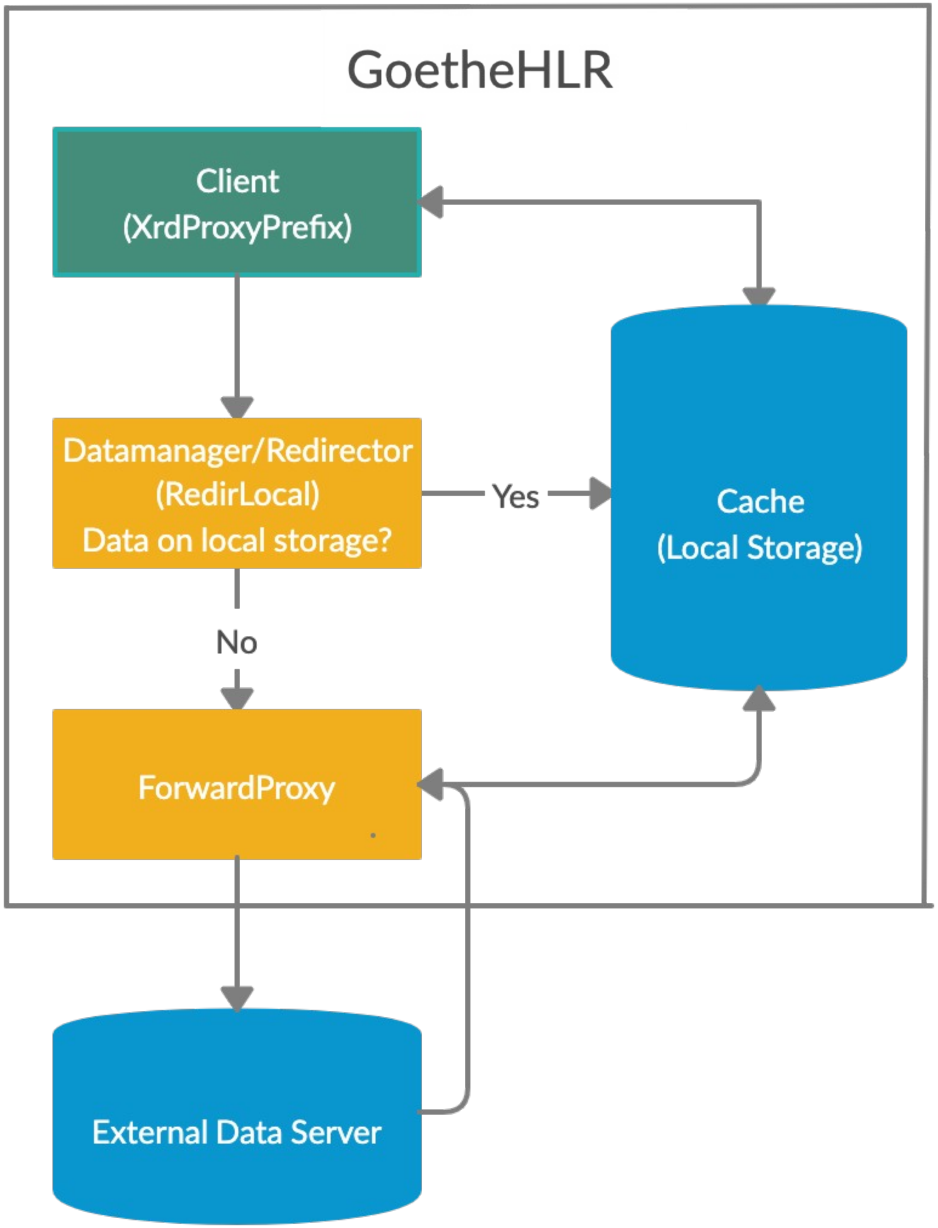
Disk-Caching-On-The-Fly

Volker Lindenstruth, Kilian Schwarz, Paul-Niklas Kramp, **Serhat Atay**
Goethe University Frankfurt
GSI Darmstadt

2 April 2020

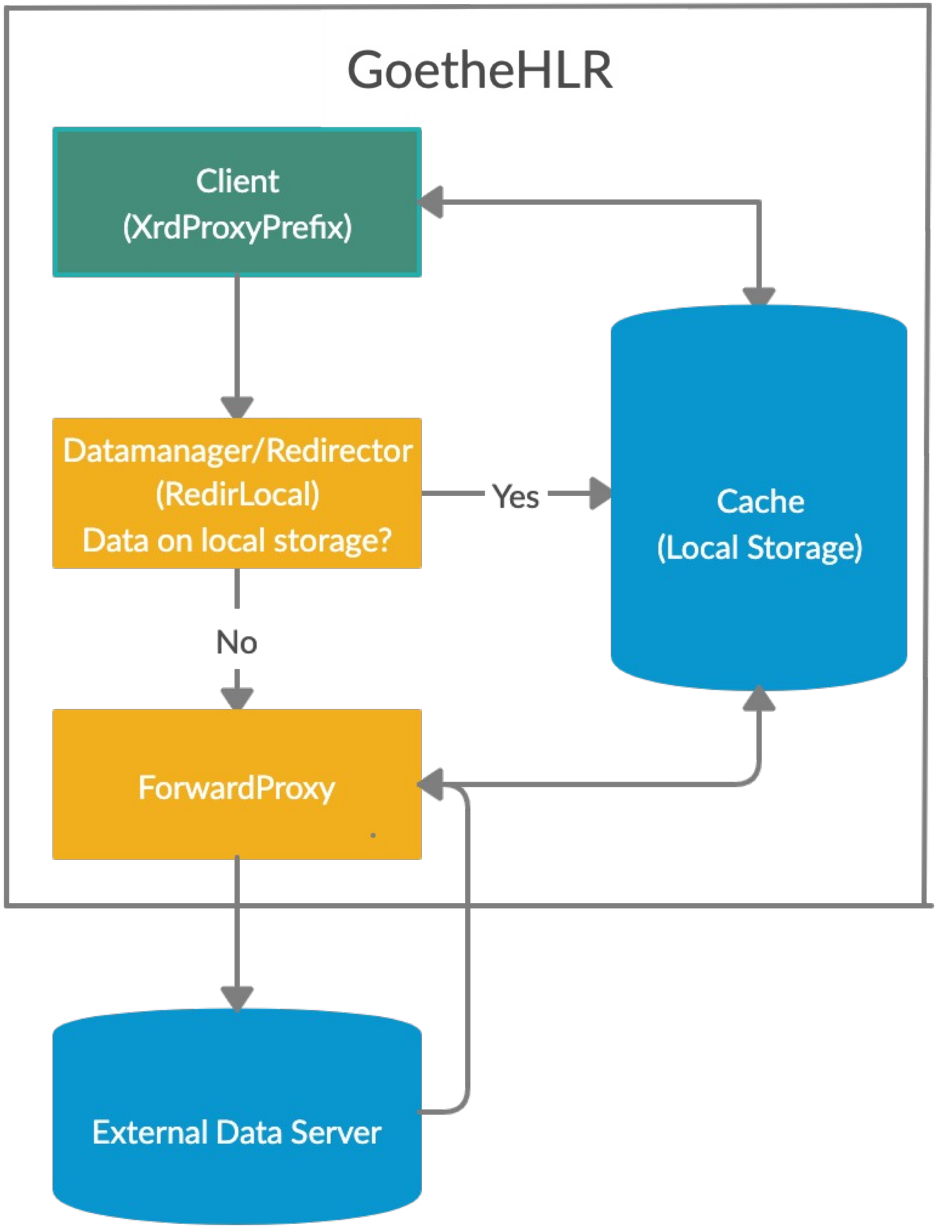
Disk-caching-on-the-fly

- Client
  - The node where data transfer command is executed
  - Requirement of the deployment of XrdProxyPrefix plugin by the user
  - Prepending the address of the datamanager to direct the request to the datamanager
- Datamanager/Redirector
  - Checking of the data on local storage (cache directory)
    - If yes: replace the address with the address of the data in local storage
    - If no: prepends the address of forwardproxy to retrieve data
- Forwardproxy
  - Connection to the external data server
  - Returning the data to the client and simultaneous caching to the cache directory



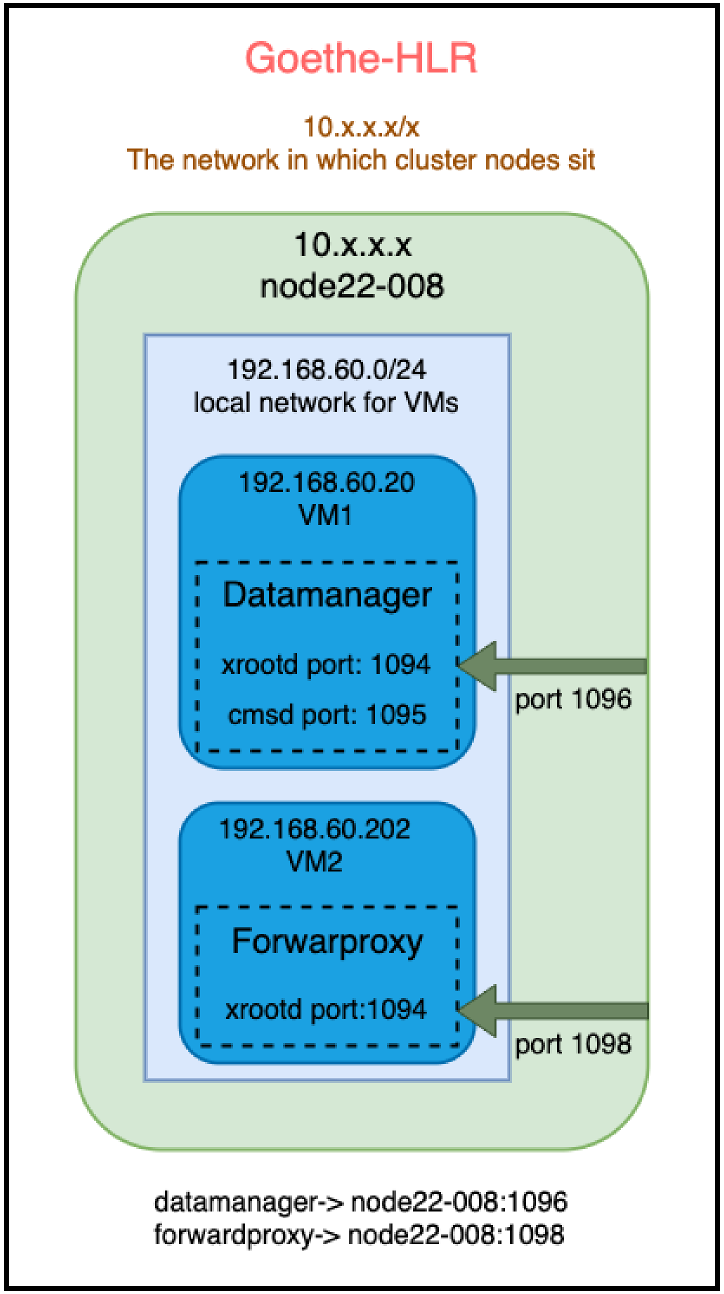
Disk-caching-on-the-fly

- Requirements
  - Plugins (already available)
    - XrdProxyPrefix plugin
    - RedirLocal plugin
  - Cache directory
    - Accessible both datamanager and forwardproxy
    - Accessible by the nodes who is employed to use disk-caching-on-the-fly
  - (At least) a node on the cluster to host XrootD servers (datamanager and forwardproxy)
    - Option 1: XrootD servers on Virtual Machines
    - Option 2: Native XrootD servers on the node
- Test tools
  - A vagrant setup with four VMs available
    - A client to request data
    - Datamanager and forwardproxy
    - An external dataserver where data will be fetched from



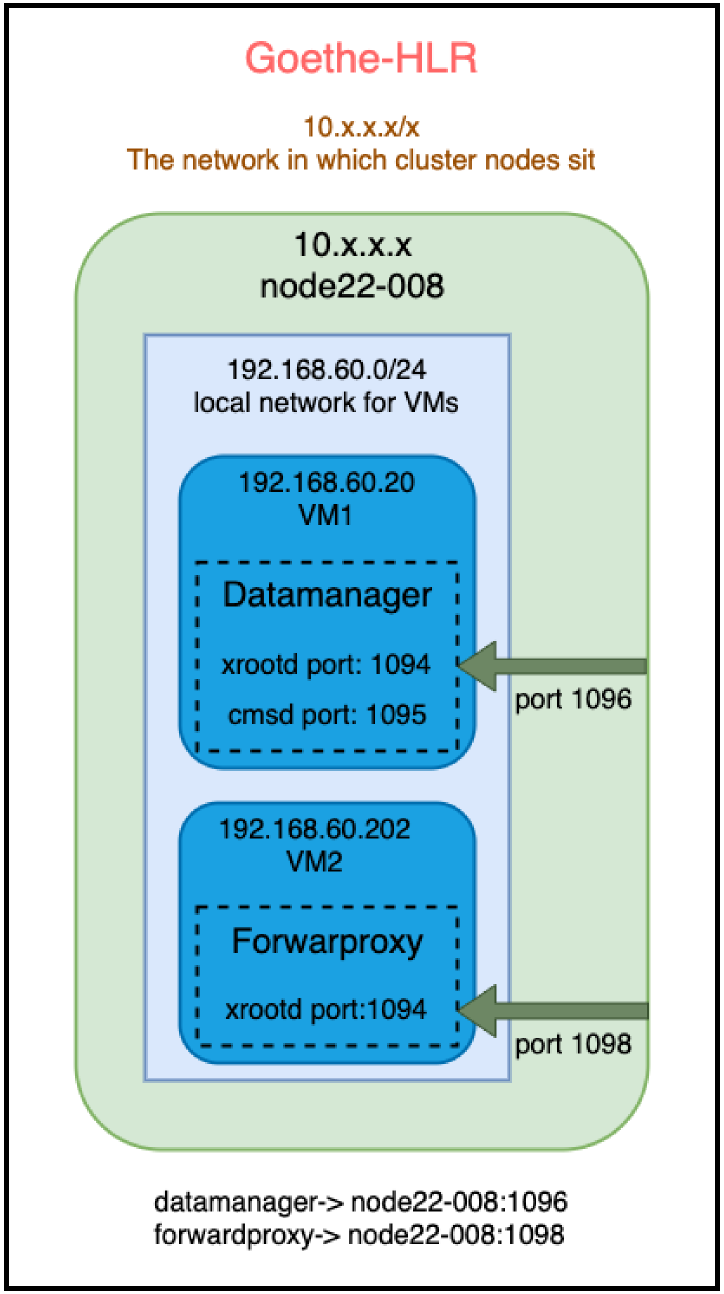
Disk-Caching-On-The-Fly on VMs (Option 1)

- Two virtual machines on the node
  - Requirement of a VM software on the node
  - Private IP addresses on private network
    - Requirement of port forwarding to be reached by the node and by other nodes
  - Synching cache directory to share filesystem
  - Option 1a: Two virtual machine with IP addresses from DHCP server of the cluster network (bridged mode)
    - No port forwarding
    - Seen as nodes on the cluster network with their own IP
    - Communication via host machine network card
    - Disadvantage: requirement of two IP entries



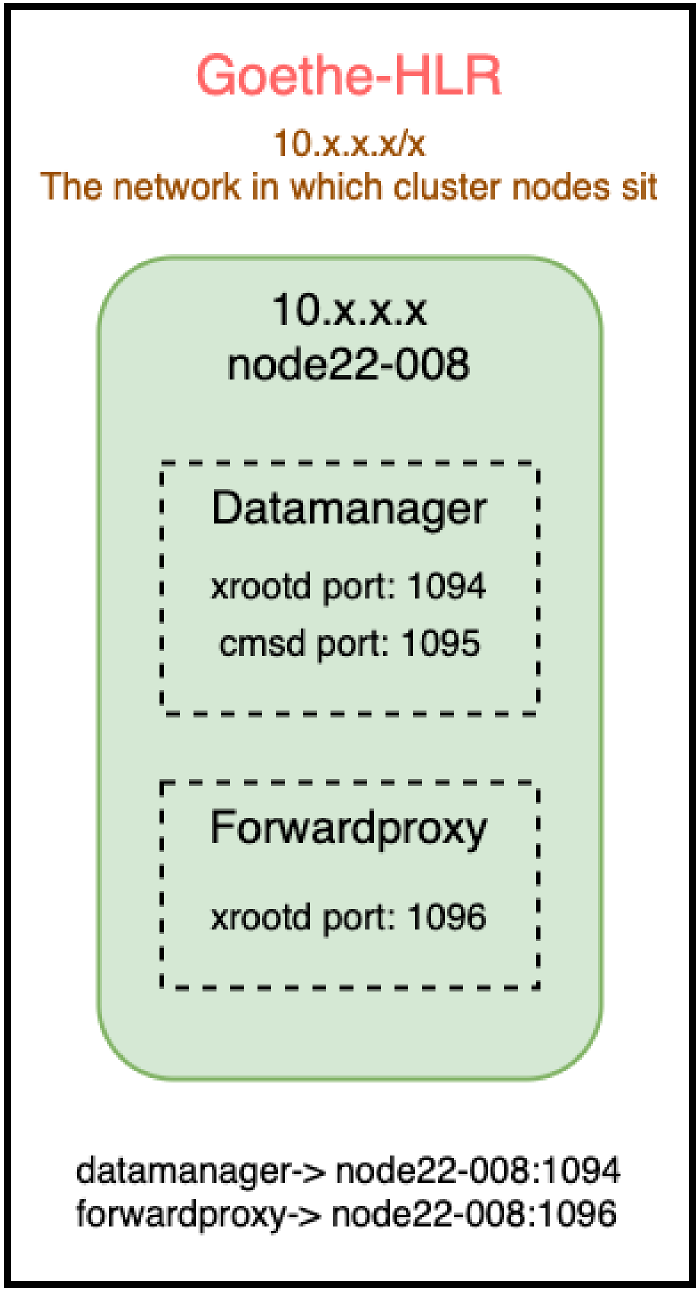
Prototype of disk-caching-on-the-fly at Goethe-HLR

- The requirement of the allocation of node22-008
  - Automatic shut down of VMs after allocation is revoked or expired
  - Accessible only by the user who allocates (until the end of allocation)
    - VMs has to run as a service (for all cluster users) always on the node without the requirement of any allocation (a dedicated node for VMs) -will be discussed with cluster admins
- The requirement of a larger cache directory
  - VMs and cache directory (at the moment both under my user space)
    - VMs must be migrated to a larger space -requested
    - Cache directory must be enlarged and be accessible for all cluster users -requested
- Effects of synching cache directories on the cluster and on VMs
  - Virtualbox suffers from synching larger files (potential performance issue of caching)



Native XrootD servers (Option 2)

- Datamanager and forwardproxy in a single node
  - Two XrootD servers in one machine with different port numbers
- Advantages
  - Much more simple to set up (via ansible from git repository)
  - No synching of files required (between cache directory on cluster and on VMs)
- Disadvantages
  - Requirement of root privileges to set it up (especially starting xrootd and cmsd services)
  - Dedicated node (this also applies to VM case)



Future plans

- Xcache installation at Goethe-HLR
- Use a data server at GSI (Darmstadt) to be accessed from Goethe-HLR (Frankfurt)
  - Performance comparison between disk-caching-on-the-fly and Xcache
  - Detailed performance measurement and scalability of disk-caching-on-the-fly
- 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

Summary

- Plugins for disk-caching-on-the-fly
  - XrdForwardProxy -already exists at GSI GitLab repository
  - RedirLocal -already exists at GSI GitLab repository
- Vagrant setup
  - with four virtual machines on a private network to test “disk caching on the fly” (client, data manager, forward proxy, external data server)
- Working first prototype of disk-caching-on-the-fly at Goethe-HLR
  - Few issues to be solved -i.e. accessibility by multiple user
  - Native installation of XrootD servers for better performance
- Xcache installation to compare performances
- Performance test and scalability of disk-caching-on-the-fly with direct WAN access to GSI

Thanks