October 19 - 25, 2024

CHEP 2024

Enhancing CMS XCache efficiency: A comparative study of Machine Learning techniques and LRU mechanisms

J. Flix, P. Serrano, A. Sikora, A. Delgado, A. Pérez-Calero, F.J. Rodriguez, J. M. Hernández
for the CMS Collaboration

PIC port d'informació científica

IFAE Ciemat RES Red Española de Supercomputación España | digital 2026 Unió Europea Fons Europeu Next Generation Gobierno de España Plan de Recuperación, Transformación y Resiliencia Next Generation Catalunya Generalitat de Catalunya Departament de Recerca i Universitats

The HL-LHC context

Currently expanding and adapting the World-Wide LHC Computing Grid (WLCG) to accommodate **increased data processing demands expected at the HL-LHC era**

- Emphasis on the need for **cost-effective solutions** to manage the growing volume of data
- **Possible consolidation of storage** resources in fewer sites
- Introduction of **Content Delivery Network (CDN)** techniques as a strategy for optimized data access and resource utilization
- Focus on deploying lightweight storage systems (**data caches**) supporting traditional (Grid) and opportunistic (Cloud/HPC) compute resources
- **Boost task execution performance** by implementing efficient data caching mechanisms in close proximity to end users

The CMS context

Default behavior for **CMS jobs** is to process data at its location, but they also possess the **capability to access data remotely** through the **CMS XRootD federation**

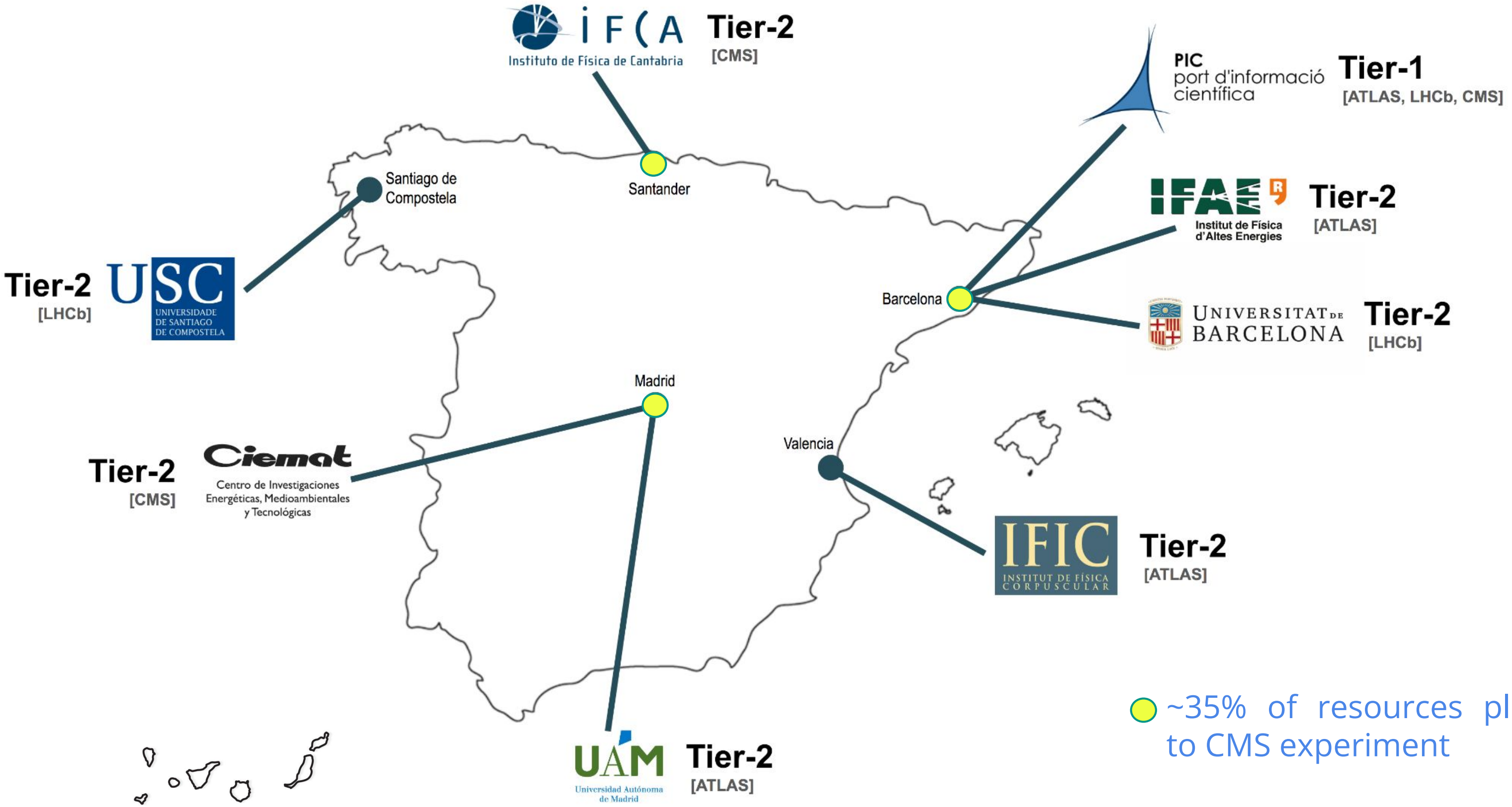
This setup offers a distinctive opportunity to evaluate the **advantages of employing data caches** to optimize CMS task execution performance

Processing campaigns are run where data is placed… but many CMS user analysis tasks read data remotely, hence **these would benefit with caching** [CHEP 2020]

We deployed an **XCache service** at **PIC Tier-1** (180 TB, embedded with regional and CMS XRootD re-directors) to cache user's data which is read from remote sites

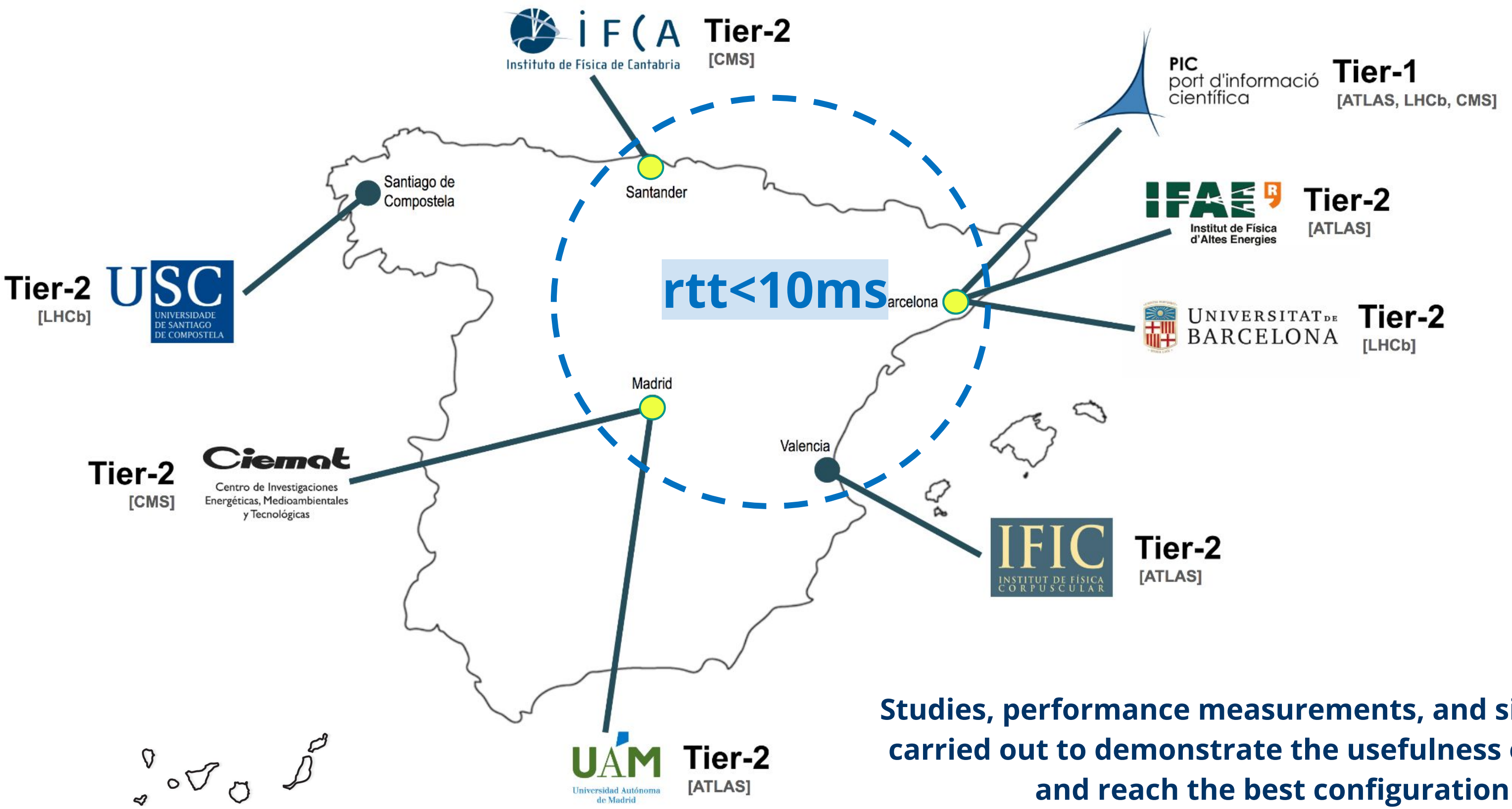
- XCache helps **reducing data access latency** and **improving CPU efficiency**: we observed a relative increase of 10% in CPUeff for analysis tasks in Spain [CHEP 2023]
- We demonstrated that analysis tasks can read from distant sites (**within rtt<10ms**) with **minimal degradations on the CPUeff** → door to deploy a regional cache in Spain

WLCG resources in Spain



~35% of resources pledged to CMS experiment

WLCG resources in Spain



Studies, performance measurements, and simulations carried out to demonstrate the usefulness of caching and reach the best configuration

Local or remote reads?

Analysis tasks logs (CRAB jobs) contain local or remote reading information, they are stored in Ceph at CERN and accessible by HTTP through **cmsweb.cern.ch** (availability: approx. last 3 months)

CERN's **SWAN** Big Data platform (Apache-Spark) gives access to the log's urls, that are downloaded and parsed at PIC using **Jupyter Notebooks + Dask**

→ **Determine # of input files** that have been read and **from where**, for each analysis job

→ **Size:** 0.5MB/log → 7.5 TB/month for all of CMS sites → 0.3 TB/month for CMS Spain (PIC+CIEMAT+IFCA)

Spain sample: **Jun. 2023-Apr.2024** [11-months]

→ **All:** 2.4M jobs **|** <file reads per job> ~2.8
6.6M file reads (27.5 PB) **|** 1.3M unique files (4.9 PB)

→ **Remote access:** 460k jobs | <file reads per job> ~ 4.6
2.0M file reads (10.8 PB) | 625k unique files (2.4 PB)

Local or remote reads?

Analysis tasks logs (CRAB jobs) contain local or remote reading information, they are stored in Ceph at CERN and accessible by HTTP through **cmsweb.cern.ch** (approx. last 3 months)

CERN's **SWAN** Big Data platform (Apache-Spark) gives access to the log's urls, that are downloaded and parsed at PIC using **Jupyter Notebooks + Dask**

→ **Determine # of input files** that have been read and **from where**, for each analysis job

→ **Size:** 0.5MB/log → 7.5 TB/month for all of CMS sites → 0.3 TB/month for CMS Spain (PIC+CIEMAT+IFCA)

Spain sample: **Jun. 2023-Apr.2024** [11-months]

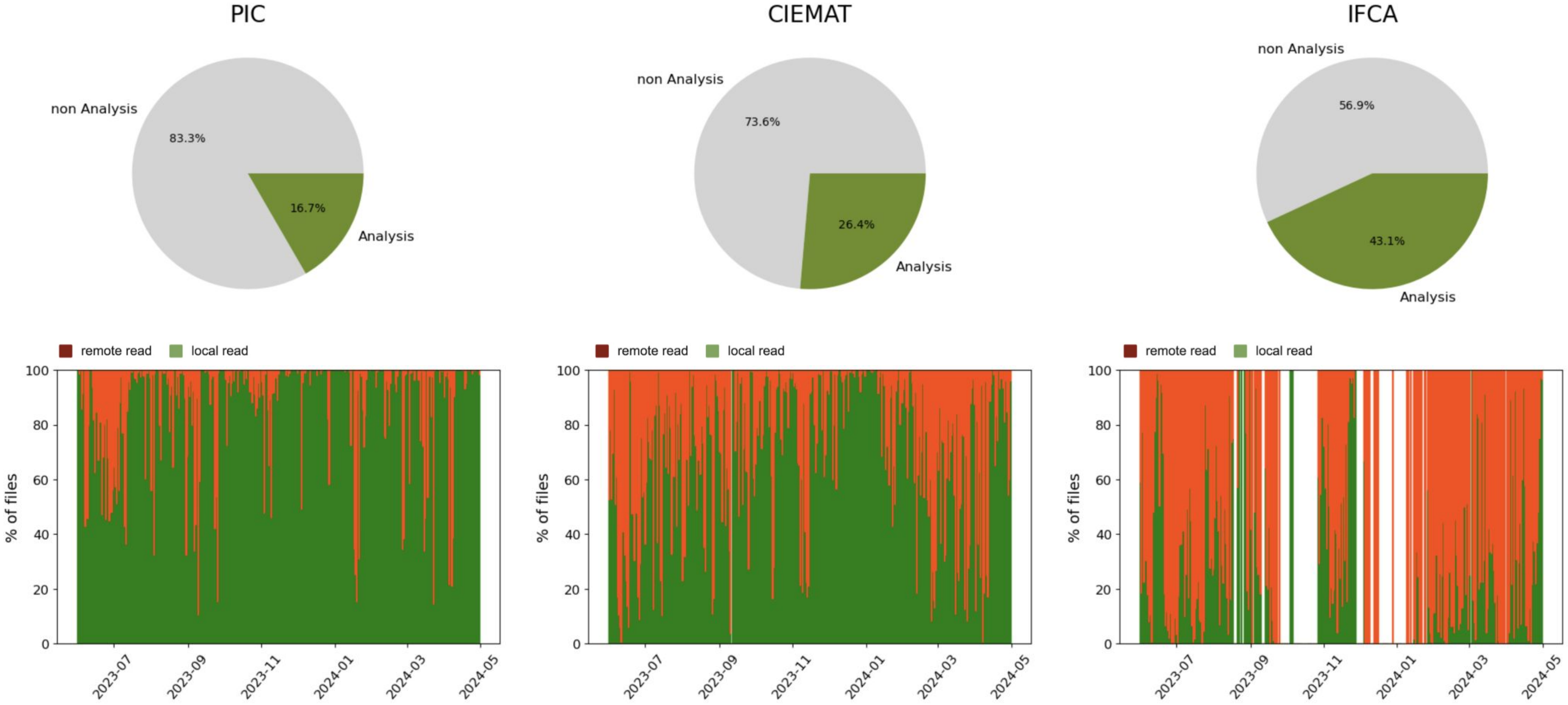
→ **All:** 2.4M jobs | <file reads per job> ~2.8
6.6M file reads (27.5 PB) | 1.3M unique files (4.9 PB)

→ **Remote access:** 460k jobs | <file reads per job> ~ 4.6
2.0M file reads (10.8 PB) | 625k unique files (2.4 PB)

~18% of remote accesses to MC PREMIX
big samples placed at FNAL and CERN to add pile-up
small reads & file re-reads are not very common
MC PREMIX excluded for caching

Remote reads: user's tasks in Spain

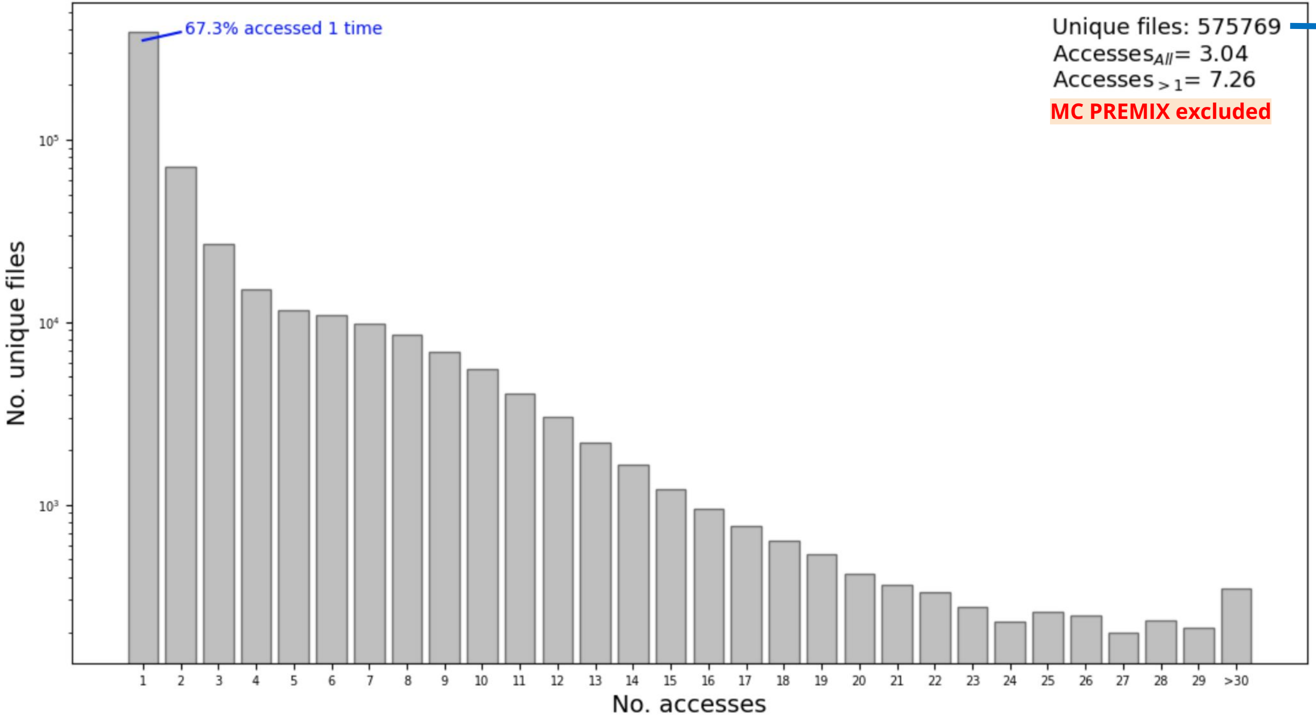
Jun. 2023-Apr.2024



Task distribution between Tier-1 and Tier-2 is **unequal**, with most analysis tasks executed at Tier-2 sites

Remote reads in Spain: file re-accesses

Jun. 2023-Apr.2024



1.52 PB (unique size)
6.36 PB (accessed size)

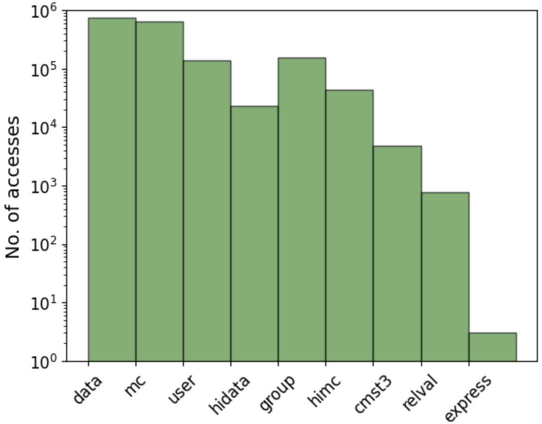
A cache of >1.52 PB
(from now on ∞-cache)
would exhibit
Hit Rate ~ 67.1%
Cache Network Out/In ~ 4.2
if caching complete files...

$$HitRate = \frac{hits}{hits + misses} = \frac{hits}{N_{accesses}}$$

Many re-reads → potential to improve user analysis tasks performance via cache

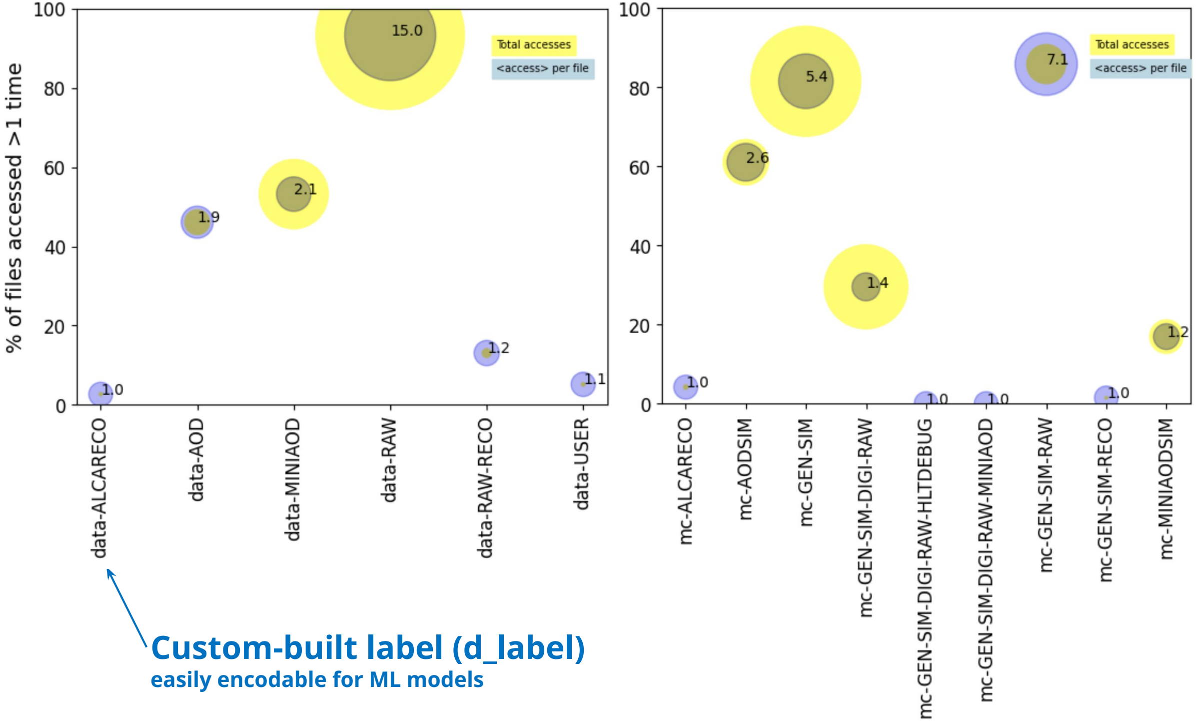
Remote reads in Spain: file re-accesses

Jun. 2023-Apr.2024



Data + mc

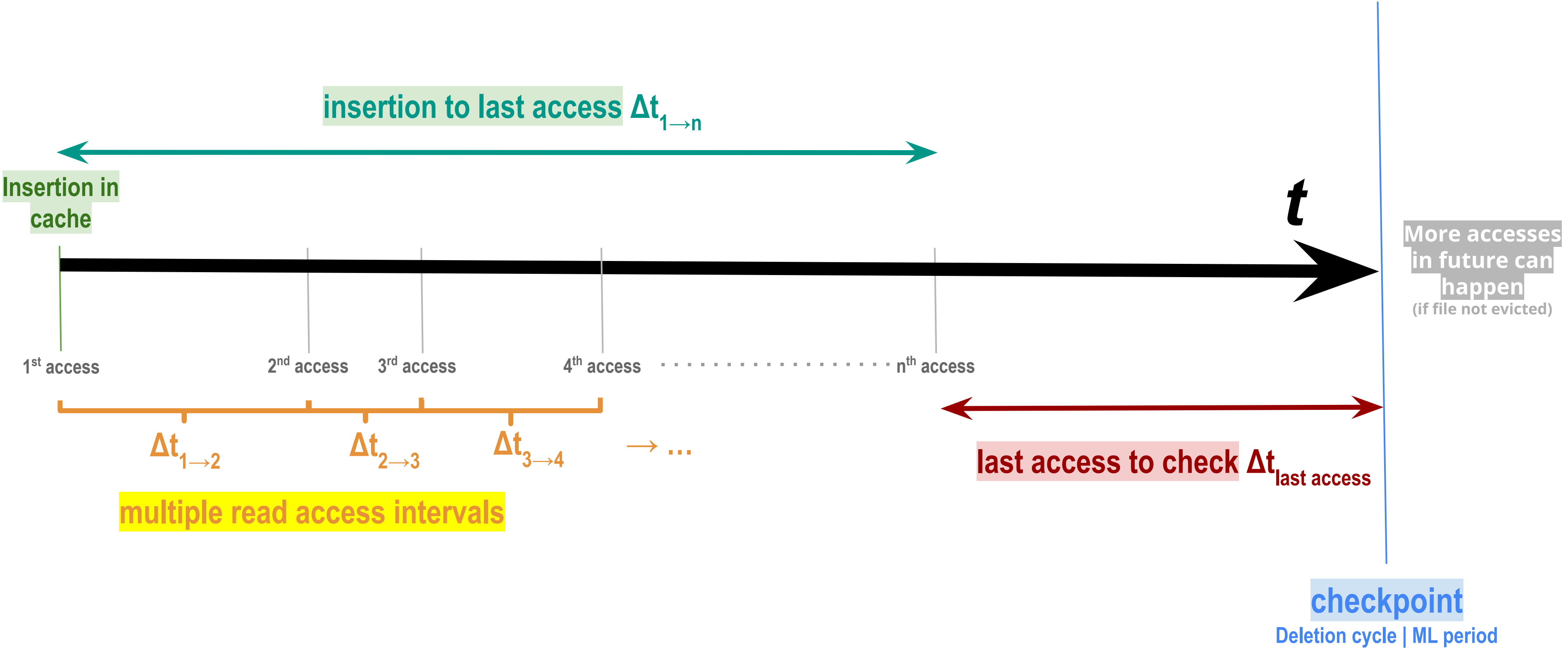
~80% of file accesses



Custom-built label (d_label)

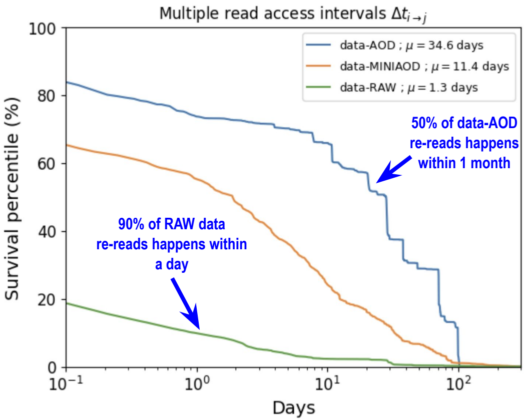
easily encodable for ML models

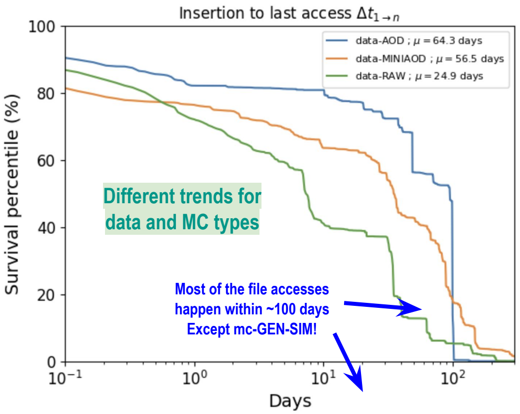
Remote reads in Spain: file re-accesses

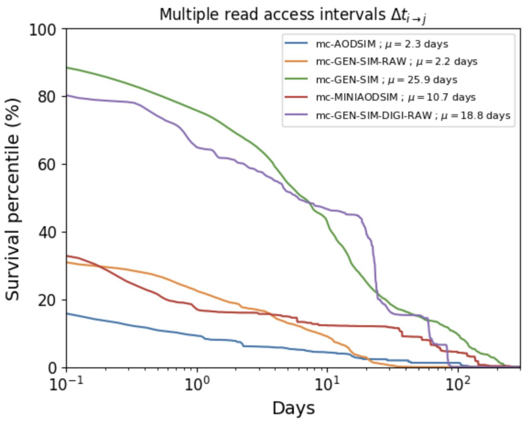


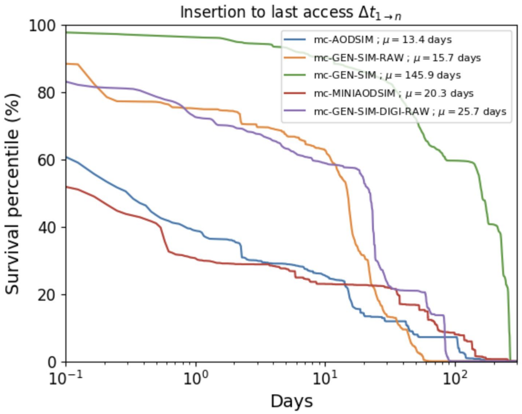
Remote reads in Spain: file re-accesses

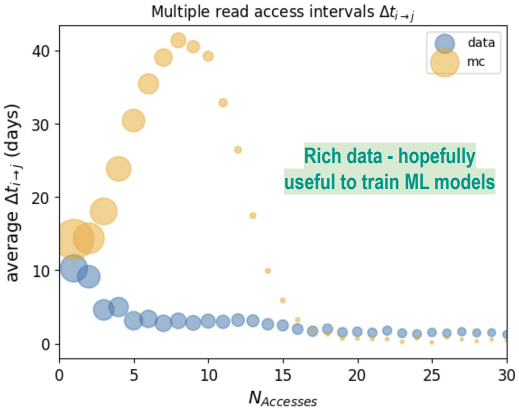
Jun. 2023-Apr.2024

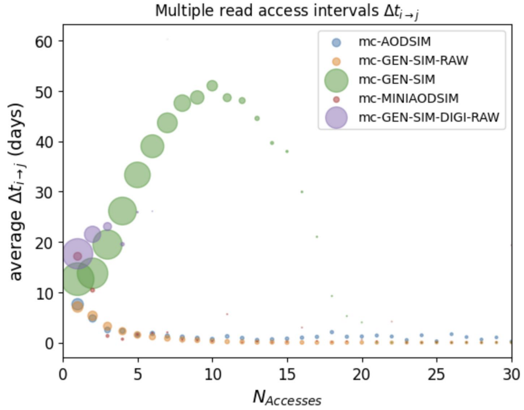












Simulating a cache for Spain

The **data access details** from all of the Spanish centers play a crucial role in determining the **optimal requirements for cache size** and **network connectivity**

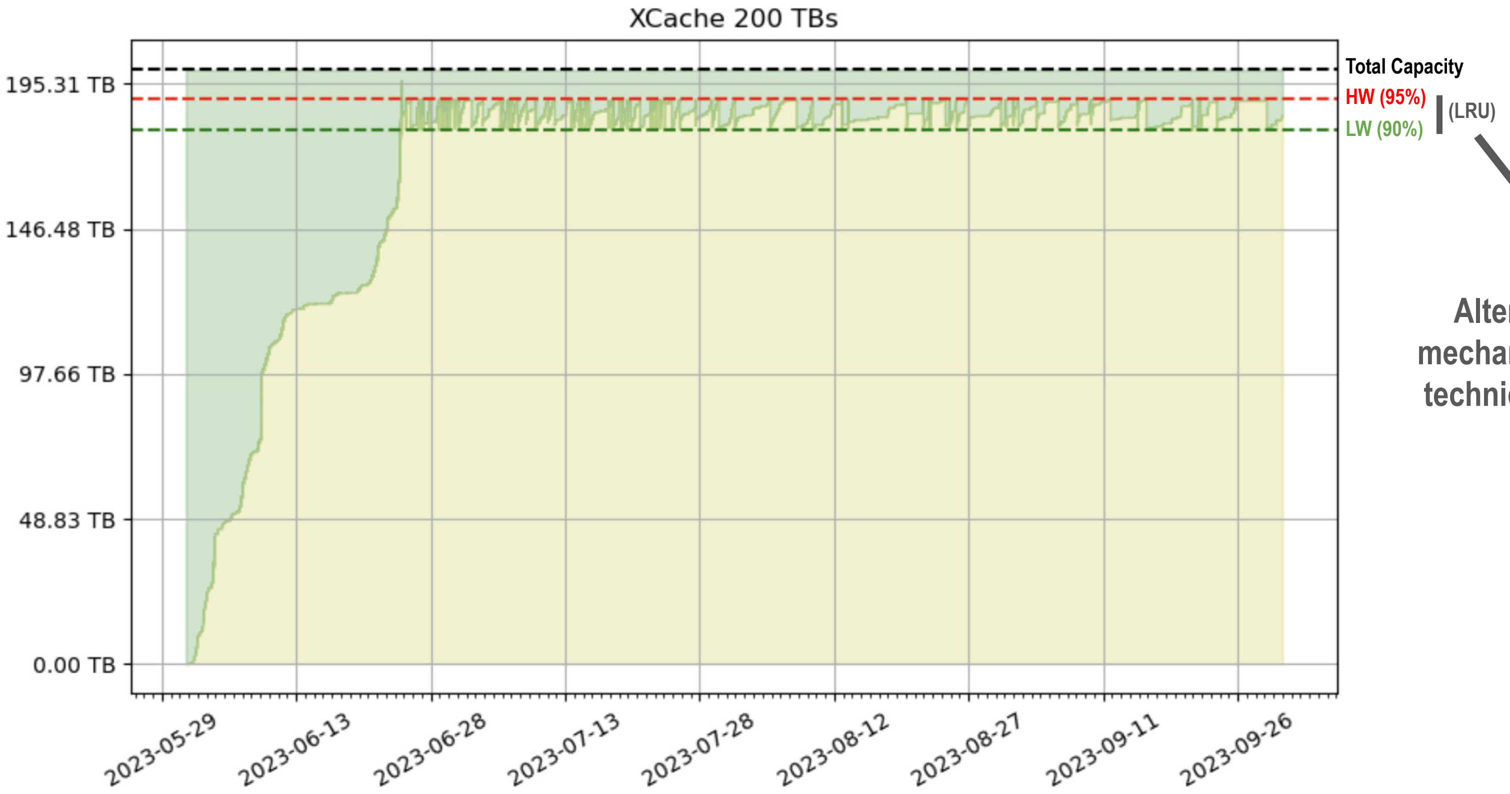
We explore all of the **user's job logs** for local or **remote** reading information

While most files are fully downloaded, **partial downloads** are considered based on insights from the production PIC XCache

We **emulate cache system population** based on these remote data accesses from production user's jobs

Deletion from the cache follows the **Least Recently Used (LRU) algorithm**: when occupancy exceeds 95% (High-Watermark - HW), file deletion is triggered until reaching the Low-Watermark (LW) of 90% for efficient space management

Simulating a cache for Spain: example



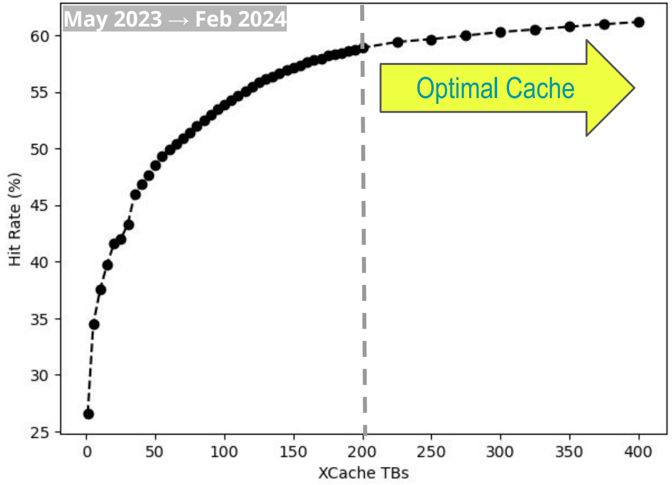
Alternative deletion mechanisms utilizing ML techniques investigated

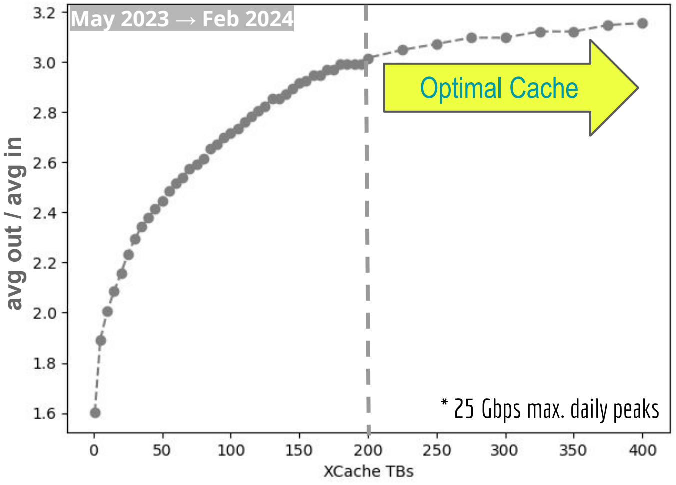
See later

Optimal cache across Spain

Simulating various cache sizes can identify the most efficient option for serving the region, determined by factors such as the **cumulative Hit Rate** (accesses to cached files over total accesses) and **network considerations [using LRU]**

EXAMPLE →



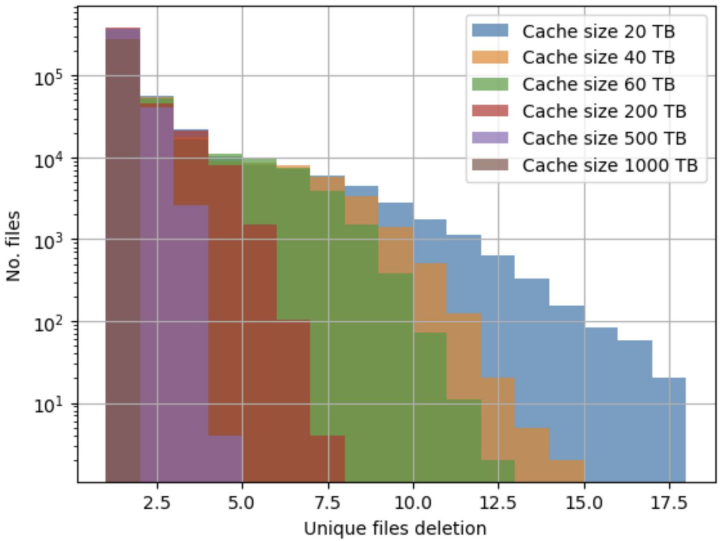


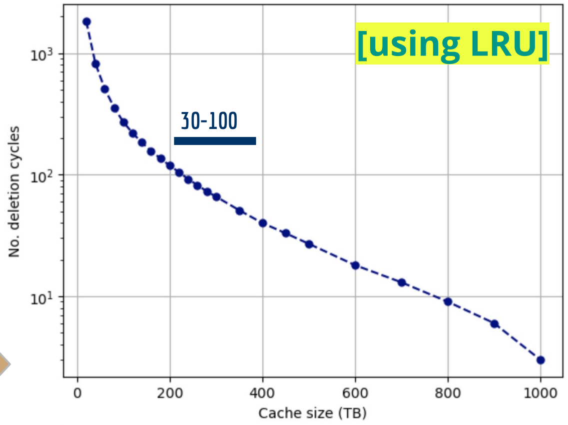
$$HitRate = \frac{hits}{hits + misses} = \frac{hits}{N_{accesses}}$$

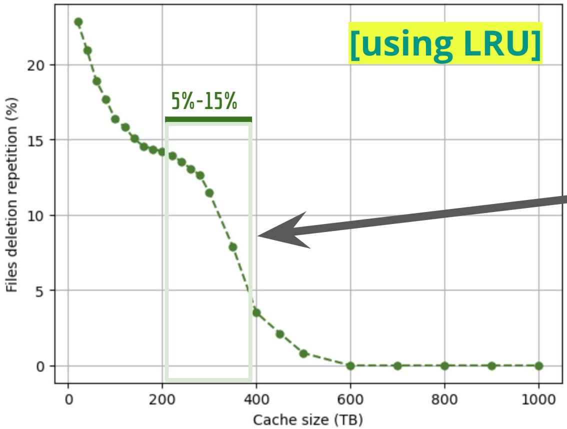
An effectively dimensioned cache typically exhibits a 3:1 ratio between outbound and inbound traffic

Simulated cache deletion cycles

Jun. 2023-Apr.2024





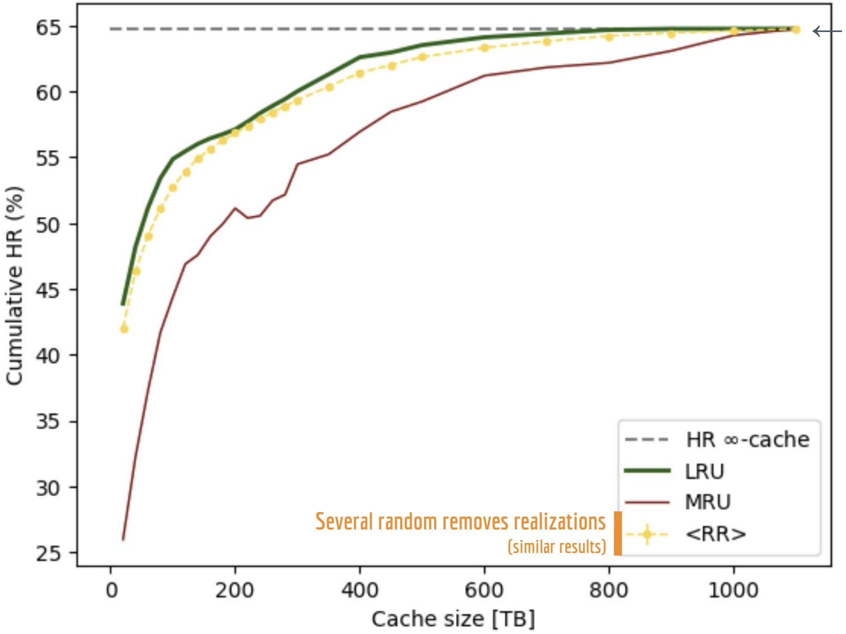


Even if the LRU mechanism is efficient, a fraction of the files that have been deleted in a deletion cycle are re-inserted at some point at the cache

How good is LRU mechanism?

Jun. 2023-Apr.2024

Most Recently Used (**MRU**) discards the most recently used items first - unlike **LRU**, which discards the least recently used items. We can also explore random removes (**RR**) at deletion cycles



← The larger the cache, the fewer deletions and reduced impact

Even if the LRU mechanism is efficient, more than 65% of the files are accessed once, hence **RR method is close to LRU**

Can we improve the cumulative hit rate by using machine learning techniques to maximize retention of the most popular datasets? By how much?

Applying ML to cache eviction policies

Neural network model (single hidden layer of 8 neurons, **ReLU** is used as the activation function for the hidden layer, and **Sigmoid** is used for the output layer), predicting if a file *in a given period* will be read in the next *n-days* **[Classification model]**

Training periods can be decoupled from deletion cycles, but should be **long enough to capture relevant data access patterns**

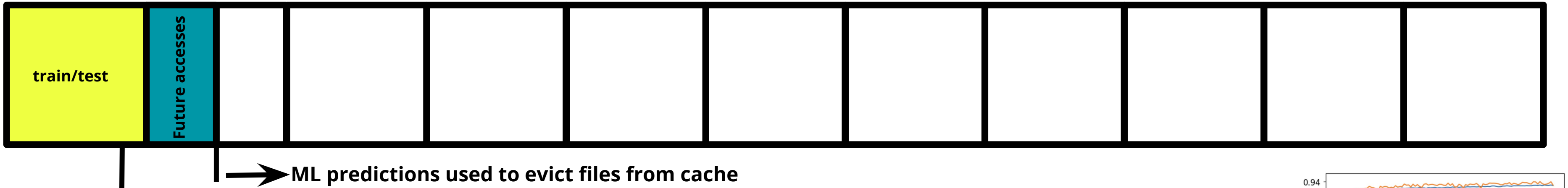
Once a period is selected, **features** from the cached files can be computed (normalized):

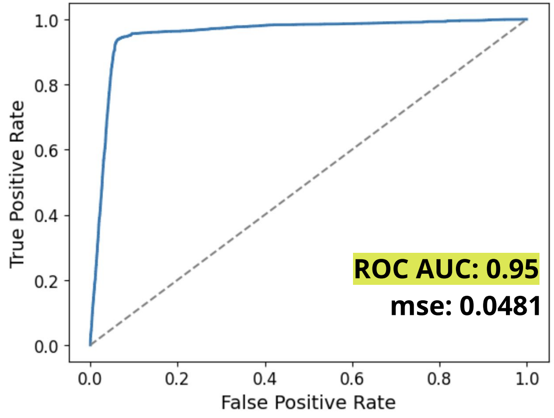
	filesize	d_label size (files)	d_label size (volume)	Total Accesses	5th last read access	4th last read access	3rd last read access	2nd last read access	last read access	deltaT_1_last	recency_1st	d_label_encoded	future_acc
1316046	0.223090	0.751239	0.434330	3	-1.00000	-1.000000	0.209650	0.055880	0.010448	0.199203	0.209650	0.400000	0
1237877	0.207711	0.751239	0.434330	1	-1.00000	-1.000000	-1.000000	-1.000000	0.102503	-1.000000	0.102503	0.400000	0
81459	0.116694	0.897340	0.346505	1	-1.00000	-1.000000	-1.000000	-1.000000	0.651558	-1.000000	0.651558	0.366667	1
1167165	0.011649	0.228312	0.007606	1	-1.00000	-1.000000	-1.000000	-1.000000	0.225011	-1.000000	0.225011	0.200000	0
1281625	0.222903	0.751239	0.434330	2	-1.00000	-1.000000	-1.000000	0.058849	0.055189	0.003660	0.058849	0.400000	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...

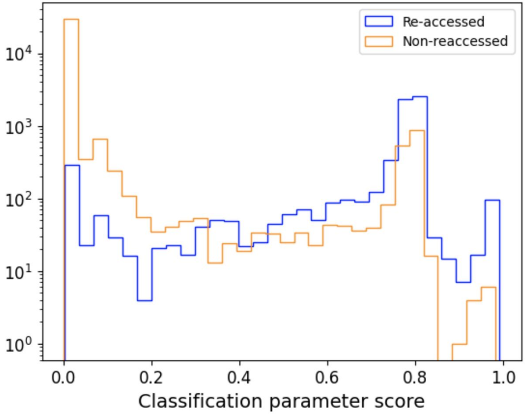
Future accesses in the next n-days
To train → predict

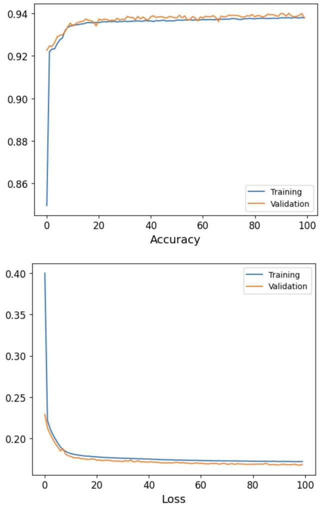
Applying ML to cache eviction policies

11-months period: **1 month to train** & **½ month to identify future accessed files**



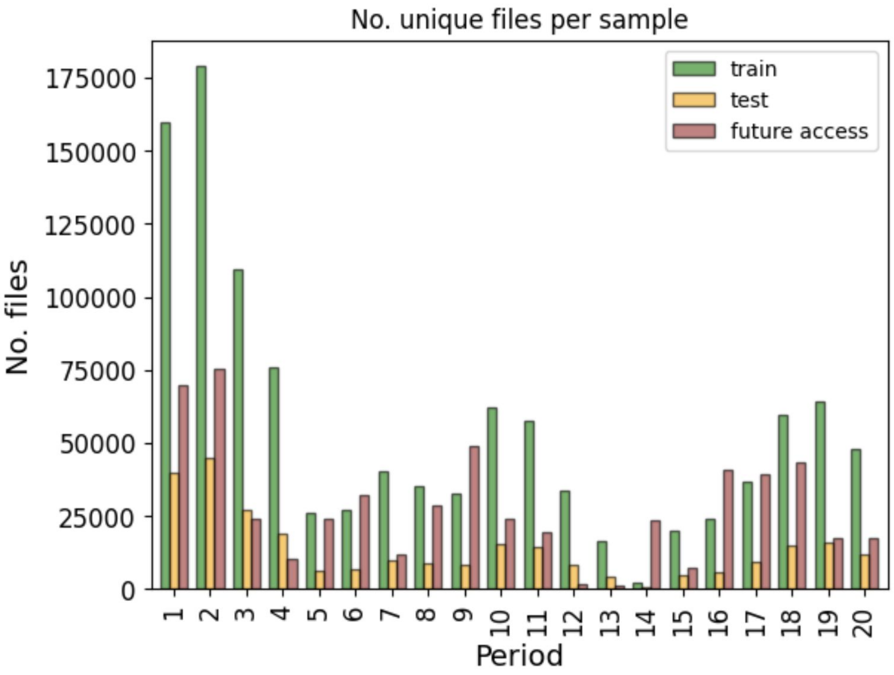


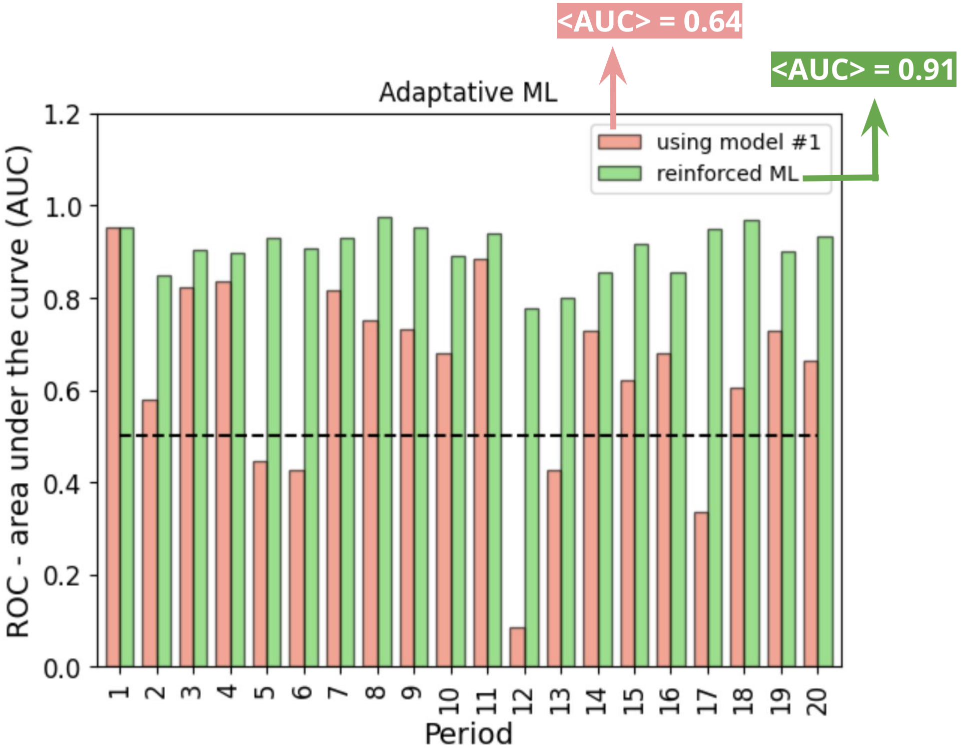




Applying ML to cache eviction policies

The types of data accessed fluctuate over time, indicating a **need for adaptive machine learning technique**





Conclusions

Our work exhibits the **advantages of using data caches** for optimized data access and resource utilization

XCache deployed in the PIC **can efficiently serve data** just as effectively as if it were read locally at each of the Spanish CMS centers (sites within 10 ms RTT)

Analyzing user job logs is essential for **simulating data caches**, identifying optimal **cache size** and **network** connectivity requirements, and exploring various cache **deletion mechanisms**

- Introducing data **caches reduces the XRootD traffic generated by 'user jobs'** remote reads by (at least) a factor of 3

Machine learning can help in **improving Hit Rate** in caches (yet to be applied in this case). **Pre-fetching datasets** could also improve cache performance and it will also be investigated

CMS

PIC
port d'informació científica

dziękuję!

Project co-funded by the Recovery and Resilience Facility (Spain), and the European Union – NextGenerationEU

GOBIERNO DE ESPAÑA — MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

Financiado por la Unión Europea NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

and the Red Española de Supercomputación (RES) through grants FI-2020-2-0025, FI-2020-3-0017, FI-2021-1-0005, FI-2021-2-0010, FI-2021-3-0025, FI-2022-1-0001, FI-2022-2-0026, FI-2022-3-0036,FI-2023-1-0019, FI-2023-2-0007, FI-2023-3-0006, FI-2024-1-0034, FI-2024-2-0038, FI-2024-3-0036

Research projects PID2019-110942RB-C21, PID2020-113807RA-I00, and PID2022-142604OB-C21 funded by:

MINISTERIO DE CIENCIA E INNOVACIÓN

Cofinanciado por la Unión Europea

AGENCIA ESTATAL DE INVESTIGACIÓN