Distributed cache with XRootD: designing, implementing and testing the model

Diego Ciangottini
on behalf of the CMS and INFN cache team

Introduction

Scope: developing a distributed disk cache model with XRootD

- the software stack is already solid and widely used adopted in WLCG and beyond
  - reduce the development and focus on a model definition and testing/evaluation
- The main objectives are:
  - implement a WLCG unmanaged storage layer
  - develop automated deployment for opportunistic/ephemeral computing
  - study AI based algos for management and optimization of data access in complex distributed setup

Activity summary

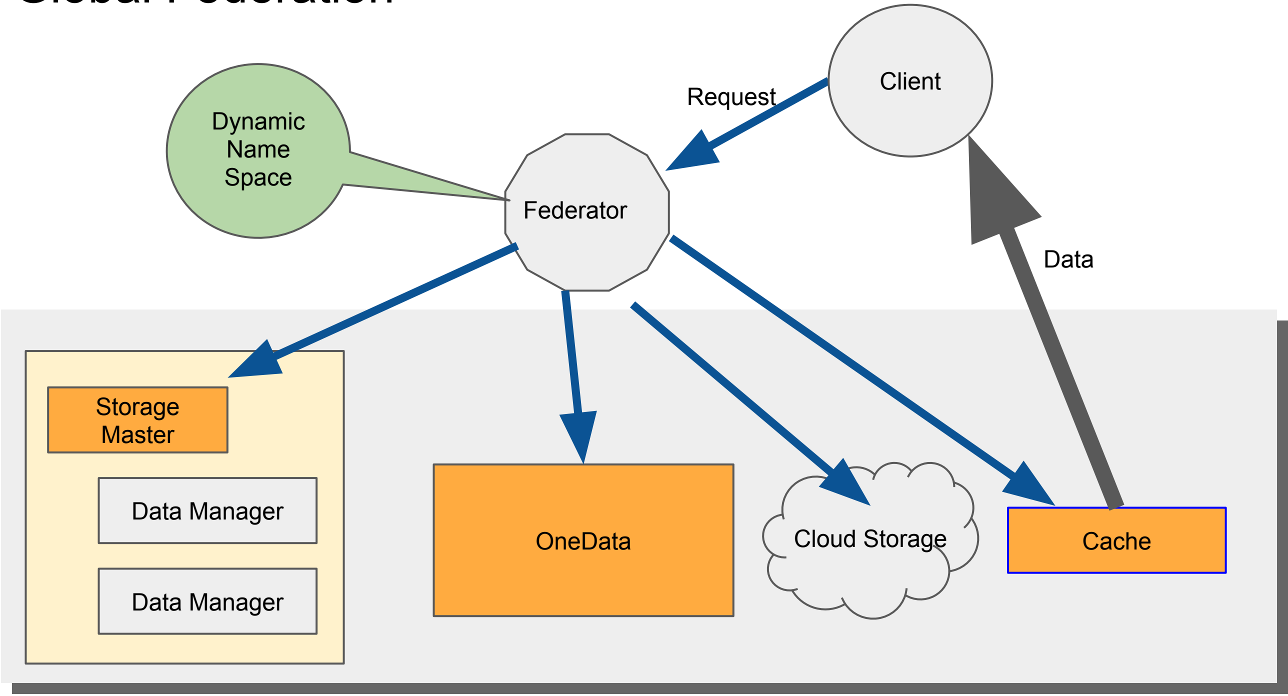
Two different cases have been studied corresponding to two reference scenarios of the WP:

- Case 1: usage of caches in a Global Federation
- Case 2: usage of Standalone caches

Case 1: Global Federation

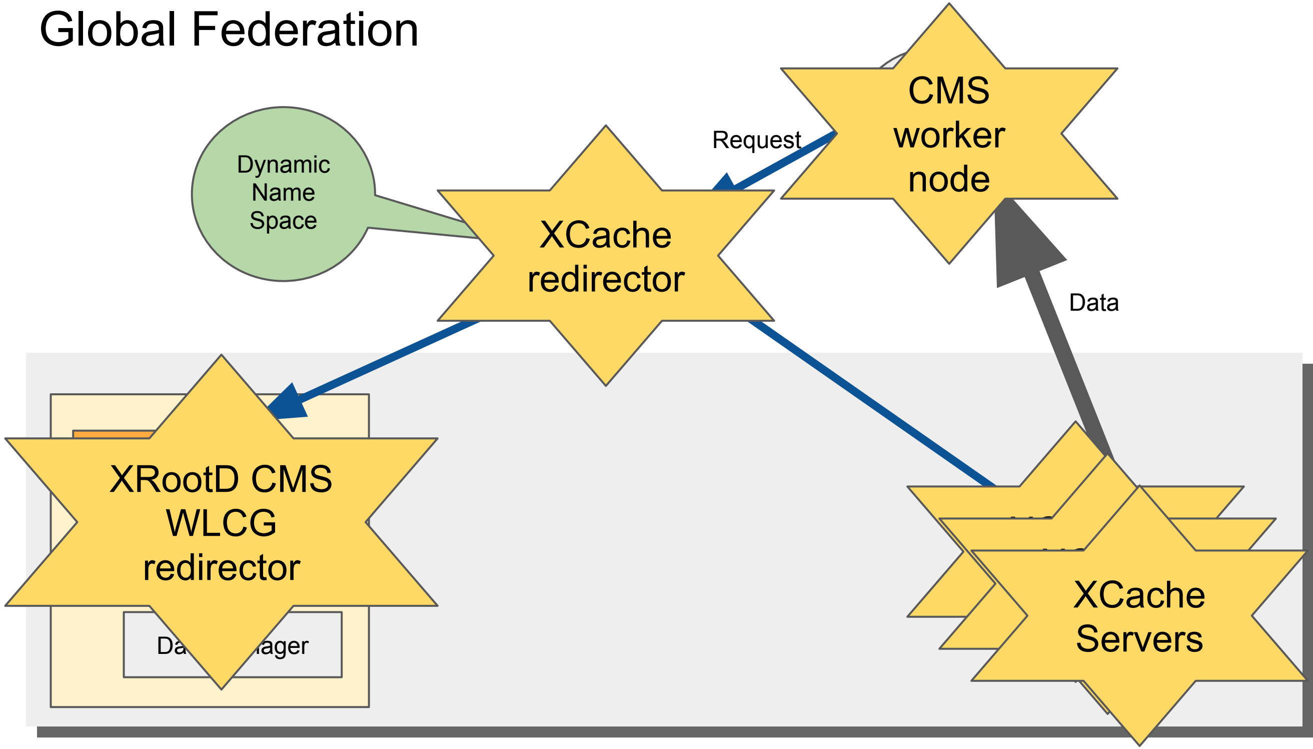
Using XRootD stack this XDC reference scenario has been integrated within the CMS computing model seamlessly.

Global Federation



Case 1: Global Federation

Schema of the XRootD equivalent scenario deployed



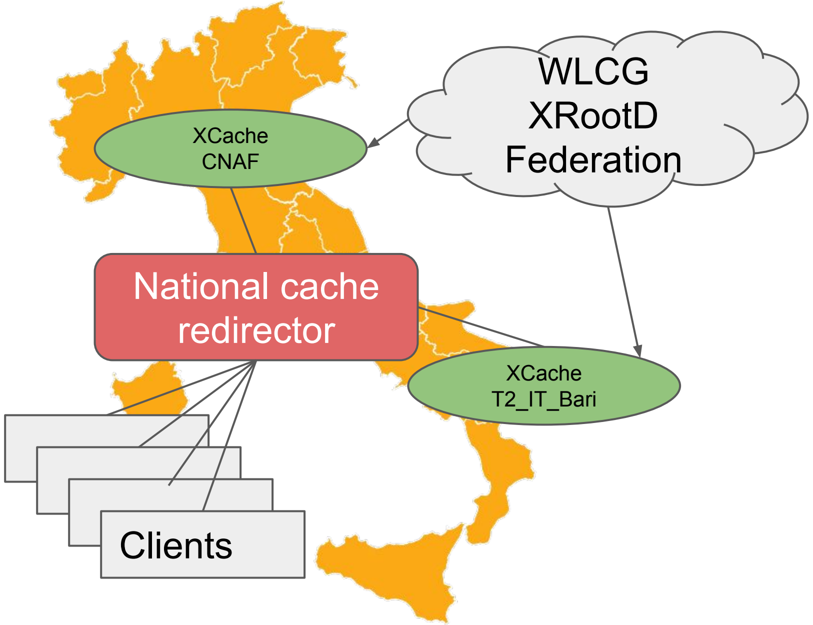
Setup - status

Details on the setup in Backup

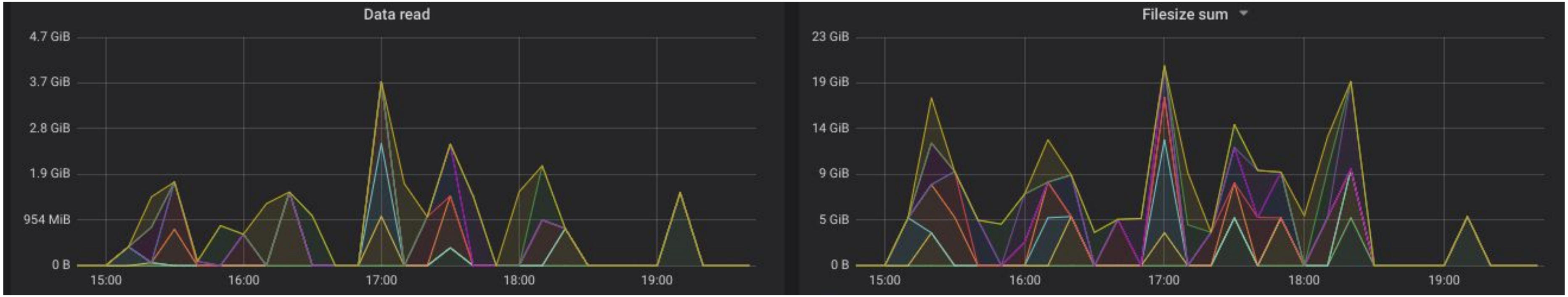
- CMS compliant and automated deployment on bare metal through Ansible:
  - https://xcache.readthedocs.io/en/latest/automated-grid.html
  - few commands needed to bring up the server with integrated system monitoring (metricbeat)

Current setup:

- CNAF XCache redirector federating:
  - CNAF XCache server
  - T2 Bari XCache server
- Currently redirecting part of the CMS analysis workflows to contact National redirector
  - based on dataset name requested



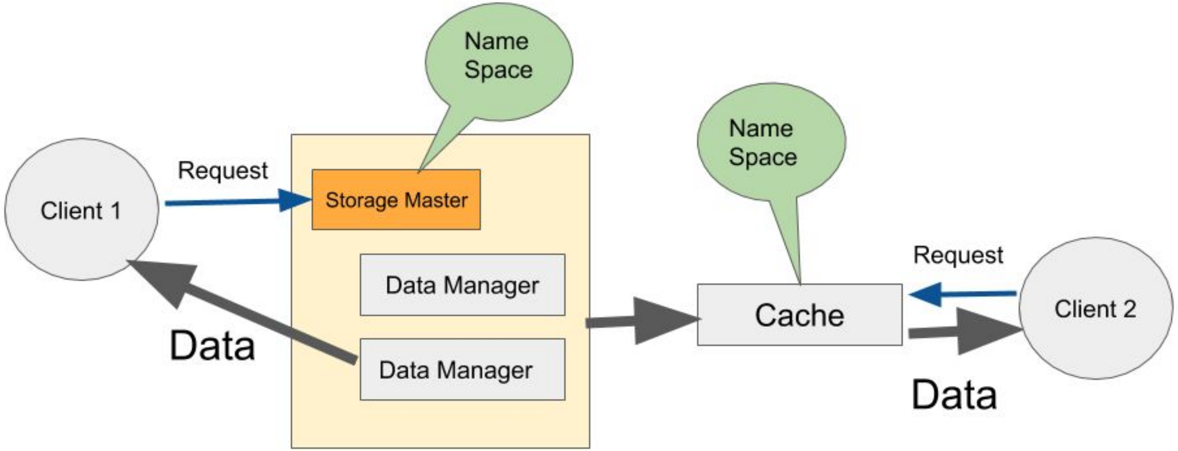
Smart/info collect metrics + automation bare



- Currently redirecting part of the CMS analysis workflows
- Collecting metrics (mainly bandwidth and data read vs filesize) with real workflows
- Started to analyze these data in the context of a IT PhD activity (Mirco Tracolli) for the modelization of a ML-based algorithm for cache eviction and writing prio

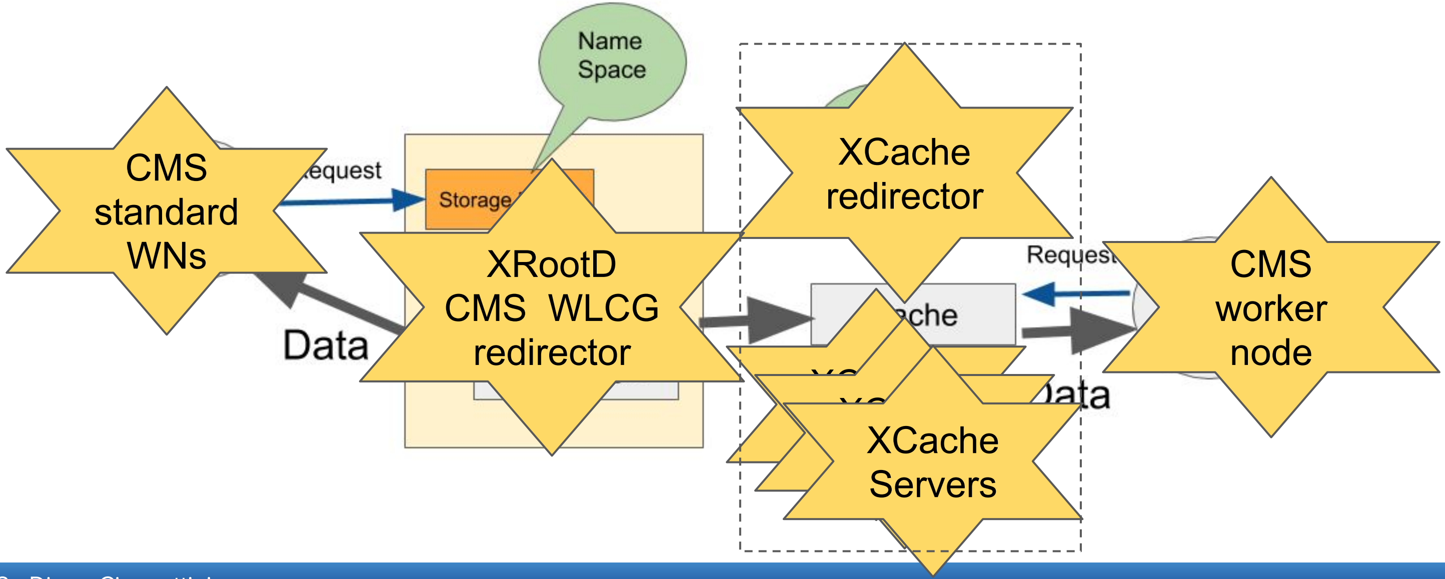
Case 2: Standalone Cache

This scenario has been investigated in the context of opportunistic computing resources (public and private cloud, HPC for HEP) that (by design) are storage-less from the experiment point of view



Case 2: Standalone Cache

Schema of the XRootD equivalent scenario deployed

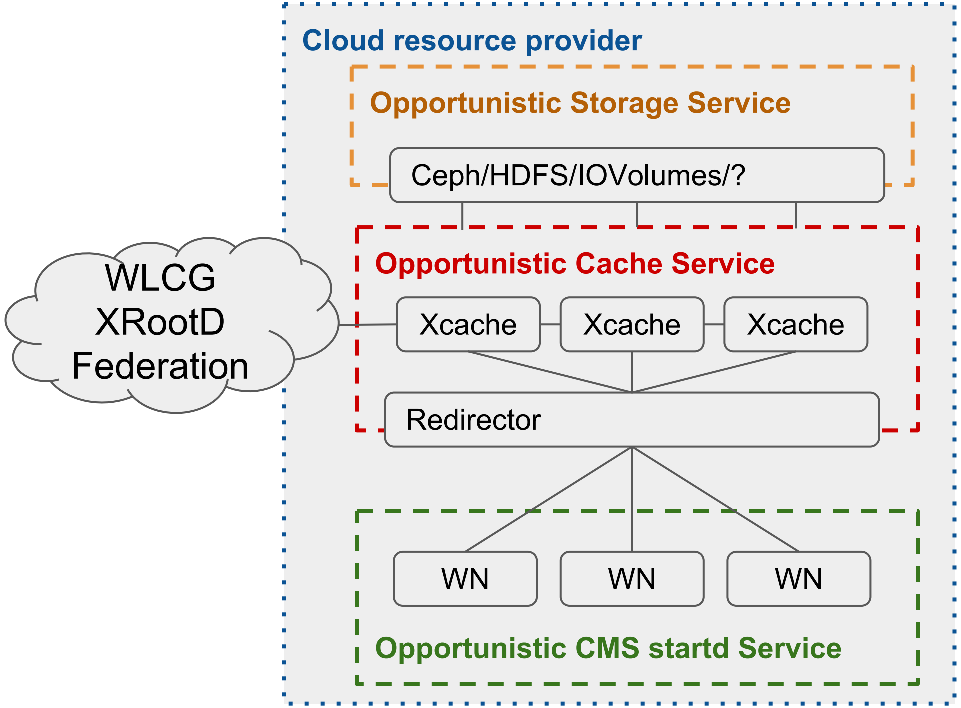


Details on the setup in Backup

Automation - cloud

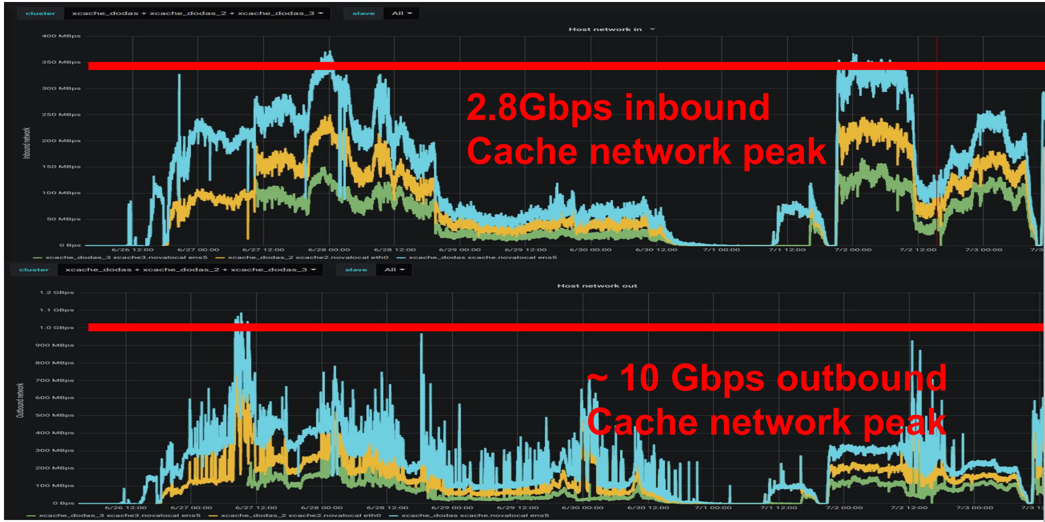
(*) https://dodas-ts.github.io/dodas-doc/

- **Tested with CMS analysis workflows**
  - 2k concurrent jobs on resources @OpenTelekomCloud (OTC)
  - ~150k of users jobs completed reading from standalone cache cluster deployed at OTC
- **DODAS (*)** Thematic Service have been used for:
  - same configuration for setup on different cloud providers
  - automated deployment through:
    - Ansible for infrastructure
    - K8s and Mesos/Marathon for container orchestration



Tests

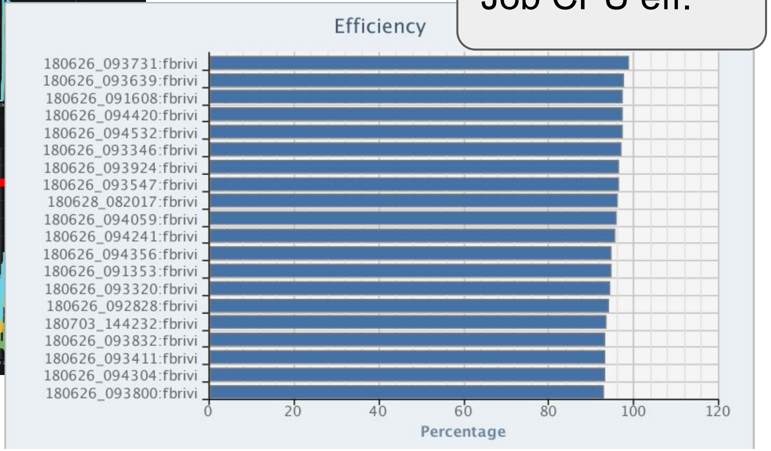
- Excellent overall performances and big amount of data collected for further analysis (next step)
- Performed also preliminary functional tests serving input files for CMS integration work on HPC resources



2.8Gbps inbound
Cache network peak

~ 10 Gbps outbound
Cache network peak

Job CPU eff.



Deploying an XCache stack: DEMO

- Recipe for K8s deployment on ~any cloud provider with DODAS
  - equivalent setup of Mesos/Marathon provided as well
- Deploy of 2 XCache server pods federated under one XrootD redirector
  - caching information from a pre installed origin server
  - sending metrics to Elasticsearch

Links here: https://cloud-pg.github.io/XDC-AH-demo/

Backup

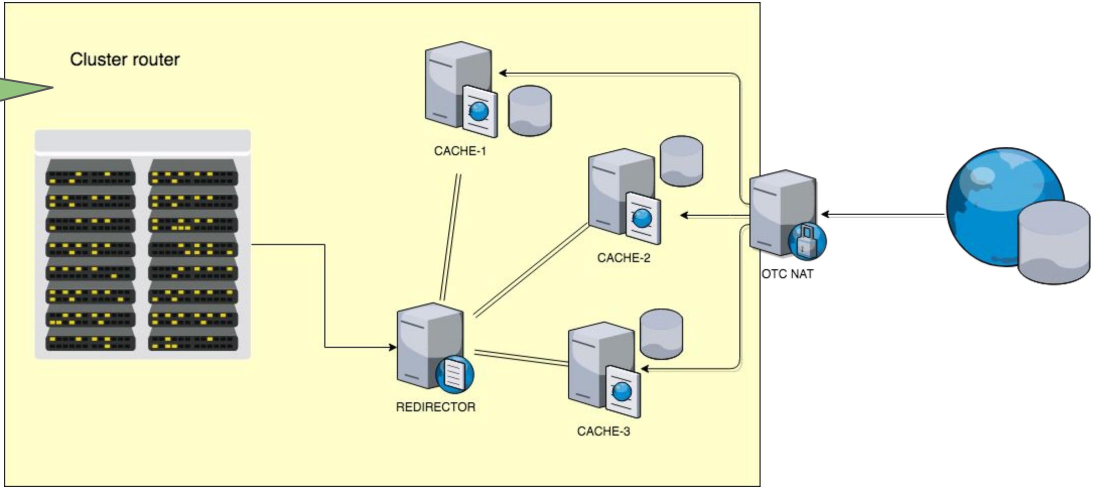
Details

- CNAF XCache redirector:
  - federating CNAF and BARI XCache servers
  - VM on cloud@CNAF
- CNAF XCache server:
  - Bare metal 32GB RAM, 10 Gbps
  - RAID5 spinning disks
  - Fully dedicated
- T2 Bari XCache server:
  - 80GB RAM, 10Gbps
  - GPFS
  - Shared machine with production xrootd

XCache setup 1/2

Ansible deployment in less than 1h. Automated procedure

- 3 cache server:
  - 12/12/8 core
  - 256/256/128GB RAM
  - Ultra I/O flavor volume
  - 20/20/10 TB
- 1 cache redirector:
  - 8 core
  - 16GB RAM


Cluster router
CACHE-1
CACHE-2
OTC NAT
REDIRECTOR
CACHE-3