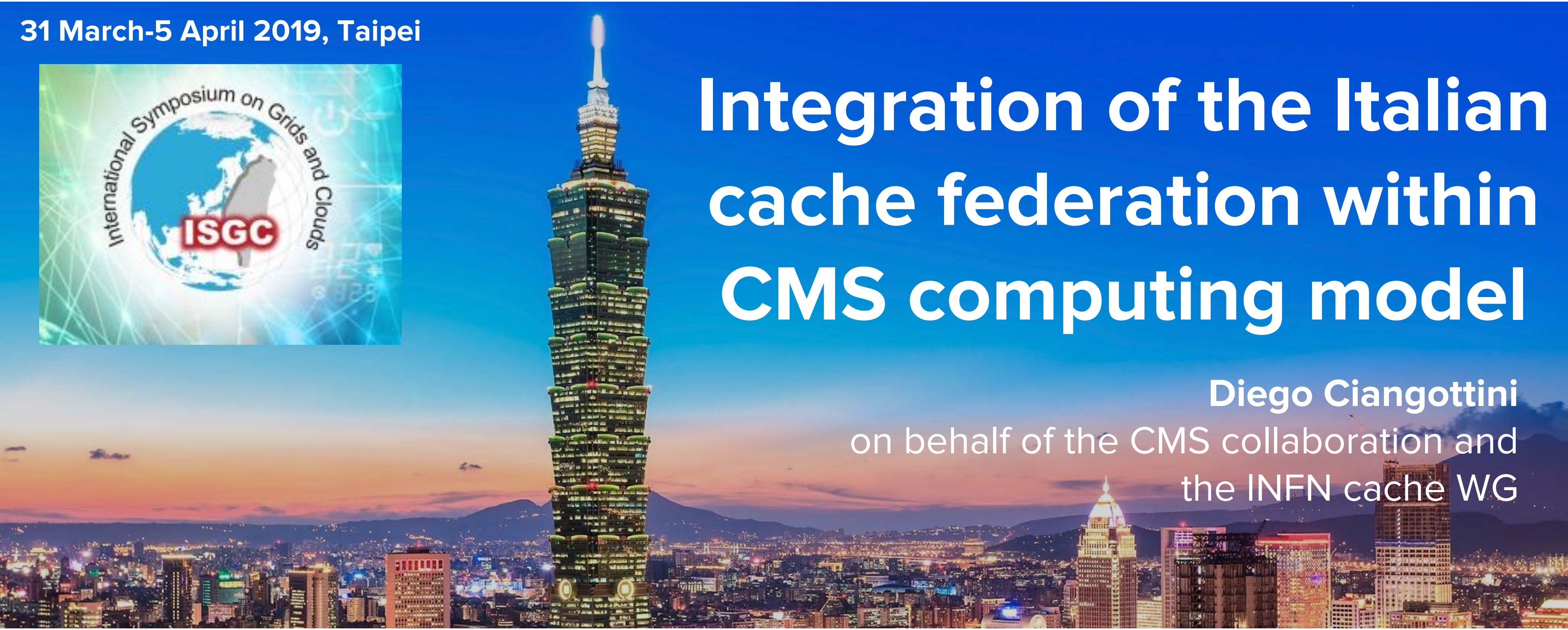


31 March-5 April 2019, Taipei

Integration of the Italian cache federation within CMS computing model

Diego Ciangottini
on behalf of the CMS collaboration and
the INFN cache WG

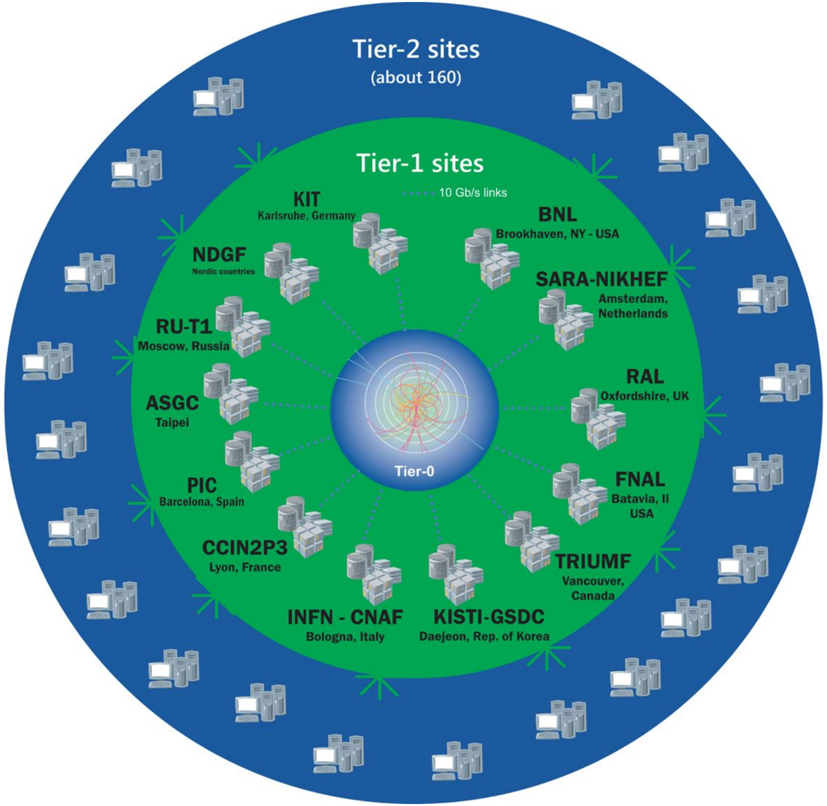
Outline

- Introduction
- CMS data access studies
- Cache federation: Italian testbed
    - setup and performance measurements
- Cache integration with a smart decision service
    - infrastructure deployment overview
- Conclusions and next steps

XCache have been used as enabling technology for the presented activities

CMS current model

- Hierarchical **centrally managed storages at computing sites** (Tier)
- Payloads **run at the site that stores** the requested data
- **Remote data access** already technically supported
    - fallback to remote in case of local read failure
    - overflow of jobs to near sites



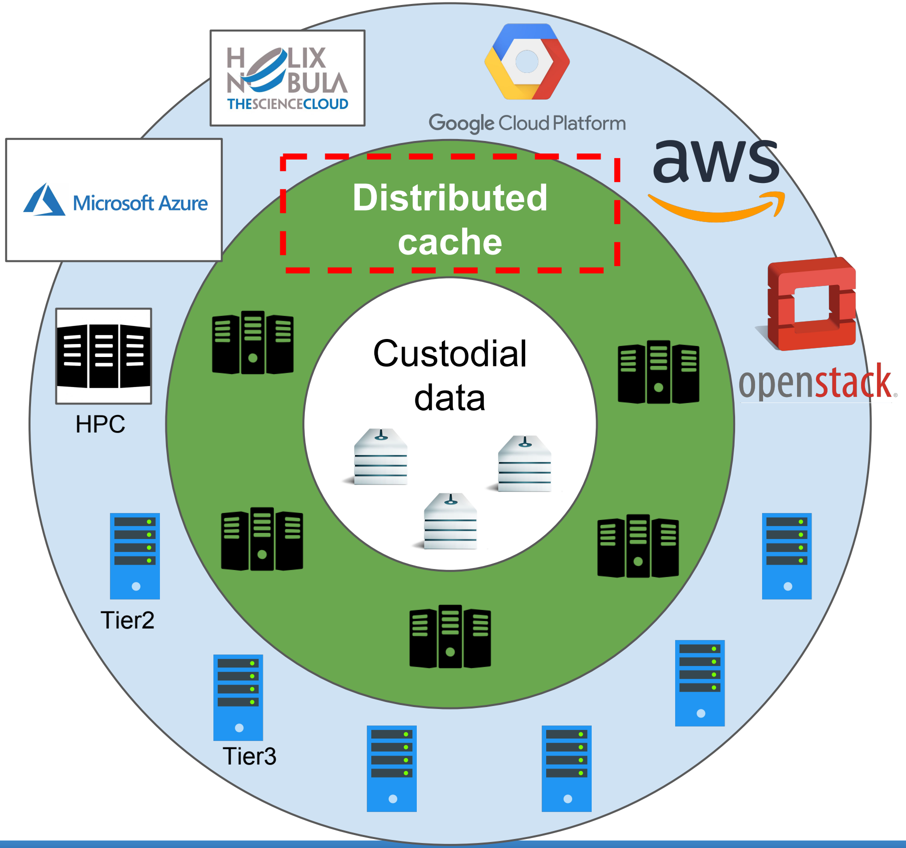
Towards “data-lake”

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:**

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



Objectives of the activity

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

Activity in the context of WLCG DOMA-Access working group

Motivation:

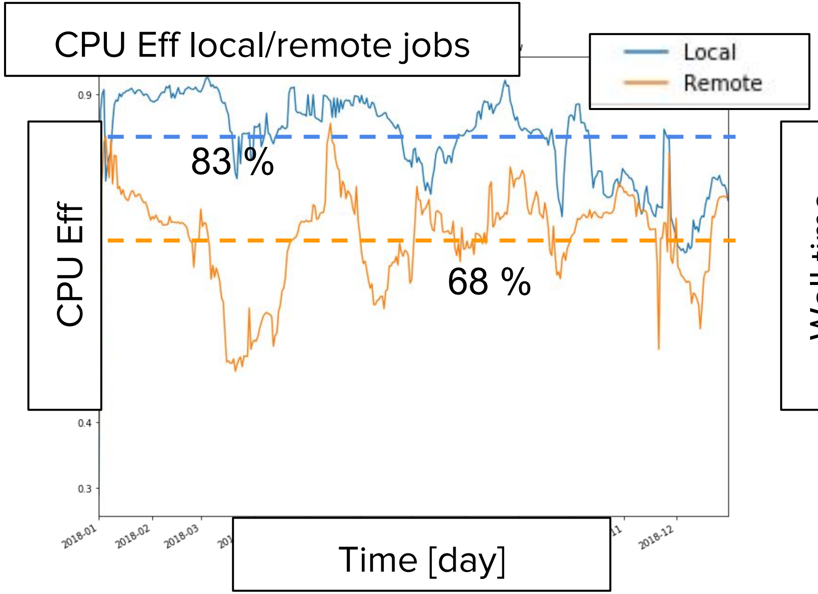
- leveraging national network to:
  - **optimize the size of stored data** at Italian Tier2's
    - adding a **layer of unmanaged storage**
      - or even replacing the current managed one
    - **reduce the redundancy** requirements (no "custodial data")
- **reduce the overall operational costs** for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

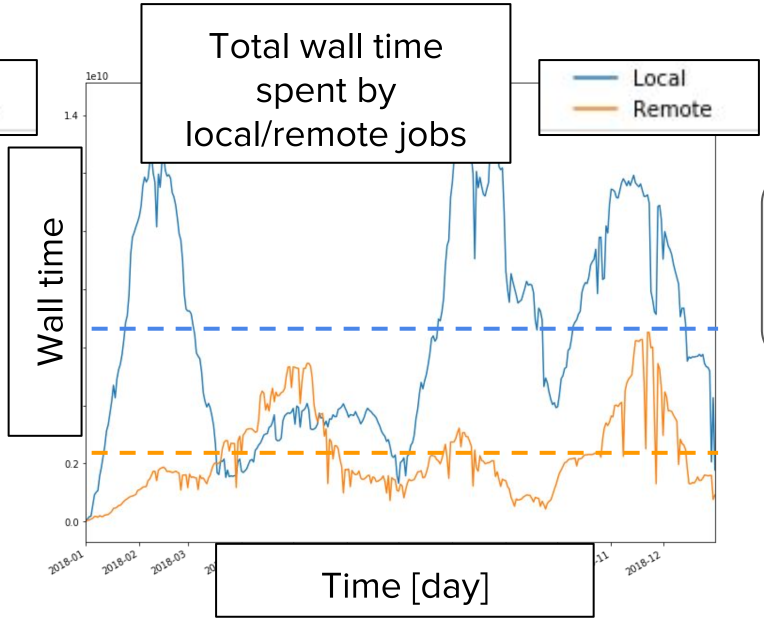
Strategy

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**
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**



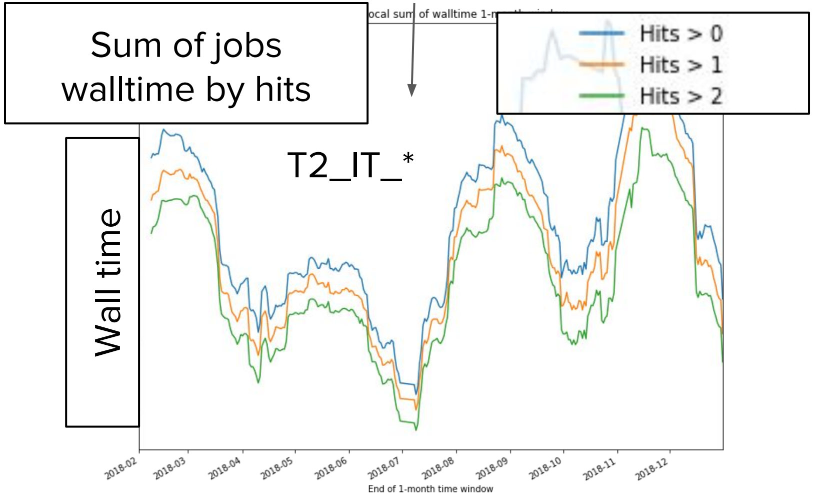


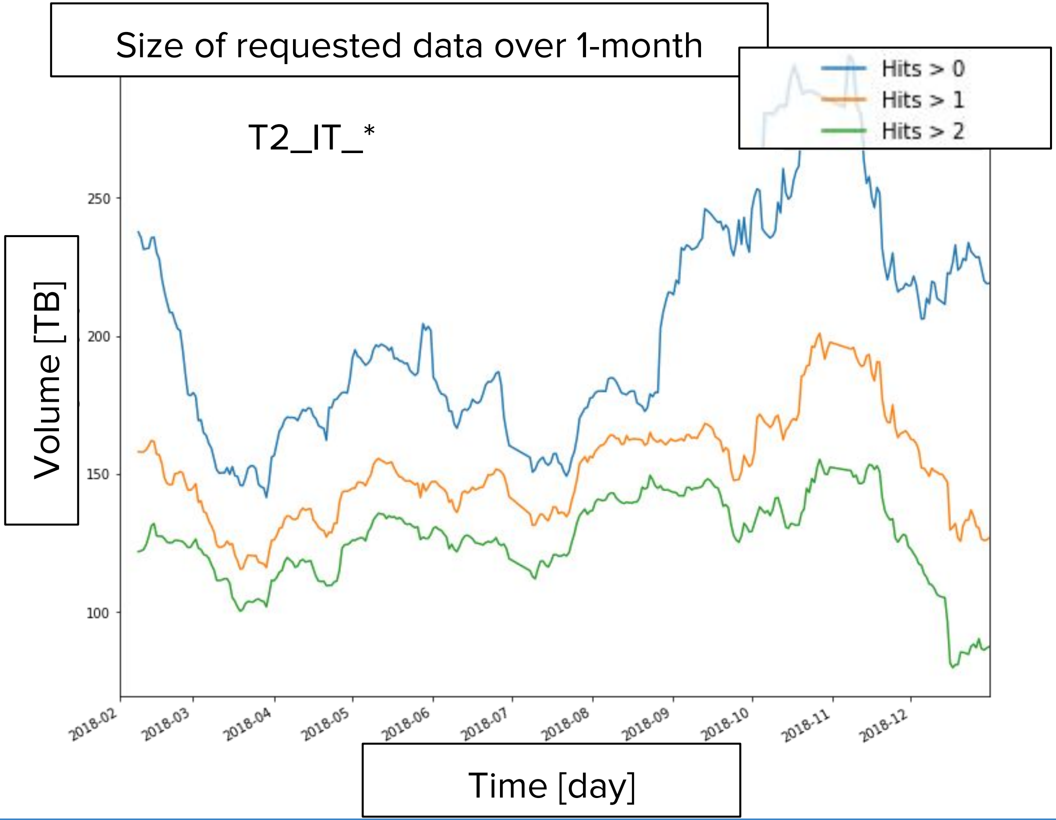
Situation in line with the overall CMS values

(*) 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 at Italian sites: hit rate

- around **40%** of total requested data are **accessed by more than one workflow** in a month (Hit)
  - in terms of CPU time the “accessed only once” is below 15%

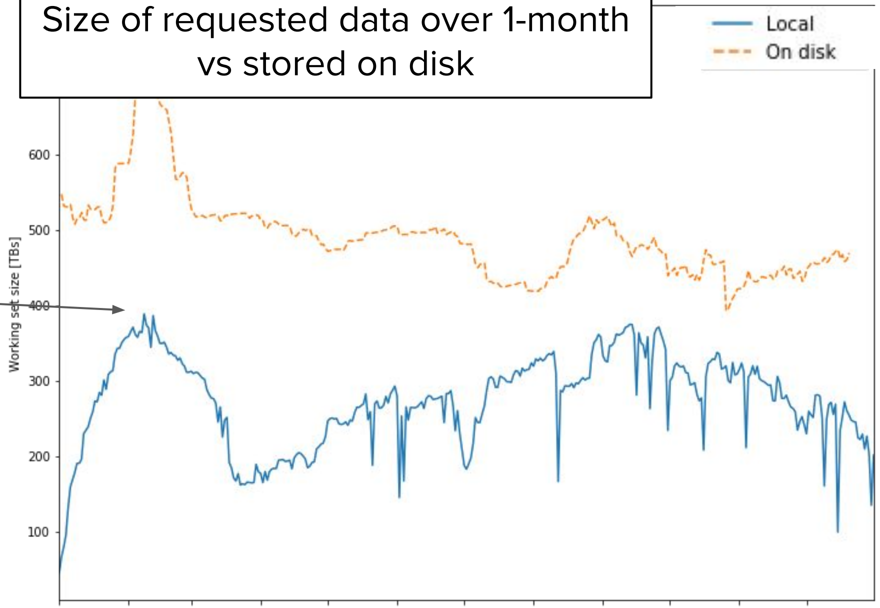




CMS user workflows: requested data volume

- **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

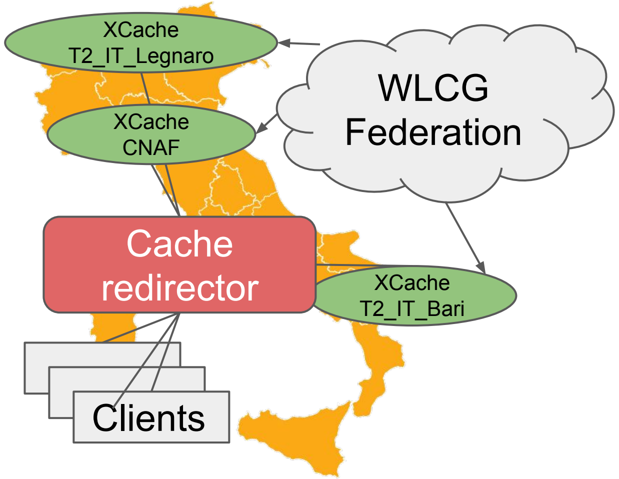
Size of requested data over 1-month vs stored on disk



- **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

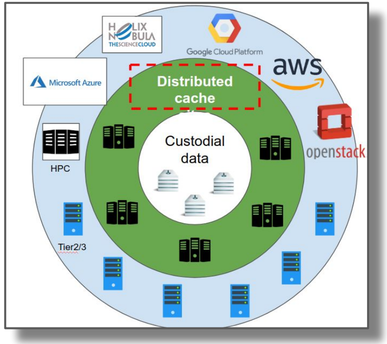
Italian CMS cache federation

- **INFN PoC for geo-distributed cache:**
  - Clients contact the **cache redirector**
  - Redirector **steers client to**
    - the **cache that actually has file** on disk
    - **If no cache has the requested file, a round robin selection** of cache server is used



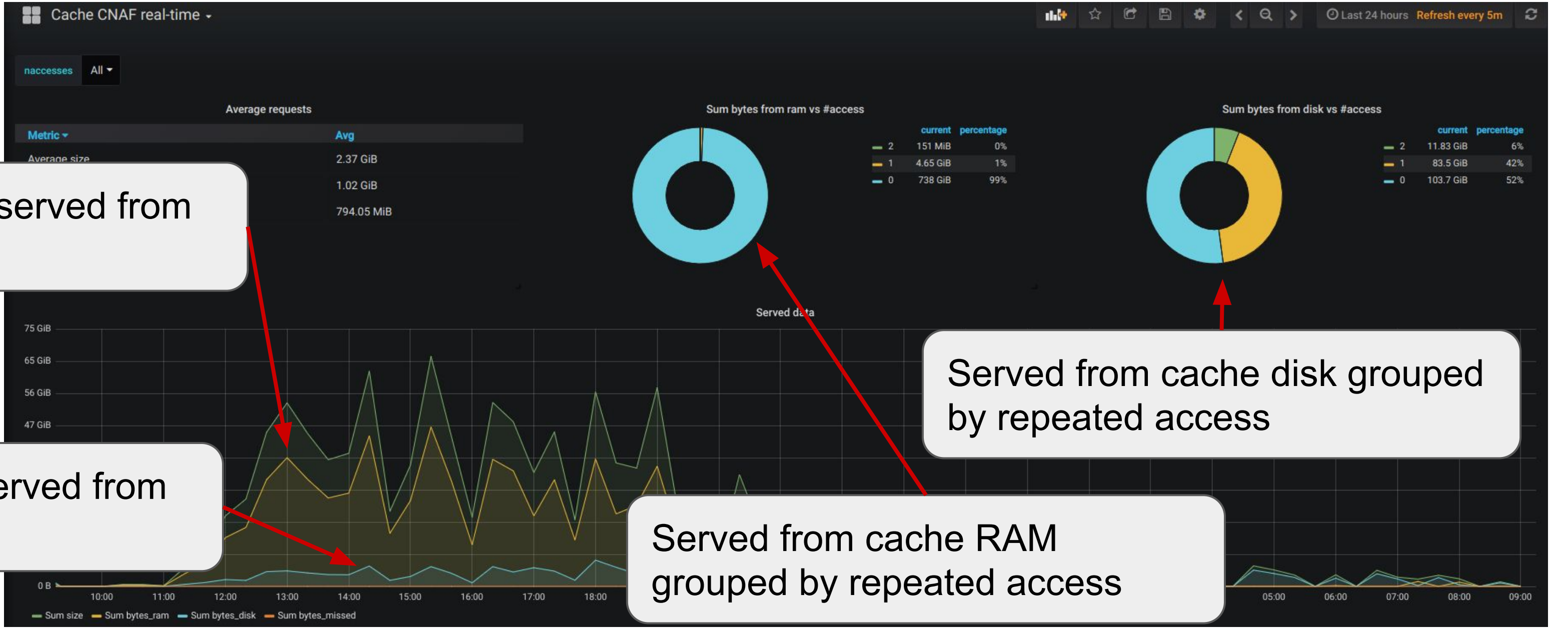
Working prototype since mid-2018 on 3 Tiers (CNAF, Bari, Legnaro) with dedicated redirector @CNAF.

Seamlessly integrated into the CMS model.
Real CMS tasks that require a set of datasets are **using the cache system in a transparent way.**



Also recipes for cloud deployment available on CachingOnDemand

Integrated cache monitor



Cache servers can be deployed through an **Ansible recipe with integrated monitor sensors** for both **host and XCache internal metrics** (example above).

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

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

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

Expected improvements

From a sample of user analysis tasks the expected effect in the current model are:

- **first remote read reduced the CPU loss by ~10%** with cache introduction
    - thanks to read-ahead
- **up to 20% for repeated accesses**
    - happening within 1-month for ~40% on the data accessed

In a future **data-lake scenario:**

- **<6% CPUEff loss at first access** w.r.t. local read, but **10% better than simple remote read**
- **local-like performance** at the second access
    - happening for 40% of the cached data
- **usage of only one replica FS is possible** ➔ at least a factor 2 in space available
    - usually 2 or 3 are used depending on FS

Improving efficiency with “smart” decisions

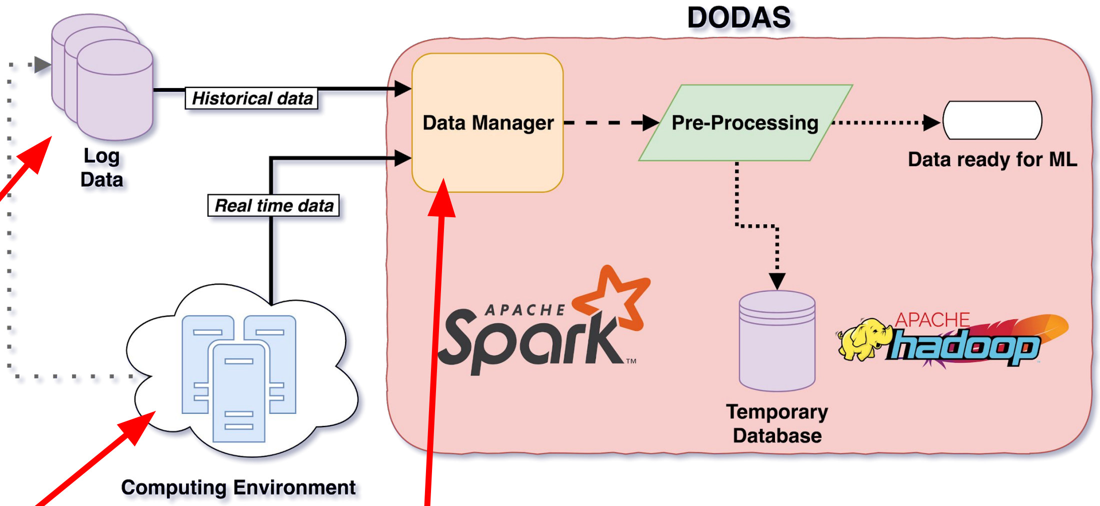
Evaluate the **use a smart decision service** for cache layer management to:

- **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

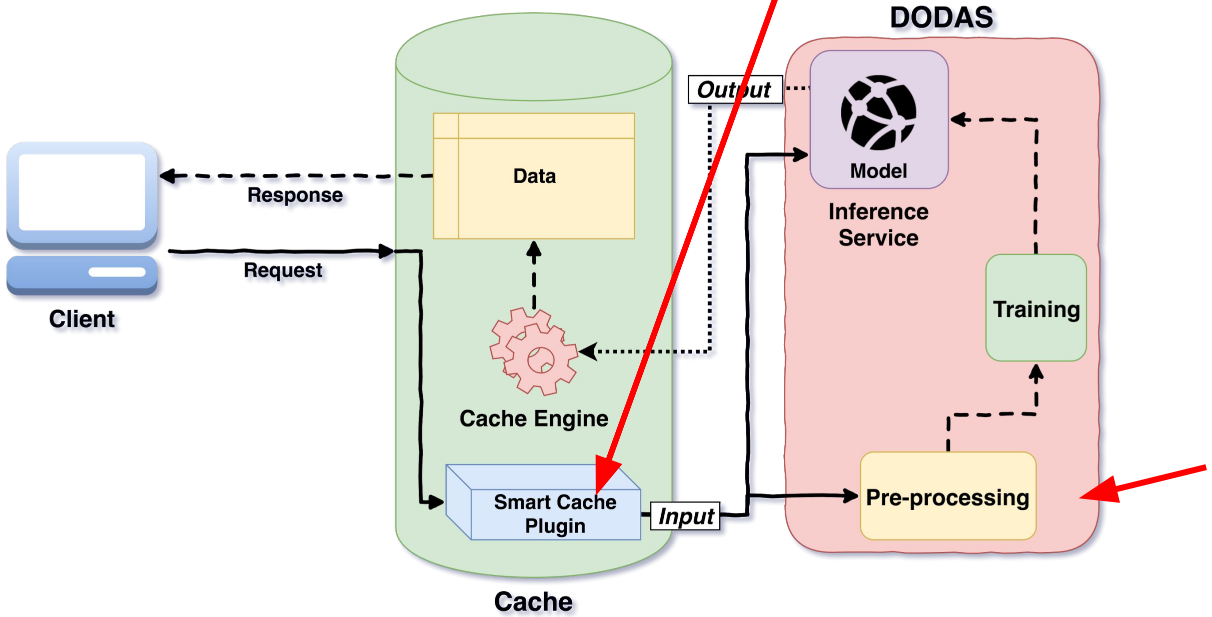
- The **CMS available logs are the key** to the success of the model development
- A **Primary data** source is historical data of infrastructure utilization:
  - Data logs are in JSON format, **stored** in a **Hadoop** file system and **serialized using Avro**.



- 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**

Conclusion

Next steps:

- **Scale up of the national testbed** towards production-like grade
- Expand the **studies also towards CMS central production workflows**
- Studies on **ML-based algorithm for smart cache decisions** in CMS
  - Use the infrastructure provided to study/simulate performance of different approaches

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
- **CMS-integrated cache federation prototype** deployed and functionally tested
- A first **INFN proof-of-concept** implementation to enable **smart data cache at CMS has been deployed**

THANK YOU FOR YOUR ATTENTION