EPJ Web of Conferences **295**, 07045 (2024) https://doi.org/10.1051/epjconf/202429507045
CHEP 2023

The Spanish CMS Analysis Facility at CIEMAT

M. Cárdenas-Montes[1,*], *A.* Delgado Peris[1], *J.* Flix[1,2], *J.M.* Hernández[1], *J.* León Holgado[1], *C.* Morcillo Pérez[1], *A.* Pérez-Calero Yzquierdo[1,2], and *F.J.* Rodríguez Calonge[1] on behalf of the CMS Collaboration.

[1]CIEMAT, Scientific Computing Unit, 28040 Madrid, Spain
[2]PIC, 08193 Bellaterra (Barcelona), Spain

Abstract. The increasingly larger data volumes that the LHC experiments will accumulate in the coming years, especially in the High-Luminosity LHC era, call for a paradigm shift in the way experimental datasets are accessed and analyzed. The current model, based on data reduction on the Grid infrastructure, followed by interactive data analysis of manageable size samples on the physicists' individual computers, will be superseded by the adoption of Analysis Facilities. This rapidly evolving concept is converging to include dedicated hardware infrastructures and computing services optimized for the effective analysis of large HEP data samples. This paper describes the actual implementation of this new analysis facility model at the CIEMAT institute, in Spain, to support the local CMS experiment community. Our work details the deployment of dedicated highly performant hardware, the operation of data staging and caching services ensuring prompt and efficient access to CMS physics analysis datasets, and the integration and optimization of a custom analysis framework based on ROOT's RDataFrame and CMS NanoAOD format. Finally, performance results obtained by benchmarking the deployed infrastructure and software against a CMS analysis workflow are summarized.

1 Introduction

The future of high energy physics (HEP) experiments will witness a substantial increase in data volumes. The most important example of this is the High-Luminosity phase of the Large Hadron Collider (LHC), which will provide a much higher collision rate and produce significantly more complex events, thus requiring much larger samples of Monte Carlo simulated events than today. All in all, the data managed by the experiments will increase by an order of magnitude or more, while available computing resources are not expected to grow at the same rate [1]. New analysis models must therefore adapt to this challenging scenario. In particular, the software libraries and frameworks used by the HEP community must be modernized to improve sustainability, ease parallelization, and benefit from current hardware architectures (large number of cores, vector instructions and accelerators), and adapt to the new reduced analysis data formats produced by the experiments (e.g. for the Compact Muon Solenoid experiment [2], CMS, the NanoAOD format [3]).

*Projects PID2019-110942RB-C21 and PID2020-113807RA-I00 funded by MCIN/AEI/10.13039/501100011033 and by ERDF A way to make Europe.

In parallel, the landscape of computational analysis tools is undergoing significant changes. Data intensive analysis is growing in importance beyond the HEP community, and new tools and paradigms have emerged. These include columnar analysis, parallel data consumption, declarative and functional programming models and the integration of powerful machine learning models. Also, an important part of the community has adopted Python as the main programming language for analysis, while Jupyter notebooks have become the principal interactive work environment for many users [1].

The concept of Analysis Facility (AF) has been recently adopted by the HEP community to describe a dedicated infrastructure that will provide the proper environment to perform the final stages of a physics analysis in the described context of massive data samples [4]. Such infrastructure should not only include high-performance hardware, but also offer the adequate services to enable efficient data access and storage. In parallel, AFs should promote the adoption of modern software and techniques that ensure an efficient usage of hardware resources. Key objectives of such facilities are ease of use, performance, scalability and sustainability. Several initiatives are already demonstrating the first experiences of analysis facilities deployments for LHC users, both in the USA [5] and Europe [6].

2 The Spanish Analysis Facility

The CMS group at CIEMAT, in Spain, comprises scientists and technologists actively taking part in diverse areas within the CMS experiment, including participation in many CMS analyses, but also the development, deployment and operation of the LHC computing infrastructure (WLCG). The computing infrastructure located at the CIEMAT headquarters in Madrid includes a CMS Tier-2 Grid site. Its capacity is currently being expanded in order to deploy an AF for the Spanish CMS community, a highly performant infrastructure based on new hardware and services, but also enabling new analysis techniques and programming paradigms, in line with current trends in the HEP community. The ultimate goal is to provide support and increase collaboration with the local CMS research community.

The architecture of the Spanish AF, shown in Figure 1, can be described in terms of its constituent hardware and services, and the analysis software available to users.

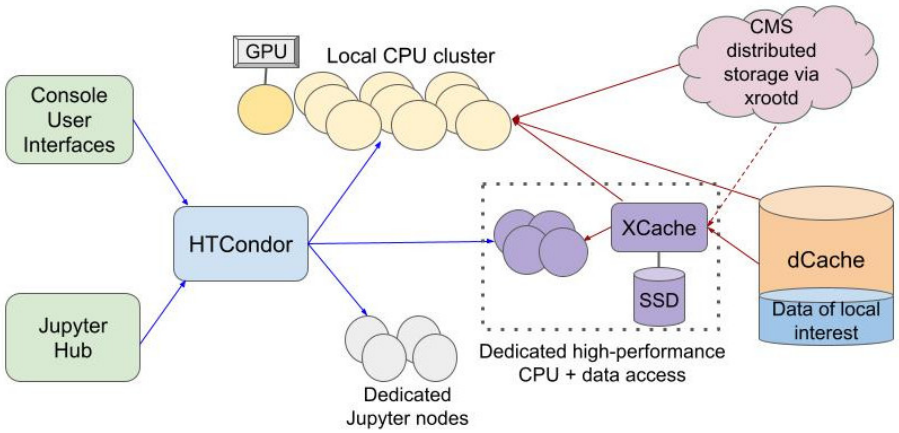


Figure 1: Architecture of the Analysis Facility for the CMS experiment, at CIEMAT.

Hardware. Two new powerful workstations, devoted exclusively to analysis tasks, have been recently deployed. We will refer to them as *AF nodes*. These AF nodes are equipped

with 128 and 172 CPU cores respectively, and 180 TB of NVMe solid-state drives (SSD) each. They have been integrated in the existing CIEMAT HTCondor [7] pool of resources. A few hundred CPU cores of the already installed resources (batch *worker nodes* (WN) with hard-disk drives) are also used for analysis. For comparison, the HS06 benchmark [8] value of the first AF node is 3000 (20.6 HS06 per job slot), while the most powerful WN reaches 1120 HS06, and the WNs used for the performance tests of Section 4 average 14.1 HS06 per job slot. Regarding data storage committed to analysis, 1 PB of disk has been allocated at the site's WLCG storage element (SE) and is dedicated to hosting replicas of the most interesting NanoAOD samples for the local CMS community. Replication of these datasets from other WLCG SEs is managed via Rucio [9] subscriptions. Finally, a NVIDIA HGX server with 4 A100 graphic processing units (GPU) has also been deployed.

Services. An analysis-only XCache [10] service is run on the first of the two new AF nodes. XCache enables the transparent caching of data consumed from the CMS XRootD data federation [11], thus facilitating the re-use of this data in consecutive iterations of the analysis process. Cached data is stored in the node's SSD space to maximize read performance. A JupyterHub service has been also set up on top of the HTCondor batch system in order to enable the execution of notebooks in any of the supported resources, namely the new AF nodes, the traditional batch worker nodes, or the HGX GPU server. Users can thus choose between two work interfaces: traditional login nodes and Jupyter notebooks. The deployment of a Dask [12] service that enables job submission from Jupyter notebooks to the batch is planned for the near future.

Software. The CMS group at CIEMAT has developed a general-purpose framework for CMS analysis using NanoAOD data [13]. This software, being used for several analyses at the CIEMAT AF, will be described in more detail in the following section.

People. The activities related to the Spanish CMS AF effort have fostered a tighter collaboration between CIEMAT computing and analysis groups. Both groups have joined in the design and evaluation of the architecture for our AF infrastructure, having contributed enhancements for the NanoAOD analysis framework as well. Naturally, the computing group offers support and advice on efficient resource utilization.

3 The CIEMAT analysis framework

Motivated by the current trends in the HEP community and CMS in particular, the CIEMAT CMS group is adapting its analysis workflows to use NanoAOD data formats and modern computational tools and paradigms, like the declarative style of ROOT's RDataFrame (RDF) [14]. Consequently, the CIEMAT analysis framework has been developed with the main objectives of being general, flexible, fast and user-friendly, and with contributions from both the analysis and computing groups. The framework can consume either NanoAOD or flat ROOT tuples as input, and uses RDF for filtering and selection, and for the creation of new variables. The new framework is based on the existing Luigi Analysis Framework [15], which is leveraged in particular for data access, task management, and command-line interface support.

The CIEMAT analysis framework can manage the execution of a complete analysis, from the acquisition of input data to the production of the final plots, and can handle event reweighting, correction factors, systematics, etc. An explicit goal of the framework is to make it easy to reuse general-purpose modules, while also supporting the addition of analysis-specific features. It supports task execution either locally or in HTCondor, provides built-in parallelization, and supports both local and XRootD-based input data file access. User code is written mostly in Python, and code version control is encouraged as part of the standard workflow for an analysis.

4 Performance measurements

The CIEMAT analysis framework is currently being used in a number of analyses. We have reused an $HH \rightarrow bb\tau\tau$ analysis for the functional and performance evaluation of the new infrastructure. We have measured the total turnaround time required to run the main (*processing*) stage of the analysis, but limited to 3 input datasets (251 files, total size of 393 GB), which represents around the 20% of the input sample used in the full analysis. This is done to avoid scheduling queues in our tests (since the framework will run one HTCondor job per input file of a given dataset). This analysis requires the application of many filters on the data, including the likelihood-based *SVFit algorithm*, for the reconstruction of the Higgs boson mass, which is quite CPU intensive.

Several test scenarios were considered. In general, the diverse configurations represented in Figures 2 through 5 are labeled indicating where the analysis was executed (*AF* for the first deployed AF node, or *WN* for a standard batch worker node), and what sources were used to fetch the input data (e.g. the *XCache* service or an *SE*), always using the XRootD protocol. In some cases, additional information is encoded in the labels, as indicated in the captions. Each configuration was tested with several executions, and consequently all figures show average turnaround times and standard deviation (error bars). Results have been normalized relative to the best case for each plot, i.e. that with the lowest execution time. The overall best case was achieved when running in the AF node reading input data from the pre-filled XCache service in the same node (*AF_XCache* label). This simulates the scenario in which the analysis had already been run once, and thus the input data had been cached. With this configuration, the processing stage took on average 44.6 minutes (using 85 cores), i.e. 1.75 K events/s/core, including the job scheduling and framework initialization phases. This case is shown in Figure 2 compared with the execution in standard worker nodes (50% worse turnaround time) and data read from the site's SE (similar turnaround time).

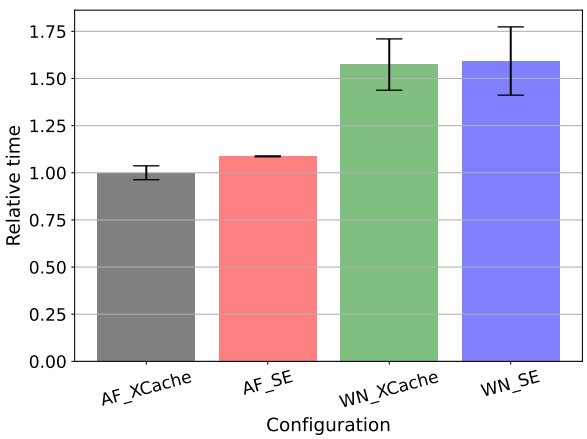


Figure 2: Relative turnaround time for execution at the AF node, reading input data from the XCache service (*AF_XCache*) or the site's SE (*AF_SE*), and for execution at standard worker nodes, again reading from XCache (*WN_XCache*) or the SE (*WN_SE*).

This result suggests that reading data from the SE disks does not impose a significant performance penalty in comparison to co-hosted SSDs, which indicates that the local network is not a bottleneck for data consumption, and that the analysis can then be thought of as

CPU-bound. The same conclusion can be deduced from Figure 3, which shows a one-to-one comparison of the same execution in the AF node and in a standard batch node, where resulting times have been normalized to the power of each node's processor, considering their HS06 benchmarks. After the normalization, both local data access (*AF*) and LAN data access (*WN*) produce quite similar results.

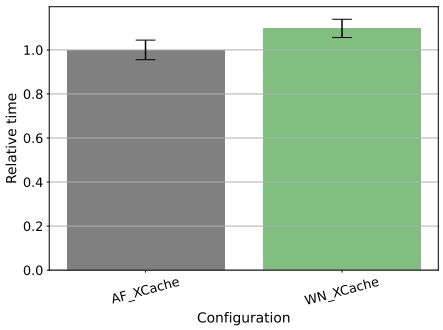


Figure 3: Relative HS06-normalized turnaround time for analysis execution at the AF node (*AF_XCache*) and standard worker node (*WN_XCache*), reading input data from the XCache.

The results in Figure 4 consider a scenario where the analysis is run in the AF node and data is read from the co-located XCache service, but for different initial conditions. In the first case (*AF_filled*), the data is already present in the cache when the analysis is started, but in the other cases the cache is initially empty, and it is configured to retrieve the data from the site's SE. The XCache fetches data in blocks (*read-ahead* technique) using a configurable block size: the *prefetch* value. The default prefetch setting is 5 MB, and as Figure 4 shows, when this value is used, the analysis turnaround time doubles. Such a bad result was unexpected, given that the network was not causing a significant performance penalty in the previous tests. The initial interpretation of this result is that it is due to a large number of interactions taking place between the XCache and the SE for these data reads. Performance can then be greatly improved by using larger prefetch sizes, reducing the required number of data exchanges between the two services, as shown by the results obtained using 100 MB and 500 MB.

Figure 5 displays the results of a more dramatic comparison. Like in the previous example, the analysis is run in the AF node in all the configurations, but data is read from different sources in each case. The best results are once again achieved reading from the pre-filled local XCache server (*AF_filled*). The worst execution times (100 times longer) are obtained when data is read directly from a remote SE, in this example located in the Fermilab laboratory, in Chicago, USA (*AF_Xrootd_US*). In contrast, much faster turnaround can be achieved when data is read from the XCache, even if the cache is initially empty, and the required files are fetched from the same remote source. Figure 5 shows three results for the described XCache set-up, with the only variation of the configured prefetch setting: 5 MB (default), 100 MB or 550 MB (*AF_XCache_**). This reveals that the insertion of the XCache layer, and the consequent use of the read-ahead technique, causes a clear reduction in the analysis completion time. The improvement is higher the larger the prefetch value in use.

5 Support to other scientific communities

An additional declared goal for the AF infrastructure at CIEMAT is to be useful for other communities beyond HEP. Even if the execution of CMS analyses is the main objective of

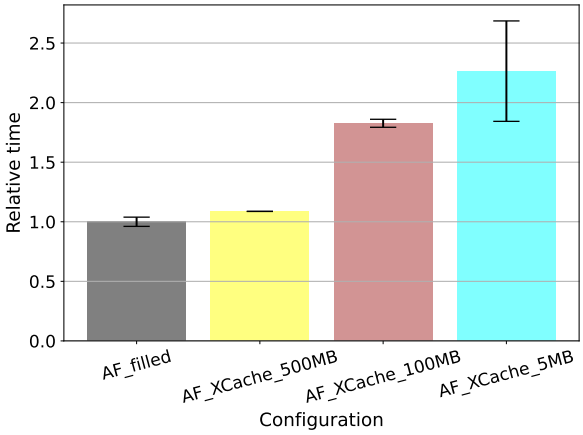


Figure 4: Relative turnaround time for execution at the new AF machines for data read from a pre-filled local XCache (*AF_filled*) or an empty XCache fetching data from the local SE, using different prefetch values (*AF_XCache_<prefetch>MB*).

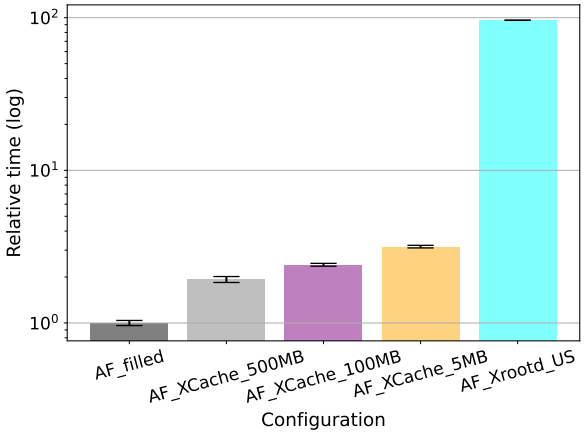


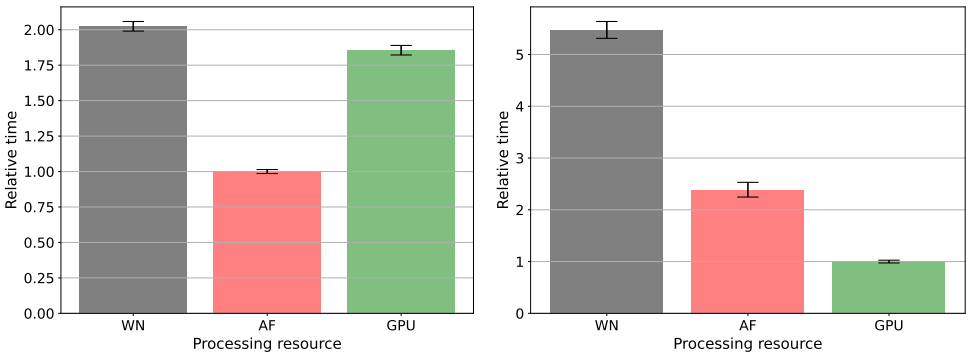
Figure 5: Relative turnaround time for execution at the new AF machines, and data read from a pre-filled local XCache (*AF_filled*), from an empty XCache fetching data from a remote SE, using different prefetch values (*AF_XCache_<prefetch>MB*), and directly from a remote SE (*AF_Xrootd_US*).

the AF, other CIEMAT research groups involved in data-intensive scientific studies, such as medicine, cosmology, astronomy or astroparticle physics, can also benefit from these computing resources and services. This is possible thanks to the modern user-friendly Jupyter interface offered by the AF, and by the support of a standard and convenient protocol for data access (NFS). Moreover, the GPU-equipped server makes the infrastructure appealing for the increasingly frequent machine learning workflows for scientific studies (both within and outside HEP).

In fact, several machine learning applications have already been ported to the new infrastructure, and they have been or are being tested. These include pollution forecasting, detection of dark matter in liquid argon and gravitational waves classification. Furthermore, the new infrastructure is expected to be very relevant for projected oncology studies.

In order to assess the suitability of the new AF for machine learning applications, a series of tests have been carried out. In these tests, a set of 40 different time series are converted into *wavelet* images, and these images are then classified with a neural network with half a million configurable parameters. For this process, two different stages are considered and their running time measured independently: the creation of the wavelet images and the training of the neural network used for the image classification. All the tests use a Jupyter notebook as interface, but three different underlying resources are considered for the execution of that notebook: a standard batch worker node (*WN* label), the new AF node (*AF*), and another standard worker node, but, in this case, with an accessible GPU attached to it (*GPU*).

Figure 6 shows the results of the described machine learning tests. The new AF, with more powerful CPUs and SSD local storage, performs significantly better than the other nodes for the image creation stage. For the second stage, the node with the attached GPU outperforms the other two. This was expected since hardware accelerators are known to be really efficient for neural network training.



(a) Image creation performance

(b) Training performance

Figure 6: Relative execution time for wavelet image creation (a) and neural network training (b), for different test execution infrastructures: standard worker node (*WN*), new AF node (*AF*), and worker node with an attached GPU (*GPU*).

6 Conclusions and outlook

The CMS group at CIEMAT is preparing for the challenging future of the HEP experiments, where much larger data volumes than today will be produced and will have to be managed and analyzed. The group is adopting recommended practices for analysis today, in particular by using samples of reduced data format when possible, aggressively applying data caching and read-ahead techniques, and embracing new software tools and paradigms in a newly developed analysis framework.

Moreover, new hardware and services for LHC analysis support have been deployed alongside the CMS computing Tier-2 at CIEMAT, thus offering an Analysis Facility to the

local CMS community. This infrastructure is already being used for several analyses. Furthermore, the new facility will support also other, non-HEP, research groups at CIEMAT, and first validation tests of some of their applications have already been conducted successfully. Finally, performance studies have been carried out, exhibiting promising results for the application of the new infrastructure to fast-turnaround studies. The results also highlight the importance of the application of read-ahead caching techniques when accessing remote data.

The operation of the AF is a continuous process. On the one hand, ongoing support on efficient usage of available computing tools and resources must be offered to the physicists. On the other hand, the infrastructure will be progressively enhanced, both in terms of deployed hardware and offered services. As examples of this, we can cite the recently deployed second SSD-equipped workstation, and the planned Dask service, which will soon enable job submission from Jupyter notebooks to the batch nodes.

References

[1] J. Albrecht, A.A. Alves, G. Amadio, G. Andronico, N. Anh-Ky, L. Aphecetche, J. Apostolakis, M. Asai, L. Atzori et al., Computing and software for big science **3**, 7 (2019)

[2] The CMS Collaboration, Journal of Instrumentation **3**, S08004 (2008)

[3] M. Peruzzi, G. Petrucciani, A. Rizzi, for the CMS Collaboration, *The NanoAOD event data format in CMS* (IOP Publishing, 2020), Vol. 1525, p. 012038, https://dx.doi.org/10.1088/1742-6596/1525/1/012038

[4] G.A. Stewart, P. Elmer, G. Eulisse, L. Gouskos, S. Hageboeck, A.R. Hall, L. Heinrich, A. Held, M. Jouvin, T.J. Khoo et al., *HSF IRIS-HEP Second analysis ecosystem workshop report* (2022), https://zenodo.org/record/7418818

[5] D. Benjamin, K. Bloom, B. Bockelman, L. Bryant, K. Cranmer, R. Gardner, C. Hollowell, B. Holzman, E. Lançon, O. Rind et al., *Analysis facilities for HL-LHC* (2022), arXiv 2203.08010, https://doi.org/10.48550/arXiv.2203.08010

[6] D. Ciangottini, T. Boccali, A. Ceccanti, D. Spiga, D. Salomoni, T. Tedeschi, M. Tracolli, EPJ Web Conf. **251**, 02045 (2021)

[7] D. Thain, T. Tannenbaum, M. Livny, Concurrency - Practice and Experience **17**, 323 (2005)

[8] J.L. Henning, ACM SIGARCH Computer Architecture News **34, 4**, 1 (2006)

[9] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, D. Cameron, D. Christidis, D. Ciangottini et al., Computing and Software for Big Science **3**, 11 (2019)

[10] A. Hanushevsky, H. Ito, M. Lassnig, R. Popescu, A. De Silva, M. Simon, R. Gardner, V. Garonne, J. De Stefano, I. Vukotic et al., EPJ Web Conf. **214**, 04008 (2019)

[11] L. Bauerdick, D. Benjamin, K. Bloom, B. Bockelman, D. Bradley, S. Dasu, M. Ernst, R. Gardner, A. Hanushevsky, H. Ito et al., Journal of Physics: Conference Series **396**, 042009 (2012)

[12] Dask Development Team, *Dask: Library for dynamic task scheduling* (2016), https://dask.org

[13] J. Leon Holgado, *nanoaod_base_analysis* (2023), https://doi.org/10.5281/zenodo.8187177

[14] D. Piparo, P. Canal, E. Guiraud, X.V. Pla, G. Ganis, G. Amadio, A. Naumann, E. Tejedor, EPJ Web Conf. **214**, 06029 (2019)

[15] M. Rieger, Y. Rath, L. Geiger, P. Fackeldey, E. G., J.L. Holgado, Nollde, F. von Cube, B.P. Kinoshita, D. Savoiu et al., *riga/law* (2023), https://doi.org/10.5281/zenodo.3703205