Deutsches Klimarechenzentrum
(German Climate Computing Centre)
DKRZ

Carsten Beyer (DKRZ) & Anna Fuchs (Universität Hamburg)

German Climate Computing Centre

Non-profit limited company since 1987

- Share-holders MPG (55%), FHH/UHH (27%), AWI (9%), Hereon (9%)
- 100+ staff at DKRZ
- 4+ staff at university research group

HLRE-4 – Levante (2022-2028)

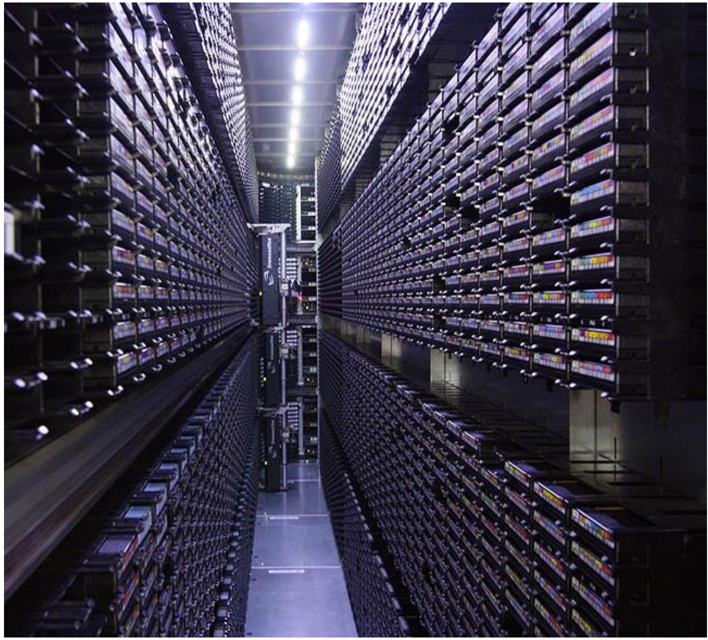
BullSequana, 3,000+ nodes, 370,000+ cores, AMD Milan, 14 PFLOPS
815 TB main memory, 130 PB disk storage,
60 GPU nodes (visualization, machine learning, faster codes)
hot liquid cooling with high efficiency

Current Lustre Storage in Levante

- HOME
  - 120 TiB NVMe
    - Home directories and software tree (User Quota)
    - Small files, fast access
- PROJECT
  - 118 PiB HDD based storage
    - Project directories (Lustre Project Quota)
    - SCRATCH directories of user (Lustre Project Quota)
- FASTDATA
  - Hybrid storage 200 TiB NVMe / 3 PiB HDD
    - Collaboration with DDN for testing new workflows / concepts

High Volume Data Archive

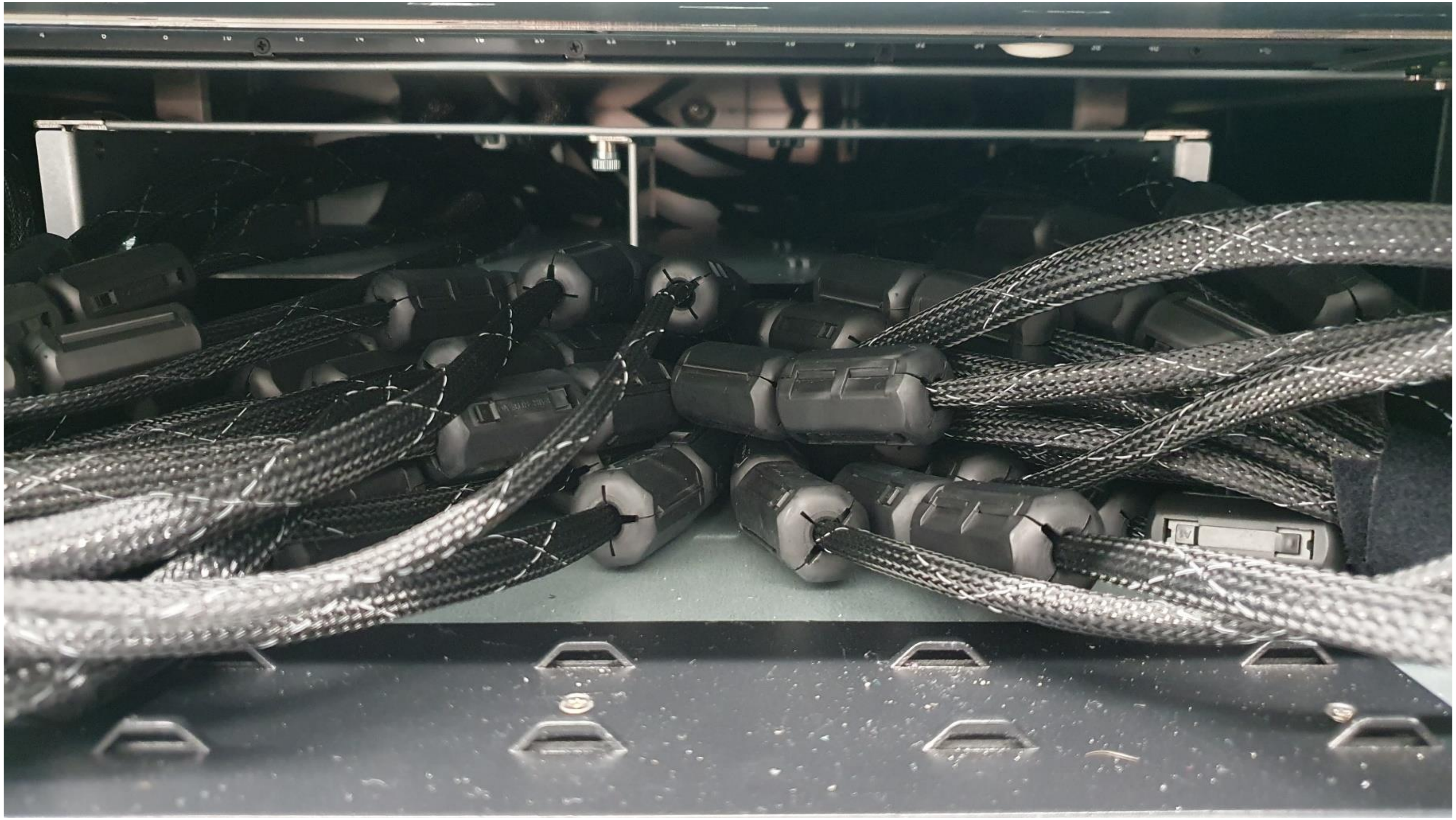
- 80,000 slots for tapes in Hamburg (10,000 remote)
- 158+ PB of climate data, increase 25 PB in 2022
- 700+ PB capacity



IB problems after start of Levante

- Unstable IB fabric
  - IB flaps for compute nodes
  - Lustre hickups
  - Lost SLURM connections / jobs
- Actions
  - Restriction on CPU frequency and disabling Turbo mode on compute nodes
  - New NVIDIA firmware on Switches (L2 / L3)
  - Ferrit Cores on IB cables

IB problems after start of Levante



Runtime deviations due to Lustre IO

- Single Jobs with toxic Lustre IO affect IB fabric
  - Causing large runtime deviations for e.g. larger MPI jobs

- Actions
  - Separate / isolate nodes to single rack / switch(es)
  - Separate Lustre IB traffic with Virtual Lanes
  - More details later from Anna Fuchs

Collaboration with ATOS/DDN

- Hot- / Cold-Pooling with NVMe/HDD
  - Lustre filesystem for testing containing 200 TiB NVME (hot) and 3 PiB with HDD (cold)
  - End of May start of nextGEMS-Hackathon for analysis of climate data on the hybrid filesystem with DDN Hot-/Cold-Pool system

    https://nextgems-h2020.eu/
  - How this could improve the workflow
    - with often accessed data on hot pool
    - Low or not accessed data on the cold pool
    - Automated two way migration between the both pools

Collaboration with ATOS/DDN (continued)

- Currently no monitoring in place by vendor which files / data was migrated and in which direction
  - Hopefully available in future release

- Creating reports about access (atime) data via scan (lipe_scan2) of MDT‘s and self written scripts
  - DKRZ provides scanned metadata (every 6 hours)
  - Max-Planck-Institute does the analysis

Reporting with LIPE

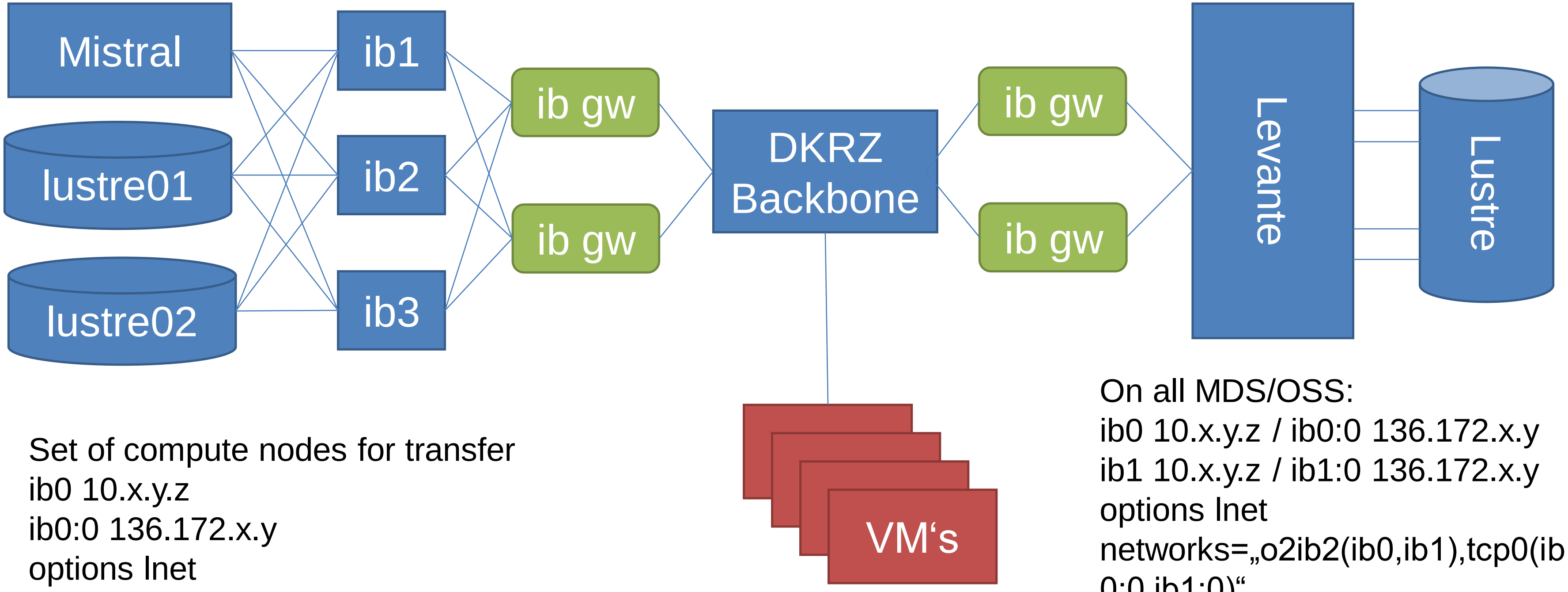
- Replacement of Robinhood
- Using the **L**ustre **I**ntegrated **P**olicy **E**ngine
  - Scanning the Metadata servers once per day
  - Analysis of the JSON output with Python script for data usage of each user in each project
    - Runtime for scan => up to 8 hours directly on the MDS/MDT
    - Runtime for analysis => 4 hours with single CPU SLURM job (currently 1.6 billion JSON lines, collected from 8 MDT‘s)
    - No extra hardware needed as for Robinhood

Reporting with LIPE

- Reports for old data
  - Reuse of data from LIPE MDT scans with modified Python script
  - Currently more then 36% of the data is not accessed for more than 1 year on the large filesystem

  - Unknown percentage of data maybe duplicated ?
  - Any experience in workflow / tools how to find out ?
  - Deduplication in Lustre, could that be possible ?

Access to Lustre without LNET Router



Set of compute nodes for transfer
ib0 10.x.y.z
ib0:0 136.172.x.y
options lnet
networks=„o2ib(ib0),tcp0(ib0:0)“

tcp0 lnet

On all MDS/OSS:
ib0 10.x.y.z / ib0:0 136.172.x.y
ib1 10.x.y.z / ib1:0 136.172.x.y
options lnet
networks=„o2ib2(ib0,ib1),tcp0(ib0:0,ib1:0)“

Access to Lustre without LNET Router

- Purpose
  - 1. Copy of approx 45 PiB of data from previous Lustre filesystems to new HPC Cluster / filesystem
  - 2. Access from VM‘s / server outside the HPC Cluster and IB fabric
- Solution
  - DDN/ATOS came up with idea of creating a scond public IP on IB interfaces of the storage servers
  - Creating a separete TCP LNET on all storage servers with these IP adresses

Lustre nodemaps

- Why nodemaps ?
  - VM's / server exposed to the internet need access to data on Lustre for different services to the Climate Community
  - Enforcement on Lustre server side for read-only access of Lustre clients
  - User / Group could be mapped to special defaults
  - Hosts could be added/removed easily with ‚lctl` command
  - Hosts could be separated to read-only / read-write access

Challenges in transfering large amounts of data

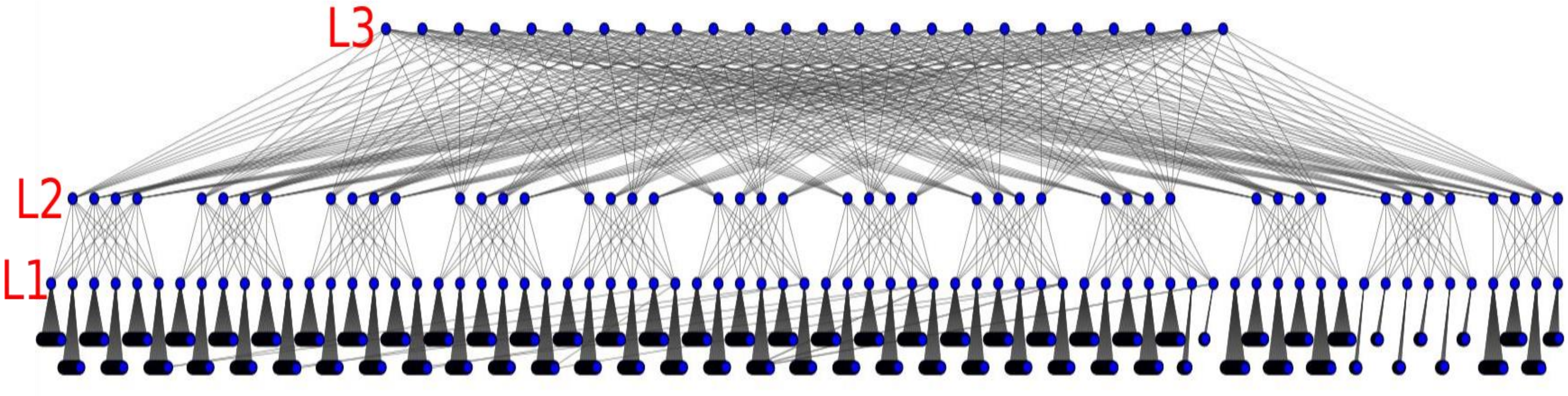
- Oops, we have project quotas
  - Each climate project has a uniqe Lustre project quota ID and limit enforcement
  - Users can‘t move (‚mv‘) data directly from one project to another. It looks like a long taking copy (‚cp‘).
  - For large movements we tweak recursively the Quota ID on the source directories with the ID of the target and it‘s a Metadata operation again .

Challenges in transfering large amounts of data

- For large transfers between filesystems we use ‚mpifileutils'
- Challenge for the nextGEMS-Hackathon was
  - Directories for a simulation run were from 1,7 TB to 480 TB
  - From 200,000 to 944,000 subdirectories per simulation run
  - From 600,000 to 35,000,000 files per simulation (between 5 and 15 MB in size as average)
  - 33 simulation runs to copy
  - How to setup this with number of nodes / MPI tasks per node / chunking / restart

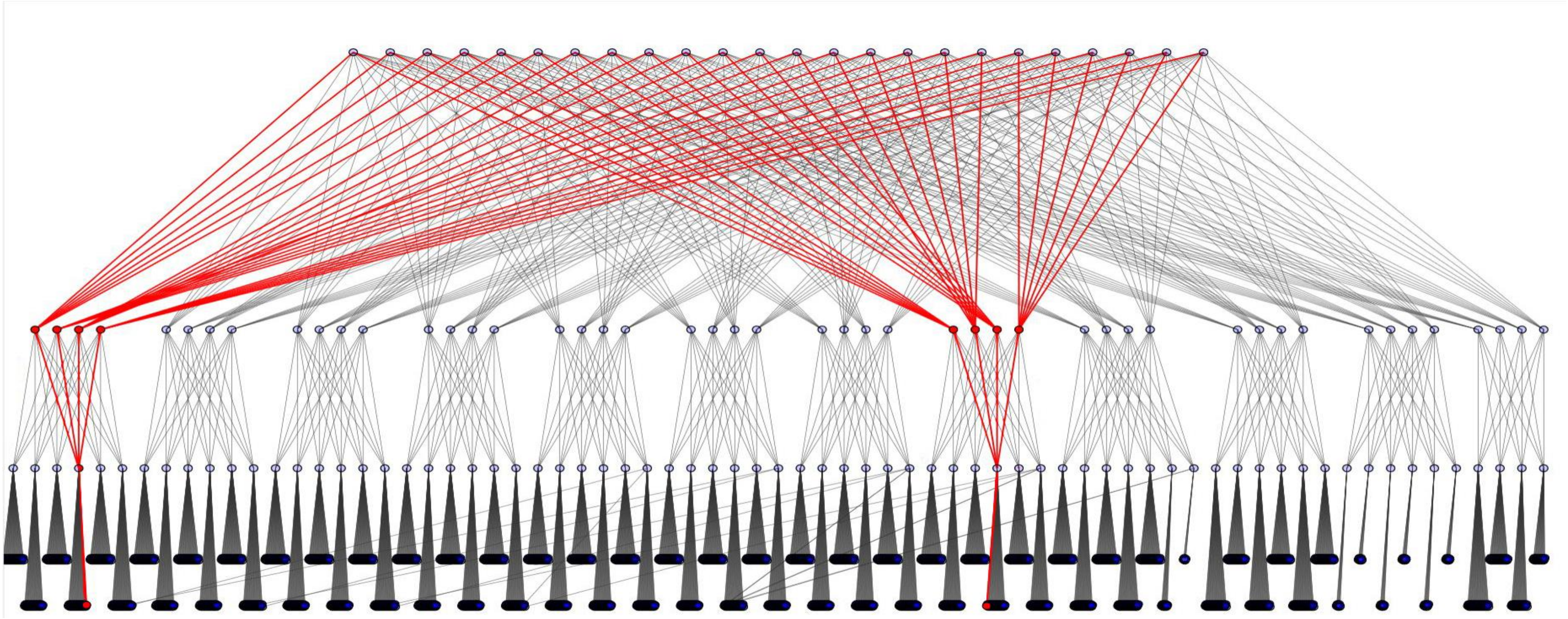
DKRZ network

- Fat trees
- Three levels of switches
- Compute nodes bundled in cells per L2 group
- Storage connected to L1 (not in picture)
- Compute node: 100 Gb/s, storage server 200 Gb/s



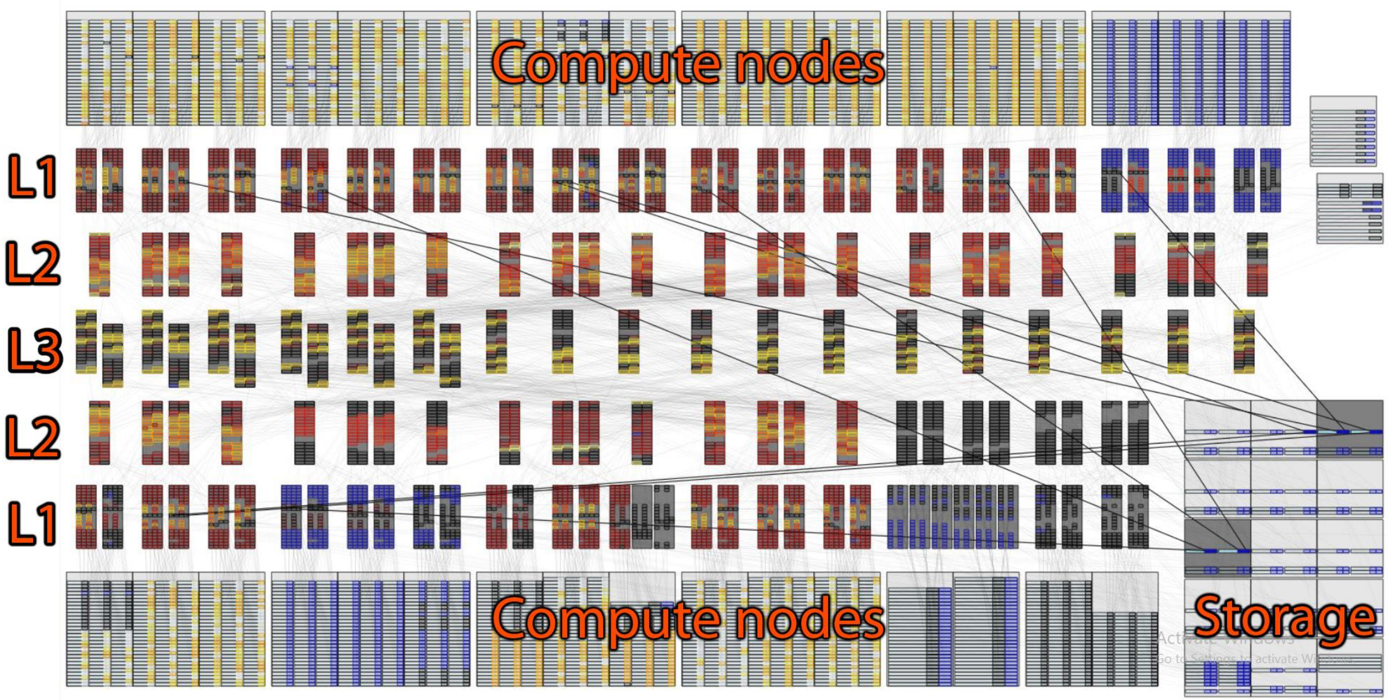
Network paths

- Communication ways between 2 compute nodes in different cells



Storage network

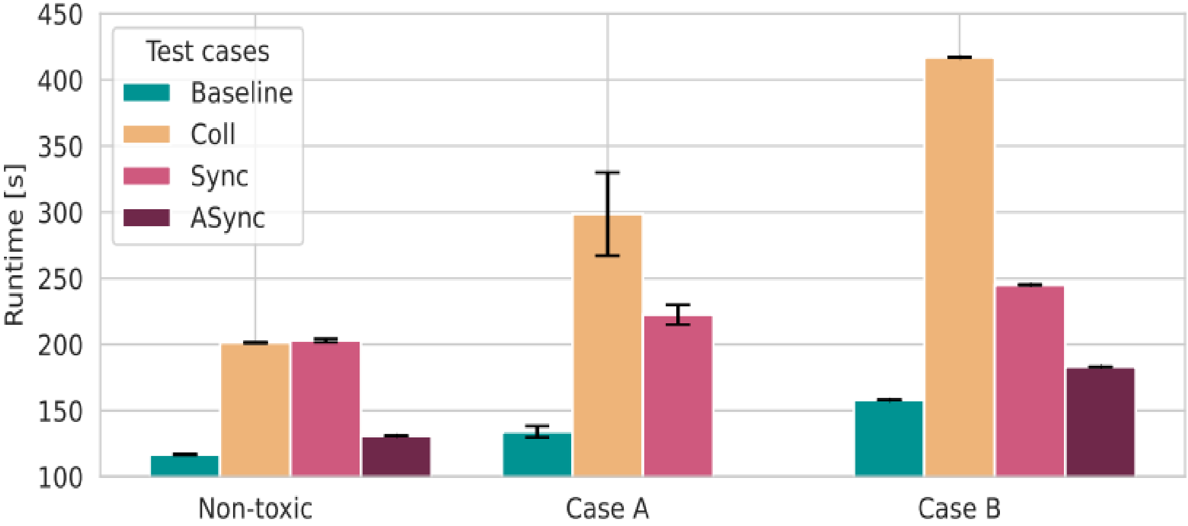
- Every storage server connected to some L1 switches
- Hops to L2 and L3 switches
  - if compute node on L1 switch does not match Sserver connection



Network congestion

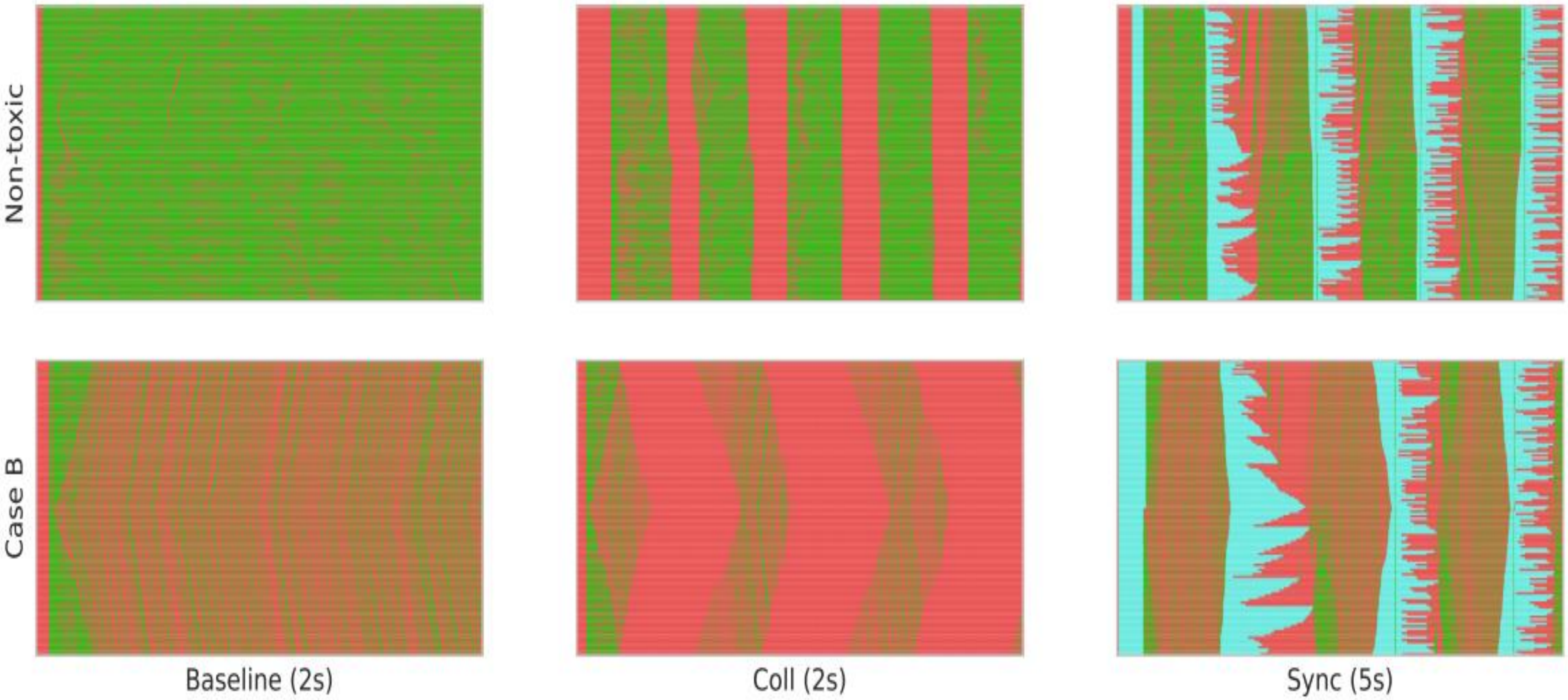
- Reference job (baseline): numio benchmark (PDE solver+); 128 procs 2 Nodes, 1 per **cell 2** and **9**
  - computation + halo-exchange (pairwise 190 kiB) , 35k iterations
  - **Coll**: + MPI collectives 265 KiB Allgrather every 70 iterations
  - **Sync/async**: MPI-IO 1 GiB every x iterations
- Background job: toxic IO (**A** and **B**), read file per process, md5sum per file, 1 node, **cell 9**

- **A**:
  - 3000 files
  - 100 MB - 7 GB each
  - stripe count 160
- **B**:
  - 365 files
  - 167 GB each
  - stripe count 16



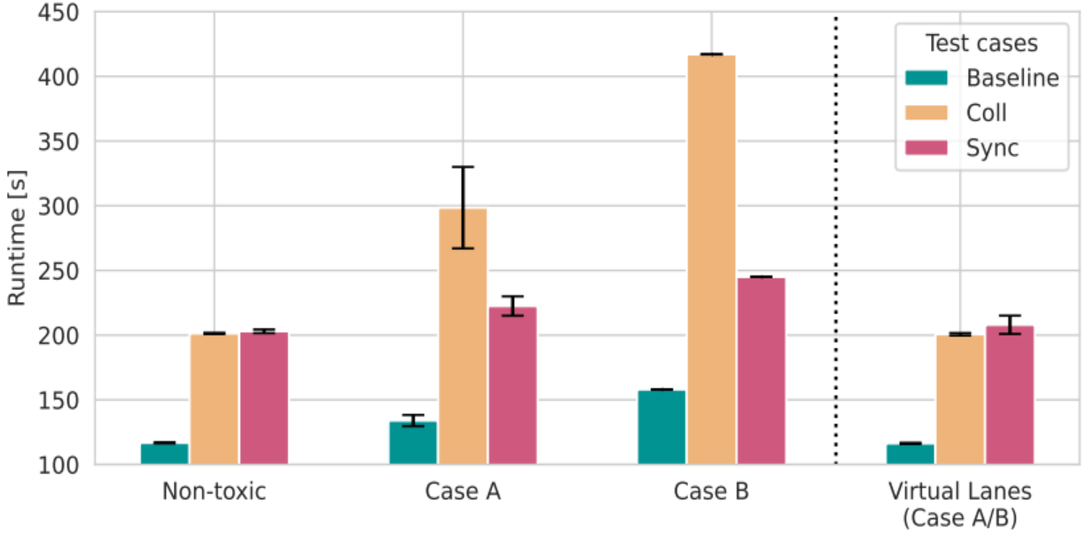
Network congestion: traces

- Latency-sensitive jobs susceptible to delays -> idle waves
- Application’s IO not latency-sensitive, therefore no substantial delay

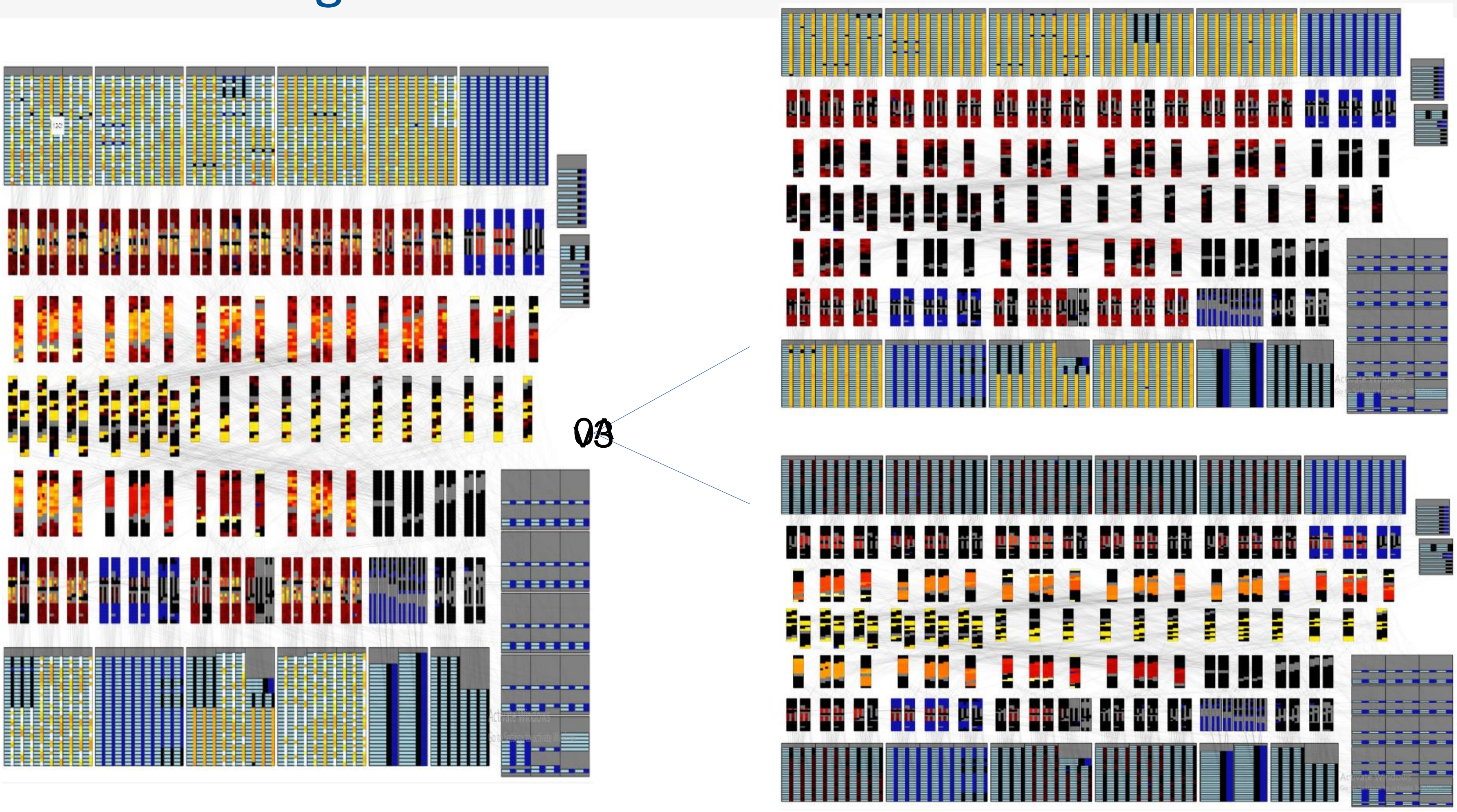


Network congestion: solution

- Virtual lanes on same IB hardware
  - Separate buffer in all network components, service level etc.
- Separate latency-sensitive traffic from large IO traffic
- Though, separation not based on assessment of requests' latency
  - Assumption: latency is critical between compute nodes, not within FS

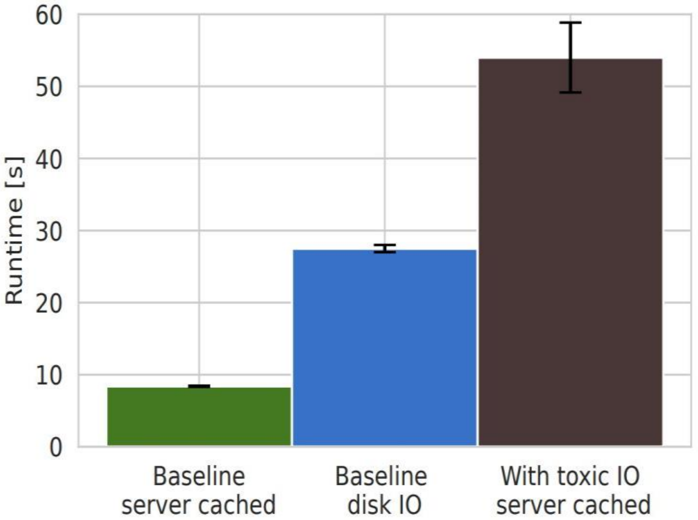


Network congestion: solution



Latency-sensitive IO traffic

- Small sparse (random) IO traffic can be latency sensitive
  - ML applications, post-processing
  - Read + checksum 4 byte every 1 MB of a 16 GB file
    - Reference job cached server-side + toxic IO B
- Solutions?
  - Reject to recognize such access patterns as HPC use cas
    - Duplicate all data for different storage systems?
  - Fix applications: improve patterns?
    - Python stack complex and transparent
  - Lustre NRS policy: TBF (token bucket filter)
    - Restricts request quantity, not quality?
  - „Virtual lanes“ inside of Lustre?



- Latency-sensitive IO reference job isolated and with toxic IO bachground

- Acks to Jannek Squar, Carsten Beyer, Jan Frederik Engels, Natalja Rakowsky, Niclas Schroeter

Questions

Questions ?

Carsten Beyer (beyer@dkrz.de)

Anna Fuchs (anna.fuchs@uni-hamburg.de)