Statusbericht
Johannes Gutenberg-Universität Mainz

André Brinkmann

20. Oktober 2022

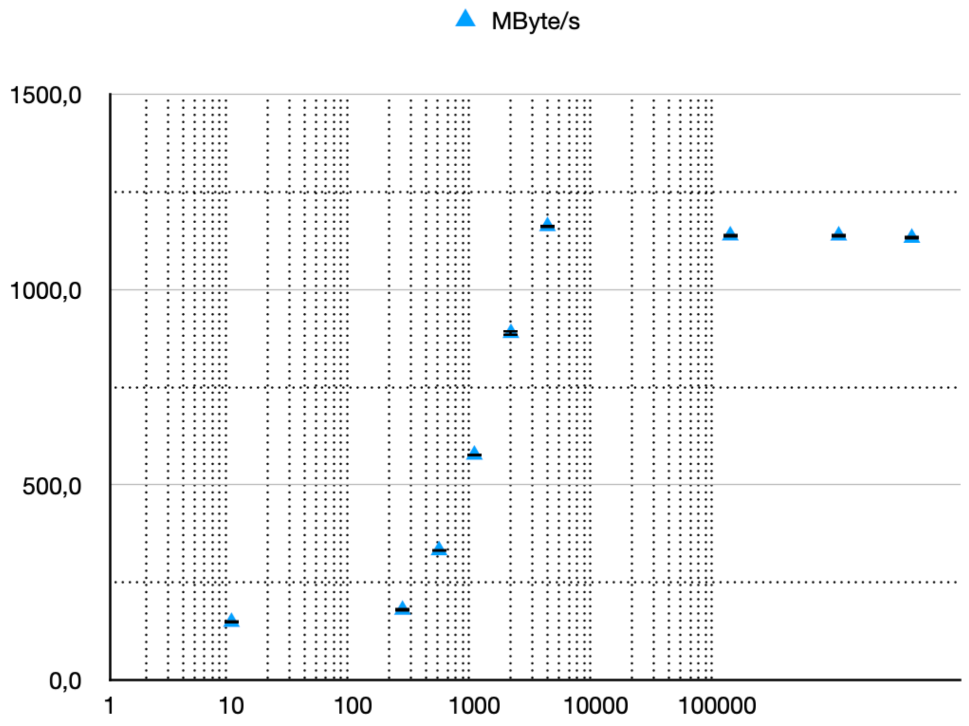
Fidium Konsortialtreffen

Hamburg - DESY

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Infrastrukturmaßnahmen JGU – GSI

- Test-Lustre vom GSI wurde erfolgreich über direkte „Tbit"-Verbindung inklusive UID-Mapping mit den entsprechenden Zugriffsrechten auf einem Clusterknoten in Mainz erfolgreich getestet
- Mount wird zurzeit auf allen Cluster-Knoten von Mogon II eingerichtet
- GSI eruiert notwendige Umstellungen, um Produktions-Lustre mounten zu können

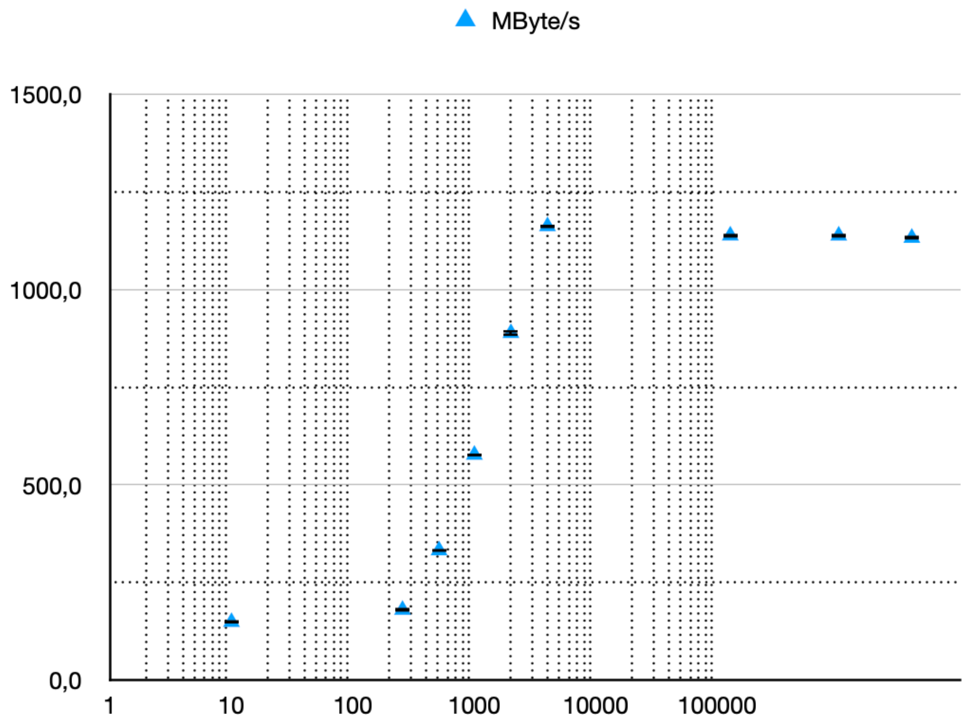


JG|U

Infrastrukturmaßnahmen JGU – GSI

Organisatorische nächste Schritte:

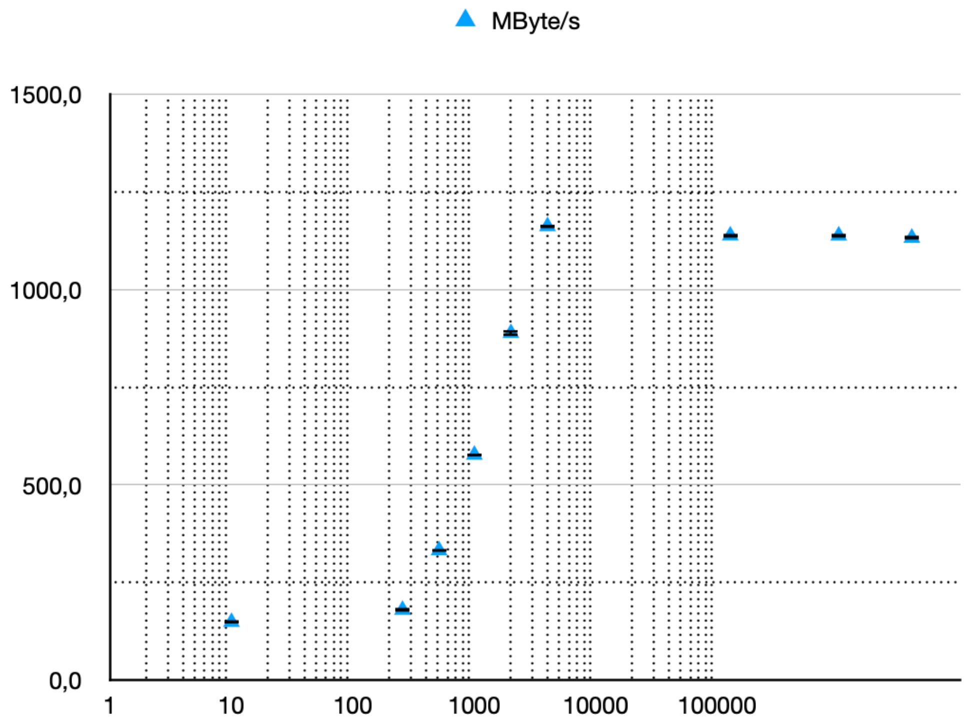
- Austausch administrativer Informationen (IP-Adressen, Account-Listen, …)
- Mount des Dateisystems auf allen Mogon II und Mogon III Knoten
- Mount des Lustre auf einem Login-Knoten
- Integration in SLURM Prolog-Skript



Infrastrukturmaßnahmen JGU – GSI

Offene Fragen:

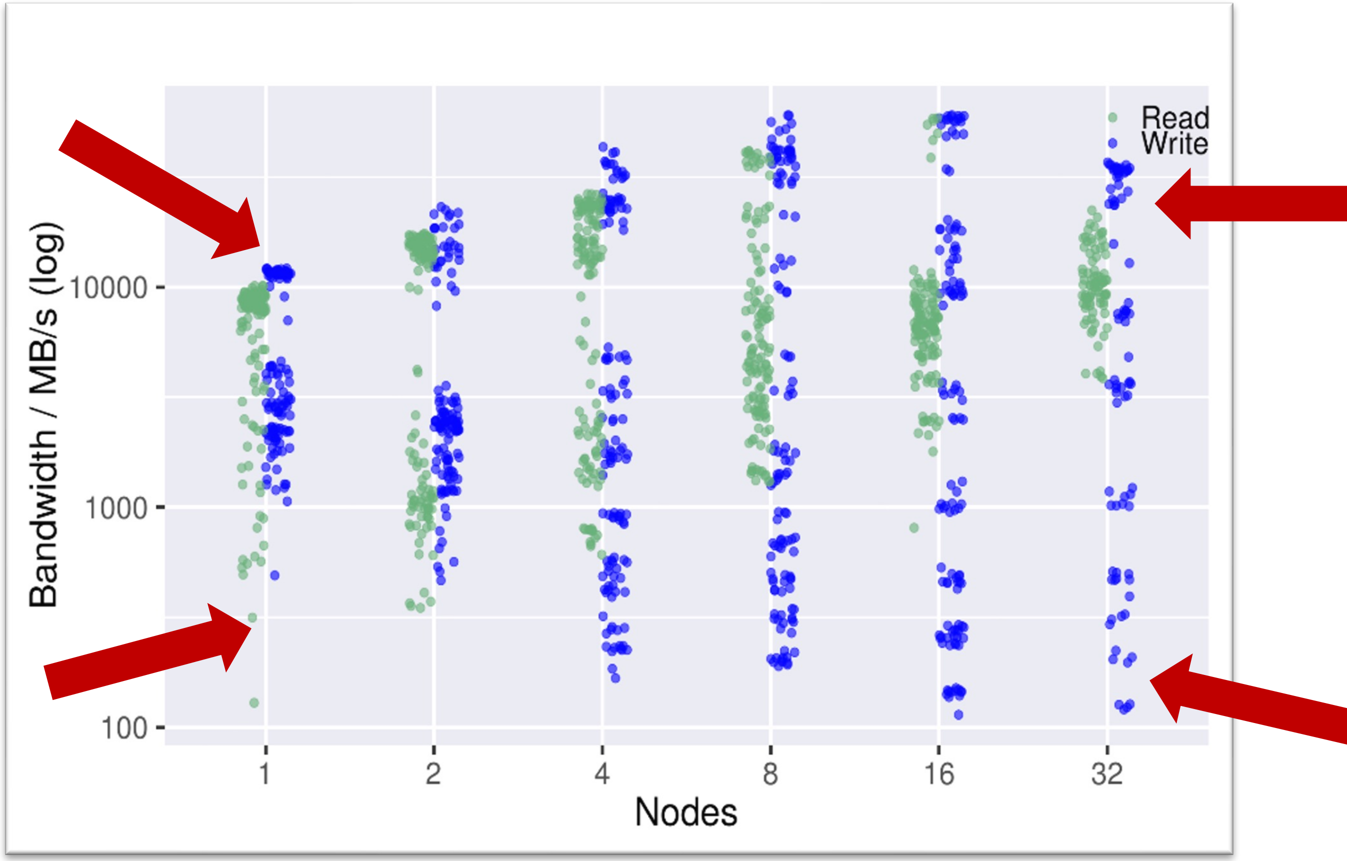
- Integration weiterer L-Net-Router auf Mainzer Seite?
- Ist es möglich, den Zugriff auf Dateisystem für nicht gemappte Nutzer zu verbieten? Aktuell sind nicht gemappte User „nobody" und können z.B. world readable Dateien lesen und damit ggf. zu viel Traffic erzeugen



The cost of using the parallel file system

I/O performance varies
wildly for identical workloads

Applications suffer due to PFS load!



Motivation

MareNostrum 4
Peak I/O bandwidth:
Read: 204,96 GB/s
Write: 120,89 GB/s

PFS BW per node (avg. 3456 nodes):
Read: 60,72 MB/s
Write: 35,81 MB/s

vs

Node-local Intel s3520 SSD:
Read: 450 MB/s
Write: 380 MB/s

From S. Moré, “Storage in MareNostrum 4: Petaflop System Administration” PATC 03/2019

- Minimize uncoordinated PFS usage
- Minimize redundant data movement and schedule transfers to reduce PFS contention
- Improve data locality: Do work where data lives!

GekkoFS

Available here https://storage.bsc.es/gitlab/hpc/gekkofs/

Core challenges to be addressed

1. **Scalability**
   - No central components
   - Linear scaling with # number

2. **Fast deployment**
   - Wall time is important
   - <10 seconds for deployment

3. **User space**
   - User decides
   - No administrative support

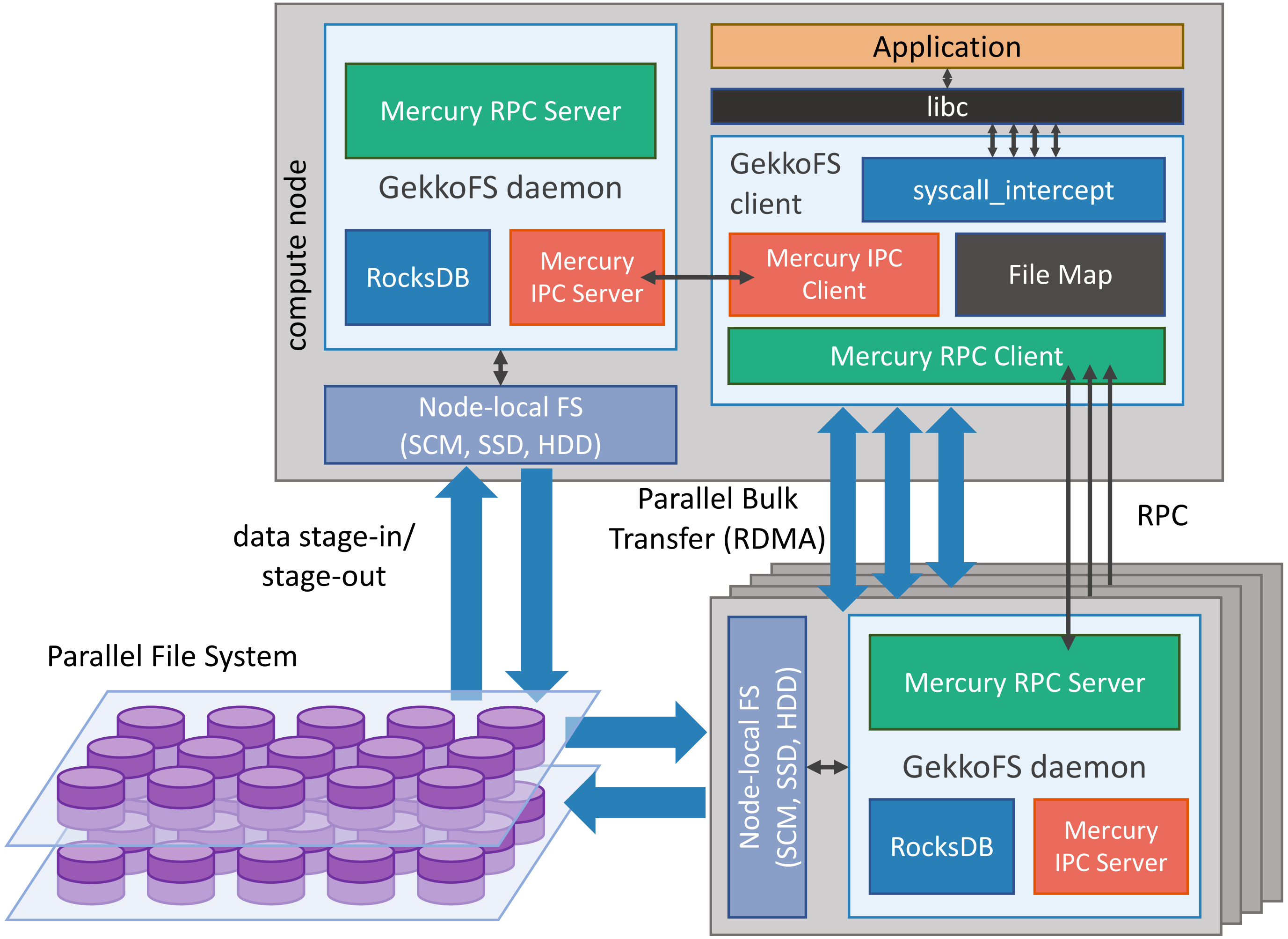
4. **Hardware independence**
   - Use accessible storage
   - Use fast network fabrics

GekkoFS architecture

Mercury
A high-performance RPC framework from ANL
https://mercury-hpc.github.io

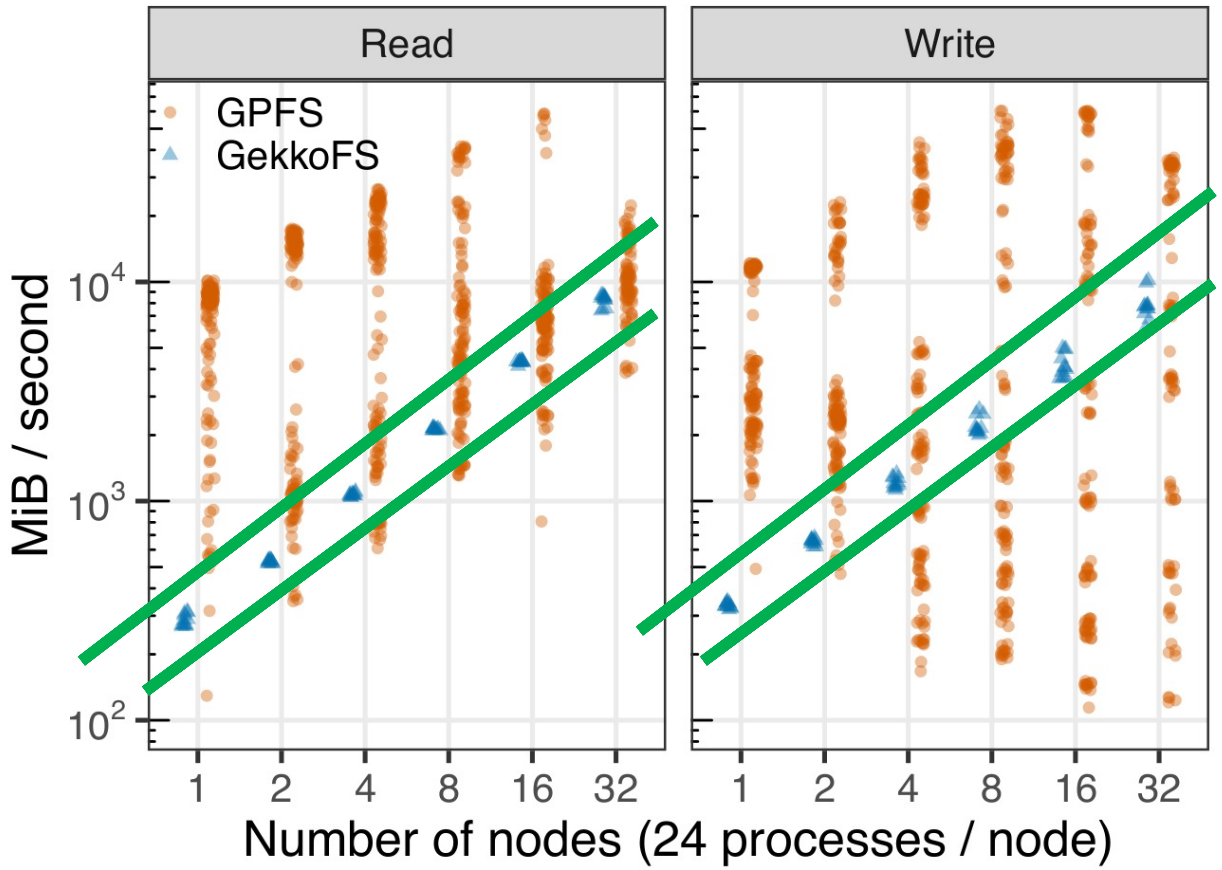
RocksDB
A persistent key-value store for fast storage from Facebook
http://rocksdb.org

syscall_intercept
A system call interception library from Intel
https://github.com/pmem/syscall_intercept



M.-A. Vef, N. Moti, T. Süß, M. Tacke, T. Tocci, R. Nou, A. Miranda, T. Cortes, A. Brinkmann.
GekkoFS – A Temporary Burst Buffer File System for HPC Applications. In Journal of Computer Science and Technology (JCST), 2020

Performance variability revisited

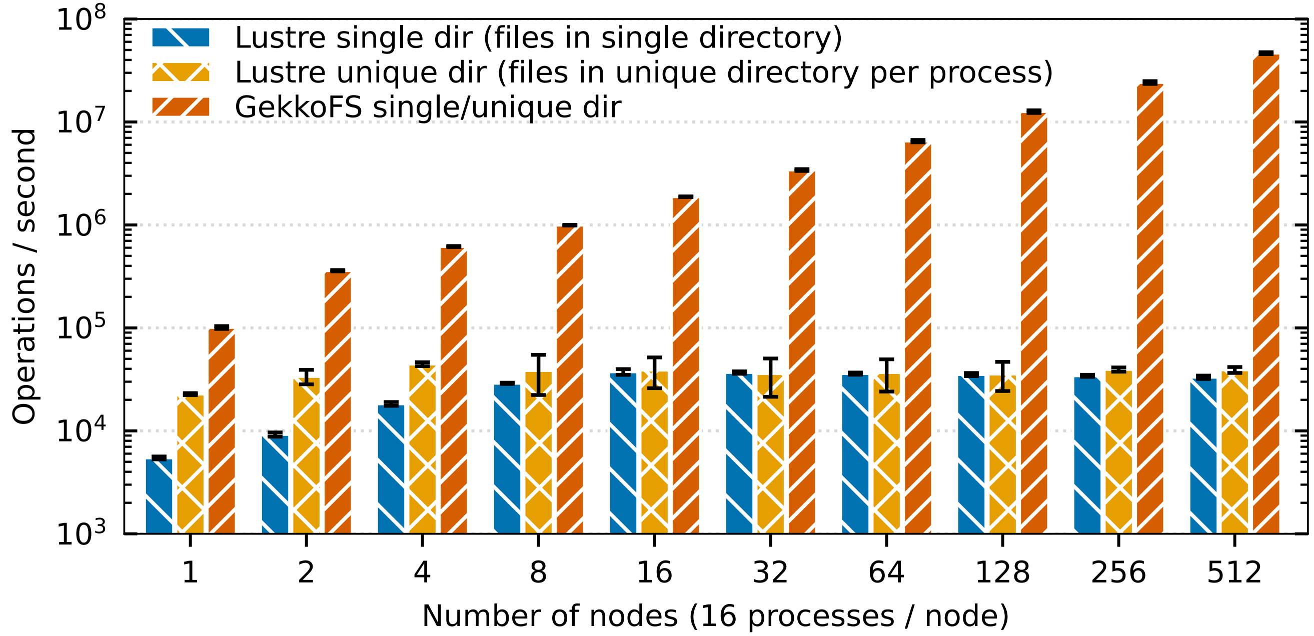


I/O performance variability is greatly reduced

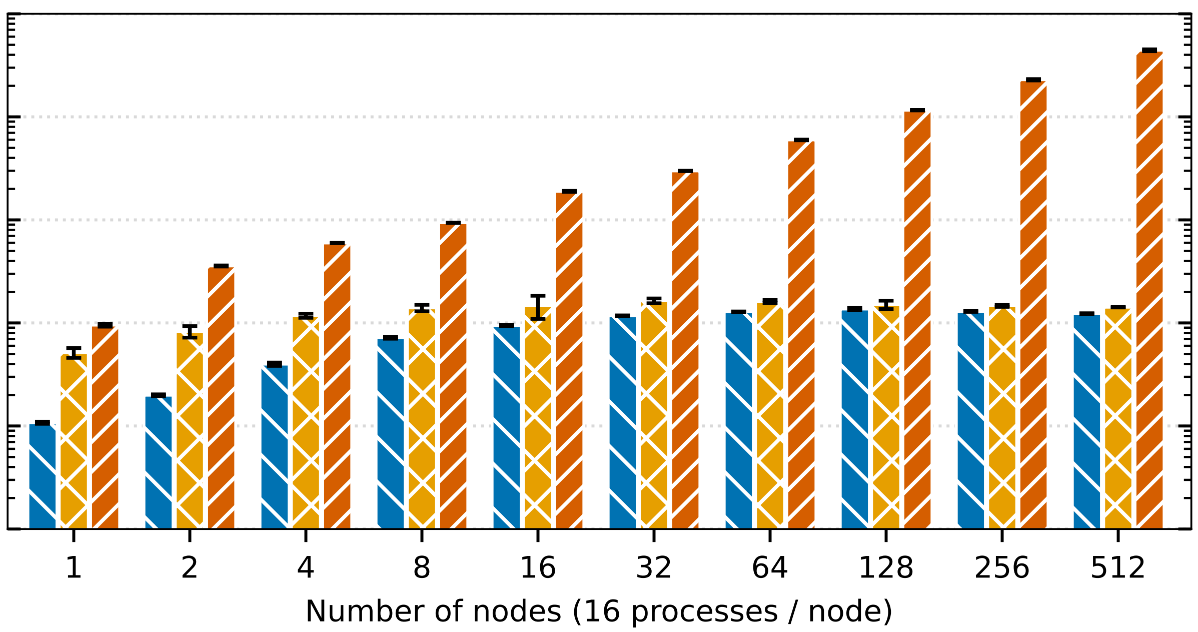
Metadata performance

GekkoFS vs. Lustre @ MOGON II

- GekkoFS weakly scaled (100K files per process)
  - More than 819 million files in total at 512 nodes for GekkoFS



File create performance



File stat performance

Ranked 4th in IO500 10-node challenge @ SC'19

JG|U

Ad hoc file systems in real life

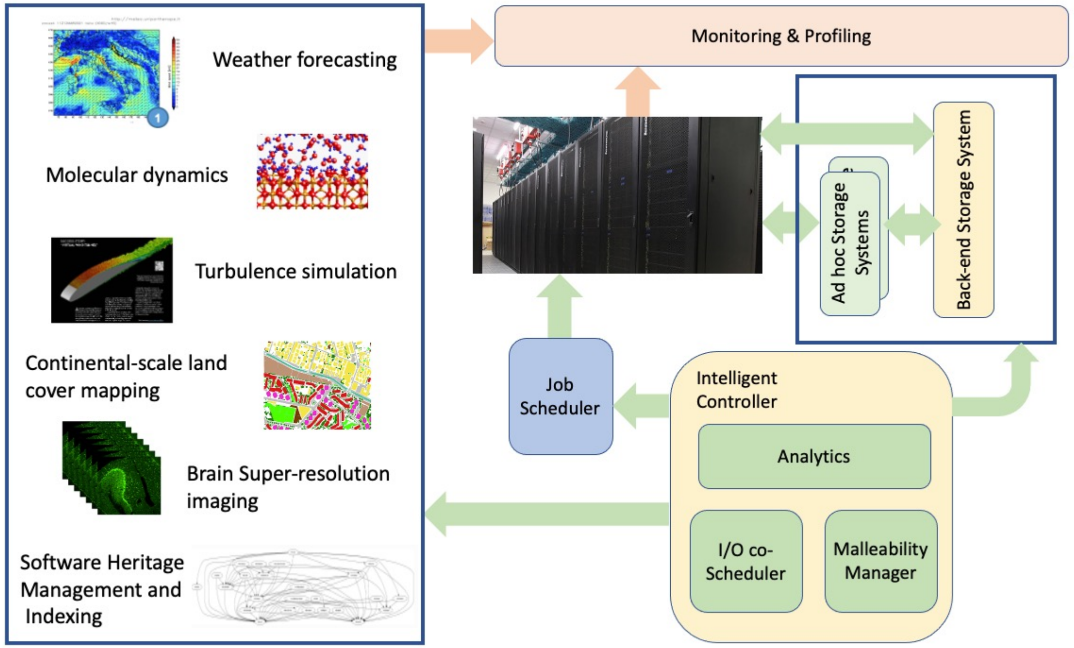
Challenges and possible solutions

- **No** transparent usage and requires user interaction
  - Starting and stopping ad hoc file system
  - Data staging
  - Data is stored at two locations (threat of overwriting)
- The EuroHPC **ADMIRE** project
  - Adaptive multi-tier data management
  - Computational and I/O malleability
  - Focus on ad hoc storage systems
  - Lustre integration (DDN and JGU collaboration)

Proposal:
Combine the benefits of Lustre HSM, PCC, and ad hoc file systems

EuroHPC Joint Undertaking

ADMIRE malleable data solutions for HPC



EuroHPC ADMIRE project architecture.
https://admire-eurohpc.eu

Lustre
Hierarchical Storage Management
&
Persistent Client Cache

Lustre Persistent Client Caching (LPCC)

Motivation and goals

- Node-local storage media often remain **unused**
- **Transparently** include fast node-local storage into Lustre
- **Increase** I/O performance for I/O workflows and **decrease** I/O interference

Features

- LPCC integrates into established HSM mechanisms
- Layout lock mechanism to provide consistent cache services
- Maintain global unified namespace
- Two caching modes
  - RW-PCC: read-write cache on **single** client
  - RO-PCC: read-only cache on **multiple** clients

LPCC limitations

- LPCC offers caching in the context of a single node
- RW-PCC: One node can use the same resource
  - **No conflicting access allowed**
  - ➢ No parallel I/O from many nodes possible
- RO-PCC: Multiple nodes can cache the same resource
  - **Same access allowed but redundant data**
  - ➢ Can cause severe I/O overhead on parallel file system when many nodes cache the same data
- Cache capacity and I/O performance **restricted** by node-local storage
- Metadata (except file size) is only **partly** cached

Distributed ad hoc file systems can offer a solution to these limitations

Zusammenfassung

- Erfolgreicher Test zum Aufbau eines gemeinsamen Data-Lakes zwischen der GSI und der JGU auf Basis von Lustre
- Ausbau und Übernahme des Data Lakes in den Produktivbetrieb in der Umsetzung

Zielsetzung:

- Aufbau von Workflows zur Kopplung zwischen externen Daten, die im Data-Lake vorgehalten werden, dem lokalen Lustre-Dateisystem an der JGU sowie mit Knoten-lokalem Speicher über GekkoFS

We greatly appreciate any feedback!

Thank You

JGU

- Marc-André Vef vef@uni-mainz.de
- Maysam Rahmanpour mrahmanp@uni-mainz.de
- André Brinkmann brinkman@uni-mainz.de

Acknowledgements: Supported by

ADMIRE malleable data solutions for HPC

EuroHPC Joint Undertaking

FIDIUM

GekkoFS

Gitlab-Repo: https://storage.bsc.es/gitlab/hpc/gekkofs/

Some of the icons in this presentation have been designed using resources from flaticon.com by the authors Freepik and Bingge Liu