



INFN XCache experience: status and plans

INFN Perugia:

[D. Ciangottini](#)

M. Tracoli

D. Spiga

INFN Pisa:

T. Boccali

TESTBED INFRASTRUCTURE

CNAF:

D. Cesini

A. Falabella

INFN Bari:

G. Donvito

INFN Legnaro:

M. Biasotto

Outline

- Activity context
- Model in mind
- Studies on analysis data access
- Implementation
- Ongoing activities

Activity context



- **WLCG DOMA** (data organization, management, access):

- XCache as core solution for Data Lake model implementation in WLCG
- interest in historical data access modellization



- **eXtreme DataCloud:**

- CachingOnDemand:
 - deployment automation of XCache stack on cloud resources (k8s, ansible)



- **ESCAPE**

- cache as data access tool for scientific communities data lake prototype



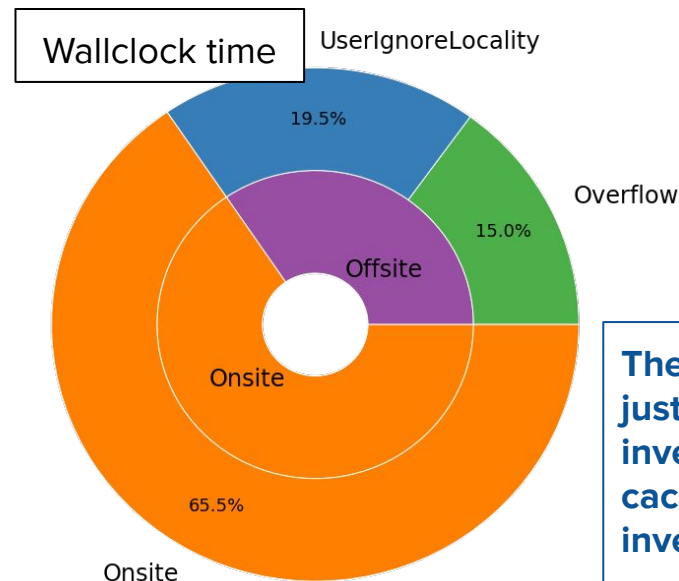
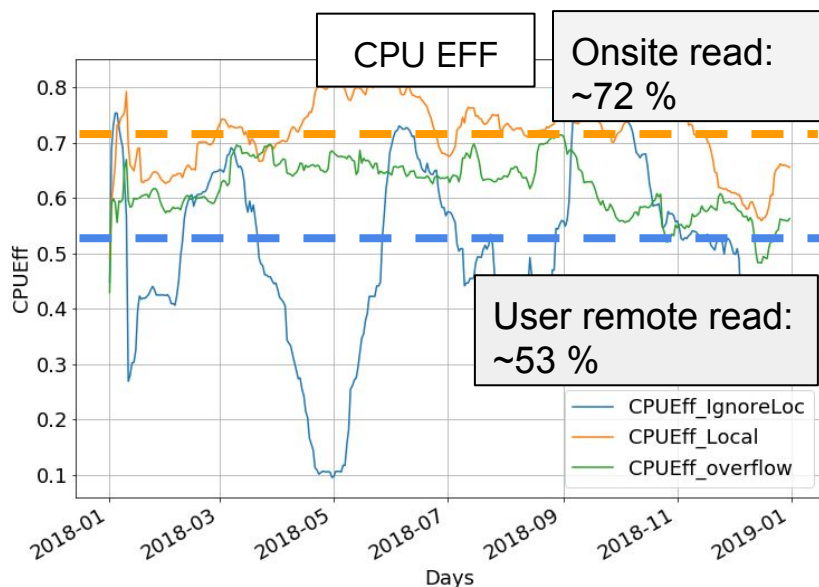
- **IDDLs**

- INFN+GARR national experimentation for a data-lake infrastructure

Studying the analysis data access



- **2018 CMS analysis workflows** running on **Italian Tier2's**:
 - on average **lost more than 15% of CPU time^(*)** when reading data remotely w.r.t. onsite
 - spent around **1/3 of the wallclock time** on jobs with **remote reading**

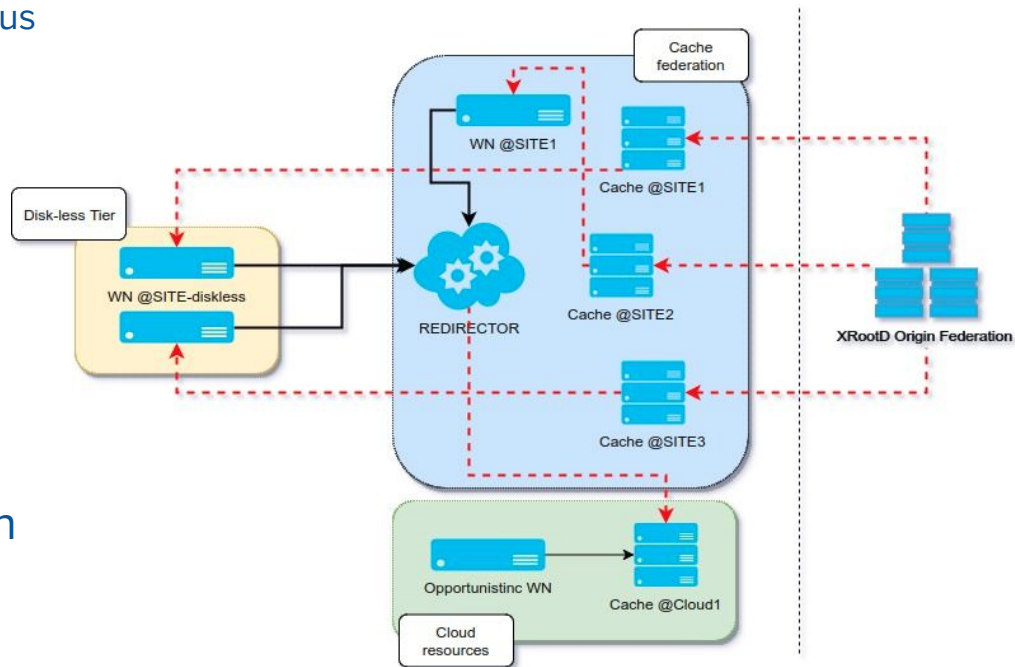


These numbers justifies the investigation of a cache prototype investigation @INFN

Reference model

We aim to the implementation of a regional cache layer for supporting the access to analysis data, thus enabling:

- reduction of operational efforts → no custodial data + LRU eviction
 - service availability thanks to multiple geo-distributed instances and ~stateless behavior
- optimized disk space → only one replica of the same file on caches with low latency links
 - replica-1 fs is possible → factor 2-3 in hw needed “per TB”

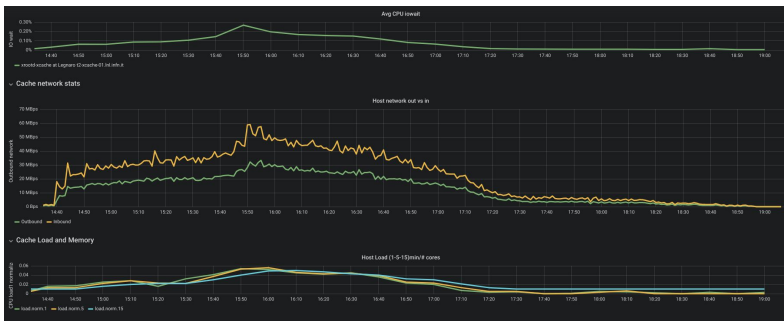
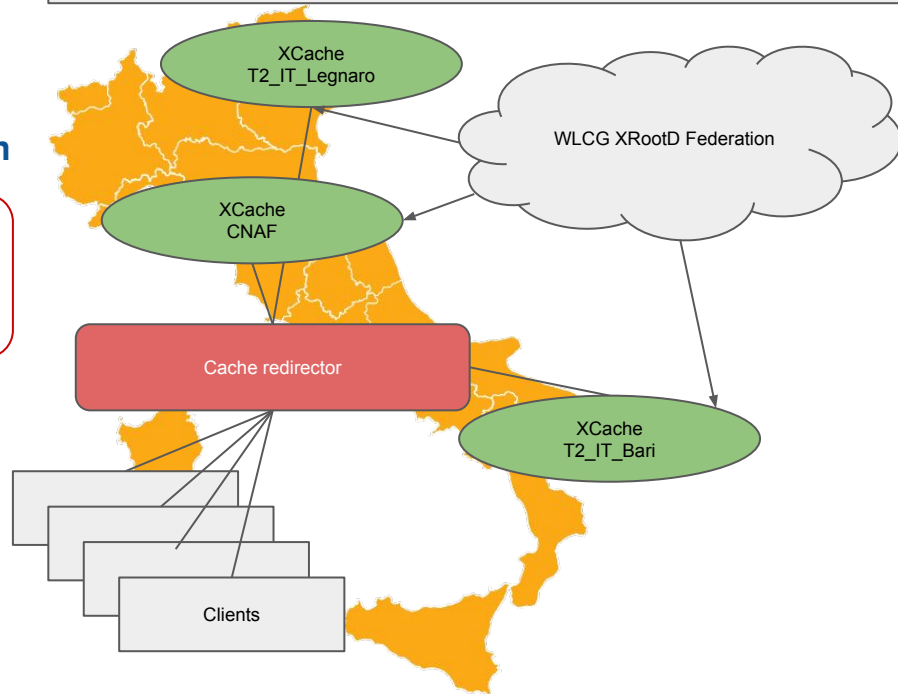


INFN distributed cache status

- Using **already available resources** (with minimal requirements) on **volunteer Italian Tiers** for a distributed **cache-layer PoC**
- Setup **integrated with CMS workflows**
- Starting from empty cache → fully client request driven

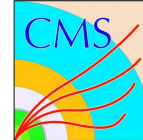
Working prototype since mid-2018 with a limited amount of real tasks using it.

N.B. infrastructure provisioning in collaboration with CNAF, Legnaro and Bari working group



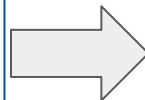
Cache server integrated monitoring - so far host based metrics - xrd metadata not yet there

Measurements



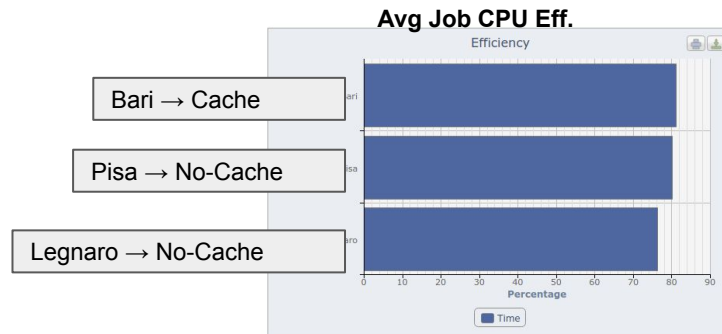
Current **functional test** setup:

- CNAF XCache redirector federating 3 servers:
 - CNAF XCache server (5TB spinning)
 - T2 Bari XCache server (10TB gpfs)
 - T2 Legnaro XCache server (22TB spinning)
- **Redirecting small part of the CMS analysis workflows** to contact National redirector
 - **based on dataset name requested**



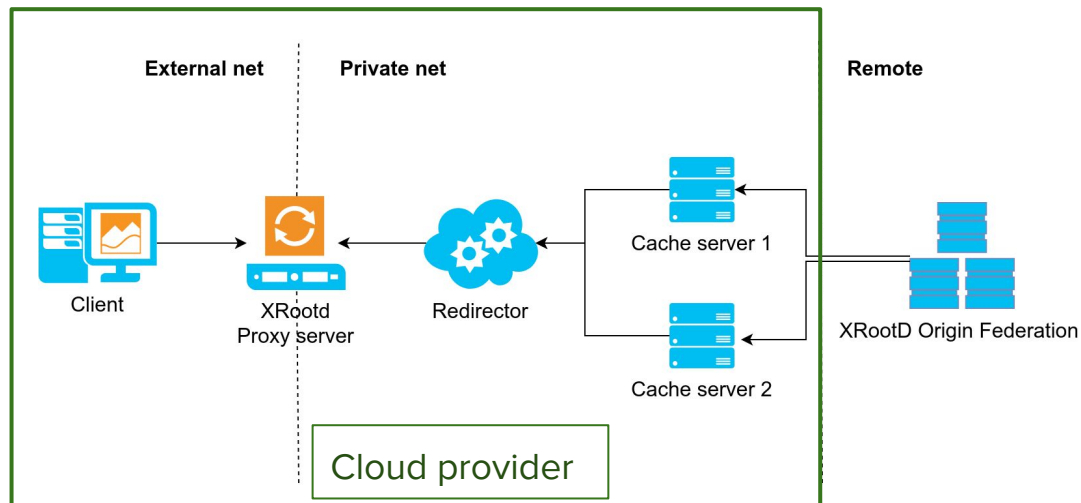
- **Satisfied with the functionalities for CMS workflows**
 - **latency hiding** effect on job CPUEff is noticeable for higher latency links

In case of heterogeneous resources, it is also possible to set load balancing “weights” on the redirector to scale the load on each instance accordingly.



XCache automated deployment

- Automatic procedures for the creation of an **XCache cluster on-demand**
 - **ready-to-use recipe** for both bare metal and cloud environment/orchestration
- Deployment available with:
 - Ansible for bare metal installation
 - Docker compose
 - good end-to-end example for new communities
 - Kubernetes
 - deployment at scale on cloud resource
 - DODAS
 - TOSCA templates for end-to-end automatic deployment on cloud resources



Work in progress: federation

The idea is to use the prototype in place as a playground for evaluating improvements at different level:

- implement and/or test routing decision plugin (on redirector) to:
 - check first on a whitelist of endpoints before going to the cache
 - react to network/hw issues providing alternative routes (either different caches or remote read)
- test the management of QoS through a federation hierarchy
 - fast cache in front of bigger but slower storage
 - include this information as input for decisions above

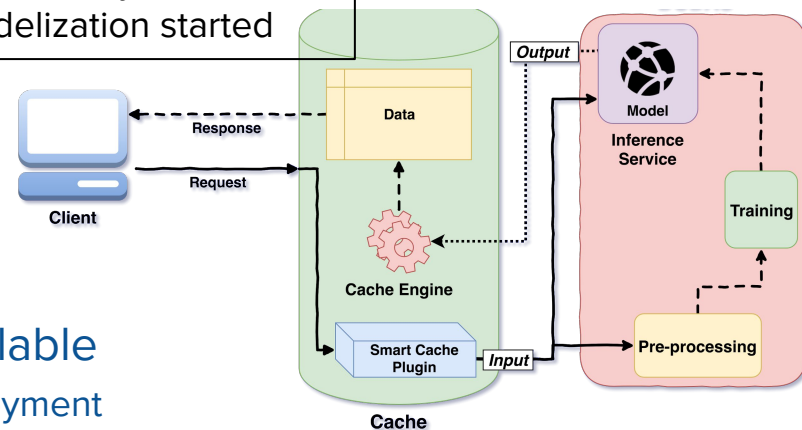
not started yet

functional evaluation
ongoing

Work in progress: cache servers

- Test a cache server plugin for decision on different cache operation mode (proxy vs disk cache)
 - interface towards an external service/catalogue with the optimal action to take based on:
 - current network topology
 - data popularity estimation
 - Provide feedbacks to the external service for strategy update
 - A first **INFN proof-of-concept smart data cache at CMS has been deployed**
- Evaluate a plugin for least used file eviction
- Test token-based auth model as soon as available
 - interesting integration also with k8s automatic deployment

Functionally evaluated.
Modelization started



Summary



- Data access patterns studies confirm the expectations, latency impact on CMS analysis jobs is not negligible and may increase in the future
- XCache shows satisfactory functionalities from latency hiding to disk space optimization
- An ansible/container/k8s based solution has been tested for on-demand deployment
- The current INFN testbed will be used to try out possible improvements that we are evaluating, from architectural design to decision optimization plugins