

IDT-UM Status Report

Volker Lindenstruth, Kilian Schwarz, Paul-Niklas
Kramp, Dirk Hutter, **Serhat Atay**

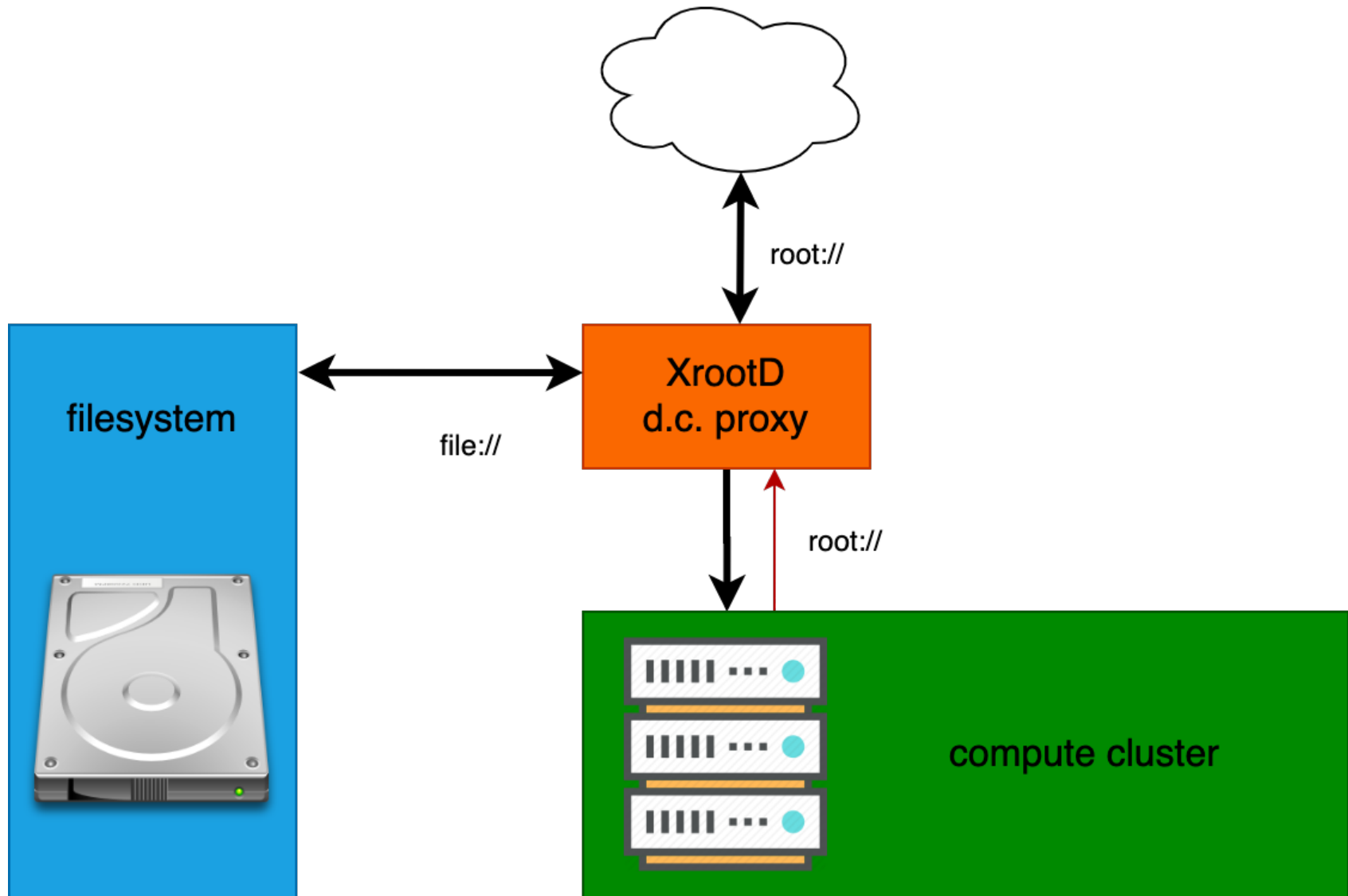
Goethe University, Frankfurt

GSI, Darmstadt

22 September 2020

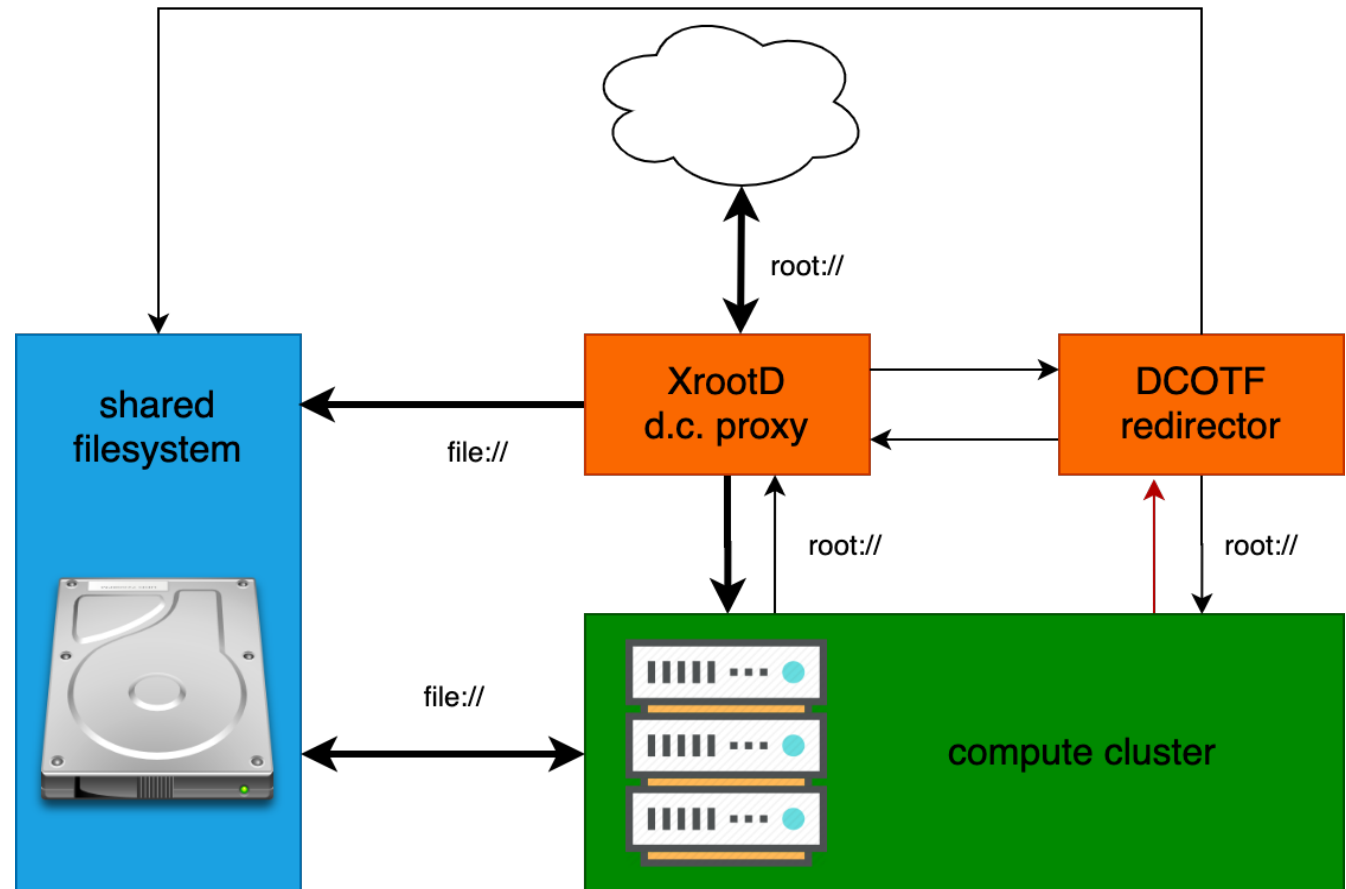


XCache (Standalone)



Disk-Caching-on-the-Fly*

- Provides a local redirection to the shared filesystem for XCache when the requested file is in the cache
 - Based on changes on both client and server side of XrootD made by GSI (LocalFileHandler)
 - Available from XrootD version >4.8.0**
 - A Local Redirect plugin developed by GSI to make use of LocalFileHandler in a special redirector
 - Available from June 2017***



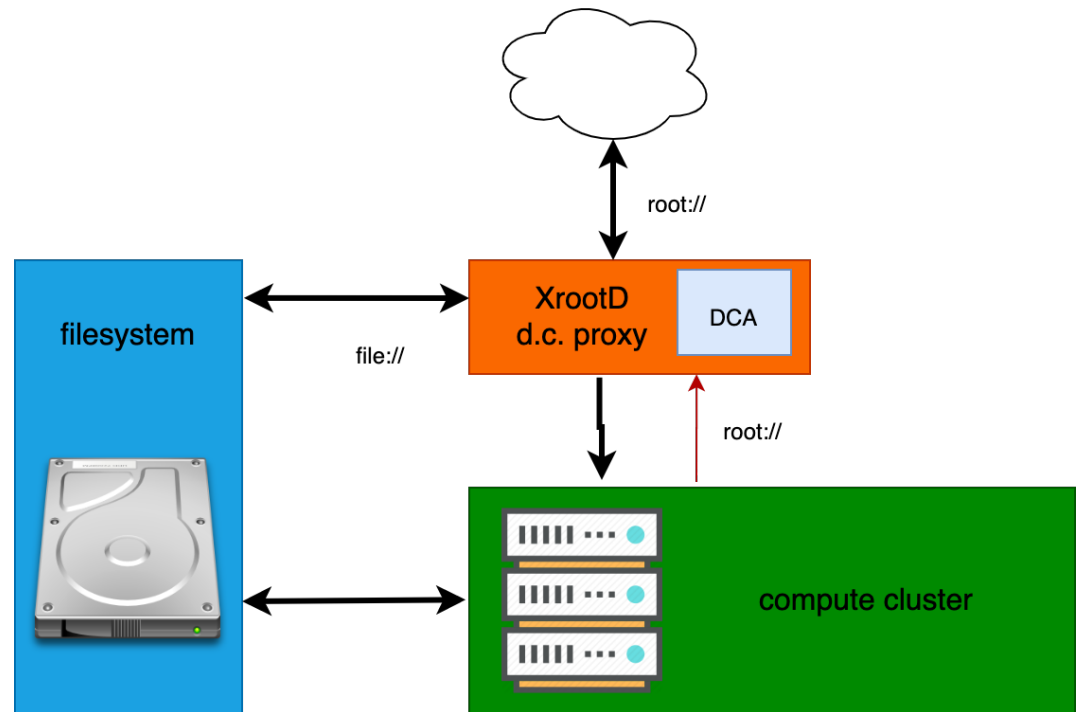
* <https://git.gsi.de/atay/xrootd-disk-caching-on-the-fly>

** Related paper: 10.1051/epjconf/201921404005

*** <https://github.com/pkramp/RedirPlugin/tree/kit-proj>

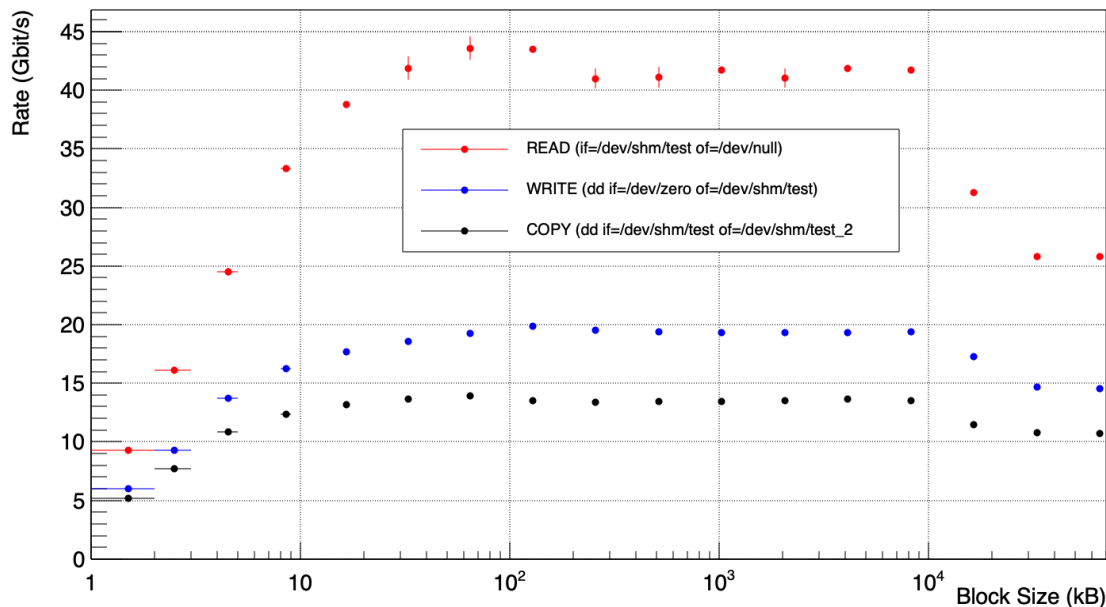
XCache direct cache access (DCA)

- Provides direct cache access to XCache when the requested file is fully cached (same as in DCOTF)
 - Based on changes on both client and server side of XrootD made by GSI (LocalFileHandler)
 - No requirement for a special redirector
 - Just an additional directive in disk caching proxy to make use of LocalFileHandler
 - Developed by a (if I remember correctly) USA institute
 - Available from XrootD v4.10-rc1 (committed to XrootD github repository on March 2019)
- Marking an access record to avoid purging of the file (missing on DCOTF)
- Saving cached data files with UNIX permission to allow direct access for other users in DCA mode (missing on DCOTF)



RAM disk Benchmark

- RAM disk benchmark as a baseline (on tmpfs) for generic read/write/copy (plot on left)
- **DCOTF** vs. **XCache** vs. **XCache Direct Cache Access** (Table on right) (XrootD copy -xrdcp)
 - In a single machine
 - No network activity
 - Storage on RAM disk (/dev/shm)



Platform	Copy Rate (Gbit/s)
DCOTF (v4.12.x)	16.16
XCache (v4.12.x)	13.22
XCache dca (v4.12.x)	16.18
XCache (v5.0-rc)	13.22
XCache dca (v5.0-rc)	13.29

Further Tests

- Tests for Disk-Caching-on-the-Fly (DCOTF) (in progress)
 - A sample SLURM job at Goethe-HLR to check the initialization of the DCOTF and the read rates
 - Just to read and cache data from GSI
 - To read data from the cache
 - A sample Grid job at Goethe-HLR
 - Submitted manually to the local Goethe-HLR's batch system
 - To analyze data using the DCOTF
 - The comparison of performance of the Grid job with and w/o DCOTF
- Test for singularity container (accomplished)
 - Installation of singularity package at Goethe-HLR
 - SLURM jobs running inside a singularity container at Goethe-HLR
- Combined Tests (in progress)
 - The sample Grid job analyzing data using DCOTF inside a singularity container at Goethe-HLR
- Detailed results and documentation of the planned tests

Future plans

- Setting up a new server to be added to the Goethe-HLR in Frankfurt with root privileges
 - 120Gbit/s private link to GSI (to be reestablished)
 - Already established a 10Gbit/s link (more links to be established to aggregate the bandwidth)
 - 100Gbit/s link to the cluster filesystem (unshared by other users)
 - Few TBs of NVMe SSD as a local filesystem in case the cluster filesystem is slow. (The cluster filesystem is always shared)
- Improvement and debugging of XrootD configuration files with the feedback from performance tests
- Development on the local redirect plugin to make use of XCache features, i.e.:
 - Caching blocks of files instead of complete files -not implemented in the local redirect plugin
 - Status of data in cache (parsing .cinfo files) -not implemented in the local redirect plugin
 - Managing UNIX permissions of cached data files to allow direct access
- Submission of local redirect plugin to XrootD core package after debugging and aforementioned improvements
- Investigation of performance differences of DCOTF vs. XCache vs. XCache Direct Cache Access

Milestones

- A1:
 - M1 (June 30,2019) (singularity container at GSI, disk image management system)
 - M2 (June 30, 2020) (singularity container at Goethe-HLR)
- A2:
 - M1 (June 30, 2018) (availability of all tools for “disk caching on the fly”)
 - M2 (Dec 31 ,2019: postponed June 30, 2020 in agreement with BMBF) (first prototype of disk caching on the fly at Goethe-HLR)
 - M3 (Dec 31, 2020) (disk-caching-on-the-fly in Freiburg with CMS experiment)
- B1:
 - M1 (Dec 31, 2019) (packaging and documentation for disk caching on the fly)
 - M2 (June 30, 2020: postponed Dec 31, 2020 in agreement with BMBF) (detailed performance measurement and scalability of disk-caching-on-the-fly)
 - M3 (Dec 31, 2020) (simplified installation and documentation including the container system)
- B4
 - M1 (June 30, 2020: postponed Dec 31, 2020 in agreement with BMBF) (combined usage of jobs in singularity containers at Goethe-HLR while reading data to be analyzed from the disk-caching-on-the-fly-at-Goethe-HLR)
- Optional Milestone G1: comparison of XCache and DCOTF in terms of performance
- Outlook: on track in terms of milestones

Legend: **accomplished**, **upcoming**, **providing support**

Thanks