

## Resource provisioning and workload scheduling of CMS Offline Computing

---

### A. Perez-Calero Yzquierdo<sup>1,2,\*</sup> and the CMS Collaboration

<sup>1</sup>*Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT),  
Madrid, Spain*

<sup>2</sup>*Port d'Informació Científica (PIC),  
Barcelona, Spain*

*E-mail:* [aperez@pic.es](mailto:aperez@pic.es)

The CMS experiment requires vast amounts of computational capacity in order to generate, process and analyze the data coming from proton-proton collisions at the Large Hadron Collider, as well as Monte Carlo simulations. CMS computing needs have been mostly satisfied up to now by the supporting Worldwide LHC Computing Grid (WLCG), a joint collaboration of more than a hundred computing centers geographically distributed around the world. However, as CMS faces the Run 3 and High Luminosity LHC (HL-LHC) challenges, with increasing luminosity and event complexity, growing demands for CPU have been estimated. In these future scenarios, additional contributions from more diverse types of resources, such as Cloud and High-Performance Computing (HPC) clusters, will be required to complement the limited growth of the capacities of WLCG resources. A number of strategies are being evaluated on how to access and use WLCG and non-WLCG processing capacities as part of a combined infrastructure, successfully exploit an increasingly more heterogeneous pool of resources, efficiently schedule computing workloads according to their requirements and priorities, and timely deliver analysis results to the collaboration, which are described in this paper.

*40th International Conference on High Energy physics - ICHEP2020  
July 28 - August 6, 2020  
Prague, Czech Republic (virtual meeting)*

---

\*Speaker

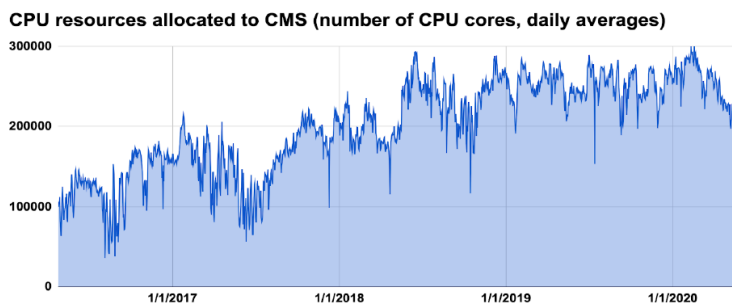
## 1. Introduction

The Submission Infrastructure (SI) Group of the CMS experiment at CERN runs the computing infrastructure in which all processing, reconstruction, simulation, and analysis of physics data takes place. In order to do that, we manage GlideinWMS [1] and HTCondor [2] tasks in CMS, operating a Global Pool [3] of computing resources. The SI team activities include overcoming current operational limitations, preparing for future scales and requirements, integrating new resource types and workload management (WM) systems, and communicating CMS priorities to the HTCondor and GlideinWMS development teams, to stimulate and integrate new features as needed by the experiment.

As part of our mission, a successful scheduling of all CMS workloads is paramount. A balanced scheduling needs to take into account the diversity of CMS tasks (from detector data reconstruction and production of simulated collisions to physics analysis) and their resource requirements (processing on CPU and/or accelerators, memory, input and output data). It is essential to note that what “good scheduling” is can be interpreted in several ways, such as ensuring that all available resources are efficiently used all the time, a fair share of resources between groups of workload submitters, the completion of tasks according to their prioritization and with minimal job failures and manual intervention, and the flexibility to handle uncommon requests (e.g. tasks demanding unusually high amounts of memory). Our efforts are dedicated to ensuring that these diverse goals are reasonably covered simultaneously, even under evolving scenarios, as described in the following sections. An example of a simple metric of success in our operations is the efficiency in using the Global Pool resources, at 95% or more.

## 2. The scalability challenge

The Global Pool has been under sustained growth, doubling in size in the past years, as Fig.1 shows, driven by increasing resource requests during the LHC Run 2 (2015 to 2018). Our infrastructure currently aggregates nearly 300k CPU cores, as during LHC Run 2 and then in the 2019 to 2021 shutdown period (LS2), we have been progressively adding opportunistic and non-standard resources. Key examples of this expansion have been the use of the CMS High Level Trigger (HLT) farm as opportunistic resource when not in use for data taking, and the increasing proportion of beyond pledge Grid slots, along with HPC and cloud resources being added into the pool.



**Figure 1:** Global Pool size in number of CPU cores averaged daily over the last 4 years

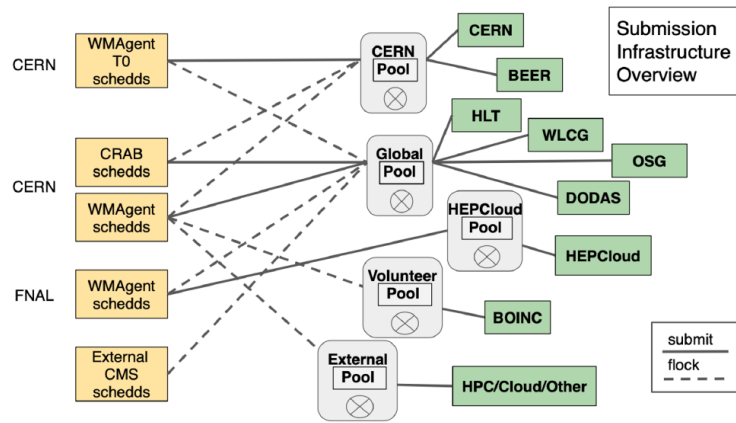
Looking on at the growing scales of data in the HL-LHC phase, driven by increasing trigger rates and event complexity, CMS has estimated that computational needs will be at least one order of magnitude higher than current capacities [4]. However, this gap is likely to be reduced by ongoing work on CMS software and data formats. In addition, further job parallelization is still expected, which should reduce the load on the SI (handling more work with fewer tasks). These uncertainties propagate into the SI realm, with unclear answers to the questions of how big of a pool will we need to manage or how many job schedulers will we have to employ in order to execute future workloads. Moreover, making predictions even harder, compute resources growth will unlikely come from increasing capacities at Grid sites. Indeed, HPC resources are expected to provide a fraction of the resource mix, adding complexity to the infrastructure, as discussed in the following section.

In order to preventively tackle potential scalability limitations, the SI team has been periodically running dedicated tests to explore the growth capacity and stability of our infrastructure resulting from the continuous evolution of our tools (find original tests described in [5]). A testbed HTCondor pool is stressed to the limits of its capacity by successively increasing the number of simultaneously running tasks. An analysis of failure modes is performed, and its results are discussed with the development team. In our latest test, performed in July 2020 using HTCondor version 8.9.7, our test pool reached about 400k concurrent running jobs. The bottleneck was identified to be the *collector*, a component of the pool holding the information about its status, which is consumed by some essential SI services. An analysis ensued on how to reduce the collector load, for example minimizing the impact from external queries, linked to monitoring services, on the collector.

### 3. Increasing complexity in CMS SI

Starting with a single HTCondor pool, the CMS SI has evolved over the recent years into a model of multiple *federated* pools of resources (see Fig.2). A number of specialized submission nodes (*schedds*), located at CERN and FNAL are primarily attached to one of the pools, however can *flock* (submit) work into secondary pools when the resource demands are not satisfied on the primary. Resources in the Global Pool are still mainly acquired with GlideinWMS pilots [1], although the SI now also includes locally instantiated slots (e.g DODAS [6]), BOINC [7] (CMS@Home [8]), opportunistic resources (such as the CMS HLT), etc. The HEPCloud [9] pool, joining cloud and HPC resources in the USA, is managed from the FNAL site. The CERN pool includes on-site resources allocated to CMS, but also opportunistic BEER [10] and Azure cloud slots. External pools can be federated into the SI in order to execute CMS tasks.

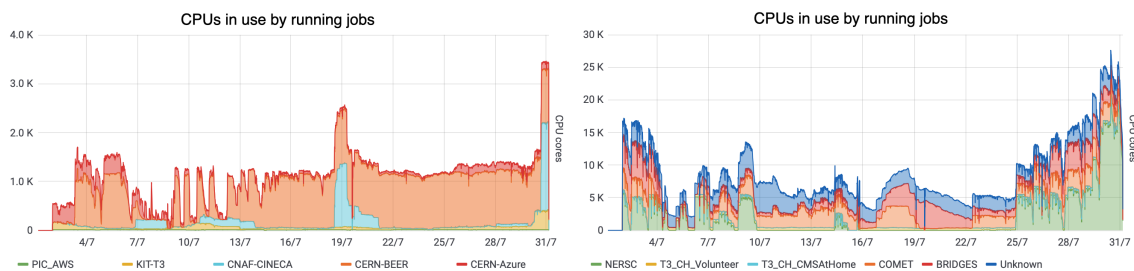
Regarding the use of HPC resources, CMS aims at increasing their usage on the mid to long term future (Run3 and HL-LHC) [11]. Growing funding for HPC infrastructures, evolving in many cases into Exascale machines, makes them very attractive for the LHC collaborations, being considered a key part of computing model evolution towards the HL-LHC. However, contrary to pledged resources deployed with standardized grid technologies, each HPC installation is different, both in terms of technical and political matters, which need to be resolved case by case, demanding big integration efforts. Provisioning HPC resources into the CMS SI can thus proceed via a number of routes, from pilot-based (e.g. on CINECA's Marconi [12]), to federation with the HEPCloud pool. Last, the exploitation of HPC resources also presents a number of operational challenges,



**Figure 2:** CMS SI current configuration, including multiple federated pools of resources allocated from diverse origins (green boxes) and sets of distributed job schedders (schedds)

as, for example, not all workflows are suitable for HPC resources (e.g. due to diverse constraints related to input data access). Automatically selecting suitable workflows for each HPC requires an additional effort on producing an enhanced characterization of both resources and workloads.

The integration of opportunistic and dynamic resources has seen recent progress, in most cases in the form of seamless extensions of Grid sites, e.g. CERN into Azure [14] and Spanish Tier-1 at PIC into AWS [15] clouds, and into available research centre or university campus clusters, such as the recent expansions of German Tier-1 at KIT [13] and US Tier-2 Purdue capacities. In some cases, these non-standard resources require enhanced workload-to-resource matchmaking, which has been implemented by using the so-called site-customizable pilots. Pilots enabled for local customization read additional constraints, set by local site admins, from a standard location in CMS file space, in order to exclude the types of jobs that can't run efficiently on their opportunistic extensions. Figure 3 presents some success cases, the integration and use of NERSC [16] resources being the most relevant of them, in terms of the compute capacity acquired for CMS use.



**Figure 3:** CPU cores in use by jobs running in several extensions of European CMS sites, including CERN (left), or HEPCloud and US-based based resources, e.g. NERSC HPC (right)

#### 4. Support for heterogeneous resources

CMS needs to be ready to utilize non-CPU resources, expected to comprise an increasingly larger fraction of the computing power available, in particular at the HPC facilities, many of

which are built on heterogeneous architectures (CPU+GPU). Using the non-CPU capacity requires the evolution of the CMS software, but also of the CMS WM and SI tools. Specifically, CMS needs to setup mechanisms to properly validate and tag those workloads that can/must use GPUs. Concerning SI, CMS already has access to a small amount of GPU resources in the Global Pool (mostly at US sites). Both GlideinWMS and HTCondor provide us with tools to manage GPUs (e.g. the HTCondor GPU discovery tool), along with use of tags to completely characterize the slots for proper matchmaking to job requests. For example GPU-oriented jobs can include in their description:

```
request_gpus = 1
requirements = (CUDARuntimeVersion >= 4.0) && (CUDACapability > 4.0)
```

## 5. Ongoing and future work

The main CMS SI team goals include continuing with the integration of non-Grid resources and refining our tools to enable CMS access and effective use of GPUs. Regarding SI performance, we are committed to the continuous improvement of our monitoring tools for a better understanding and mitigation of any sources of scheduling inefficiency. With the support of the HTCondor team, our work will include new rounds of the scalability and stability tests. We'll start exploring the use of token-based authentication, already natively supported by HTCondor, as a substitute to x509-based authentication. In short, SI aims at providing CMS with a flexible and efficient infrastructure for the execution of any type of workloads.

As an example of CMS SI flexibility, we adapted the infrastructure in order to enable a quick integration of non-CMS workloads. A CERN task force was formed in March 2020 to coordinate support from the Particle Physics community for Covid-19 related work, endorsing the donation of CERN and WLCG computing power, specifically to Folding@Home [17] (F@H) tasks, a collaborative computing project that simulates protein folding, in particular those of the SARS-CoV-2 virus. CMS SI performed the integration of F@H payloads into the CMS Global Pool, allowing them to run as low-priority HTCondor jobs consuming steadily 4K CPU cores during the period April-June 2020. Apart from altruistically providing resources for this humanitarian cause, a number of useful lessons were learned, such as how to inject directly via HTCondor a non-standard executable and the use of low priority generic workloads to fill unused slots in our pool. This is a role that certain CMS jobs, such as CMS@Home tasks, could play in the future, thereby improving the global efficiency in the utilization of CMS computing resources.

## 6. Conclusions

The CMS SI team is responsible for the good scheduling of CMS workflows, according to diverse criteria described and in a continuously evolving environment. Looking into the late 2020's, CMS expects increasing scales for computing needs and a bigger component from non-Grid providers, leading to a more complex infrastructure. Our team works on detecting bottlenecks preventively, assessing SI scalability in relation to CMS projected computing needs. The integration of HPC resources, key for CMS success into the HL-LHC era, is proceeding via a number of

technical routes. Ongoing and future work will be dedicated to endowing CMS computing with further flexibility and reach.

The author acknowledges support by Spain's Ministry of Economy and Competitiveness grant FPA2016-80994.

## References

- [1] The Glidein-based Workflow Management System, <https://glideinwms.fnal.gov/doc.prd/index.html>
- [2] HTCondor public web site, <https://research.cs.wisc.edu/htcondor/index.html>
- [3] J. Balcas et al. "Using the glideinWMS System as a Common Resource Provisioning Layer in CMS", J. Phys.: Conf. Ser. **664** 062031 (2015).
- [4] <https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSOfflineComputingResults>
- [5] J. Balcas et al. "Pushing HTCondor and glideinWMS to 200K+ Jobs in a Global Pool for CMS before Run 2", J. Phys.: Conf. Ser. **664** 062030 (2015).
- [6] D. Spiga et al. "Exploiting private and commercial clouds to generate on-demand CMS computing facilities with DODAS", EPJ Web of Conferences. **214**. 07027 (2019).
- [7] D. P. Anderson. "BOINC: A Platform for Volunteer Computing", J Grid Computing (2019).
- [8] LHC at Home, <https://lhcatome.cern.ch/lhcatome/>
- [9] S. Timm et al. "Virtual machine provisioning, code management, and data movement design for the Fermilab HEPCloud Facility", J. Phys.: Conf. Ser. **898** 052041 (2017).
- [10] D. Smith et al. "Sharing server nodes for storage and computer", EPJ Web of Conferences. **214**. 08025 (2019).
- [11] A. Pérez-Calero Yzquierdo et al. "CMS Strategy for HPC resource exploitation", EPJ Web of Conferences **245**, 09012 (2020).
- [12] CINECA <https://www.cineca.it>
- [13] M. Schnepf et al. "Dynamic Integration and Management of Opportunistic Resources for HEP", EPJ Web of Conferences **214**, 08009 (2019)
- [14] D. Giordano et al. "CERN-IT evaluation of Microsoft Azure cloud IaaS, "<https://zenodo.org/record/48495#.X9I7JC8rx24>
- [15] J. Flix et al. PIC report to HEPIX Fall 2019 workshop, <https://indico.cern.ch/event/810635/contributions/3592939/>
- [16] National Energy Research Scientific Computing Center (NERSC) <https://www.nersc.gov>
- [17] <https://foldingathome.org>