

HPC resources for CMS offline computing: an integration and scalability challenge for the Submission Infrastructure

Antonio Pérez-Calero Yzquierdo^{1,2,*}, Marco Mascheroni³, Edita Kizinevic⁴, Farrukh Aftab Khan⁵, Hyunwoo Kim⁵, Maria Acosta Flechas⁵, Nikos Tsipinakis⁴, and Saqib Haleem⁶ on behalf of the CMS Collaboration.

¹Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain

²Port d'Informació Científica (PIC), Barcelona, Spain

³University of California San Diego, La Jolla, CA, USA

⁴European Organization for Nuclear Research (CERN), Geneva, Switzerland

⁵Fermi National Accelerator Laboratory, Batavia, IL, USA

⁶National Centre for Physics, Islamabad, Pakistan

Abstract. The computing resource needs of LHC experiments are expected to continue growing significantly during the Run 3 and into the HL-LHC era. The landscape of available resources will also evolve, as High Performance Computing (HPC) and Cloud resources will provide a comparable, or even dominant, fraction of the total compute capacity. The future years present a challenge for the experiments' resource provisioning models, both in terms of scalability and increasing complexity. The CMS Submission Infrastructure (SI) provisions computing resources for CMS workflows. This infrastructure is built on a set of federated HTCondor pools, currently aggregating 400k CPU cores distributed worldwide and supporting the simultaneous execution of over 200k computing tasks. Incorporating HPC resources into CMS computing represents firstly an integration challenge, as HPC centers are much more diverse compared to Grid sites. Secondly, evolving the present SI, dimensioned to harness the current CMS computing capacity, to reach the resource scales required for the HL-LHC phase, while maintaining global flexibility and efficiency, will represent an additional challenge for the SI. To preventively address future potential scalability limits, the SI team regularly runs tests to explore the maximum reach of our infrastructure. In this note, the integration of HPC resources into CMS offline computing is summarized, the potential concerns for the SI derived from the increased scale of operations are described, and the most recent results of scalability test on the CMS SI are reported.

1 The CMS Submission Infrastructure

The Submission Infrastructure (SI) team in Offline and Computing for the CMS experiment [1] is in charge of operating a set of federated HTCondor [2] pools which aggregates resources from about 70 Grid sites distributed worldwide, plus non-Grid resources, where reconstruction, simulation, and analysis of physics data takes place. In total, the CMS SI manages approximately 400k CPU cores (see Figure 1), regularly supporting the simultaneous execution of nearly 200k CMS computing tasks.

*e-mail: aperez@pic.es

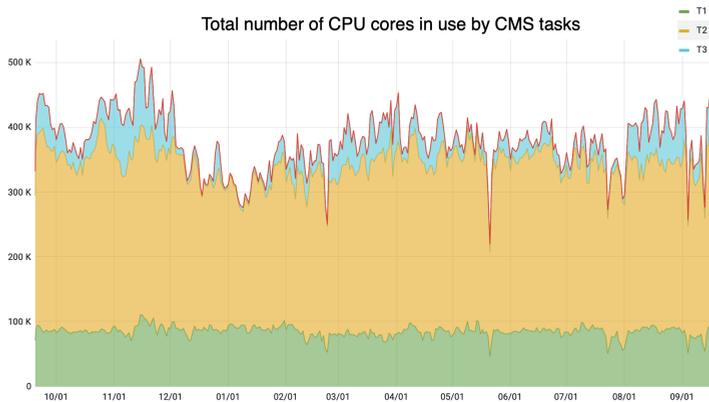


Figure 1. Daily averaged number of CPU cores in use by CMS tasks at each of the WLCG Tier layers for the last 12 months.

A schematic view of this infrastructure is presented in Figure 2. Despite its increasing complexity in recent years [3], its main component continues to be the Global Pool [4], which joins the major fraction of resources managed by the CMS SI.

The challenges of operating such an infrastructure reside in managing an ever growing collection of computing resources, connecting new and more diverse resource types (including non-x86 CPU architectures and GPUs) and resource providers, such as WLCG [5] and OSG [6] infrastructures, High Performance Computing (HPC) facilities, as well as Cloud and volunteer clusters, while simultaneously using all of our resources efficiently, maximizing data processing throughput and enforcing task priorities according to CMS scientific program.

2 HPC growing contribution to CMS computing

HPC centers can help supporting the computing needs of CMS, given the substantial national and supranational investments made and planned for the immediate future. HPC clusters are indeed already part of the scientific computing infrastructure in High Energy Physics (HEP), and it is expected that their contribution will only grow in the coming years. In fact, several examples of HPC resources integration to the CMS computing model have already been completed with success [9].

As shown in Figure 2, CMS follows two main models for the integration of HPC resources to our computing resource portfolio: firstly, in the US, the HEPCloud [10] infrastructure, managed by the FNAL team, supports a separated HTCondor pool, which is however federated to the central CMS SI, being delegated the execution of CMS tasks [11]. Secondly, a method of WLCG site-extension (transparent from the CMS perspective) is mainly employed for the EU HPC facilities [12] [13]. In this second model, GlideinWMS [14] pilot jobs [15] submitted by the CMS SI to WLCG sites' Compute Elements [16] (CE), are locally redirected onto the corresponding access point of the associated HPC cluster.

Figure 3 shows the evolution of HPC resources contribution to CMS offline computing effort over the recent years. A continuous increase in the aggregated CPU usage can be observed, indicating that the compute capacity utilized by CMS at HPC facilities could be

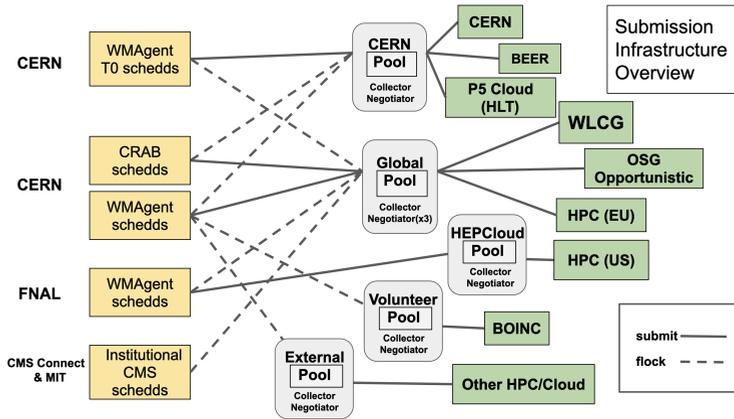


Figure 2. CMS SI current configuration, including multiple federated pools of resources aggregated from diverse providers (green boxes) and sets of distributed job schedders (schedds), handling the Tier-0 and centralized production workloads (WMAgent), as well as analysis job submission (CRAB). A number of opportunistic resources are shown, including Batch on EOS Extra Resources (BEER) at CERN [7] and the CMS Run 2 High Level Trigger farm, reconfigured to support CMS offline computing (P5 Cloud [8]).

sustained at its current level, or even exceeded, in the future years. Moreover, Figure 3 also displays when new facilities are gradually integrated to the CMS HPC resource pool, initially mainly in the USA (e.g. the National Energy Research Scientific Computing center [17], NERSC, as well as the Supercomputing Center at the University of San Diego [18], SDSC, and the Pittsburgh Supercomputing Center [19], PSC). More HPC facilities were then also incorporated to the CMS pool in the EU countries (e.g. with clusters at CINECA [20] in Italy, HoreKa [21] in Germany and the Barcelona Supercomputing Center [22], BSC, in Spain).

3 Challenges in the integration and use of HPC resources by CMS

Incorporating any new HPC facility to our SI, in order to enable CMS making use of its resources, presents a number of challenges. First, as each HPC facility is different, CMS has to manage a diversity of technical requirements and policy constraints. This affects how access to compute resources, transfers of datasets from/to CMS distributed storage infrastructure, deployment of required software (e.g. CVMFS [23]) and services, such as access to remote databases, etc, are technically solved. The integration of HPC clusters thus frequently results in each of these elements having to be resolved case by case. Integration efforts are coordinated centrally by CMS Computing, with the support of the national teams negotiating the use of their respective HPC clusters for CMS offline computing.

The second challenge arises from the scale of potentially available resources at HPCs, and the fact that CPU allocation at these facilities typically differs from resource provisioning at WLCG sites. Indeed, under sufficient workload pressure from CMS, grid sites tend to supply a constant amount of CPU power, generally in agreement with the capacity each site has pledged to the experiment. On the contrary, HPC facilities often allocate CPU resources to their users in shorter yet more intense bursts. An example of this pattern, at NERSC, is shown in Figure 4, including peaks of up to 100k CPU cores in use by CMS lasting for a few hours of duration.

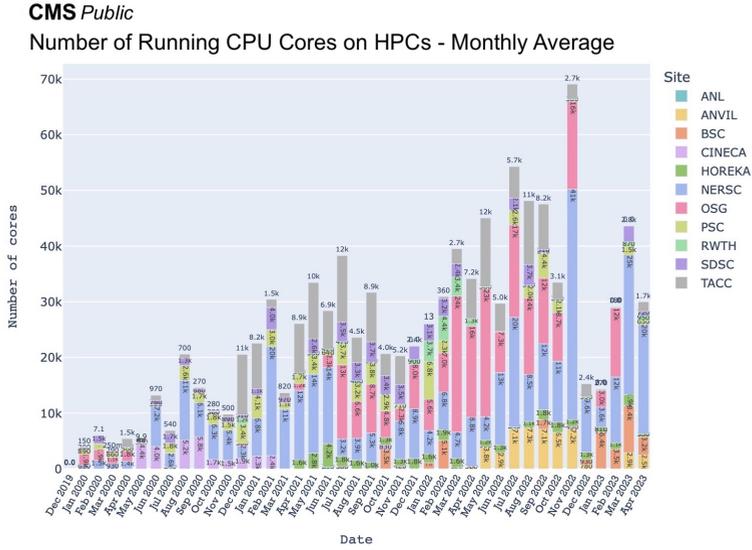


Figure 3. HPC contribution to CMS compute resources over the recent years, expressed as a monthly average of CPU cores in use, for each contributing resource provider.

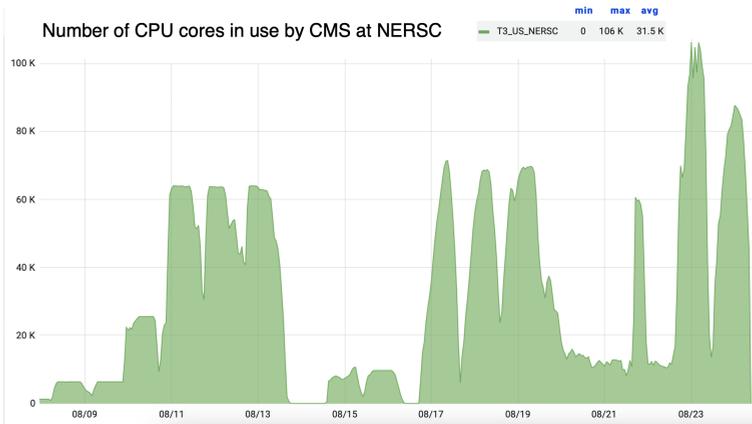


Figure 4. Number of CPU cores allocated at NERSC for CMS jobs execution as a function of time, demonstrating a pattern of resource provisioning in peaks usually found in relation to HPC facilities.

The CMS Workload Management (WM) and SI systems must therefore be appropriately dimensioned to manage and distribute workloads in order to leverage these enlarged pools of resources, profiting from the HPC resource peaks in addition to the WLCG resource baseline. Ensuring the stability and flexibility of the WM and SI infrastructures and services is crucial for effectively harnessing the increased compute capacity provided by HPCs, which will be discussed in the following sections.

4 Scalability limits in the CMS SI

Operating away from any scalability limiting factor is a critical aspect for a system such as the CMS SI, designed to perform in a dynamic environment, adapting itself to the growing CMS resource demands, the fluctuating resource availability from the WLCG and other providers, and the variable mix of workloads the SI has to manage. The SI team has therefore proactively run scalability tests over the years in order to find those limits for every dimension of our system, and, consequently, evolve the infrastructure to push the scalability limits further away.

The assessment of SI scalability involves firstly analyzing the computing capacity that the Central Manager (CM) element of our HTCondor pools can effectively harness. This is in turn derived from the *collector* capability of real-time processing the stream of status updates arriving from the slots across the pool, as well as keeping the *negotiator* matchmaking cycle time under control. Secondly, once the resources are available and can be effectively negotiated, the active pool of scheduler nodes (*schedds*, see Figure 2) must be tested on their aggregated maximum concurrent job execution capacity. This metric determines the number of workflows and the scale of resources that can be efficiently managed by our infrastructure.

4.1 Pushing the limits of the CMS Global Pool

The SI team is continuously working on detecting and solving scaling bottlenecks to the infrastructure, with the support of the HTCondor and GlideinWMS developers teams. As a consequence of the accumulated experience, our setup currently includes multiple "non-standard" customized settings, most of them aimed at avoiding the saturation of the main collector service of the pool, as described in previous reports [24]. Some of these specialized settings are:

- (a) Our HTCondor Connection Broker (CCB) service is running on a separate host to the CM, and configured with an enlarged pool of available connection sockets.
- (b) Our Global Pool CM employs multiple negotiator daemons, running in multi-threaded mode, to speed up the negotiation cycle.
- (c) Multiple secondary collector processes are deployed in our CM, supporting the task of the main top collector daemon.
- (d) The stream of slot update messages to the collectors has been optimized to only propagate certain status transitions (such as the slot becoming *unclaimed* by any user, thus subject to negotiation in the following cycle), filtering out the rest.
- (e) Slot update messages use UDP protocol instead of TCP (also the CM is configured with an enlarged UDP buffer size).
- (f) Pool status queries reaching the collector from the negotiator are classified as high-priority, as opposed to those launched by other services in the SI, and from the monitoring processes. Non high-priority queries are subsequently redirected to a secondary collector for the pool (in fact, the backup infrastructure running at FNAL, see [25])

4.2 The SI 2023 scale tests

The main goal for the Spring 2023 tests of the CMS SI was to assess the potential scalability of our Global Pool, considering the following updates since our latest tests (2021, see [24]). These include the evolution in HTCondor software (tested version 10.0.1), a new

physical host for the CM processes (AMD EPYC 7302 at 3 GHz), the adoption of token-based authentication for HTCondor services in our SI [26], and the incrementally improved configuration of our infrastructure. As in previous tests, *uberglideins* [27], running multiple HTCondor *startd* processes per glidein, are used to generate enlarged copies of our Global Pool. A secondary collector daemon is employed as a source of monitoring data for the test pool, reducing the stress from non-essential queries on the main collector.

Initial results indicated a first bottleneck was found in the total capacity of our pool of schedds to supply jobs to the resource pool. At 1 MB of RAM consumed per running job, our schedds, hosted on VMs configured with 50 GB of memory, saturated at approximately 50k running jobs each. With an initial set of 10 identical schedds, the saturation of our test infrastructure was observed at 500k simultaneously running jobs, comparable to our previous result. The set of schedds was therefore enlarged with new machines, such that the new aggregated capacity was, in principle, capable of supporting approximately 1M running jobs.

However, a second bottleneck immediately appeared in the following tests, caused by the CCB service, initially hosted in a VM deployed over a shared hypervisor, which was effectively limiting the maximum number of TCP connections that the broker could manage. As a consequence, open slots could not be matched, therefore constraining the pool from reaching higher scales. The CCB service was then redeployed to a physical node under SI team responsibility, which could then optimize its network configuration, opening again the possibility of reaching higher pool scales.

Ultimately, the saturation of the primary pool collector set the boundary for further scaling of our infrastructure, in a sequence that can be summarized as follows: the collector, which processes slot status updates, becomes increasingly stressed as the size of the pool, and thus the rate of updates, keeps growing. When the collector daemon becomes fully saturated (indicated in Figure 5 by collector duty cycle approaching 1), it can't provide further updated slots status information to the negotiator. Jobs to slots matchmaking process becomes therefore inefficient, leaving many slots unused. This effectively limits the total number of running jobs that our test infrastructure can manage while still using the pool resources efficiently. As Figure 5 shows, our latest tests pushed the scalability of our Global Pool to about 800k simultaneously running jobs. This represents a potential for a factor 4 growth in comparison to our current number of running jobs.

5 Conclusions and future work

The contribution of HPC resources to CMS Computing has increased in the recent past and is expected to continue growing in the coming years. The integration of these resources presents a number of challenges to our infrastructure, including that of scalability, caused by the increasing total amount of compute resources, but more importantly, due to the frequently observed pattern of CPU allocation at HPCs, in bursts.

The CMS SI team, in collaboration with HTCondor and GlideinWMS developers, performs regular assessments of infrastructure scalability. This proactive approach aims to anticipate future scaling challenges and ensures that we remain ahead of any potential limitations. Our Spring 2023 scalability tests demonstrated the capacity of our SI to support up to 800k simultaneous running jobs, which indicates a potential for fourfold growth compared to CMS current number of running jobs. The LHC program extends well into the future, therefore continuously pushing the SI for higher scales, as required by CMS needs, while maintaining stability and efficiency, is a critical element of the future CMS computing program success.



Figure 5. Final results of the Spring 2023 scalability tests of the CMS HTCondor Global Pool, reaching the simultaneous execution of approximately 800k jobs. The collector duty cycle indicates the fraction of seconds over a predefined time interval that the collector process was busy, i.e. not idle waiting to process incoming requests.

Acknowledgements

This work was partially supported by the Spanish Ministry of Science and Innovation under grants PID2019-110942RB-C21, PID2019-110942RB-C22 and PID2020-113807RA-I00, which include FEDER funds from the European Union, and by the US National Science Foundation under Grant No. 2121686.

References

- [1] CMS Collaboration, The CMS experiment at the CERN LHC, JINST 3 (2008) S08004, doi:10.1088/1748-0221/3/08/S08004.
- [2] The HTCondor Software Suite public web site, <https://research.cs.wisc.edu/htcondor/index.html>, accessed September, 2023.
- [3] A. Perez-Calero Yzquierdo et al. “Evolution of the CMS Global Submission Infrastructure for the HL-LHC Era”, EPJ Web of Conferences 245, 03016 (2020).
- [4] J. Balcas et al. “Using the glideinWMS System as a Common Resource Provisioning Layer in CMS”, J. Phys.: Conf. Ser. 664 062031 (2015).
- [5] The Worldwide LHC Computing Grid <http://wlcg.web.cern.ch>, accessed August, 2023.
- [6] The Open Science Grid, <https://osg-htc.org>, accessed September, 2023.
- [7] D. Smith et al. “Sharing server nodes for storage and computer”, EPJ Web of Conferences 214, 08025 (2019).
- [8] M. Mascheroni et al. “Repurposing of the Run 2 CMS High Level Trigger Infrastructure as an Cloud Resource for Offline Computing”, these same proceedings.
- [9] J Adelman-McCarthy et al, “Extending the distributed computing infrastructure of the CMS experiment with HPC resources”, J. Phys.: Conf. Ser. 2438 012039 (2023).
- [10] P. Mhashilkar et al, “HEPcloud, an Elastic Hybrid HEP Facility using an Intelligent Decision Support System”, EPJ Web of Conferences 214, 03060 (2019).
- [11] D. Hufnagel et al. “HPC resource integration into CMS Computing via HEPcloud”, EPJ Web of Conferences 214, 03031 (2019).
- [12] M. Fischer et al, “Effective Dynamic Integration and Utilization of Heterogenous Compute Resources”, EPJ Web of Conferences 245, 07038 (2020).

- [13] J. Flix et al. “Integration of the Barcelona Supercomputing Center for CMS computing: towards large scale production”, these same proceedings.
- [14] The Glidein-based Workflow Management System, <https://glideinwms.fnal.gov/doc/prd/index.html>, accessed September, 2023.
- [15] I. Sfiligoi et al. “The Pilot Way to Grid Resources Using glideinWMS”, 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 2009, pp. 428-432, doi: 10.1109/CSIE.2009.950.
- [16] I. Bird, “Computing for the Large Hadron Collider”, Annual Review of Nuclear and Particle Science, 61, 99-118 (2011).
- [17] National Energy Research Scientific Computing Center, <https://www.nersc.gov/systems/>, accessed September, 2023.
- [18] San Diego Supercomputing Center at the University of California San Diego, <https://www.sdsc.edu/services/hpc/>, accessed September, 2023.
- [19] Pittsburgh Supercomputing Center, <https://www.psc.edu>, accessed September, 2023.
- [20] High Performance Computing at CINECA, <https://www.cineca.it/en/hpc>, accessed September, 2023.
- [21] Hochleistungsrechner Karlsruhe at the Karlsruhe Institute of Technology, <https://www.scc.kit.edu/en/services/horeka.php>, accessed September, 2023.
- [22] Barcelona Supercomputing Center, <https://www.bsc.es/marenostrum/marenostrum>, accessed September, 2023.
- [23] The CernVM File System, <https://cernvm.cern.ch/fs/>, accessed September, 2023.
- [24] A. Perez-Calero Yzquierdo et al. “Reaching new peaks for the future of the CMS HTCondor Global Pool”, EPJ Web Conf. 251 (2021) 02055.
- [25] A. Perez-Calero Yzquierdo et al. “Stability of the CMS Submission Infrastructure for the LHC Run 3”, to be published in the proceedings of the 21st International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Bari, It, 24 - 28 Oct 2022, CMS-CR-2023-033 (2023).
- [26] M. Mascheroni et al. “Adoption of a token-based authentication model for the CMS Submission Infrastructure”, to be published in these proceedings.
- [27] E M Fajardo et al. “How much higher can HTCondor fly?”, J. Phys.: Conf. Ser. 664 062014 (2015).