EPJ Web of Conferences **245**, 03013 (2020) https://doi.org/10.1051/epjconf/202024503013
CHEP 2019

Lightweight site federation for CMS support

C. Acosta-Silva[1,2], *A.* Delgado Peris[3,*], *J.* Flix[1,3], *J. M.* Guerrero[3], *J. M.* Hernández[3], *A.* Pérez-Calero Yzquierdo[1,3], *F. J.* Rodriguez Calonge[3], and *J.* Gómez-Pulgar[3] **

[1]PIC, Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona), Spain
[2]IFAE, Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona), Spain
[3]CIEMAT, Scientific Computing Unit, 28040 Madrid, Spain

Abstract. There is a general trend in WLCG towards the federation of resources, aiming for increased simplicity, efficiency, flexibility, and availability. Although general VO-agnostic federation of resources between two independent and autonomous resource centres may prove arduous, a considerable amount of flexibility in resource sharing can be achieved in the context of a single WLCG VO, with a relatively simple approach. We have demonstrated this for PIC and CIEMAT, the Spanish Tier-1 and Tier-2 sites for CMS, by making use of the existing CMS xrootd federation infrastructure and profiting from the common CE/batch technology used by the two centres. This work describes how compute slots are shared between the two sites, so that the capacity of one site can be dynamically increased with idle execution slots from the remote site, and how data can be efficiently accessed irrespective of its location. Our contribution includes measurements for diverse CMS workflows comparing performances between local and remote execution, and can also be regarded as a benchmark to explore future potential scenarios, where storage resources would be concentrated in a reduced number of sites.

1 Introduction

Federating computing resources is integrating them so that they can be transparently used through a single access point. Virtual Organizations (VOs) of the Worldwide LHC Computing Grid (*WLCG*) [1] are exploring the possibility of consolidating its resource centers, *sites*, into a series of federations [2]. A lower number of access points should allow for simpler management and reduced operational effort, while gaining flexibility in the internal organization of resources within a federation. Moreover, geographical redundancy enables high availability for selected services, while unnecessary duplicated ones can be removed.

While a general solution for resource federation between independent and autonomous sites may prove difficult, our work explores a more restrictive scenario, in which the resources of two Spanish WLCG sites, CIEMAT-LCG2 and PIC, which use the same relevant technologies for job management (HTCondor [3]) and data access (xrootd [4]), are federated on the framework of a single WLCG experiment, CMS [5]. These sites hold 10 Gbps links to

*e-mail: antonio.delgadoperis@ciemat.es
**The authors acknowledge support provided by Spanish funding agency SEIDI, grant FPA2016-80994-C2-1-R, and by the Spanish Ministry of Economy and Competitiveness through the Unidad de Excelencia María de Maeztu CIEMAT - Física de Partículas, grant MDM-2015-0509.

the WLCG-dedicated network, *LHCONE*, are relatively close to each other, with an average round trip time (*RTT*) of 10 milliseconds, and benefit from a long-standing collaboration.

The purpose of our work is to demonstrate that compute and storage resources may be logically shared between the sites (one site's capacity dynamically expanded with resources at the other one), to ensure that data can be transparently accessed at either site, from both local and federated compute nodes, and to study the impact of such federation on the execution of unmodified CMS computing tasks (*jobs*) in a production environment.

2 Implemented Solution

2.1 Compute resources

Both PIC and CIEMAT use HTCondor as their technology of choice for both the Computing Element (*CE*) and the batch system. HTCondor supports several mechanisms to federate resources between independent pools, e.g., merging site pools, *flocking* jobs, or using *glide-ins* [6]. For our work, we have chosen a method that we call *job re-routing*, and that is illustrated by Figure 1. When CMS jobs reach one of the site CEs, a fraction of them are resubmitted as Grid *Condor-C* jobs to the remote site's CE. To accomplish this, the *Job Router* daemon, which normally routes incoming CE jobs into the local batch pool, is configured to send some jobs to the other site instead. At the remote site, the jobs are run routinely, although they are identified as *re-routed*, so they might be especially treated if desired. In practice, the site is expanding its available resources with those of the federated peer.

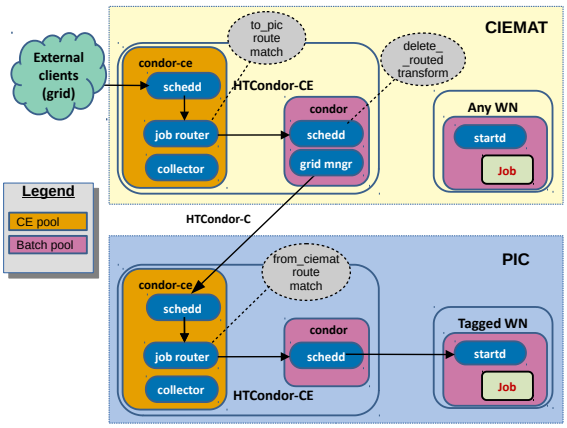


Figure 1: Overview of HTCondor job re-routing mechanism (*CIEMAT to PIC* case).

This mechanism was chosen because it is the simplest configuration offering the desired functionality. It allows us to control the share of federated resources, by limiting the number of jobs that can be re-routed simultaneously; it does not require modifying the network connectivity of the sites, or sharing HTCondor configuration, password or users. In the future, though, other solutions may be considered. For example, a single HTCondor pool encompassing all resources might be more appropriate for a consolidated long-standing federation.

In the current configuration, any CMS job is subject to being re-routed, as long as the configured quota has not been reached, since, from a CE's perspective, all CMS jobs are equal. This is due to CMS using a late-binding scheduling model, in which *pilot jobs* are sent to sites, while *payload jobs* are transparently scheduled to already-running pilots [7].

However, since CMS payloads match pilots considering their target site (where they were *meant to run*), a re-routed pilot will match jobs meant for that site, rather than the one where it is really running. Therefore, re-routing pilots is equivalent to re-routing payload jobs.

Although the described technique works well and enables us to federate resources, it suffers from a few limitations. First, downtimes of any of the involved sites prevent pilot re-routing in any direction. Fortunately, the unused shared resources may be occupied by normal pilots, so no resources are lost from the VO's perspective; and, once the inaccessible site recovers, the re-routing is automatically enabled again. Second, a bug was found on the *GridManager daemon* of HTCondor: in some cases, a single problematic pilot job could prevent the processing of further (sound) jobs. This was reported and fixed by the developers on *HTCondor 8.8.7*. As a final technical note, the `routed` attribute must be removed from re-routed jobs, or they will not be matched by the remote Job Router, and consequently run.

2.2 Access to data

Regarding data access, the existing CMS xrootd federation [8] has been used. This federation enables transparent location discovery and access of CMS data by making use of the xrootd protocol and a chain of *xrootd redirectors*. When a client requests a file, an xrootd redirector will point it to a known server holding the demanded data, or, if not found, instruct it to go upstream in the redirectors hierarchy, to look for the file elsewhere.

We have added a *regional* redirection layer, to favour accessing data stored in one of the federated sites over data located elsewhere. The existing site redirectors subscribe to this new server, which, in turn, subscribes to the European redirector. In addition, the CMS logical to physical filename translation service, the Trivial File Catalog (*TFC*), has been set up so that jobs try to read files at their execution site but fall back to the regional level if the data is not available locally. In the latter case, files are looked for at the federated SE, and, only if not found there, the rest of the CMS infrastructure is searched. This setup is shown by Figure 2.

Jobs targeted at a site are expected to find most of their required input files at that site, and, thus, read data locally. The described redirection setup was designed to favour that re-routed jobs find their input files on their original target site. The files are directly streamed from the remote server, but the impact of latency is kept moderate thanks to several strategies implemented by CMS jobs (vector reads, caching, and read-ahead), and the relative closeness of the two federated sites. Moreover, this configuration also enables sites to transparently *lend* storage space to the federated peer, though we have not explored this possibility yet.

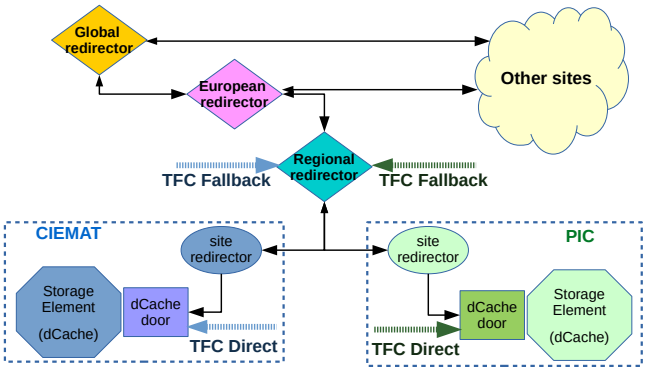


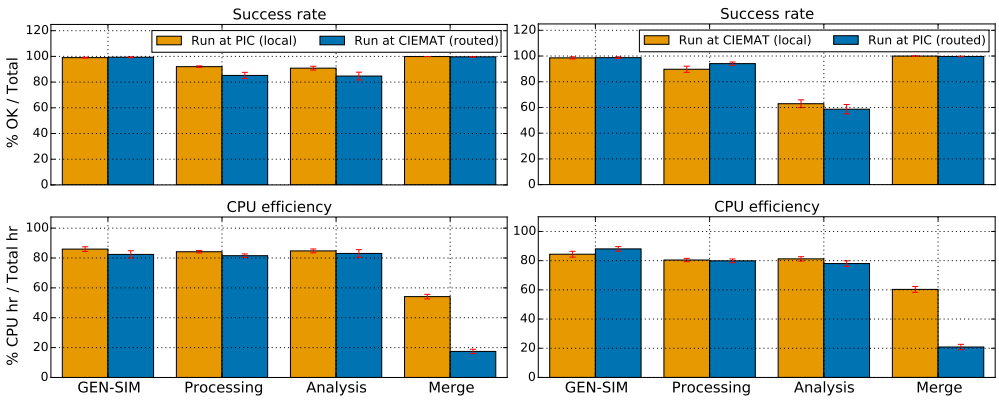
Figure 2: Scheme of xrootd federation hierarchy with added regional redirector.

3 Evaluation of results

The described setup has been deployed on the production CMS infrastructure of CIEMAT and PIC. During 6 months, CMS pilot jobs have been routinely re-routed from one site to the other. Ten 8-core job slots at each site have been devoted to the federated share, so up to eighty 1-core payload jobs have been simultaneously run on re-routed pilots.

Since one of our objectives was to assess the impact of an increase in the amount of remote data reads (due to pilots re-routing), we have compared the success rate and CPU efficiency of CMS jobs run on *normal* (executed on their original target site) and *re-routed* (executed on the federated site) pilots, for both resource centers, and for different job types. For this analysis, the 6 months span has been divided into a series of 3-day periods, and the success/efficiency ratio has been calculated for each of these periods. The bar plots in Figures 3, 4 and 5 show average values and uncertainties for those ratios, for different conditions.

Figure 3 shows the overall results for CMS jobs run at the sites. No relevant degradation can be noted between the results of payloads run on re-routed and normal pilots, except for *merge* jobs, which combine the output files of individual jobs, and are I/O-bound. This suggests that the overall effect of re-routing is not significant in most cases. One can however appreciate a lower success rate for CIEMAT's analysis jobs than for PIC ones. The reasons for this are unclear, but it cannot be caused by re-routing, since normal pilots are also affected.



(a) Jobs meant for PIC (b) Jobs meant for CIEMAT

Figure 3: Success rate (top) and CPU efficiency (bottom) by job type and target site, for normal (left/orange bars) and re-routed jobs (right/blue bars).

Figure 4 compares the previous CPU efficiency values with those of PIC jobs run at the Amazon Web Services Cloud (*AWS*). The latter are the results of a hybrid cloud test conducted at PIC, where AWS resources were used to transparently run some of PIC's CMS jobs [9]. As we can see, analysis jobs run on re-routed pilots at CIEMAT obtain significantly better results than the same jobs run at Amazon's Frankfurt site. We can identify two main reasons: the lower RTT from PIC to CIEMAT (10 ms) than to Frankfurt (40 ms); and the fact that jobs run at CIEMAT may find some input files locally if a replica of those files is stored there, while jobs at AWS must read every file remotely, since AWS does not hold CMS storage.

We can take a closer look at analysis jobs, the ones showing greater differences between CIEMAT and AWS results. We have classified jobs according to the relative location of the data they have really read: from the local site, from the peer site (regional), or from any other CMS site. Figure 5 shows these results. We find that CPU efficiency degrades as jobs read

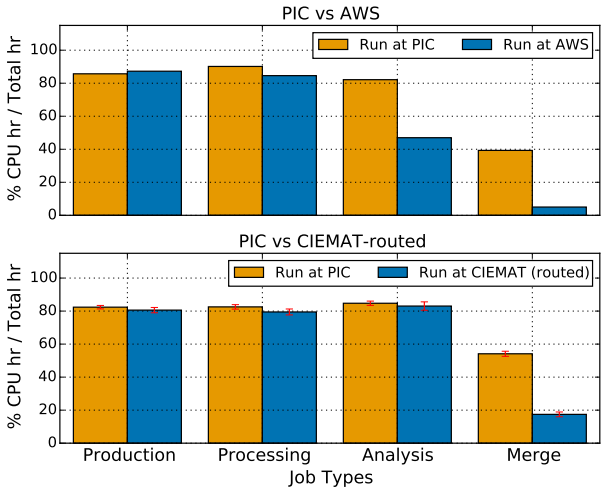


Figure 4: CPU efficiency by job type, for normal PIC jobs (left/orange bars) and PIC jobs run elsewhere (right/blue bars): AWS (top plot) or CIEMAT (bottom plot).

from farther sources. In particular, the efficiency for jobs reading from the federated peer shows low but non-negligible decline. Regarding the success rate, the previously indicated low numbers for CIEMAT analysis jobs also affect the results of jobs re-routed to PIC, which causes an abnormally low value for jobs run at PIC and reading regionally.

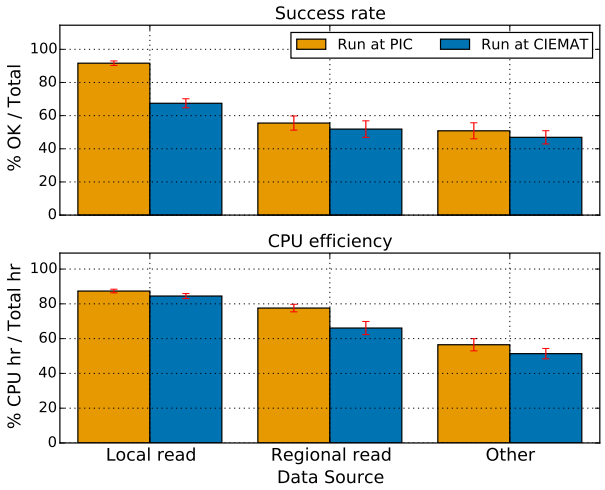


Figure 5: Success rate (top) and CPU efficiency (bottom) per relative location of input data, for jobs run at PIC (left/orange bars) and run at CIEMAT (right/blue bars).

We have also studied how the success rate and CPU efficiency evolve with time. We do not include figures showing those for reasons of space. We can mention, however, that rates are generally flat with time, but there are multiple, generally brief, periods where external conditions affect the results, sometimes in a dramatic way. Problems in a site's SE, or disrup-

tions of network links have, at times, caused a considerable increase in the job failure rate. Moreover, these problems may affect re-routed pilots only, or, perhaps, just normal pilots on one site, but re-routed in the other, or any other combination. As further examples, a single user's code caused thousands of analysis job failures at CIEMAT during a period of a couple of days. On another occasion, errors in the workflow definition by CMS central production caused problems on most CMS jobs.

A particularly interesting factor was the impact of general CMS xrootd activity. Both local and remote CMS jobs use the xrootd protocol to access CIEMAT and PIC data. It has been observed that *storms* of external xrootd requests occasionally hit the SEs. This causes the queuing of file requests, affecting overall performance. CIEMAT and PIC mitigate this by keeping separate transfer queues (*movers*) for local and external access. Thus, jobs reading data locally are not influenced by heavy CMS xrootd activity, but remote jobs may be. Should this happen, the effect would be more noticeable on re-routed pilots.

Figure 6 compares the number of failed jobs executed at CIEMAT and reading from PIC with the external xrootd transfer queues at PIC. To ease visual inspection, outliers—three standard deviations from the mean—in the number of failed jobs have been removed. The plot gives a feeling of the correlation between the two magnitudes. It suggests that external activity may be indeed affecting jobs on re-routed pilots; namely, long xrootd queues at PIC may be causing jobs at CIEMAT to fail, since they may time out reading input data from PIC.

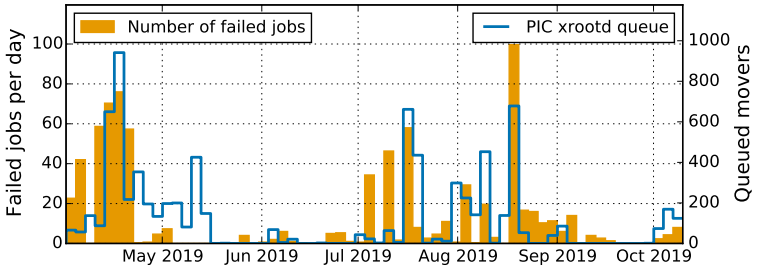


Figure 6: Number of failed jobs (orange bars) vs queued xrootd movers at PIC (blue line).

4 Future work and conclusions

The deployed lightweight federation has proven to operate successfully. Most jobs running on re-routed pilots just work, with very limited efficiency degradation, while a fraction of the analysis jobs on those pilots (those reading mostly from the federation peer) suffer slightly higher degradation, but at still tolerable levels.

Merge and other auxiliary jobs, which are I/O-bound, experience a higher performance penalty, and it would be best to avoid them being matched by re-routed pilots. In fact, since CMS has recently enabled the possibility to use site-customizable pilots [10], we are planning to test this feature to constrain the types of jobs that the re-routed pilots are allowed to match.

For larger scales, a dedicated regional xrootd transfers queue on the sites SE might be advisable, in order to avoid the impact of external CMS xrootd activity.

As further future work, we intend to test the effect of data caching on the execution nodes, and to improve monitoring to allow for finer-grain studies. Finally, we are willing to collaborate with other CMS regions interested in deploying a similar federation solution.

References

[1] WLCG, *Worldwide LHC Computing Grid*, http://wlcg-public.web.cern.ch/

[2] I. Bird, S. Campana, M. Girone, X. Espinal, G. McCance, J. Schovancová, *Architecture and prototype of a WLCG data lake for HL-LHC*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 04024

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

[4] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, WSEAS Transactions on Computers **1**, 348 (2005)

[5] T.C. Collaboration, Jinst **803**, S08004 (2008)

[6] G. Thain (2019), *Federating HTCondor pools*, at European HTCondor Workshop 2019, https://indico.cern.ch/event/817927/contributions/3570512/

[7] J. Balcas, S. Belforte, B. Bockelman, D. Colling, O. Gutsche, D. Hufnagel, F. Khan, K. Larson, J. Letts, M. Mascheroni et al., *Using the glideinWMS system as a common resource provisioning layer in CMS*, in *Journal of Physics: Conference Series* (IOP Publishing, 2015), Vol. 664, p. 062031

[8] K. Bloom, CMS Collaboration et al., *CMS use of a data federation*, in *Journal of Physics: Conference Series* (IOP Publishing, 2014), Vol. 513, p. 042005

[9] J. Casals, C. Acosta, F. López, V. Acín (2019), *Hybrid batch system deployment with AWS spot instances*, at IBERGRID 2019, https://indico.lip.pt/event/575/contributions/1854/

[10] A. Pérez-Calero Yzquierdo et al., *Evolution of the CMS Global Submission Infrastructure for the HL-LHC Era*, to be published in these proceedings