PROCEEDINGS OF SCIENCE

Content Delivery Network solutions for the CMS experiment: the evolution towards HL-LHC

José Flix,[a,b,*] Carlos Pérez,[a,b] Anna Sikora,[c] Paula Serrano[c] and the CMS Collaboration

[a] *CIEMAT*
Basic Research, Scientific Computing Unit, 28040, Madrid, Spain
[b] *PIC*
08193, Bellaterra (Barcelona), Spain
[c] *Universitat Autonoma de Barcelona (UAB)*
08193 Bellaterra (Barcelona) Spain

E-mail: jflix@pic.es

PoS(ISGC2024)041

The Large Hadron Collider (LHC) at CERN in Geneva is undergoing a significant upgrade in anticipation of a tenfold increase in proton-proton collisions expected in its forthcoming high-luminosity phase, starting by 2029. This necessitates an expansion of the World-Wide LHC Computing Grid (WLCG) within a constant budgetary framework. While technological advancements offer some relief for the expected increase, numerous research and development projects are underway. Their aim is to bring future resources to manageable levels and provide cost-effective solutions to effectively handle the expanding volume of generated data. In the quest for optimised data access and resource utilisation, the LHC community is exploring Content Delivery Network (CDN) techniques to optimize data access and resource utilization. A comprehensive study focuses on implementing data caching solutions for the Compact Muon Solenoid (CMS) experiment, particularly in Spanish compute facilities, revealing benefits for user analysis tasks. The study details the implementation of a data caching system in the PIC Tier-1 compute facility, discussing its positive impact on CPU usage and exploring optimal requirements and cost benefits. Furthermore, it investigates the potential broader integration of this solution into the CMS computing infrastructure.

International Symposium on Grids and Clouds (ISGC2024)
24 -29 March, 2024
Academia Sinica Computing Centre (ASGC), Institute of Physics, Academia Sinica Taipei, Taiwan

[*] Speaker

1. Introduction

In response to the escalating demands for computing power and storage during the HL-LHC period, LHC experiments have launched an extensive R&D program [1]. This program aims to propose novel ideas and techniques that will shape the evolution of their computing models and services, focusing not only on optimizing existing tools and technologies but also on creating innovative solutions for data management and processing capabilities. The primary objective is to ensure that experiments can derive maximum scientific output from the expected vast data generated during the HL-LHC phase.

Projections of necessary computing resources for the HL-LHC, compared with estimated resource availability under a flat-budget model where no additional funding is expected, have been undertaken by the experiments. While technological evolution suggests an annual resource increase of 10-20% at a constant budget, this rate appears insufficient to cope with the requirements at the HL-LHC era. Without the implementation of novel ideas, the available resources might not meet the future computational requirements in the World-Wide LHC Computing Grid (WLCG).

To address this challenge, R&D efforts are focusing on improving compute power through GPU integration, partial application vectorization, and utilization of opportunistic resources and HPC centers. However, storage services present complex challenges. A new approach is being considered to optimize cost-efficient operations and storage by consolidating resources into fewer WLCG sites, forming the WLCG Data-Lake model [2, 3]. This model involves a few centers managing most LHC data and serving smaller centers (or even Analysis Facilities) through simplified data caches.

Inspired by Content Delivery Networks (CDNs), this innovative approach optimizes hardware and operational costs through strategically placed caches near high data demand points. High-bandwidth networks interconnecting WLCG sites, namely LHCONE and LHCOPN [4], must be appropriately dimensioned to handle data flow within the Data-Lake. Ongoing R&D tasks focus on network virtualization and programmable networks [5] to ensure agile, swift, and cost-effective network infrastructures. To complement these efforts, a key focus lies in deploying lightweight storage systems, specifically data caches, which can seamlessly support both traditional (Grid) and opportunistic (Cloud/HPC) compute resources. This approach ensures flexibility and scalability in accommodating diverse computing environments. Moreover, to elevate task execution performance, there is a concerted effort to implement efficient data caching mechanisms in close proximity to end users. This proximity facilitates quicker data retrieval and processing, thereby enhancing overall system responsiveness and user experience.

The paper is organized as follow. Section 2 describes the the Compact Muon Solenoid (CMS) experiment and the current data caching solution (XCache) deployed in Spain. Section 3 presents a comprehensive analysis and impact of remote reads on the performance of CMS user jobs. Section 4 proposes an optimal dimensioning of a single CMS XCache for Spanish Tiers. Finally, Section 5 concludes the proposed approach and Section 6 indicates future work.

2. The CMS context

The default operational protocol for CMS (Compact Muon Solenoid [6]) jobs involves processing data at its designated location. However, CMS also boasts the capability to remotely access data via the CMS XRootD federation [7]. This unique setup presents a golden opportunity to explore the benefits of integrating data caches into the CMS ecosystem, thereby optimizing task execution performance. Given that major processing campaigns are typically conducted where data is, the implementation of CDN techniques become particularly advantageous for CMS user analysis tasks. These techniques ensure streamlined data accessibility and processing efficiency across the CMS network.

To fortify these initiatives, we have strategically implemented an XCache service in both the PIC Tier-1 and CIEMAT Tier-2 facilities. This service acts as a channel for storing user data from remote sites, significantly decreasing data access delays, improving CPU utilization, and potentially minimizing storage needs in the area. A comprehensive array of studies, performance measurements, and simulations has been conducted to elucidate the utility of the XCache service and ascertain the optimal configuration settings. These endeavors underscore our commitment to refining CMS data management processes and maximizing operational efficacy.

2.1 The CMS XCache deployed in Spain

With its seamless integration into the XRootD protocol, XRootD proxy cache (XCache [8]) is considered as the preferred caching service for scientific data within WLCG [9], playing a pivotal role in the construction of a CDN-based infrastructure. Serving as a proxy-based data cache system, XCache utilizes a physical cache to efficiently manage frequently accessed data. When a data request is made, the proxy swiftly checks the physical cache for the required data and promptly delivers cached content to the client. Should the data not be cached, the proxy server retrieves it from the appropriate storage server, leveraging a hierarchy of re-directors, and subsequently caches it for future requests, ensuring optimal data accessibility and performance.

After successful functional tests and service validation, the production deployment of an XCache server at the PIC WLCG Tier-1 in Barcelona was finalized by 2021. This server boasts a current deployed capacity of 175 TiB, achieved through the aggregation of 6TB HDDs in RAID6 configuration. It is powered by 2xCPUs E5-2650L v3 (with HT enabled, totaling 48 cores), 128 GB RAM, and an active-active 10 Gbps Network Interface Card (NIC). Operating on the latest XRootD version, specifically XRootD 5.5.1, this service has been instrumental in facilitating the studies presented in this work.

Figure 1 provides an overview of how the XCache implementation at PIC Tier-1 integrates with both regional and other higher-level CMS XRootD redirectors. In this configuration, when the required input data for a job executing on a compute node is not locally available, the fallback mechanism first engages with the XCache service. If the data is already cached, it is promptly served; otherwise, it is fetched from a remote site using the CMS XRootD redirector infrastructure and then delivered from the data cache to the compute node. Although the primary role of XCache is file retrieval, it can also be customized to proactively retrieve data with read-ahead capabilities. The current setup allows for data retrieval in blocks of 10x, each consisting of 50 kB in size.

The XCache service implements the Least Recently Used (LRU) deletion algorithm, a robust method for efficiently managing outdated and unused data [10]. The LRU algorithm organizes cached files based on their usage and timestamps, identifying files that have remained untouched for an extended period as candidates for removal. Deletion is triggered by watermarks, which represent specific occupancy thresholds. When the occupancy exceeds the High-Watermark (HW) threshold

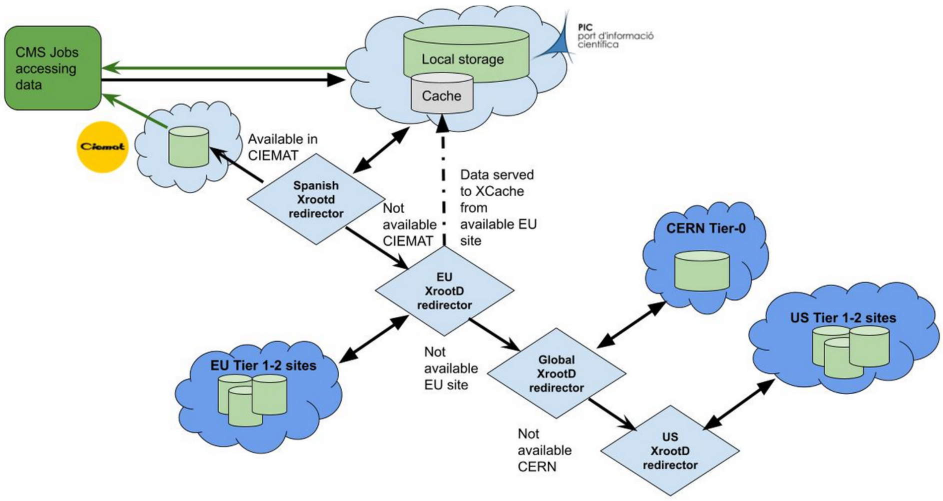

Figure 1: PIC Tier-1 XCache integrated with the regional and other higher-level CMS XRootD redirectors.

of 95%, the algorithm initiates file deletion until reaching the lowest occupancy range, known as the Low-Watermark (LW), set at 90%.

CMS utilizes local configuration files to establish algorithmic rules for mapping Local File Names (LFNs) to Physical File Names (PFNs) at each site. These files also outline the access protocols and data servers to be employed, determined by regular expressions applied within them, accommodating both local and remote accesses. Through this tailored CMS configuration, frequently accessed data of specific types can be directed to the XCache service, while data not matching these rules can utilize CMS's global redirection infrastructure as a fallback mechanism. This configuration enables the XCache service to effectively retain frequently accessed files, enhancing its overall performance. It's worth emphasizing that accurate knowledge of data popularity is crucial for ensuring optimal functionality.

The current configuration of the PIC XCache service encompasses caching all CMS data types, with the exception of test files, intermediate files, and input pile-up sample files. These latter files, which are sizable and accessed randomly by simulation tasks, can reach up to 1 PB in data samples and are located at both FNAL Tier-1 and CERN Tier-0. The PIC XCache service supports both the PIC Tier-1 and fifty percent of the compute nodes at the CIEMAT Tier-2 facility (in Madrid), collectively supporting 4500 actively utilized CPU cores by CMS across both sites. On a daily basis, the XCache service efficiently manages access to approximately 5000 files, effectively serving an average of 15 TB of data per day.

Figure 2 depicts the file access pattern within the XCache system, delineating between hits (files already cached) and misses (files not cached, subsequently fetched to the data cache on request). As illustrated, popular files consistently occupy the cache, leading to a continual increase in hit rates. It is noteworthy that roughly half of the accessed files during this period are already cached, underscoring the efficiency of the caching mechanism.

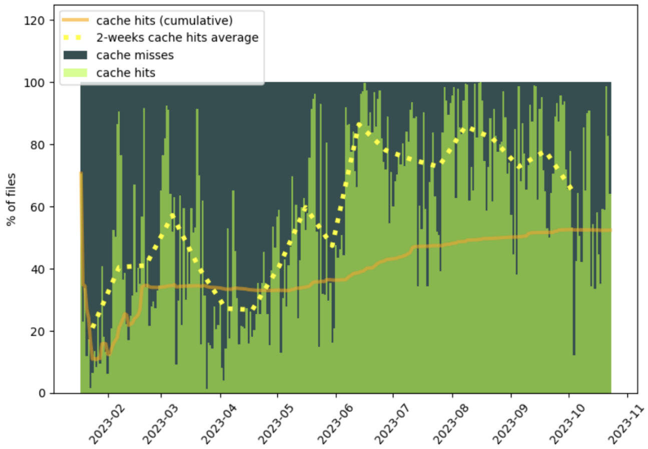

Figure 2: PIC XCache hits and misses over five months. The number of hits increases with popular files remaining within the cache.

3. Effects of remote reads on user job performance

The impact of remote reads on CMS user jobs may potentially cause lag due higher latencies in data retrieval from remote servers. These remote reads entail fetching data from storage locations that are not locally available, requiring additional network communication and potentially encountering latency issues. Consequently, user jobs may experience longer execution times, decreased throughput, and overall diminished performance. Understanding and mitigating the effects of remote reads is essential for optimizing the efficiency and productivity of CMS workflows.

3.1 A controlled assessment

For optimal performance and efficiency of the XCache system, careful consideration must be given to the types of files suitable for caching, and an exploration of the potential benefits of employing the service. Drawing on prior research and usage patterns, AOD files accessed by users' analysis jobs have been identified as the most appropriate files for storage in a data cache [11]. CMS AOD files contain a reduced set of reconstructed Physics objects, tailored for higher-level analysis, while MINIAOD files encapsulate only the pertinent information for swift processing and rapid analysis. The latter files, MINIAOD, prove to be the most accessed when conducting final analyses for research. Given that Analysis jobs access MINIAOD, these are the jobs identified as most likely to benefit from implementing XCache in a production environment. To evaluate the potential advantages of caching MINIAOD samples, a comprehensive set of controlled Analysis jobs was executed in a production-like environment.

The focal point involved employing a benchmark job that required MINIAOD as input data, both with and without the XCache infrastructure deployed at PIC. For that purpose, a real Analysis job from CMS was elected, using the MuonAnalyzer package [12] within the ROOT framework [13]. Using C++ programming language, it conducts muon data analysis, employing a technique known as tag-and-probe. The tag-and-probe method, executed within an event loop, selects a "tag" particle based on specific criteria, facilitating in-depth analysis, from identifying particle types to meeting precise energy requirements.

During the execution, certain tasks extended beyond the main event loop, such as initialization and output file writing. While a smaller event count reduced stage-out times and outputs, the execution task overhead remained relatively constant, irrespective of the number of processed events. This overhead was well-defined, occurring both before and after the event loop.

In the initial research phase, the goal was to identify the optimal number of events for analysis from a selected MINIAOD file, with a focus on optimizing CPU efficiency within the main event loop. The benchmark job, using a single-core at an isolated compute node at PIC, demonstrated an initialization time of approximately 28 seconds before entering the event processing loop. Processing the entire set of 110,323 events in the 2.9 GB template MINIAOD file took around 6.1 ± 0.5 HS06·hours[1], with a high CPU efficiency of 98.3 ± 0.2%. Peak memory usage reached 1.47 ± 0.05 GB, and the average input file read throughput during execution was approximately 2.46 ± 0.07 MB/s.

After selecting the benchmark job and gaining initial insights into its characteristics, a controlled environment was established at PIC. The objective was to submit jobs accessing MINIAOD files from both local and various remote sites to evaluate how the CPU efficiency of the benchmark job deteriorates during data reads from distant locations. To prevent conflicts or interference with other running jobs in the compute node, the PIC compute node designated for these experiments was isolated and removed from the local HTCondor pool. In the initial phase of results analysis, the CPU efficiency for a series of sequentially executed analysis jobs at PIC was scrutinized. Once the controlled environment was validated, approximately 25 sites were meticulously selected for further investigation to expand the scope of results. The benchmark job continued to utilize a single CPU core and relied on the LHCOPN and LHCONE networks for accessing the input MINIAOD data. We assessed the average CPU efficiency of the benchmark jobs when reading data remotely from these sites.

The entire test spanned 25 days, totaling 6.5k HS06·hours in the PIC compute node designated for the test. Figure 3 illustrates the average CPU efficiency of the benchmark jobs conducted at PIC when reading data either from local storage (labeled as T1_ES_PIC, with data stored in PIC XCache) or from remote storage systems for sites placed in Europe (EU) and outside Europe (non-EU), as a function of latency (round-trip time, in ms). Apart from a few sites displaying significantly poor or outstanding performance despite their latency, a discernible trend of CPU efficiency degradation emerges for tasks reading from remote sites.

Accessing data from sites in France, Italy, or CERN while the job runs at PIC results in a noticeable decline in CPU efficiency, dropping from 98% to 80-85% levels. These sites, located at distances of 650 km, 1,100 km, and 1,000 km, respectively, exhibit similar round-trip time (rtt) values. The FNAL [15] Tier-1 site in Chicago, USA, situated approximately 7,000 km away from the PIC center (with a latency of 150 ms), experiences a significant reduction in the mean CPU efficiency of these jobs, decreasing to approximately 65%. The furthest site tested in this study was in South Korea (KISTI [16]), at a distance of approximately 10,000 km from PIC.

While remote data access over transatlantic or transpacific networks is not the typical CMS procedure, this study was conducted to evaluate the benefits of bringing data from exceedingly

[1] HEP-SPEC06 (HS06) benchmark assesses CPU core performance in high-energy physics [14]. Conversion to walltime (core·h) for CMS analysis tasks depends on workload and system factors, approximating 12.06 HS06·core per 1 walltime core·hr at Tier-1s.

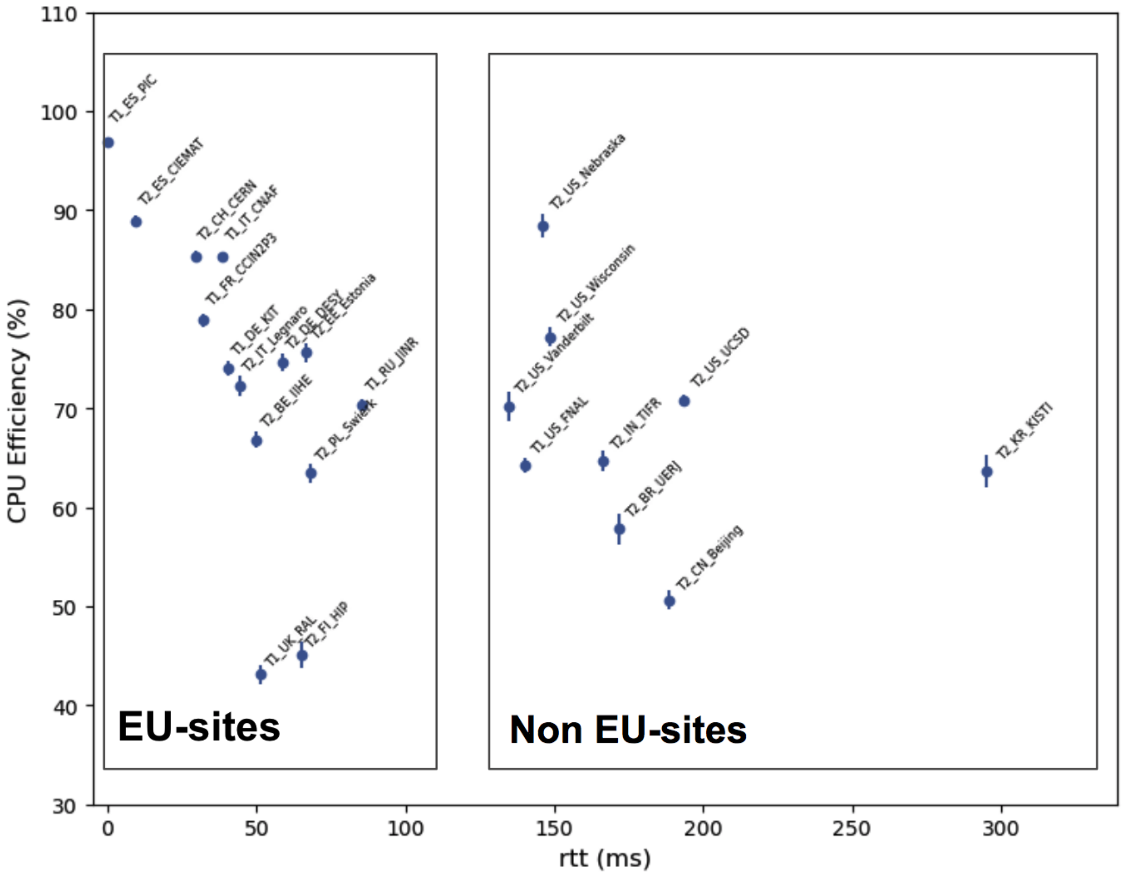

Figure 3: Average CPU efficiency of the benchmark analysis jobs executed at PIC when reading data from local, EU and non-EU sites, as a function of the site's latencies (round-trip-time (rtt), in ms)

distant locations closer to compute nodes.

3.2 Effects on real user's jobs

Half of the CIEMAT compute nodes were configured to utilize the PIC XCache as a fallback mechanism, while the remaining nodes directly interfaced with the CMS XRootD redirector system. In both configurations, if input data files were locally available, they were directly read from CIEMAT's local storage. Otherwise, jobs reading remote data encountered two possible scenarios: half of them accessed the XCache server at PIC, benefiting from caching techniques, while the other half still fetched data from various remote (and distant) sites. This setup enabled us to examine the impact of caching techniques compared to the standard operational approach, allowing for a performance comparison across all types of user jobs submitted by CRAB [17].

Over a span of 100 days (from January to April 2023), we conducted measurements and comparisons of the average CPU efficiency for user's jobs submitted with CRAB executed on CIEMAT compute nodes with XCache enabled or disabled. Subsequently, the computed results revealed that the average CPU efficiency for jobs with XCache enabled was 77.2 ± 0.9%, while it was 70.4 ± 1.0% for jobs with XCache disabled. The normalized CPU efficiency distribution of

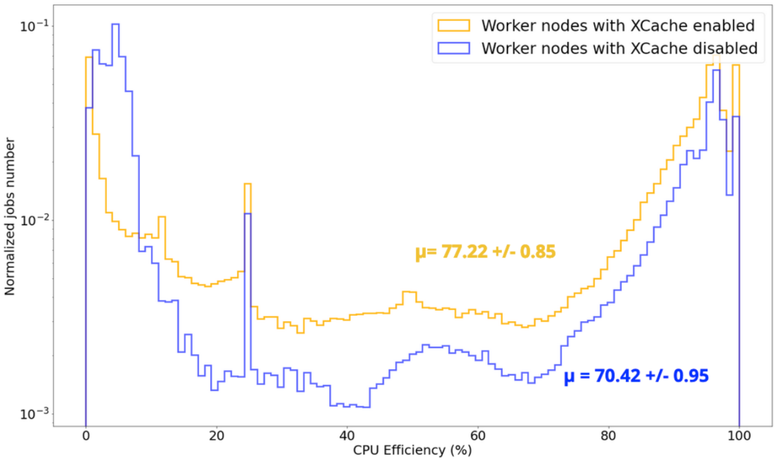

Figure 4: CPU efficiency distribution for CRAB jobs executed at CIEMAT compute nodes when XCache is enabled or disabled, in the period covering 100 days (from January to April 2023)

CRAB jobs run on both sections of the CIEMAT compute farm is depicted in Figure 4. Remarkably, the ability of XCache to serve popular files with reduced latency significantly enhanced the overall efficiency of analysis tasks on CIEMAT compute nodes when XCache was enabled. This highlights the potential for performance improvement and resource savings in the region through the utilization of the XCache service. Either CIEMAT can perform +11% more computational work, or deploy 11% less CPU resources to perform the same work as previously done without using the XCache service.

This study benefited from the readily available information of CIEMAT compute nodes where CRAB jobs were executed, accessible via central CMS job monitoring. However, the monitoring of CRAB jobs posed a challenge due to its inability to accurately trace whether jobs read input files locally or remotely. To tackle this issue, we devised a solution to track the origin of file reads, distinguishing between local and remote accesses. The CRAB logs contain pertinent details regarding file read origins. The CRAB logs do contain information about the origin of file reads, either local or remote. These logs are stored in Ceph at CERN and can be accessed via HTTP through *cmsweb.cern.ch*, typically covering the past three months. To harness this valuable data, we have leveraged CERN's SWAN Big Data platform [18], powered by Apache Spark, to access the log URLs. At PIC, we have developed a process using Jupyter Notebooks and Dask to download and parse these logs. This approach enables us to precisely determine the number of input files read and their source for each analysis user job executed in CIEMAT.

Figure 5 provides a breakdown of the average CPU efficiency for CRAB jobs conducted at CIEMAT, categorized by data access from local storage systems, XCache at PIC, or remote CMS sites. Remarkably, when reading data locally (T2_ES_CIEMAT), the average CPU efficiency soars to approximately 95%. Equally noteworthy is the close resemblance in CPU efficiency observed when data is fetched from the PIC XCache and served to CIEMAT, mirroring the efficiency of local storage access. The figure also accentuates a decline in CPU efficiency when data is retrieved from remote and distant sites. Beyond network latency, various factors such as the load on the remote storage system and WAN configuration/network load can influence CPU efficiency. Nonetheless,

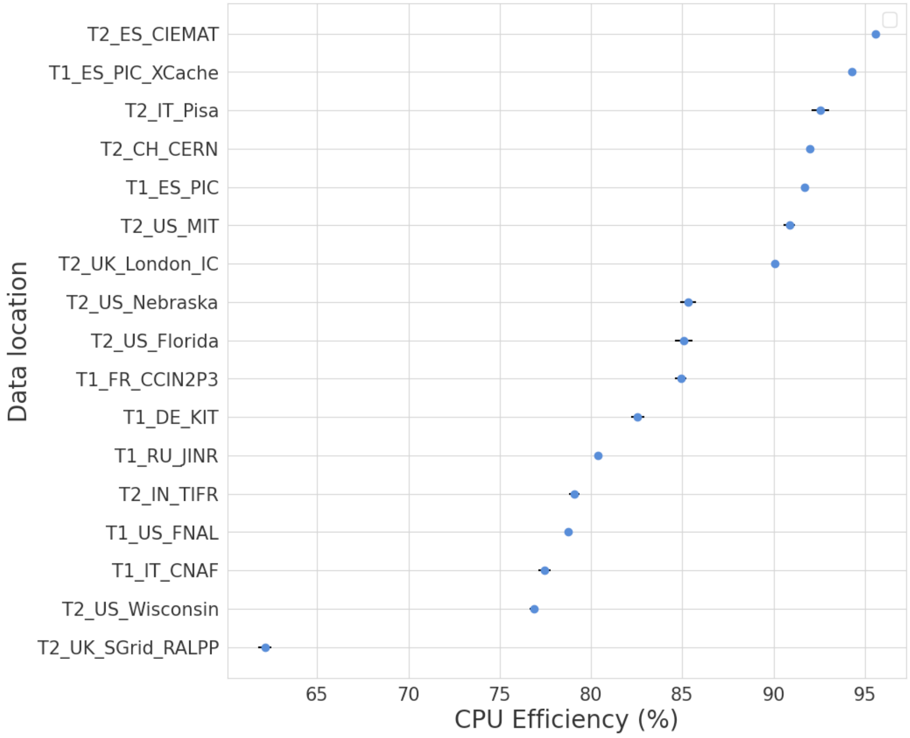

Figure 5: Average CPU efficiency of CRAB jobs executed at CIEMAT when reading data from local,XCache or remote sites, in the period covering 100 days (from January to April 2023).

it's worth noting the relatively insignificant degradation in CPU efficiency for CRAB jobs accessing data from PIC XCache, highlighting its robust performance despite the remote access.

The findings of this study offer valuable insights into enhancing CPU efficiency for CMS tasks through the utilization of an XCache service. Specifically, our results underscore that CMS CRAB jobs conducted at CIEMAT compute nodes, leveraging remote reads from PIC XCache, exhibit superior performance compared to analogous jobs using the CMS global XRootD re-director infrastructure. Moreover, our analysis reveals a striking similarity in CPU efficiency between tasks executed at CIEMAT accessing local storage and those fetching data from PIC's XCache (provided the data is already cached). This suggests that a unified cache deployed at PIC Tier-1 could efficiently cater to data needs across all Spanish CMS Tier-2 sites without imposing a notable strain on application performance.

4. Optimal dimensioning of a single CMS XCache in Spain

The streamlining of data extraction and processing procedures concerning CRAB job access logs has been a significant achievement in the course of this research. Leveraging this refined dataset enables us to model data cache behavior accurately and forecast the effects of diverse cache configurations on performance metrics.

While previous studies have explored various cache algorithms and configurations for CMS [19], they often lack crucial information readily available in CRAB logs. To bridge this gap, our study harnesses SWAN's Big Data infrastructure and Dask's parallelization framework to efficiently process CRAB user logs from Spain. By doing so, we aim to construct a more realistic model of data cache behavior, shedding light on the applicability of traditional caching algorithms for CMS regional XCaches and determining optimal cache sizes. To emulate the impact of a cache serving data across the entire Spanish region, we conducted an analysis of CRAB jobs executed within Spanish sites over a span of 4 months. On average, the daily job count stood at approximately 9.5k, with peak periods surging to around 30k jobs. Notably, CIEMAT contributed 50% of these jobs, while PIC and IFCA (a Tier-2 in Santander) handled 34% and 16% of the workload, respectively.

Using methodologies outlined in the preceding section, we extracted information regarding the input files for each CRAB job from log files. On a daily basis, an average of approximately 25k files were accessed, translating to roughly 2.7 input files per job. Figure 6 illustrates the daily proportion of files accessed from local and remote storage systems across all CMS Spanish sites. On the other hand, around 22%, 33%, and 77% of input files for PIC, CIEMAT, and IFCA, respectively, originated from remote storage systems. In total, approximately 3.1M files were accessed, with 1.1M sourced from remote locations in this period.

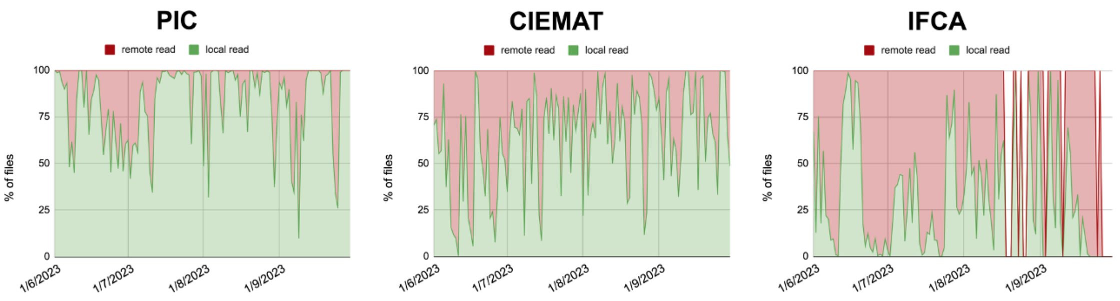

Figure 6: Percentage of input files that have been read from local storage systems or from remote sites (red) in PIC, CIEMAT and IFCA, for the period from June to October 2023.

4.1 Simulating XCache Implementation for Spanish CMS Tiers

The data access specifics from every Spanish CMS center are pivotal in determining the optimal requirements for cache size and its network connectivity. To explore these complexities further, we analyze all user job logs to discern whether data access occurs locally or remotely. While the majority of files are fully downloaded, we also consider partial downloads, drawing on insights gathered from the production PIC XCache service. Leveraging this knowledge, we replicate the population of our cache system to reflect real data access patterns from production user jobs. Furthermore, to maintain efficient space management within the cache, we implement the same algorithm for file deletion (LRU), applying the same low and high watermarks for file deletions as those used in the production XCache.

Figure 7 displays an example of a simulated XCache with a size of 200 TB (black dashed line). The cached data deletions are handled by an LRU with 95%-90% watermarks, shown in the figure with red and green lines, respectively. It takes less than 1 month to saturate the XCache disk with the number of files created in the simulated XCache.

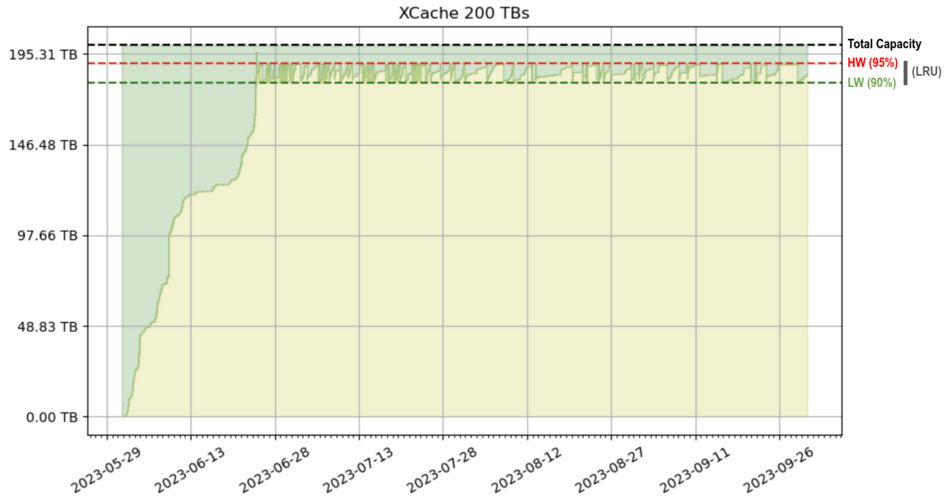

Figure 7: Simulation of a 200 TB XCache that caches all of the remote reads from CRAB jobs executed in PIC, CIEMAT and IFCA, in the period from June to October 2023.

An important aspect of a data cache, aside from its size, is the data import and export, since it conditions the network connectivity the cache server should have available. By means of these simulations, we observed that the average daily import or export throughput surpasses the capacity of the current 10 Gbps Network Interface Card (NIC) installed in the PIC production XCache service. Through simulations like these, we can identify potential bottlenecks and anticipate configurations that would align with expected usage if all Spanish sites were to utilize the PIC XCache service. In this simulated scenario, where a single XCache service serves the entire Spanish region, it's imperative to equip it with a robust 25 Gbps NIC (active-active) to ensure seamless operation. Additionally, an efficiently performing XCache, housing frequently accessed data, should prioritize exporting data to regional compute nodes rather than importing data from remote sites. The ratio of in/out average data rates and the maximum in and out data rates are key metrics for determining the optimal size of the XCache to be deployed in the region. These measures enable us to make informed decisions regarding the capacity and configuration of the XCache system, ensuring its effectiveness in meeting the demands of the CMS infrastructure.

Ensuring an optimally sized cache involves striking a delicate balance between housing frequently accessed data files and minimizing the presence of non-accessed ones. Even with the LRU algorithm managing data deletion, deploying a cache that's too small would prove inefficient. In such cases, popular files may not be retained adequately, leading to frequent cache re-population with previously stored popular files—a process that incurs unnecessary overhead. Conversely, an excessively large data cache risks retaining outdated and unaccessed data, which may persist through multiple LRU data deletion cycles. Thus, characterizing the XCache is paramount to determining the most suitable size for deployment.

Introducing the *Hit Rate* measure becomes pivotal in this context. It quantifies the effectiveness of the cache by representing the ratio of cache hits (i.e., accesses to files already present in the cache) to the total number of accesses (comprising both cache hits and cache misses). A cache miss occurs when a file is not found in the cache and needs to be cached anew. The *Hit Rate* can be expressed in percentage, and it can be calculated in a cumulative way, since the data cache starts being populated:

$$HitRate = \frac{hits}{hits + misses} = \frac{hits}{N_{accesses}} \tag{1}$$

4.2 Optimal XCache across Spain

Simulating different cache sizes is instrumental in pinpointing the most efficient solution for serving the region. This process takes into account various factors, including the *cumulative Hit Rate*, which measures the proportion of accesses to cached files compared to the total number of accesses. Additionally, network considerations play a crucial role in determining the optimal cache size, ensuring that the chosen configuration can effectively manage data transfer and accommodate the demands of the user base. By analyzing these factors in tandem, we can identify the cache size that maximizes performance and resource utilization while minimizing overhead and inefficiencies.

In Figure 8 (left), we observe the *cumulative Hit Rate*. Remarkably, the cache reaches saturation, achieving around 60% in *cumulative Hit Rate* for data cache sizes exceeding 200 TB. Beyond this threshold, the relative gains in *cumulative Hit Rate* diminish significantly. Figure 8 (right) illustrates a characteristic feature of an optimally sized cache: a balanced 3:1 ratio between outbound and inbound traffic. This ratio underscores the efficient distribution of data from the cache to users, reflecting an effective utilization of resources and network bandwidth.

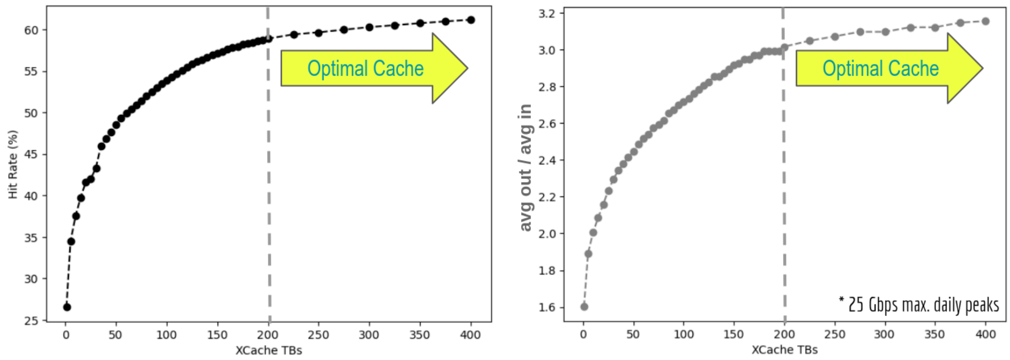

Figure 8: *Cumulative Hit Rate* (left) and ratio between outbound and inbound traffic at the XCache (right) for different simulated XCache sizes.

These findings suggest that deploying a data cache of around 200 TB appears to be sufficient for optimizing performance from the perspective of *cumulative Hit Rate*. This size strikes a balance between accommodating the caching needs of the region and avoiding diminishing returns associated with excessively large caches.

5. Conclusions

Our work demonstrates the advantages of utilizing the XCache service for optimized data access and resource utilization. XCache deployed in the PIC proves to efficiently serve data to all of the Spanish CMS sites, matching the effectiveness of local access at each Spanish CMS center, since all of these sites are located within a 10 ms round-trip time (RTT). The analysis of user job logs plays a crucial role in determining the optimal requirements for cache size and network connectivity. In our case, this analysis suggests the need for a cache size exceeding 200 TB, coupled with a NIC capacity of over 25 Gbps.

The integration of data caches holds promise beyond our immediate region, particularly in areas experiencing heightened remote data accesses by user tasks. These insights underscore the broader applicability and potential impact of data caching technologies in optimizing data workflows across diverse environments. In certain regions, remote reads surpass those observed in the Spanish sites. We have estimated that the traffic generated by remote reads from user jobs can reach up to 10 GB/s, a figure comparable to the global File Transfer Service (FTS [20]) traffic generated by CMS worldwide. Consequently, the incorporation of data caches presents an opportunity to alleviate load on network resources and minimize overall traffic congestion. Data caches elsewhere could reduce the XRootD traffic generated by these user jobs' remote reads by (at least) a factor of 3.

6. Future work and outlook

The implemented data cache solution has delivered promising results, establishing itself as a production-like service in the Spanish region. Currently undergoing a comprehensive evaluation at scale during the ongoing LHC operation phase, this initiative aims to further enhance its capabilities. This research underscores the significant advantages of introducing a CDN and centralized cache system to facilitate data access within a region or country. Not only does this deployment bolster job efficiency, but it also yields cost savings and reduces workload completion times. The insights garnered from this endeavor carry far-reaching implications for other regions and data-intensive scientific endeavors, positioning the service deployment as a strategically significant project.

Future work should extend benchmark studies to multiple CMS sites to objectively assess the savings achieved by incorporating XCache in each region. The deployment in Spain has already demonstrated significant benefits in production, suggesting that a global expansion of XCache deployment could unveil its true impact on production workflows.

Cache simulations hold immense potential to surpass the presented models by delving into the exploration of optimal replacement algorithms. This includes leveraging Machine Learning techniques to identify relevant features and optimize cache performance for specific use cases. Configuration studies should encompass multi-node XCache deployments, evaluating the potential enhancements they could provide to the existing infrastructure. In conclusion, future efforts are focused on refining XCache performance within this context and assessing the feasibility of running computational resources without depending on persistent storage.

Acknowledgements

The authors of this work express their gratitude to the PIC and CIEMAT teams for their support in these studies and for deploying novel cache services for the CMS experiment in the Spanish region.

This project is partially financed by the Spanish Ministry of Science and Innovation (MINECO) through grants FPA2016-80994-C2-1-R, PID2019-110942RB-C22, DATA-2020-1-0039, and BES-2017-082665. It has also been supported by the Ministerio de Ciencia e Innovación (MCIN)

AEI/10.13039/501100011033 under contract PID2020-113614RB-C21, the Catalan government under contract 2021 SGR 00574. The deployment of the XCache service is financed by the Red Española de Supercomputación (RES) through the grant DATA-2020-1-0039.

References

[1] J. Albrecht, A. A. Alves, G. Amadio, G. Andronico, N. Anh-Ky, L. Aphecetche, J. Apostolakis, M. Asai, L. Atzori, M. Babik, G. Bagliesi, M. Bandieramonte, S. Banerjee, M. Barisits, L. A. Bauerdick, *A Roadmap for HEP Software and Computing R&D for the 2020s*, *Computing and Software for Big Science*, vol. 3, no. 7, pp. 1–39, Springer Science and Business Media LLC, Mar. 2019. DOI: 10.1007/s41781-018-0018-8

[2] J. Schovancova, S. Campana, X. Curull, M. Girone, I. Kadochnikov, G. McCance, *Understanding the Performance of a Prototype of a WLCG Data Lake for HL-LHC*, in *2018 IEEE 14th International Conference on e-Science (e-Science)*, pp. 332-333, IEEE Computer Society, 2018. DOI: 10.1109/eScience.2018.00080

[3] Ian Bird, Simone Campana, Maria Girone, Xavier Espinal, Gavin McCance, Jaroslava Schovancová, *Architecture and prototype of a WLCG data lake for HL-LHC*, *EPJ Web Conf.*, vol. 214, pp. 04024, 2019. DOI: 10.1051/epjconf/201921404024

[4] E. Martelli, S. Stancu, *LHCOPN and LHCONE: Status and Future Evolution*, *Journal of Physics: Conference Series*, vol. 664, no. 5, pp. 052025, Dec. 2015. DOI: 10.1088/1742-6596/664/5/052025

[5] Marian Babik, Shawn McKee, *Network Capabilities for the HL-LHC Era*, *EPJ Web Conf.*, vol. 245, pp. 07051, 2020. DOI: 10.1051/epjconf/202024507051

[6] CMS Collaboration, *The CMS experiment at the CERN LHC*, *Journal of Instrumentation*, vol. 3, pp. S08004, 2008. Publisher: IOP Publishing.

[7] L. Bauerdick, K. Bloom, B. Bockelman, D. Bradley, S. Dasu, M. Ernst, R. Gardner, A. Hanushevsky, H. Ito, D. Lesny, P. McGuigan, S. McKee, O. Rind, H. Severini, I. Sfiligoi, M. Tadel, I. Vukotic, S. Williams, W. Yang, *Using XRootD to Federate Regional Storage*, *Journal of Physics: Conference Series*, vol. 396, no. 4, pp. 042009, 2012. DOI: 10.1088/1742-6596/396/4/042009.

[8] L. Bauerdick, K. Bloom, B. Bockelman, D. Bradley, S. Dasu, Jeffrey Dost, Igor Sfiligoi, A. Tadel, Matevz Tadel, Frank Wuerthwein, A. Yagil, *XRootd, disk-based, caching proxy for optimization of data access, data placement and data replication*, *Journal of Physics: Conference Series*, vol. 513, pp. 042044, Jun. 2014. DOI: 10.1088/1742-6596/513/4/042044.

[9] Teng Li, Robert Currie, Andrew Washbrook, *A data caching model for Tier 2 WLCG computing centres using XCache*, *EPJ Web Conf.*, vol. 214, pp. 04047, 2019. Section: T4 - Data handling. DOI: https://doi.org/10.1051/epjconf/201921404047. Published online: 17 September 2019.

[10] Alan Jay Smith, *Design of CPU Cache Memories*, in *Proc. IEEE TENCON*, 1987.

[11] C.Perez Dengra, J. Flix, A. Sikora, J. Casals, C. Acosta-Silva, C. M. Morcillo Perez, A. Pérez-Calero Yzquierdo, A. Delgado Peris, J. M. Hernández, F. J. Rodriguez Calonge for the CMS collaboration *A case study of content delivery networks for the CMS experiment*, to be published in *Proceedings of CHEP 2023.*

[12] G. Ramirez, *Muon Analysis: Muon Analyzer*, GitLab, 2023. URL: https://gitlab.cern.ch/garamire/muonanalysis-muonanalyzer/-/tree/master/. Accessed on November 17, 2023.

[13] CERN, *ROOT*, https://root.cern/, 2023. Accessed on October 30, 2023.

[14] Domenico Giordano, Manfred Alef, Michele Michelotto, *Next Generation of HEP CPU Benchmarks*, *EPJ Web of Conferences*, vol. 214, pp. 08011, 2019. DOI: 10.1051/epjconf/201921408011.

[15] Fermi National Accelerator Laboratory, *Fermilab - America's Premier National Lab for Particle Physics and Accelerator Research*, https://www.fnal.gov/, Accessed: 1 November 2023.

[16] Korea Institute of Science and Technology, *Korea Institute of Science and Technology*, https://www.kisti.re.kr/eng/, Accessed: 11 November 2023.

[17] Daniele Spiga, S. Lacaprara, William Bacchi, M. Cinquilli, Giuseppe Codispoti, M. Corvo, Alvise Dorigo, A. Fanfani, Federica Fanzago, Fabio Farina, O. Gutsche, Carlos Kavka, M. Merlo, Leonello Servoli, *CRAB: The CMS distributed analysis tool development and design*, *Nuclear Physics B - Proceedings Supplements*, vol. 177-178, pp. 267-268, Mar. 2008. DOI: 10.1016/j.nuclphysbps.2007.11.124.

[18] Danilo Piparo, Enric Tejedor, Pere Mato, Luca Mascetti, Jakub Moscicki, Massimo Lamanna, *SWAN: A service for interactive analysis in the cloud*, *Future Generation Computer Systems*, vol. 78, pp. 1071-1078, 2018. DOI: 10.1016/j.future.2016.11.035. URL: https://www.sciencedirect.com/science/article/pii/S0167739X16307105.

[19] Daniele Spiga, Diego Ciangottini, Mirco Tracolli, Tommaso Tedeschi, Daniele Cesini, Tommaso Boccali, Valentina Poggioni, Marco Baioletti, Valentin Y. Kuznetsov, *Smart Caching at CMS: applying AI to XCache edge services*, *EPJ Web Conf.*, vol. 245, pp. 04024, 2020. DOI: 10.1051/epjconf/202024504024.

[20] File Transfer Service (FTS), https://fts.web.cern.ch/.