Connecting Restricted, High-Availability, or Low-Latency Resources to a Seamless Global Pool for CMS

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 052037 doi:10.1088/1742-6596/898/5/052037

Connecting Restricted, High-Availability, or Low-Latency Resources to a Seamless Global Pool for CMS

J Balcas[1], B Bockelman[2], D Hufnagel[3], K Hurtado Anampa[4], B Jayatilaka[3], F Khan[5], K Larson[3], J Letts[6], M Mascheroni[3], A Mohapatra[7], J Marra Da Silva[8], D Mason[3], A Perez-Calero Yzquierdo[9,10], S Piperov[11], A Tiradani[3], and V Verguilov[12], on behalf of the CMS Collaboration

[1] California Institute of Technology, Pasadena, CA, USA
[2] University of Nebraska - Lincoln, NE, USA
[3] Fermi National Accelerator Laboratory, Batavia, IL, USA
[4] University of Notre Dame, IN, USA
[5] National Centre for Physics, Quaid-I-Azam University, Islamabad, Pakistan
[6] University of California, San Diego, La Jolla, CA, USA
[7] University of Wisconsin, Madison, WI, USA
[8] Universidade Estadual Paulista, São Paolo, Brazil
[9] Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, Madrid, Spain
[10] Port d'Informació Científica, Barcelona, Spain
[11] Brown University, Providence, RI, USA
[12] Bulgarian Academy of Sciences, Sofia, Bulgaria

E-mail: jletts@ucsd.edu

Abstract. The connection of diverse and sometimes non-Grid enabled resource types to the CMS Global Pool, which is based on HTCondor and glideinWMS, has been a major goal of CMS. These resources range in type from a high-availability, low latency facility at CERN for urgent calibration studies, called the CAF, to a local user facility at the Fermilab LPC, allocation-based computing resources at NERSC and SDSC, opportunistic resources provided through the Open Science Grid, commercial clouds, and others, as well as access to opportunistic cycles on the CMS High Level Trigger farm. In addition, we have provided the capability to give priority to local users of beyond WLCG pledged resources at CMS sites. Many of the solutions employed to bring these diverse resource types into the Global Pool have common elements, while some are very specific to a particular project. This paper details some of the strategies and solutions used to access these resources through the Global Pool in a seamless manner.

1. Introduction

CMS [1], one of the four main experiments at the Large Hadron Collider (LHC) [2], a proton-proton and heavy ion accelerator at CERN in Geneva, Switzerland, was designed from the beginning as a global experiment with a distributed computing infrastructure, as described in [3]. In Run 1 of the LHC these resources were predominantly provided by a mix of Grid sites (in the context of the MONARC model [4]) and local batch resources. During the long shutdown of the LHC in 2013-2014,

cloud infrastructures, diverse opportunistic resources, and HPC supercomputing centers were made available to CMS, further complicating the operations of the submission infrastructure.

CMS transitioned to a pilot-based submission system based on glideinWMS [5] and HTCondor [6] during Run 1 of the LHC, completing the transition by late 2013 [7]. Inefficiencies in direct submission architectures due to networking and site issues drove the transition to a light-weight pilot submission system.

As shown in Figure 1, the main elements of glideinWMS are factories, which submit light-weight pilots to remote sites, and a glideinWMS frontend, which requests the pilots based on the need for resources in the underlying HTCondor pool. However, it is also possible for a trusted resource to join the HTCondor pool when instantiated by a local user or site.

The HTCondor pool itself consists of submit nodes which hold the job queues, HTCondor startd daemons which run on execute nodes, and a central manager (collector and negotiator) which negotiates matches between queued jobs and resources. GSI authentication based on Grid certificate proxies is used to establish trust between the various elements of the pool. At job run time, CMS uses glexec [8] where implemented to switch context to the proxy used to submit the job.

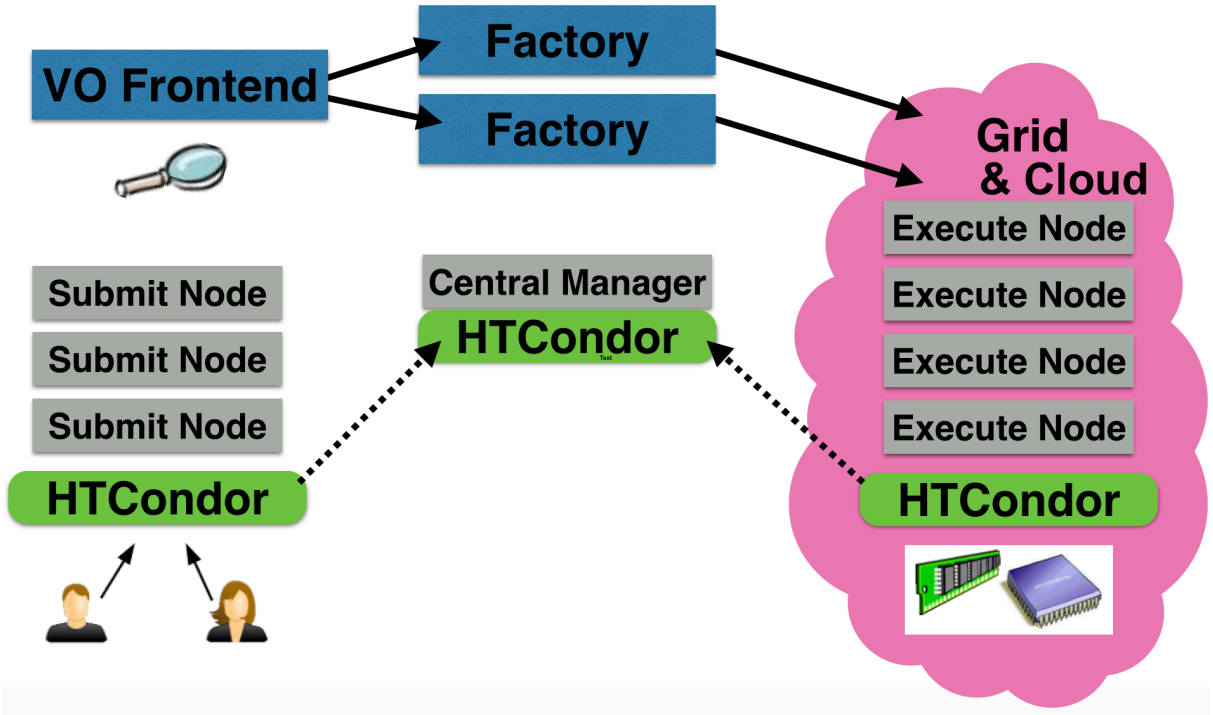

Figure 1. Architecture of a glideinWMS pilot submission system to an HTCondor pool.

During the first two years of the LHC Run 2 in 2015 and 2016, the diversity of resources that connected to the Global Pool evolved from traditional Grid resources. These included a low-latency, high availability resource at CERN called the CAF (CERN Analysis Facility) with a restricted user community, a similar analysis facility at Fermilab called the LPC CAF but with strict site security restrictions, special computing allocations at super computing centers at NERSC and SDSC, resources not pledged to CMS but accessible through the Open Science Grid (OSG) [9], and the high level trigger computing farm (HLT) of CMS, which could only be used when the CMS detector was not taking data. In addition, certain sites had restrictions on how pilots could run (Texas A&M University), or want to instantiate their own resources without the use of the glideinWMS factories, or

connect their own glideinWMS instances to flock work to the Global Pool in a seamless way. In the following sections we will discuss the strategies and solutions employed (some of them common) to connect these diverse resources to the CMS Global Pool.

2. The CAF at CERN

The CMS CERN Analysis Facility (CAF) is dedicated to latency-critical activities like detector calibration and alignment, detector and trigger commissioning, and very high priority physics analyses. Connected to the Global Pool and accessible within CRAB, the modern CMS physics analysis job submission middleware [10], the CAF resource is restricted to a small number of users. Similar in structure to the Tier-0 facility [11] on OpenStack [12], this set of resources is available on demand with very low latency. This facility was created to replace a similar resource on LSF at CERN.

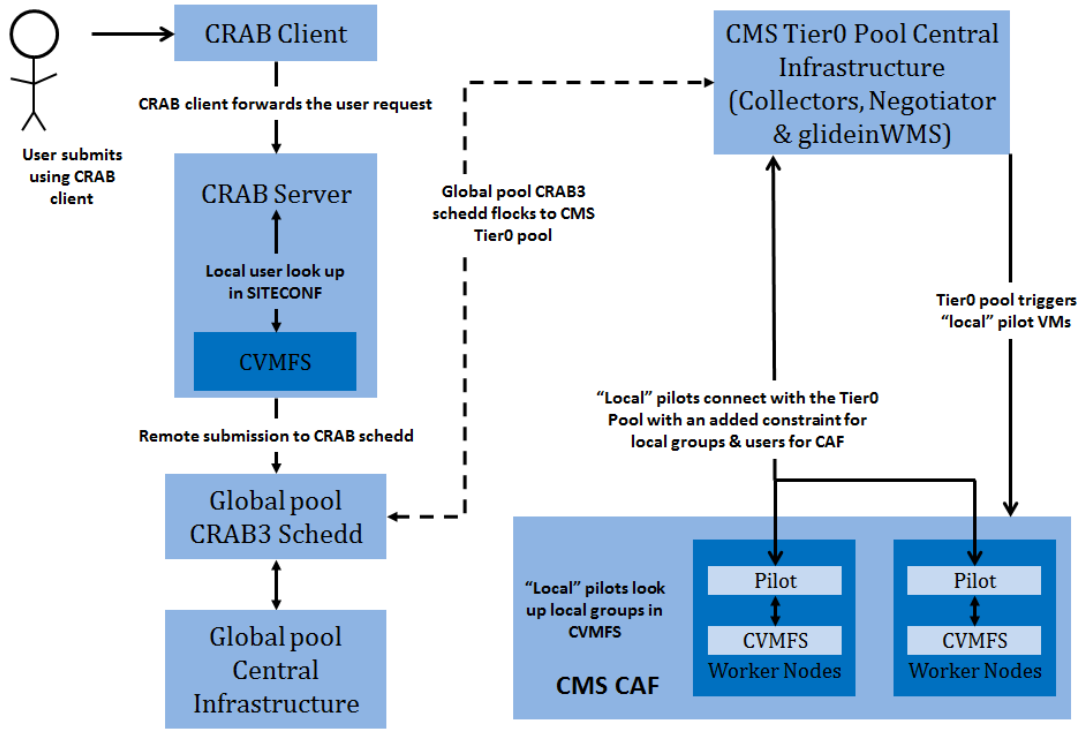

Figure 2. Diagram of the CERN CAF integration.

OpenStack Virtual Machines (VMs) for the CAF are spawned using the glideinWMS factory at CERN. Low latency (with respect to jobs) is ensured by running these VMs for a month. Since these resources might remain idle for long periods of time, they are connected to a separate HTCondor pool for the Tier-0 (which is itself a high-availability, low-latency resource), but remain available at all times to the Global Pool through the flocking mechanism of HTCondor, as can be seen in Figure 2.

In order to restrict the usage of the CAF to priority users only, we developed a configurable mechanism to implement this functionality, which is described in the next section.

3. Local User Prioritization in CRAB

Certain resources connected to the Global Pool must give priority or restrict access to a certain group of users, which we call "local users". One use case is the CAF, but many sites also have computing resources beyond their WLCG pledge that they wish to give access priority to local analysis users.

A high level block diagram of our solution to this challenge is shown in Figure 3. The site administrator can define a list of local users according to tight CERN computing username or alternatively a VOMS group (such as a national group) in a file which is published in CVMFS. When any user submits a CRAB job, the CRAB server can look up whether the user has special priority on the desired sites for job execution. If so, the CRAB server adds a ClassAd to the job JDL called "CMSGroups". When the Global Pool frontend sees such jobs in the queue, it can request special pilots of the factories with a special tag "CMSIsLocal", which can only match in the Negotiator to jobs from local users and will reject general work. It is up to the site to prioritize these pilots over generic ones from the Global Pool, or direct them to special resources.

This mechanism was deployed for the CAF as well as by several CMS Tier-2 sites in the U.S., Germany, Spain, and Italy. In addition, it was used by a Tier-3 site at Texas A&M University in conjunction with another extension, which is described in the following section.

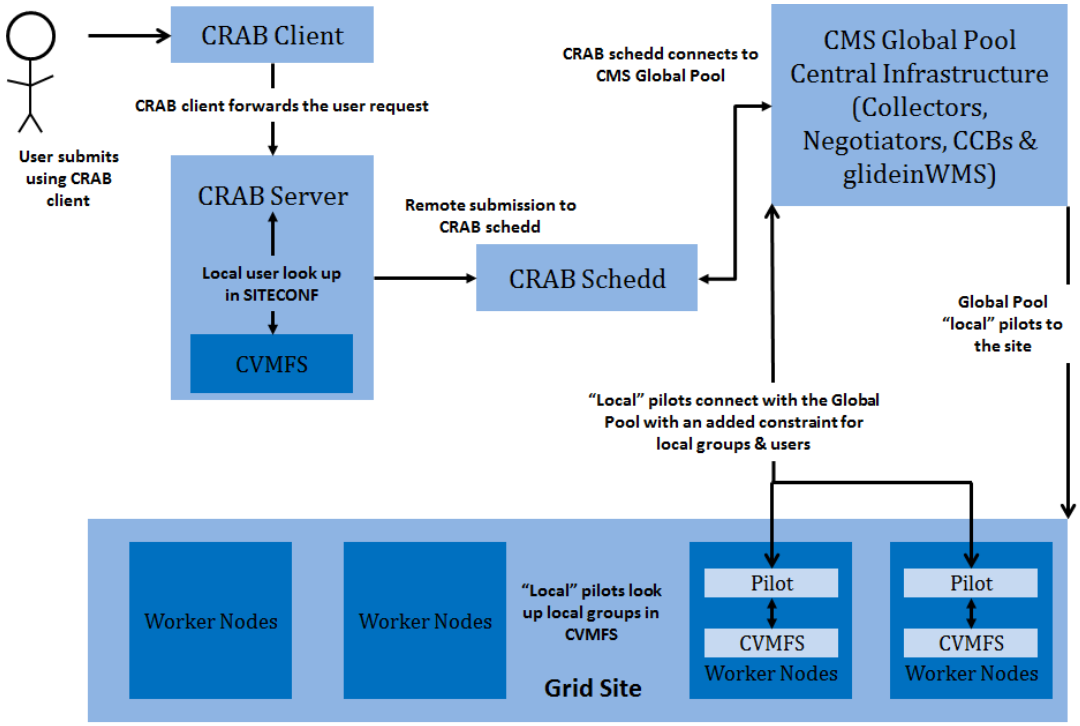

Figure 3. Local user prioritization in the CMS Global Pool.

4. Site-customized pilots

The Tier-3 site at Texas A&M University (TAMU) presented a particular challenge in that all jobs, including glideins, which are submitted to computing resources there must be run under a local user credential, not a CERN credential like normal pilots. As part of the security model, we also want only that particular user's jobs to run on such a pilot, not general work, or even the work of other local users.

CMS worked with the site administrators to write a script that queries the Global Pool job queues for TAMU local users and their idle jobs. Based on these idle job numbers, they allow pilots to run on their computing resources and based on the users with idle jobs, they set a corresponding environment

variable 'USER_DN' for the pilot running under each user's local account. When the CMS pilot runs, it checks for this environment variable and then adds an extra constraint to the start expression to only match jobs from this user.

5. The CMS LPC CAF at Fermilab

The LHC Physics Center (LPC CAF) is a regional analysis facility at the Fermi National Accelerator Laboratory (FNAL). The primary objective of the LPC CAF is to facilitate members of U.S. CMS collaboration institutes in analysis of the data being taken by the LHC. Previous versions of CRAB had a special plugin to allow users to submit work directly to the LPC CAF queues. Modern versions (CRAB3) have a central server at CERN which submits jobs to the Global Pool, whose job schedulers are primarily at CERN. The restriction here is that security policies at FNAL do not allow Grid job submission, including pilots from CERN.

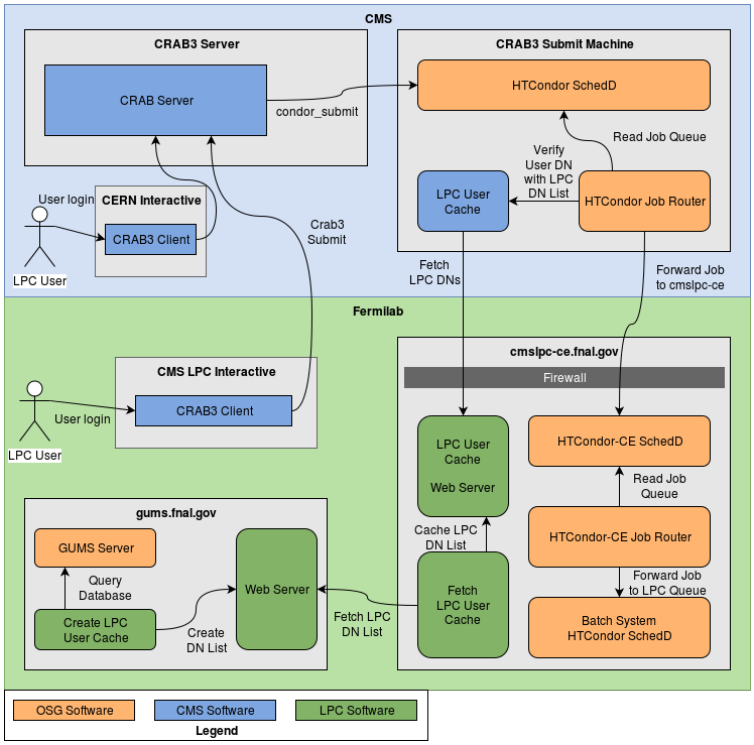

Figure 4. Local submission to the CMS LPC CAF at FNAL.

The solution we employed, which is pictured in a block diagram in Figure 4, uses an HTCondor JobRouter running on the CRAB3 schedulers at CERN to 'route' jobs of only LPC CAF users from the CRAB3 Grid schedulers to the LPC CAF using an HTCondor CE under control of the FNAL site administrators. In this way the site security policies can be respected. This CE is behind the FNAL firewall and only fetches jobs from specific trusted CRAB3 schedulers. The CE also periodically fetches a list of allowed LPC CAF users from a GUMS server, also inside the FNAL firewall, never exposing this information to the outside. Jobs from non-LPC CAF users are rejected.

6. The CMS HLT at CERN

The over 20,000 cores of the CMS High Level Trigger (HLT) farm can be utilized when CMS is not taking data. The HLT runs the trigger software on the physical machine but starts OpenStack Cloud VMs to allow production jobs. Since this represents a significant fraction of the total computing power available to CMS globally, we want to take advantage of idle cycles on this resource.

However, unlike other resources, we do not use the glideinWMS factories, but rather the VMs of this resource are instantiated in the Cloud with a server certificate trusted by the Global Pool and an HTCondor startd. A block diagram of the interaction of the Global Pool with the HLT is shown in Figure 5. Once the VMs connect to the Global Pool, they can accept jobs designated to run on the HLT.

When the HLT needs to be used as a trigger farm again, the VMs are suspended to disk. The running jobs have been configured to be able to be suspended for up to 24 hours. If the VMs come back up within that time (i.e. during the inter-fill periods of the LHC), the jobs can resume execution.

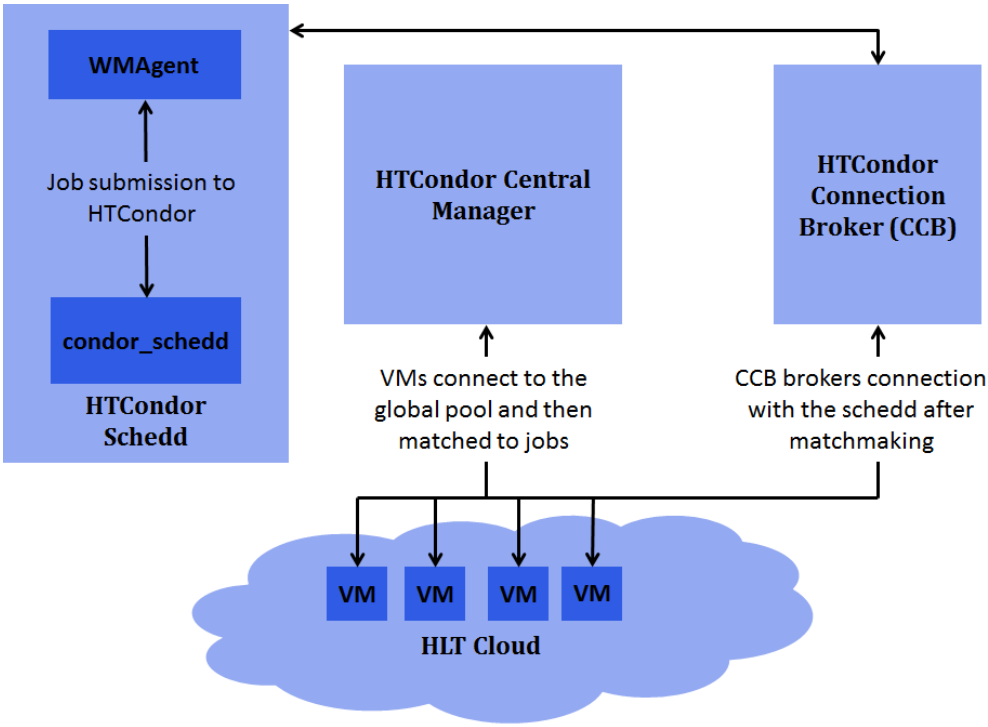

Figure 5. The integration of the HLT farm resources into the CMS Global Pool.

7. OSG Resources and other Allocations

While CMS makes use of many opportunistic resources available through existing CMS sites, the resources of the Open Science Grid [9] represent another potential source of opportunistic processing. OSG and CMS work together closely, so an effort was made to use OSG resources for CMS general purpose computing. To that end, we developed a way to address the various OSG opportunistic sites in a single coherent way in CMS. Since both the OSG and CMS use glideinWMS and HTCondor as the base of the submission infrastructure, the integration was not difficult.

During the past several years CMS institutions have obtained allocations at supercomputing centers such as NERSC and SDSC. While SDSC was relatively easy to integrate into our computing infrastructure, NERSC posed challenges for access via the glideinWMS factories and for providing a local runtime environment suitable for CMS jobs. We addressed the former via the use of BOSCO [13] to facilitate glideinWMS factory submission into the HPC batch system. A CMS compatible runtime environment is provided via the use of Parrot [14] and more recently Shifter [15].

However, the available storage options are very limited at these opportunistic sites. While we in principle want to support all types of workflows at these sites, lately we have mainly run fully integrated beginning-to-end workflows that run all steps of the Monte Carlo generation and processing

chain in a single job. Therefore, these jobs require very little input and only write minimal output data that is then staged directly from the worker node to Fermilab.

8. Connecting Institutional Schedulers to the Global Pool in a Manageable Way

In the future, we will be connecting many different types of job submitters and dynamic resources to the Global Pool. Already we are standardizing procedures and requirements to connect HTCondor schedulers run by CMS institutions to the Global Pool, as well as Cloud resources instantiated by local institutions. The latter takes advantage of tools and techniques outlined in this paper, such as restriction of locally instantiated resources to match jobs only from local users.

While CRAB handles grid job submission for analysis jobs depending on the CMS framework executable (cmsRun), late-stage analysis tasks often are independent and batch systems are more suitable to handle these type of jobs. CMS Connect [16] is used as one of the submission interfaces for bare HTCondor jobs to the CMS Global Pool. Another possibility is allowing institutionally-run HTCondor submission nodes to flock work to the CMS Global Pool. However, allowing many institutional schedulers access to the Global Pool can bring management problems, as user accounting, job reporting, and job control measures to protect the Global pool have to be implemented.

Having a central submission interface these schedulers can talk to as a gateway before going to the Global Pool is important. In order to do this, pilot jobs are submitted by the users via a python-based package called pyglidein [17], originally developed for the IceCube experiment, to CMS Connect so that these pilots can talk to their institutional submission scheduler to overflow jobs into the Global Pool, as shown in Figure 6.

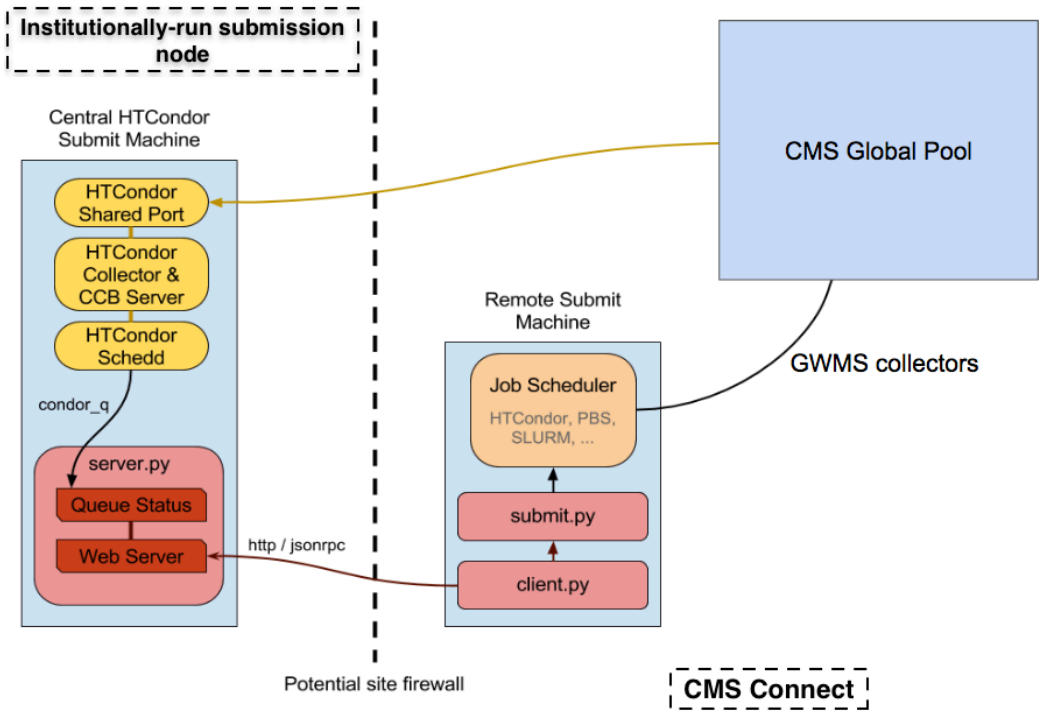

Figure 6. Pyglidein workflows as a way to allow institutionally-run submission nodes to overflow jobs to the Global Pool via CMS Connect.

9. Conclusions

CMS has developed several tools and techniques to seamlessly connect opportunistic or dedicated restricted, high-availability, or low-latency resources, to the Global Pool. The types of resources use

either instantiation in the Cloud or more traditional Grid-based submission, often have restricted user access or trust, but still can utilize the powerful CMS services such as CRAB thanks to their inclusion in the Global Pool.

References

[1] S. Chatrchyan *et al*. CMS Collaboration 2008 The CMS experiment at the CERN LHC *J. Inst*. **3** S08004
[2] Evans L and Bryant P 2008 LHC Machine *J. Inst*. **3** S08001
[3] Gutsche O *et al*. 2014 CMS computing operations during Run 1 *J. Phys. Conf. Ser.* **513** 032040
[4] The MONARC project http://monarc.web.cern.ch/MONARC/
[5] Thain D, Tannenbaum T, and Livny M, 2005 Distributed Computing in Practice: The Condor Experience *Concurrency and Computation: Practice and Experience*, **17(2-4)** 323
[6] Sfiligoi I *et al*., 2009 The Pilot Way to Grid Resources Using glideinWMS *WRI World Congress* **2** 428
[7] Belforte S *et al*. 2014 Evolution of the pilot infrastructure of CMS: towards a single glideinWMS pool *J. Phys. Conf. Ser.* **513** 032041
[8] Sfiligoi I *et al*. 2012 glideinWMS experience with glexec *J. Phys. Conf. Ser.* **396** 032101
[9] Jayatilaka B *et al*., 2015 The OSG Open Facility: A Sharing Ecosystem Using Harvested Opportunistic Resources *J. Phys. Conf. Ser.* **664** 032016
[10] Cinquilli M *et al*., 2015 CRAB3: Establishing a new generation of services for distributed analysis at CMS *J. Phys. Conf. Ser.* **396** 032026
[11] Hufnagel D *et al*., 2015 The CMS TierO goes Cloud and Grid for LHC Run 2 *J. Phys. Conf. Ser.* **664** 032014
[12] Bell T *et al*., 2015 Scaling the CERN OpenStack cloud, *J. Phys. Conf. Ser.* **664** 022003
[13] Weitzel D, Fraser D, Bockelman B, and Swanson D, 2012 Campus grids: Bringing additional computational resources to HEP researchers," in *J. Phys. Conf. Ser.* **396(3)** 032116
[14] Thain D and Livny M, 2005 Parrot: An Application Environment for Data-Intensive Computing, Scalable Computing: Practice and Experience **6**(3) 9
[15] Jacobsen D M and Canon R S, 2015 Contain this, unleashing docker for hpc, Proc. of the Cray User Group, 2015.
[16] Hurtado Anampa K *et al*., 2017 CMS Connect, to be published in these proceedings.
[17] Schultz D, Riedel B, and Merino G, 2017 Pyglidein - a simple HTCondor glidein service, to be published in these proceedings.

Acknowledgments

We would like to thank our colleagues and collaborators from CMS, the LHC, CERN, Fermi National Accelerator Laboratory, the Open Science Grid, and the HTCondor and glideinWMS development communities. The present work is partially funded under grants from the U.S. Department of Energy, the National Science Foundation, and Spain Ministry of Economy and Competitiveness grant FPA2013-48082-C2-1/2-R. The Port d' Informació Científica (PIC) is maintained through a collaboration between the Generalitat de Catalunya, CIEMAT, IFAE and the Universitat Autonoma de Barcelona.