XRootD cache Federation, status, experience and outlook next steps

Daniele Spiga, Diego Ciangottini
On behalf of CMS and INFN-Cache wg

Outline

All credits goes to Diego both for the slides and most of the work done so far..

- Introduction to the target scenario (data lake)
- CMS-historical data analysis: early results
- Tested setups and early prototypes/PoC
- A quick outlook to the next steps

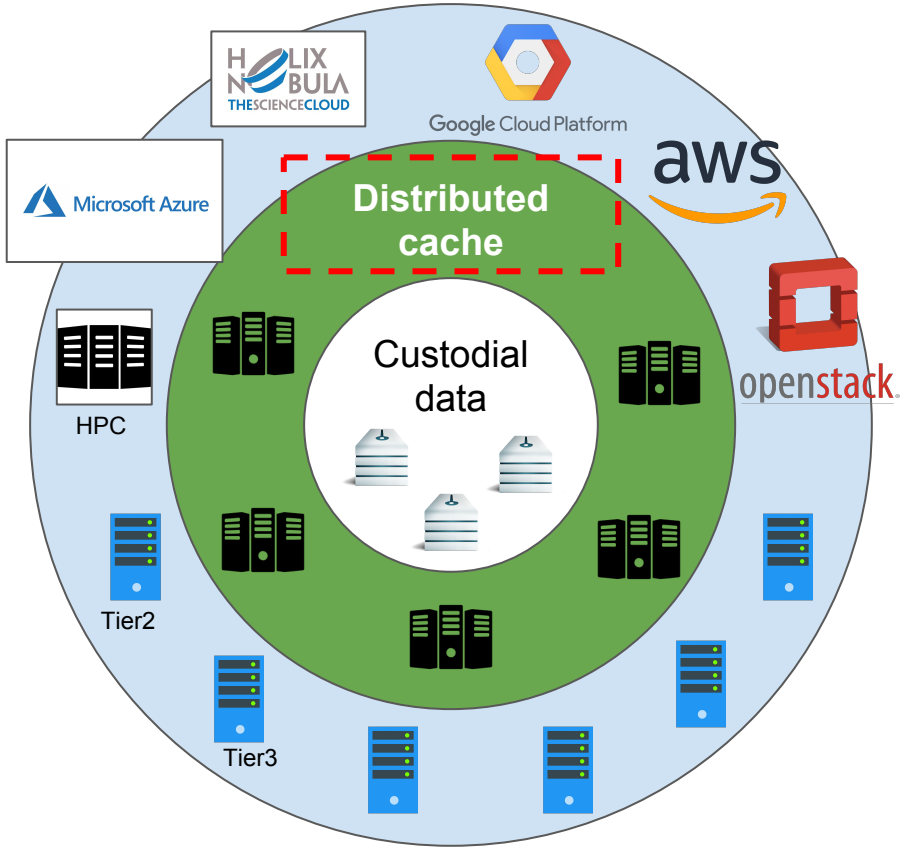
Assumption: towards “data-lake”

Few world-wide custodial centers with data replica managed by the experiment

- Computing Tiers **access data directly from closest custodial center**

Using **cache for a client-driven cache network approach:**

- no central data management - **cache content driven by client requests** (pull model)
- geo-distributed **network of unmanaged storages**
- common namespace (**no data replication**)
- **request mitigation** to custodial sites



Objectives of the activity

- Integration of a **cache layer PoC** in CMS computing model
- **Estimate and measure the benefits** of introducing such a solution

Motivation:

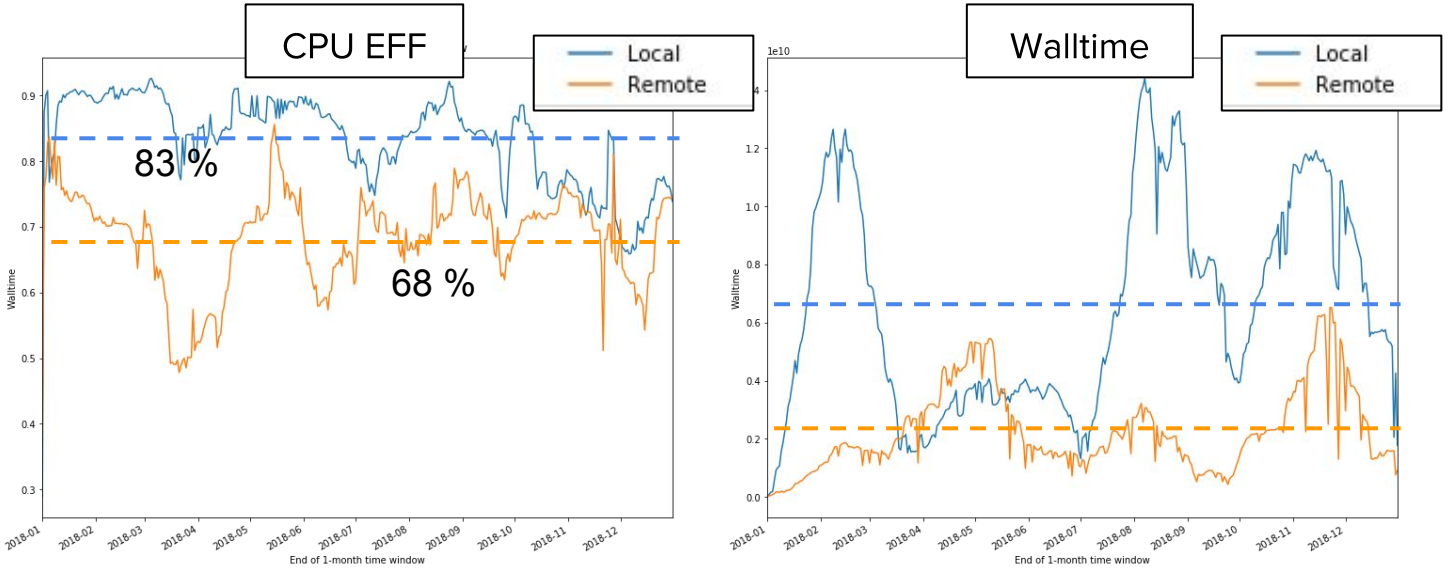
- leveraging national network to:
  - **optimize the size of stored data** (currently focussing at Italian Tier2's)
    - adding a **layer of unmanaged storage**
    - **reduce the redundancy** requirements (no "custodial data")
- **reduce the overall operational costs** for storage maintenance
  - by **adding automation**
  - introducing set of **unmanaged storage resources**

Strategy

1. **Evaluate the impact** of a cache layer on regional basis
   - studying **CMS historical job accesses metadata**
2. Setup a **PoC for a distributed cluster** of cache servers on **Italian Tier2's**
   - **And opportunistic resources we might have (HPC, Clouds etc) <--storage-less (e.g. CCC)**
3. **Measure the effect** in terms of
   - **CPU efficiency**
   - **disk space**
   - **operational efforts**
4. **R&D usage of ML-based algorithm** for further improvements
5. Deploy a **PoC for a modular all-in-one infrastructure** for smart cache decisions

CMS user workflows: CPU performances

- during **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 **⅓ of the wallclock time** on jobs with **remote reading**



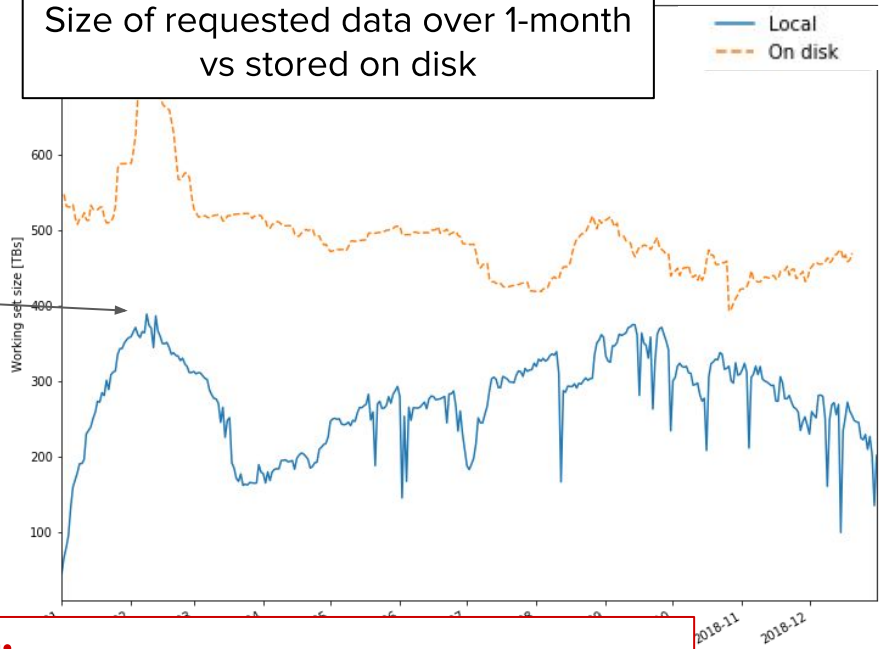
Situation in line with the overall CMS values

(*) such inefficiencies have been investigated by a dedicated WG → The motivation for that is a trade-off made b/w CPUEff loss and reduced replicas of data around

CMS user workflows: requested data volume

- **In terms of stored data:**
  - max amount of MINIAOD **data locally-read** for analysis over 1-month window is below **400TB**
  - corresponding to **~80% of what is usually stored (500TB)** on the Italian tiers for the same data format

Size of requested data over 1-month vs stored on disk



- **So, introducing a cache layer we expect:**
  - **a narrowed CPUEff difference w.r.t. local data access** (reduced latency)
  - **optimized data volume stored on disk**
    - cache only what requested frequently + no internal replica at FS level needed

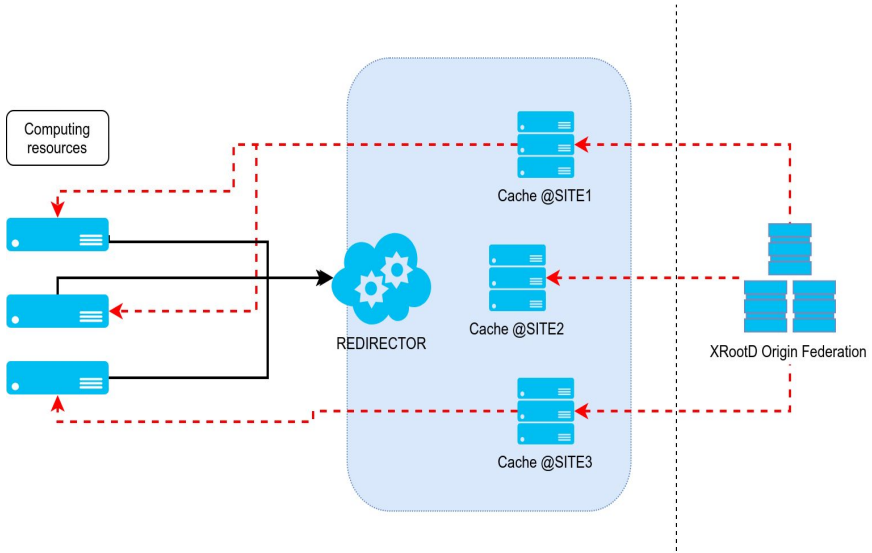
Distributed cache: recap

Prototyping a cache model for DCCs (in straw model terminology) :

- enabling **“storage-less” T2s (CCNC)**and efficient **scale out over opportunistic resources**

Using cache for a **geo-distributed layer approach:**

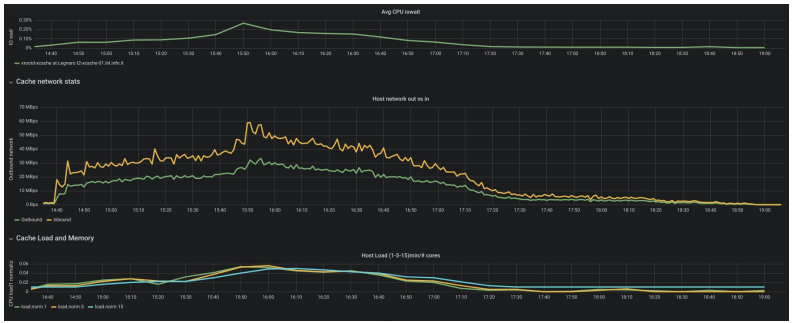
- geo-distributed **network of unmanaged storages**
- common namespace (**no data replication** across cache on DCCs)
- **request mitigation** to custodial sites

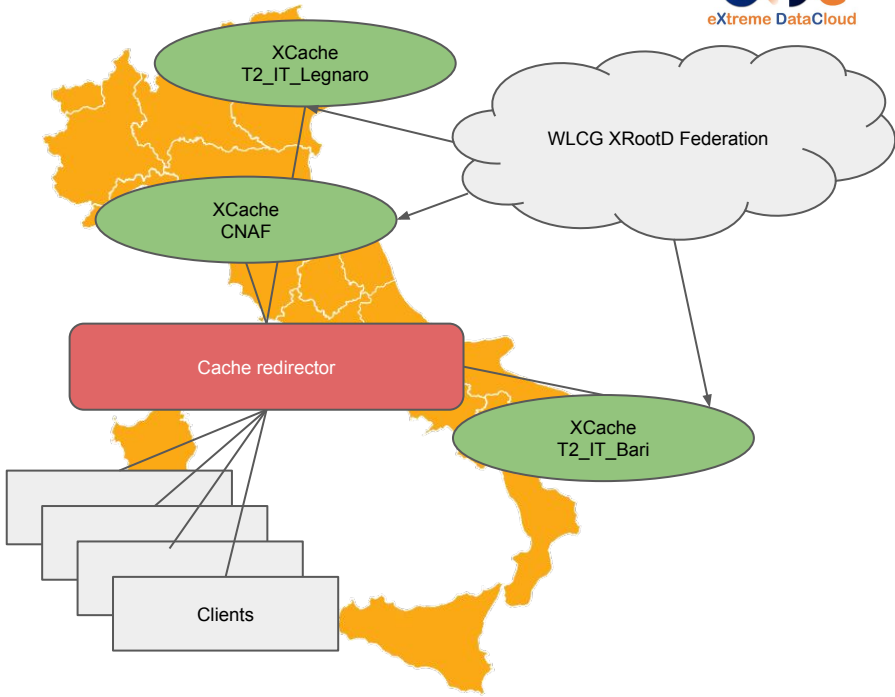


INFN distributed cache on DCCs: status

- Using **already available resources** (with minimal requirements) on **volunteer Italian Tiers** for a distributed **cache-layer PoC**
- Setup **integrated with CMS workflows**

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





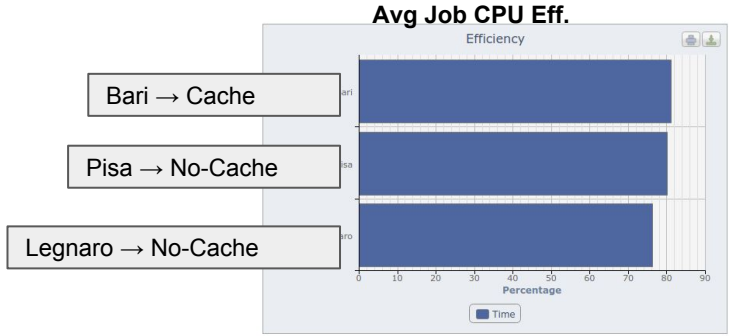
Cache server integrated monitoring - so far host based metrics - xrd metadata not yet there

INFN distributed cache: tests

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 part of the CMS analysis** workflows to contact National redirector
  - based on dataset name requested
- Tier2 at Pisa is joining the testbed

- **Verified functionalities with CMS workflows**
  - redirector **cache content awareness**
    - no data duplication
  - **latency hiding** effect on job CPUEff
  - see previous DOMA pres here



Measurements using Italian Cache Federation

190322_134029:vmariani_crab_WJets_800To1200_il_script_2	92.25%
190321_085414:vmariani_crab_WJets_800To1200_script_ign_loc	86.85%
190321_083600:vmariani_crab_WJets_800To1200_ign_loc	77.89%

Sample tasks from **real user analysis:**

- data reduction to ROOT plain tuples
  - **typical 2018 analysis use case**
  - ~0.4 MB/s per job
  - input data stored at DESY and T2_FR_IN2P3
- task monitored for three different benchmarks:
  - **No cache:** running at T2_IT_* and remote read
  - **Cold cache:** running at T2_IT_* and remote read with empty cache
  - **Warm cache:** running at T2_IT_* and remote read after cold cache

Total dataset size: 1.2 TB
Cached size: 922 GB (77%)

Summary of jobs using **cache (2nd time):**
* CPU eff: **92%** average
* Waste: 14:24:53 (2% of total)

Summary of jobs using **cache (1st time):**
* CPU eff: **87%** average
* Waste: 21:31:38 (3% of total)

Summary of jobs with **remote read:**
* CPU eff: **78%** average
* Waste: 44:28:37 (7% of total)

Deployment on cloud resources

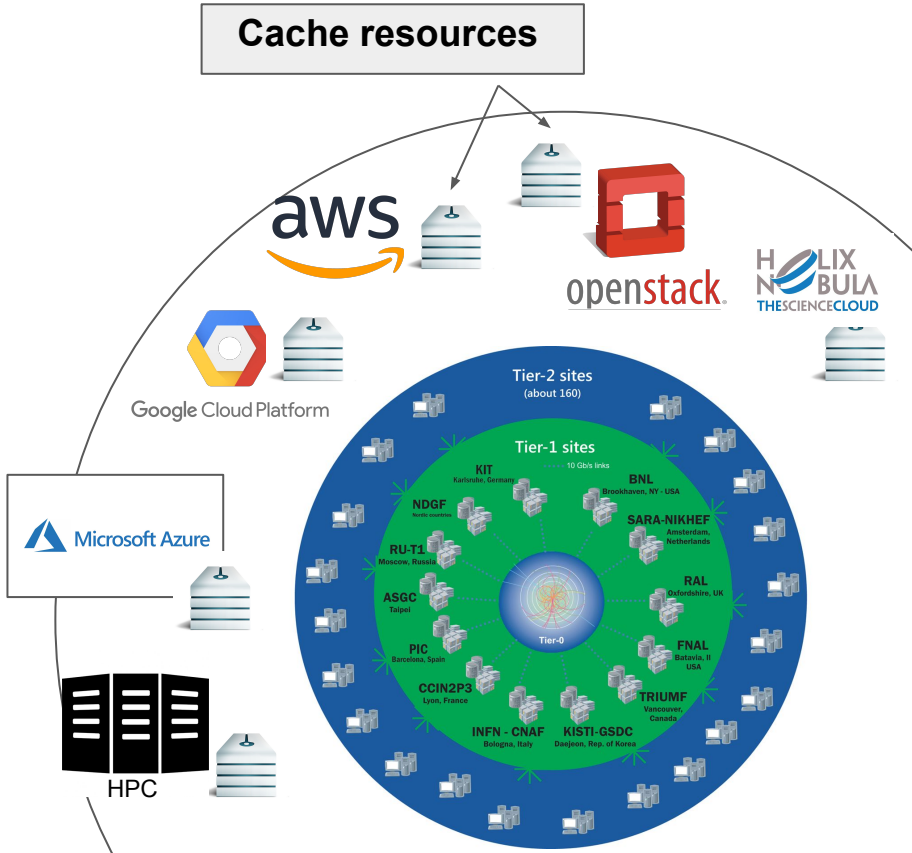
Scenario 2

Computing resources are **opportunistically deployed on cloud/HPC resources**

- **storage not necessarily available**
  - remote read **latency**
  - **I/O inefficient**

The **cache** introduction may offer:

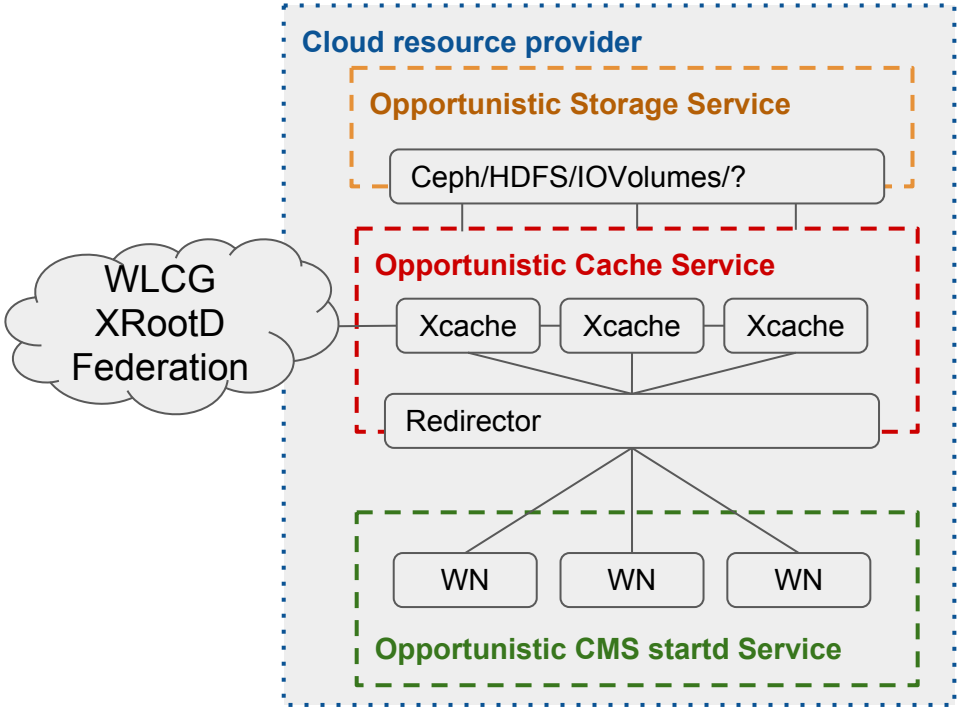
- **ephemeral storage for hot data** near the computing provider
- **optimized wan access**, only for data not already on the cache



Recipes on CachingOnDemand

XCache for CCC on cloud resources

- Automatic procedures for the creation of an **XCache cluster on-demand**
  - **ready-to-use recipe** for both bare metal and cloud environment/orchestration
    - complete high level infrastructure description → QoS configuration
- Currently available 3 deployment methods:
  - Ansible for bare metal installation
  - Marathon/Mesos
  - Kubernetes

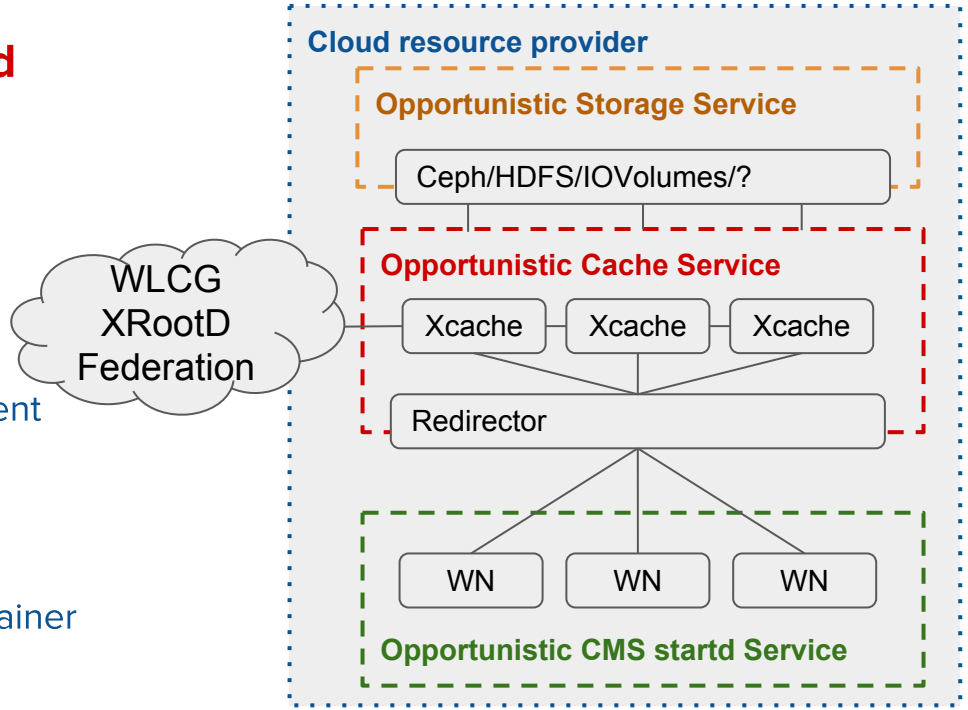


Recipes on [CachingOnDemand]

Scale tests 1/3

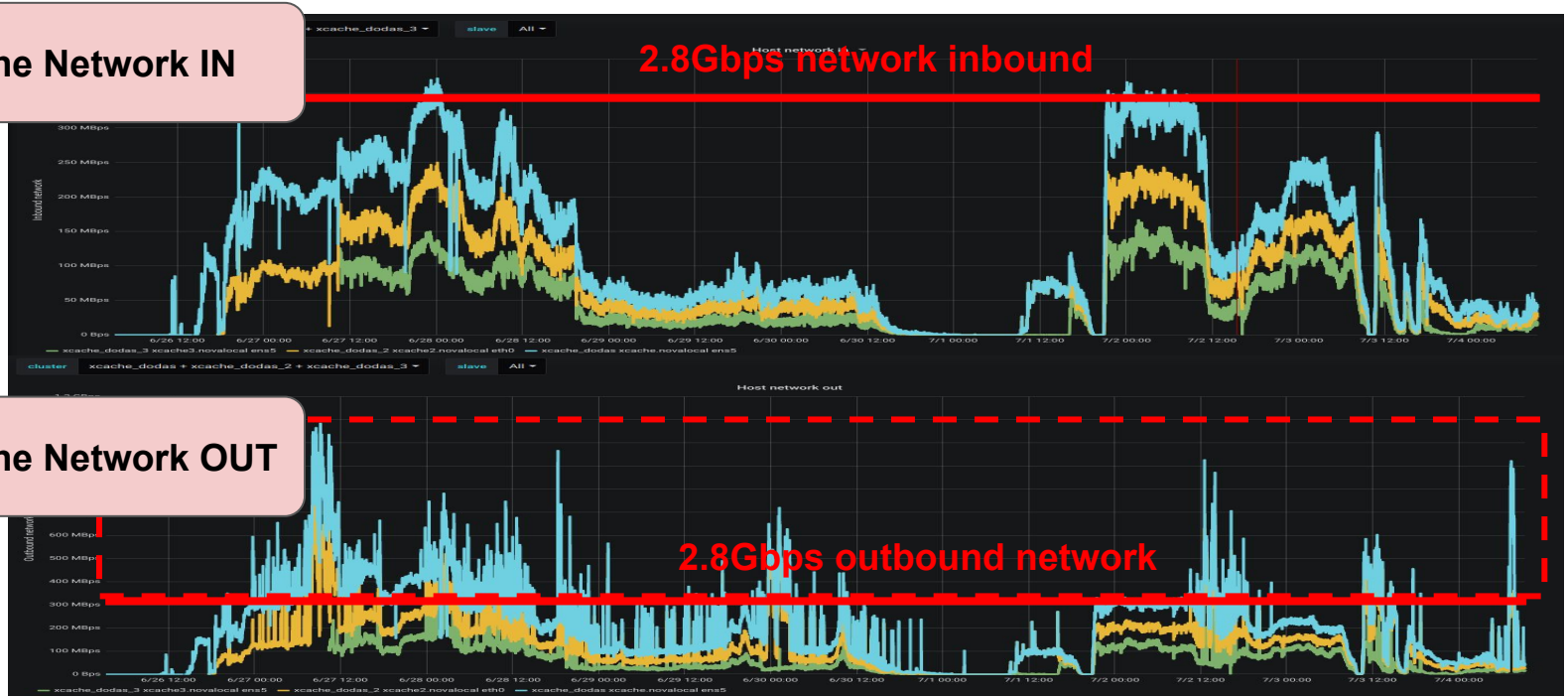
(*) https://dodas-ts.github.io/dodas-doc/

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



Scale tests 2/3

Cache Network IN



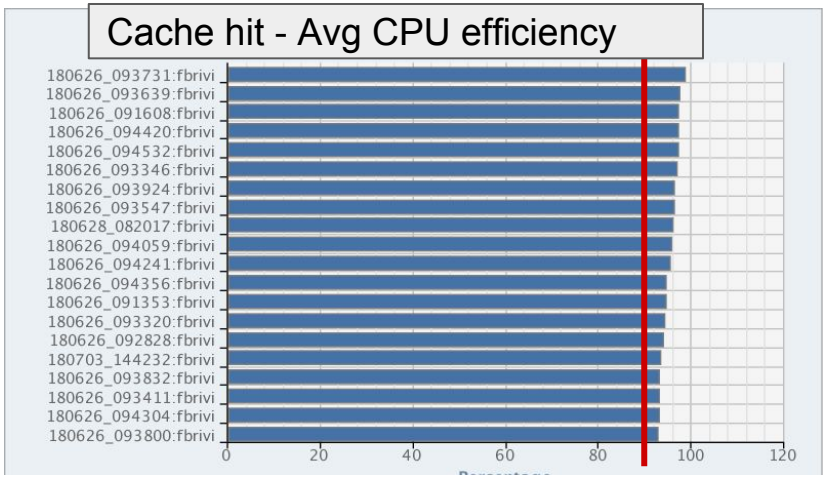
Cache Network OUT

Not negligible gain with **significant spike of direct cache disk/memory usage**

Scale tests 3/3

- **local-like performances when a cache hit occurs**
  - **No cache overhead observed with this workflow samples**
  - slight improvement due to good performances of the low latency storage flavor provided

No cache overhead observed



Local read reference

A quick outlook on the next steps..

In the pipeline

- **Scale up of the national testbed** towards production-like grade
  - Synergies with other national project (such as IDDLS)
  - Ultimate goal is to use this not only for CMS workflow, of course!
- Exploitation of QoS && Caches
  - Already existing other initiatives in US CMS (but not only I guess)
- Expand the **studies also towards CMS central production workflows**
  - AndreaS. Already performed a early analysis which seems to confirm what we expect… cache is more useful for analysis wf than production one
  - We'd like to combine / complement our analysis with AndreaS. and similar activities in CMS
    - There is a "working group"
- Studies on **ML-based algorithm for smart cache decisions** in CMS
  - Use the infrastructure provided to study/simulate performance of different approaches
  - Something in line with activities carried on by **Operation Intelligence** WG
    - See more lather

Improving efficiency with “smart” decisions

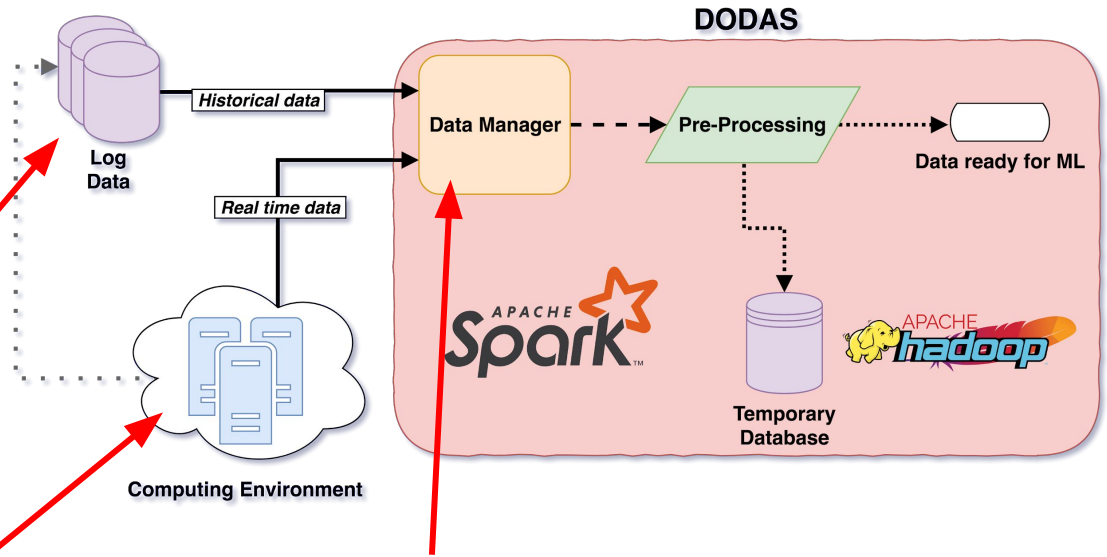
Evaluate the **use a smart decision service** for cache layer management to:

- **Further reduce latencies**
  - client-cache routing based on topological real-time information
- **Optimize the cached data volume**
  - Optimized data eviction decisions (LRU atm)
  - Decide what to save on disk based on algorithm trained over historical data
- **Lower operational costs**
  - re-adapt routing in case of link failure

The service environment implementation has been **created and packaged as a modular all-in-one solution** (data ingestion ➔ training ➔ inference) leveraging DODAS framework

Smart Cache decision service overview

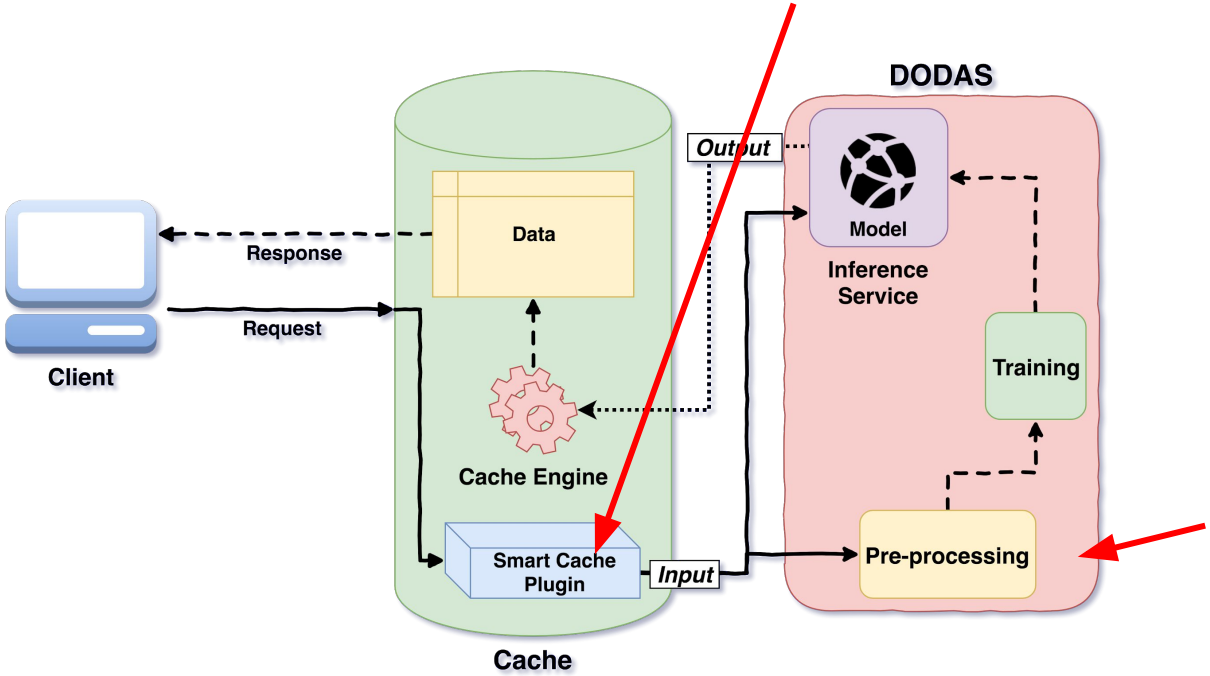
- The **CMS available logs are the key** to the success of the model development
- A **Primary data** source is historical data of infrastructure utilization:
  - **Data logs** are in JSON format, **stored** in a **Hadoop** file system and **serialized using Avro**.



- The **Secondary data** source are **real-time information**
  - Info of hardware, clusters, network and the cache system (content and status)
  - Streaming information feed
- The **Data Manager** can be customized to **prefetch data** into DODAS environment **or to get a stream** of data in real-time.

Integration with XCache

- **Extend the XRootD cache** with a specific **plugin** which queries against the deployed **AI Service** to understand **whether or not to keep data on disk.**



Preliminary tests ongoing with a PoC deployed on INFN cloud resources

Runtime information are used to **continue** the **training** of the **model**

Wrapping-up

- **Preliminary evaluation of cache layer effects** on Italian CMS Tiers done:
  - based on historical user analysis access metadata
  - measuring improvements on CPUEff from sample of real user workflows
  - Lot of room for extension/integration and sinergies
- **CMS-integrated cache federation prototype** deployed and functionally tested
  - Consolidate and increase (the size).. Move toward production quality setup
  - Extend the cache layer to other communities
- Working on a first **proof-of-concept** implementation and deployment to enable **smart data cache**
  - Measure gain in HW and operational costs