

Evolution of the pilot infrastructure of CMS: towards a single glideinWMS pool

S Belforte¹, O Gutsche², J Letts³, K Majewski², A McCrea³, I Sfiligoi³

¹INFN Univ. di Trieste, Italy

²Fermi National Accelerator Laboratory, Batavia, IL, USA

³UC San Diego, California, USA

E-mail: gutsche@fnal.gov

Abstract. CMS production and analysis job submission is based largely on glideinWMS and pilot submissions. The transition from multiple different submission solutions like gLite WMS and HTCondor-based implementations was carried out over years and is coming now to a conclusion. The historically explained separate glideinWMS pools for different types of production jobs and analysis jobs are being unified into a single global pool. This enables CMS to benefit from global prioritization and scheduling possibilities. It also presents the sites with only one kind of pilots and eliminates the need of having to make scheduling decisions on the CE level. This paper provides an analysis of the benefits of a unified resource pool, as well as a description of the resulting global policy. It will explain the technical challenges moving forward and present solutions to some of them.

1. Introduction

CMS [1], one of the four experiments at the Large Hadron Collider (LHC) [2], a proton-proton accelerator at CERN in Geneva, Switzerland, was designed from the beginning as a global experiment with a distributed computing infrastructure, described in [3]. This infrastructure is large based on GRID technologies. In the beginning of the experiment, mainly two different job submission technologies have been used to execute CMS workloads on the CPUs at the CMS sites. Predominantly in Europe, the gLite WMS system [4] was used. In the United States it was HTCondor_G [5]. These systems were operating in direct submission modes, either submitting jobs directly to the compute elements at the sites or through an intermediate workload management system. During LHC run 1 (2010-2012), inefficiencies were observed in jobs using the direct submission techniques, resulting from a large dependence on network, site and GRID middleware failure rates. To reduce the dependency, pilot-based submission systems were developed that first submit a lightweight pilot to the compute elements at the sites, and only after the pilot started on a workernode executed the actual job, reducing the failure rate significantly. In the course of LHC run 1, CMS transitioned from the direct submission systems to the pilot-based glideinWMS system [6], which is based on HTCondor. The transition of the production activity started in the beginning of 2011 and was completed in Fall 2011 (see figure 1). Analysis adopted the glideinWMS system later and reached >95% adoption by Fall 2013 (see figure 2).

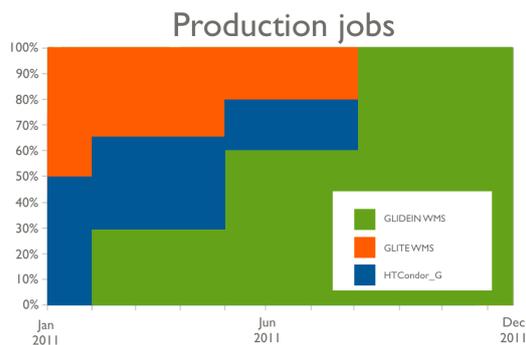


Figure 1. Transition of production activity to glideinWMS completed in Fall 2011.

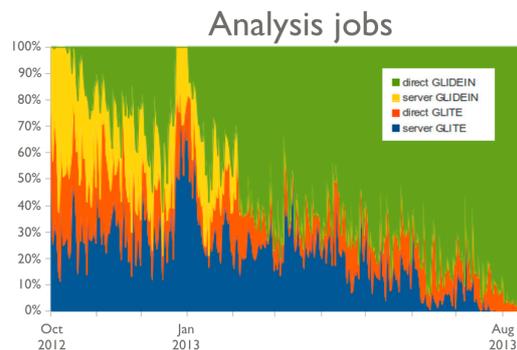


Figure 2. Transition of analysis activity to glideinWMS, reached >95% adoption in Fall 2013.

During the transition, all submission modes needed to be supported at the same time, which led to a not completely efficient setup. After the adoption of the glideinWMS system is near complete, CMS will optimize the setup to benefit from all advantages of a single pilot-based submission infrastructure. We will describe the current system and the future glideinWMS setup in the following.

2. Current glideinWMS setup

The current glideinWMS setup is designed to allow for direct submissions through HTCondor_G and gLite WMS to the sites at the same time as the usage of pilot-based submissions through glideinWMS (see figure 3).

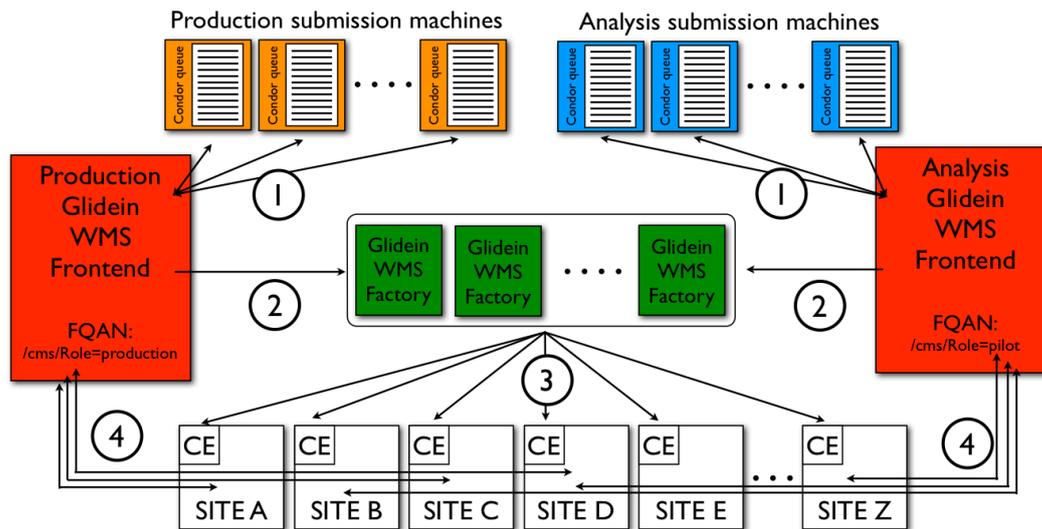


Figure 3. Current glideinWMS setup with two separate systems for analysis and production to allow for simultaneous submission to the sites through glideinWMS, HTCondor_G and gLite WMS.

CMS requires a resource sharing fraction of 50% at its Tier-2s between production and analysis. To allow for simultaneous submission through all three systems, CMS had to setup

two different complete glideinWMS systems consisting in a simplified view of frontend, factories, submission machines and compute elements (CEs) at the sites to provide GRID access to the workernodes of the CPU farms.

- (1) Both the analysis and production frontends monitor their attached submission machines for jobs to be executed.
- (2) Both frontends then request pilot submissions to sites defined in the pending jobs from the factories, while a sharing of factories is already possible in the current setup.
- (3) The factories submit pilots to the requested sites using the proxy of the frontend. The compute element (CE) at the sites then prioritize according to user and VOMS role of the proxies of the jobs. CMS allocates 50% of a sites CPU resources to the production VOMS role. The remaining 50% are shared fairly between all analysis users that submit to the site. Pilots requested by the analysis frontend can run multiple analysis jobs one after the other. If analysis pilots and direct user submission would simply run in parallel at a site, analysis jobs through the glideinWMS system would be disadvantaged and more priority would be given to directly submitted analysis jobs. To overcome this limitation, the analysis frontend uses several different analysis proxies in a round-robin fashion to boost the priority of glideinWMS analysis jobs relative to directly submitted analysis jobs.
- (4) Both frontends match jobs to pilots at the requested sites separately. These jobs mainly run within the context of the pilot. The frontends have the possibility to prioritize workflows/jobs inside the glideinWMS system, allowing for setting up fine grained rules to allow for optimal resource utilization.

The disadvantage of the current system manifests itself in the two levels of prioritization, once at CE level at the sites and again on frontend level in the glideinWMS system. This setup causes inefficiencies in resource utilization. In the case of mixed submission, the complicated multi-proxy setup of the analysis frontend cannot guarantee a fully fair resource sharing between users and also does not transparently allow for special prioritization for individual users or groups as the glideinWMS system allows.

With the transition to glideinWMS, the system can be simplified and the all the advantages of the glideinWMS system can be used as described in the following.

3. Future glideinWMS setup

The future system creates a common pool for analysis and production activity from a single glideinWMS system (see figure 4)

- (1) All submission machines both for analysis and production are monitored by only one global frontend.
- (2) The global frontend requests pilots at sites defined in the job requirements.
- (3) Factories submit pilots to the requested sites using the proxy from the global frontend. The global frontend exclusively uses a proxy generated with the pilot role. Because pilots with only one VOMS role reach the sites, no prioritization on CE level is needed anymore.
- (4) The global frontend matches jobs to pilots. The prioritization is exclusively done inside the global frontend, allowing for
 - Prioritization of analysis vs. production activity to allow for 50% analysis and 50% production activity at the sites.
 - Prioritization of analysis activity on user and analysis project level to guarantee fair share on user level. This also allows to increase the priority consistently for individual users or user groups.

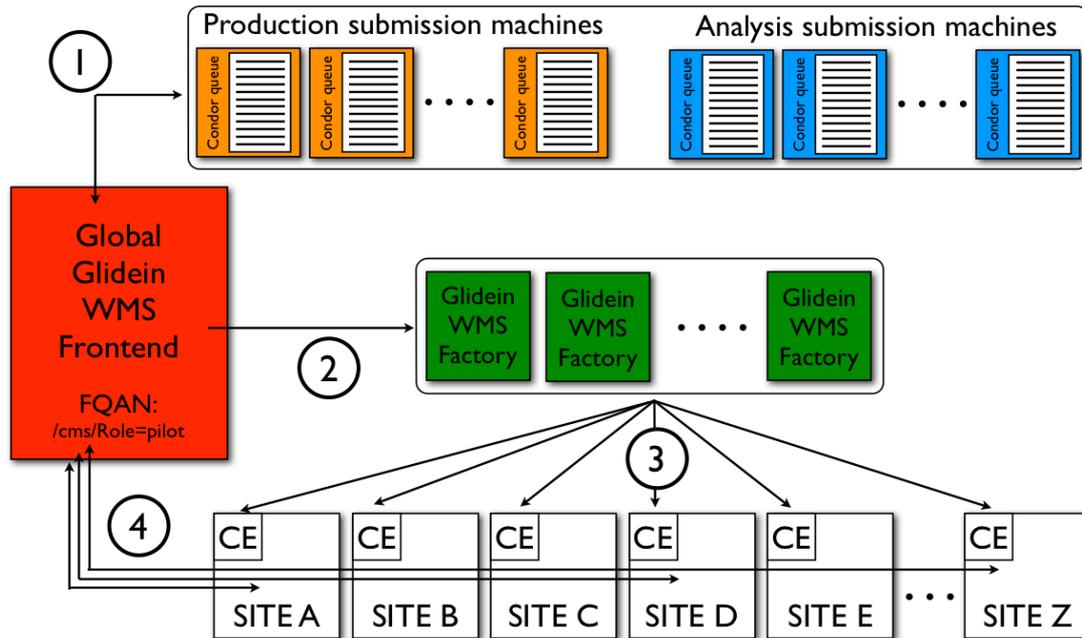


Figure 4. Future glideinWMS system of CMS using a single frontend to prioritize consistently all production and analysis activity at the same time.

- Priorities can be changed dynamically and already submitted jobs can be re-prioritized. Dynamic adaption of priority rules allow a flexible usage of all resources.

All jobs are using `glexec` [7] to switch context to the proxy used to submit the job. The proxy handling is done by HTCondor. This allows production jobs to gain rights to write to the mass storage at sites and other privileges analysis jobs should not have access to. `glexec` also allows sites and central functions to track which payloads are executed within the pilots without access to the pilot system itself.

The challenge that needs to be solved is how to expose the glideinWMS system efficiently to a sophisticated and complicated prioritization setup including resources that are reserved for only special user groups. The setup needs to be dynamic and has to have multiple levels of definition and approval of associations of users to groups used in the glideinWMS system to make priority decisions. A candidate is the aforementioned VOMS system but the decision has not been made yet.

4. Summary & Outlook

CMS transitioned from a mixed setup of direct and pilot-based submission system to a pure pilot-based submission system based on glideinWMS. The described global system will allow CMS to efficiently use all resources without the need for prioritization on the CE level at the sites. All prioritization can be performed in the global frontend of the glideinWMS system allowing for analysis vs. production prioritization as well as fair share prioritization for users and user groups. CMS plans for a fully customizable and dynamic prioritization setup that allows to adapt priorities and user group composition including the definition and usage of resources reserved for only special group of users. How to expose this information to the glideinWMS system is not decided yet.

5. Acknowledgements

We would like to thank the funding agencies supporting the CMS experiment and the LHC computing efforts.

References

- [1] CMS Collaboration 2008 “The CMS experiment at the CERN LHC”, *JINST* **3** (2008), no. 08, S08004
- [2] Evans L and Bryant P 2008 “LHC Machine” *JINST* **3** (2008), no. 08, S08001
- [3] Gutsche O et al. 2013 “CMS computing operations during run 1”, Proceedings of the Conference on Computing in High Energy and Nuclear Physics (CHEP13), Amsterdam, The Netherlands, 14 Oct - 18 Oct 2013
- [4] Cecchi M et al 2010 “The gLite Workload Management System” *J. Phys.: Conf. Ser.* 219 062039 <http://dx.doi.org/10.1088/1742-6596/219/6/062039>
- [5] Thain D, Tannenbaum T and Livny M 2005 “Distributed Computing in Practice: The Condor Experience” *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, pages 323-356, February-April, 2005
- [6] Sfiligoi I, Bradley D C, Holzman B, Mhashilkar P, Padhi S and Wrthwein F 2009 “The pilot way to grid resources using glideinWMS” *Comp. Sci. and Info. Eng.*, 2009 WRI World Cong. on 2 428-432 <http://dx.doi.org/10.1109/CSIE.2009.950>
- [7] Sfiligoi I et al 2012 “glideinWMS experience with glexec” *J. Phys.: Conf. Ser.* 396 032101 <http://dx.doi.org/10.1088/1742-6596/396/3/032101>