Status update Goethe Universität

Themenbereich III

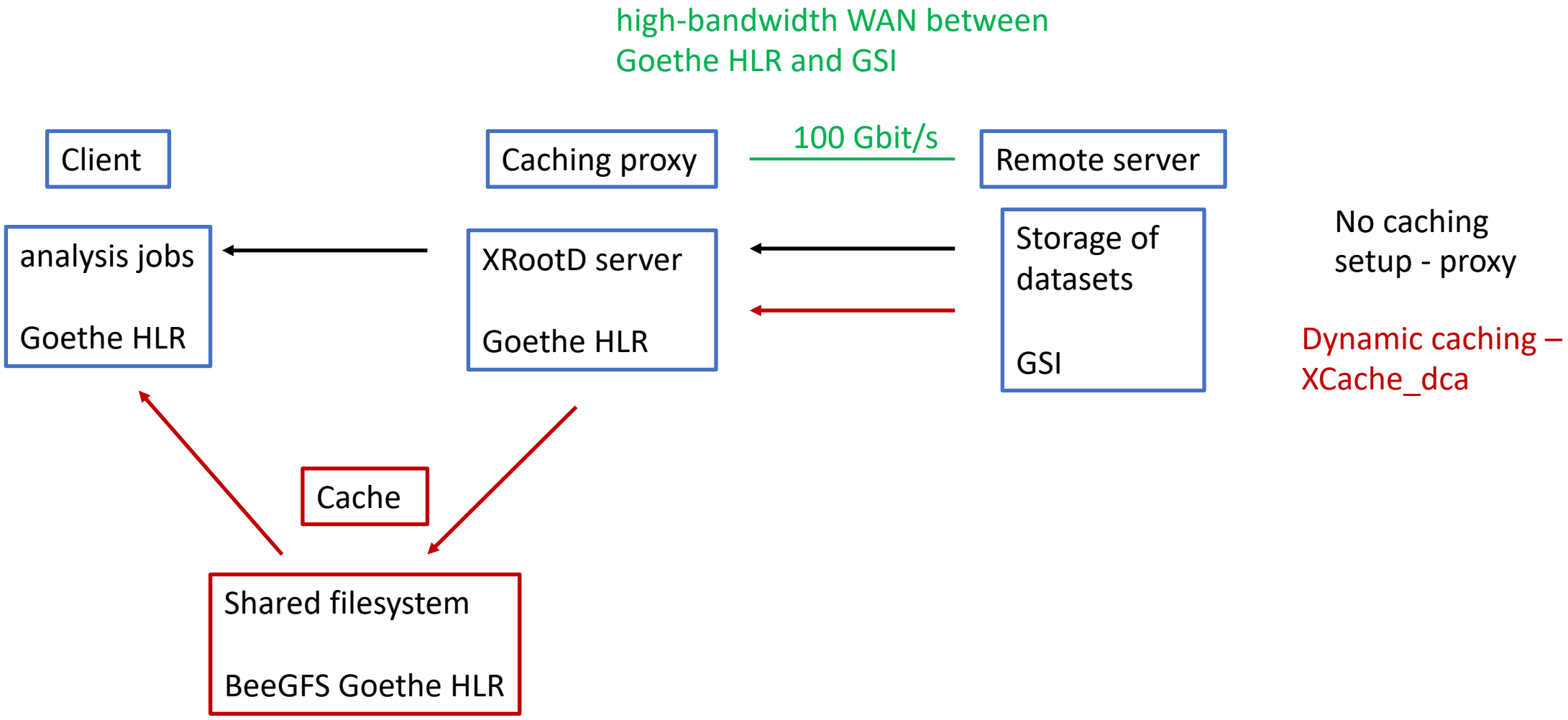
FIDIUM Meeting

21.10.2022

V. Lindenstruth, A. Redelbach

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Network and setup



Setup of caching proxy at Goethe HLR

Templated system files for relevant XRootD server setups:

- No caching (start/stop)
- XCache_dca (start/stop)

control and monitoring via systemctl

Example: systemctl status xrootd@xcache_dca.service

Setup of clients

At Goethe HLR:

- Shared file system based on BeeGFS
- Slurm scheduler for job scripts
- Analysis jobs run inside singularity container

Client nodes:
2x (20 core, 40 thread) Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz
192 GB RAM
100 Gbit/s infiniband (EDR)

Scenarios for **XCache_dca and no cache/proxy**:

- Scaling with number of clients
- Variation of numbers of tasks and nodes
- Different numbers of datasets per client

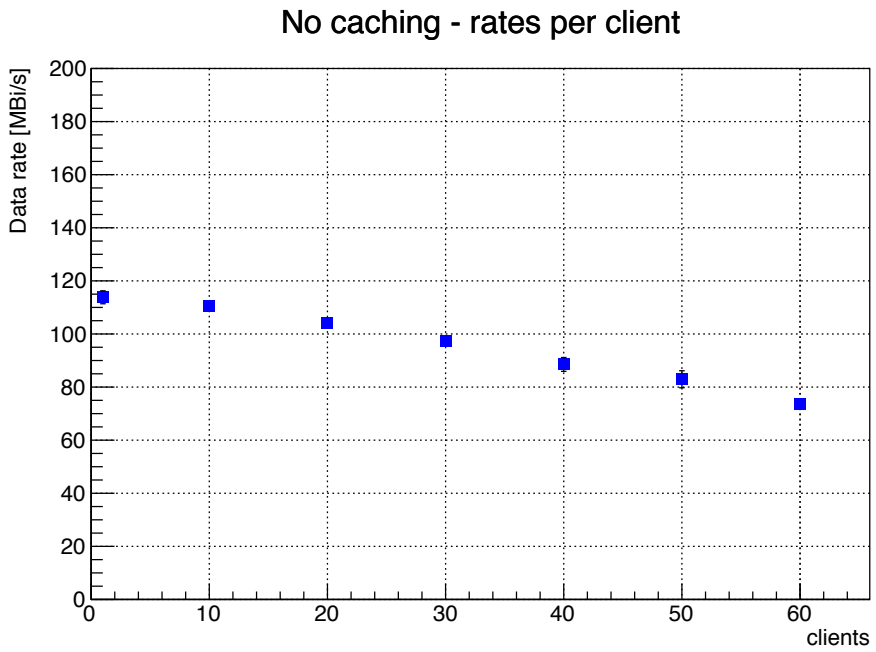
→Towards generic results beyond specific analysis

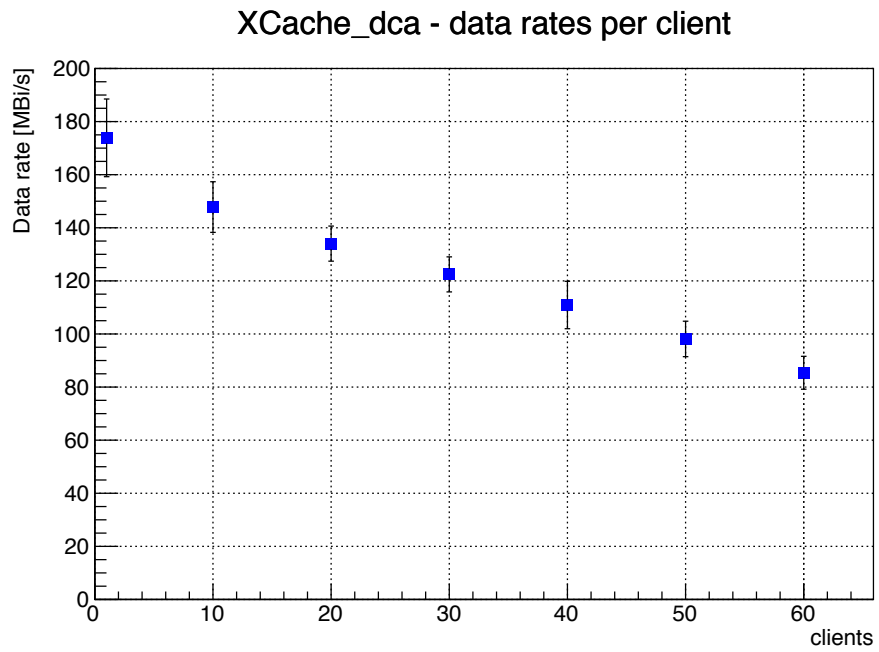
Detailed statistical **evaluations per client**

→ **To be scaled** by number of clients for total data rates

Results: Scaling with number of clients (1 to 60 tasks)

Processing of same set of ALICE AODs
Increasing number of client tasks on **one cluster node**
Evaluations of average runtime per client
→ Data rates for processing





XCache with direct cache access always faster than XRootD without caching
Actual efficiency of dynamic caching also depends on other parameters (e.g. network bandwidth)

Results: Runtimes (per client) (1 to 60 tasks)

Processing of **smaller sets of ALICE AODs**
Increasing number of client tasks on **one cluster node**
Evaluations of average runtime per client (in seconds)

15 Datasets	No caching	XCache_dca
1 client	584	385
5 clients	546	394
15 clients	595	473
30 clients	662	486
60 clients	883	781

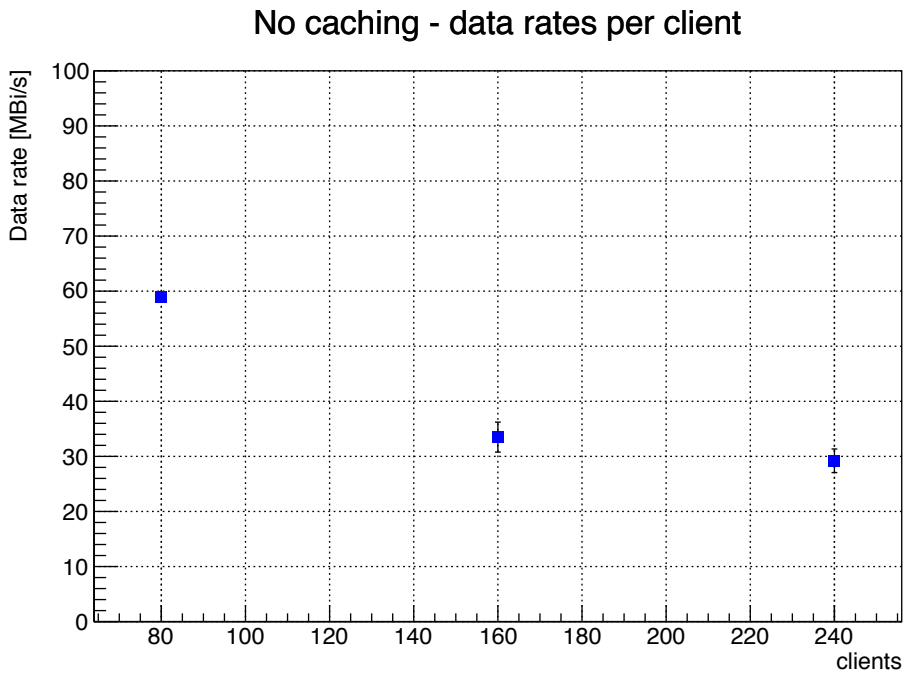
5 Datasets	No caching	XCache_dca
1 client	212	159
5 clients	200	152
15 clients	219	167
30 clients	244	182
60 clients	327	298

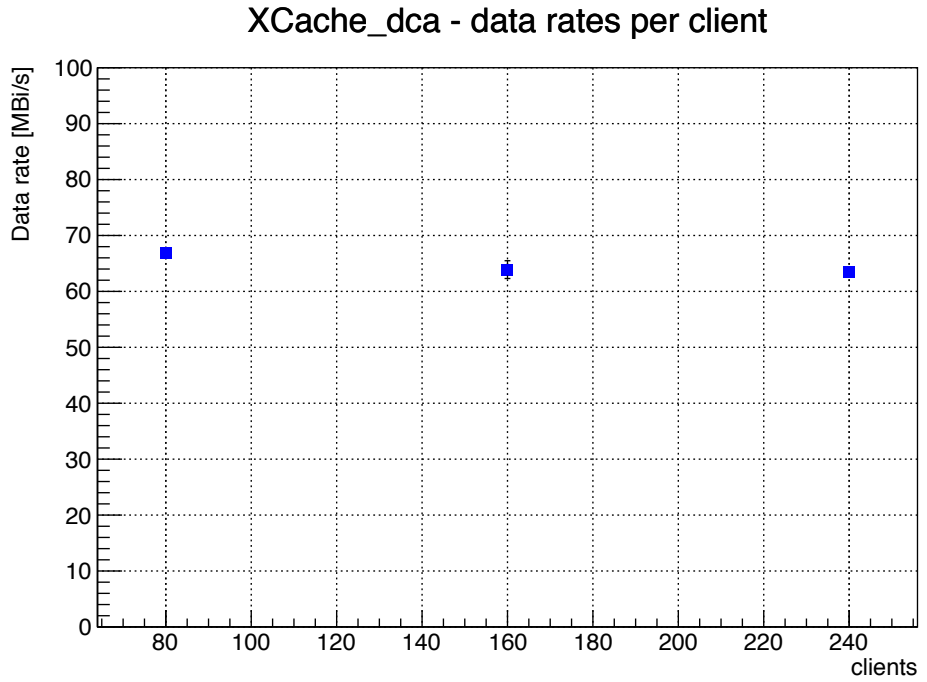
1 Dataset	No caching	XCache_dca
1 client	56	55
5 clients	58	57
15 clients	63	60
30 clients	66	64
60 clients	95	89

For smaller (size of) datasets: Runtimes needed for overhead become more relevant.

Results: Scaling with number of clients (80 to 240 tasks)

Processing of same set of ALICE AODs
Maximum number of client tasks on **1, 2, 3 cluster nodes**
Evaluations of average runtime per client
→ Data rates for processing





Runtimes ~constant for more client jobs (executed on other nodes)
Advantages for dynamic caching based on shared storage for increasing number of nodes
→ Significant potential for efficient usage of resources

Summary and next steps

Setup for high bandwidth connection/caching for efficient data transfer/analysis between Frankfurt and GSI
Focus on XCache with direct Cache Access as dynamic caching method (see also Area II update)

Milestone: Funktionale Tests (hinsichtlich Performanz und Skalierung) von Workflows mit dynamischem Caching

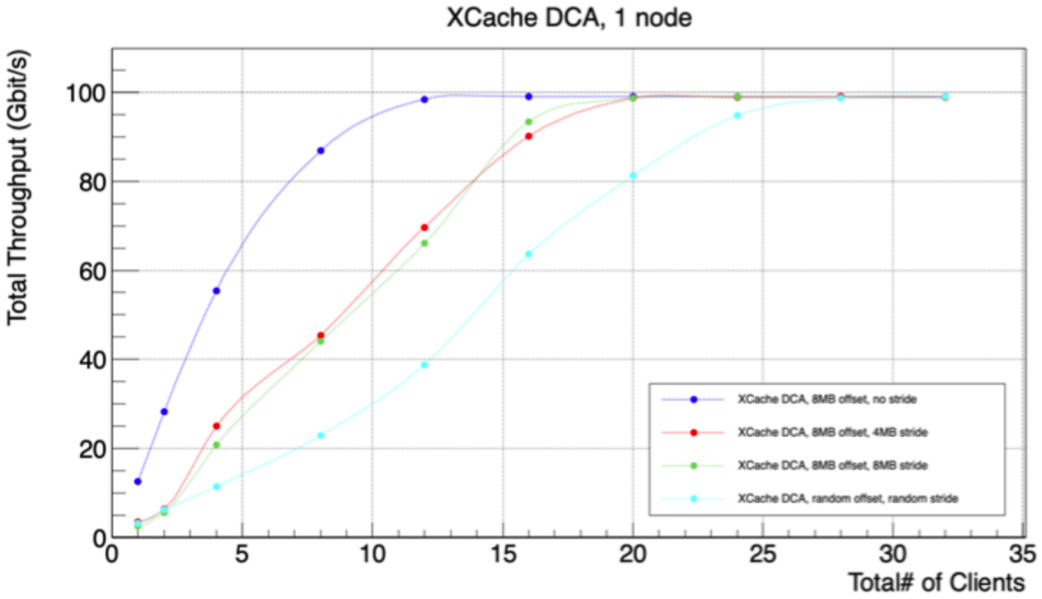
Extended previous measurements for performances:
→ Statistics
→ Granularity in number of clients
→ Number of AOD datasets

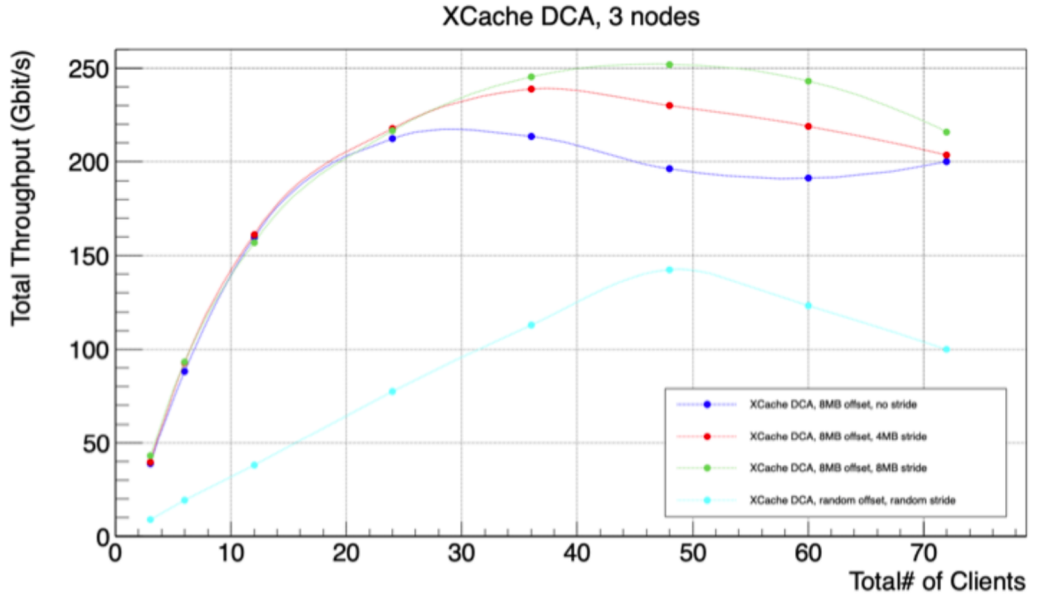
Towards better coverage/understanding of multi-dimensional parameter space:
- Hardware resources for caching
- Datasets for analysis (total throughput higher for larger datasets)
- Numbers of clients and nodes for analysis
- Fraction of overhead for runtimes
- Network bandwidths

→ Scaling of runtimes can be optimized for different parameters
→ Basis of future integration into efficient workflows in data lakes

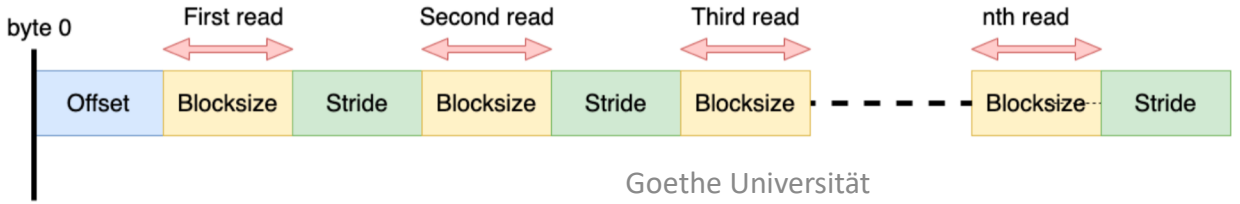
BACKUP

Measurements of total throughput





Clients running from 1 node saturate at the node bandwidth (100 Gbit/s).
Clients running from 3 nodes reach up to file system bandwidth (BeeGFS file system limits).
4 different conditions of offset and stride have been used for these benchmarks,



Blocksize for file read: 8MiB