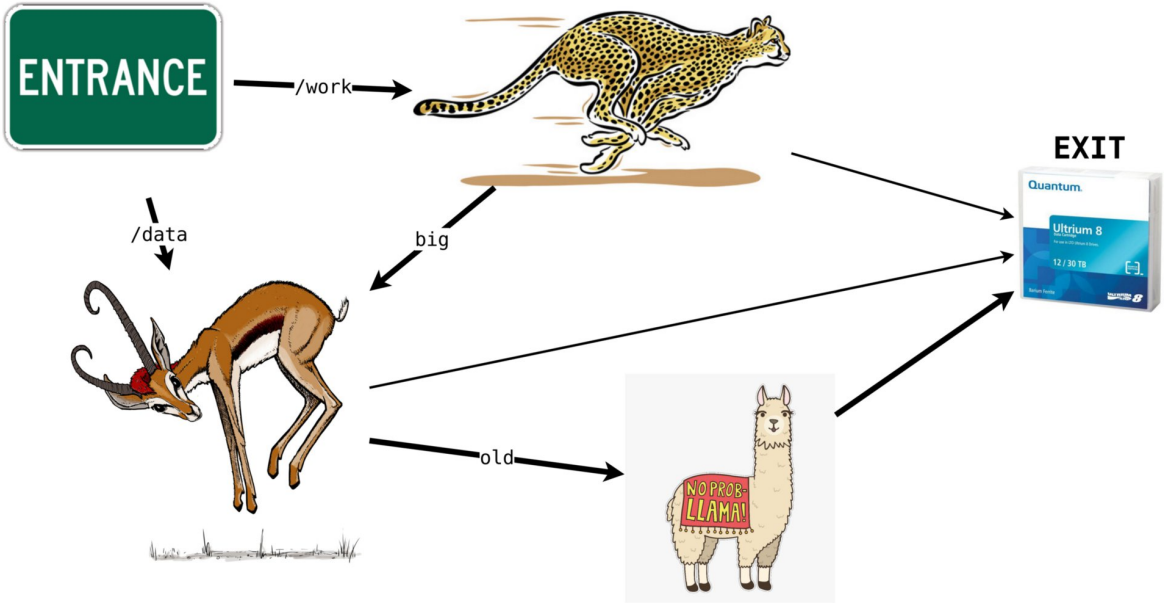
```
# Schritt 1 ist prefetchen, abwarten und Tee trinken
mmafmctl gpfs1 prefetch -j data_$project --directory /gpfs1/data/$project --enable-failed-file-list --gateway filer1-ib0

# Fileset noch gelinkt, erstmal GPFS den Cache überprüfen lassen
# Bei dem Befehl das erste mal 2x "yes" angeben
mmchfileset gpfs1 data_$project -p afmTarget=disable

# Wenn Cache OK ist, dann unlinken in gpfs1
# Ggf. forcen wenn Leute noch drauf sind
mmunlinkfileset gpfs1 data_$project -f

# Den mmchfileset Befehl nochma absetzen
# Diesmal bei der zweiten Frage "nein" auswählen
mmchfileset gpfs1 data_$project -p afmTarget=disable

# Wenn wieder keine Fehler kommen, dann ist die AFM Beziehung aufgelöst
# AFM Status überprüfen. Wenn AFM disabled ist, meckert er rum
mmafmctl gpfs1 getstate -j data_$project

# Man muss das directory nur nochmal neu linken in gpfs1
# Wenn unsicher wegen Pfad, dann einfach in $OLDFS nachguggn
mmlinkfileset gpfs1 data_$project -J /gpfs1/data/$project

# Das alte directory kann nun auf dem alten Filesystem geunlinked
mmunlinkfileset $OLDFS $fileset

# und dann kann das Directory in $OLDFS gelöscht werden
# Wenn es noch kein Fileset ist, dann muss man alternativ mit rm -rf arbeiten
mmdelfileset $OLDFS  $fileset -f
```

AFM
Abschluss

- Abschluss am 23.12.2021 mit Auflösung letzter AFM Beziehung und Löschen des alten Filesystems
- Beginn Tuning der TIERing Migrationen
- Dokumentation der neuen Möglichkeiten
- Hohes Vertrauen in AFM
- Beginn Vertrauen in TIERing Konstrukt

...

TIERing
Abschluss



- our zoo has three animals, cheetah, gazelle, and llama
- animal speed correlates with storage write speed
- read speed can be assumed to be approximately the same for all tiers
- new zoo visitors at work will visit cheetah first
- new zoo visitors at data will visit gazelle first
- big visitors will be migrated from cheetah to gazelle, due to the cheetah pen being our smallest
- old / long-term visitors will be migrated from gazelle to llama
- visitors can exit from anywhere exit, but due to llama having mostly long-term visitors, it's likely exit is done from there

Vielen Dank für die Aufmerksamkeit

ben.langenberg@ufz.de

Quellen

- [1] https://e3zine.com/exascale-supercomputer-frontier-for-the-u-s-department-of-energy/
- [2] https://www.ufz.de/index.php?de=34208
- [Picard] https://www.cnet.com/culture/entertainment/picard-memes-patrick-stewart-best-viral-star-trek-moments/
- [stor-age] https://crates.io/crates/stor-age
- [mmdu] https://crates.io/crates/mmdu
- [mmoxi] https://crates.io/crates/mmoxi