Elastic Extension of a CMS Computing Centre
Resources on External Clouds

G. Codispoti\textsuperscript{1,2}, R. Di Maria\textsuperscript{1,2,a}, C. Aiftimiei\textsuperscript{3,4}, D. Bonacorsi\textsuperscript{1,2}, P. Calligola\textsuperscript{1}, V. Ciaschini\textsuperscript{3}, A. Costantini\textsuperscript{3}, S. Dal Pra\textsuperscript{3}, D. DeGirolamo\textsuperscript{3}, C. Grandi\textsuperscript{3}, D. Michelotto\textsuperscript{3}, M. Panella\textsuperscript{3}, G. Peco\textsuperscript{1}, V. Sapunenko\textsuperscript{3}, M. Sgaravatto\textsuperscript{5}, S. Taneja\textsuperscript{1}, and G. Zizzi\textsuperscript{3}

\textsuperscript{1} INFN Bologna, Bologna, I-40127, Italy
\textsuperscript{2} Universit\`{a} di Bologna, Bologna, I-40127, Italy
\textsuperscript{3} CNAF, Bologna, I-40127, Italy
\textsuperscript{3} IFIN-HH, Magurele, Romania
\textsuperscript{5} INFN Padova, I-35131, Italy
\textsuperscript{a} now at Imperial College, London, UK

E-mail: Giuseppe.Codispoti@bo.infn.it

Abstract. After the successful LHC data taking in Run-I and in view of the future runs, the LHC experiments are facing new challenges in the design and operation of the computing facilities. The computing infrastructure for Run-II is dimensioned to cope at most with the average amount of data recorded. The usage peaks, as already observed in Run-I, may however originate large backlogs, thus delaying the completion of the data reconstruction and ultimately the data availability for physics analysis. In order to cope with the production peaks, CMS - along the lines followed by other LHC experiments - is exploring the opportunity to access Cloud resources provided by external partners or commercial providers. Specific use cases have already been explored and successfully exploited during Long Shutdown 1 (LS1) and the first part of Run 2.

In this work we present the proof of concept of the elastic extension of a CMS site, specifically the Bologna Tier-3, on an external OpenStack infrastructure. We focus on the “Cloud Bursting” of a CMS Grid site using a newly designed LSF configuration that allows the dynamic registration of new worker nodes to LSF. In this approach, the dynamically added worker nodes instantiated on the OpenStack infrastructure are transparently accessed by the LHC Grid tools and at the same time they serve as an extension of the farm for the local usage. The amount of resources allocated thus can be elastically modeled to cope up with the needs of CMS experiment and local users. Moreover, a direct access/integration of OpenStack resources to the CMS workload management system is explored.

In this paper we present this approach, we report on the performances of the on-demand allocated resources, and we discuss the lessons learned and the next steps.

1. Introduction

Scientific experiments nowadays require ever-increasing computing resources, and several of them work in a ”burst” modality in which there are periods of peak usage where resource usage greatly increases with respect to periods of ”normal” usage. Traditional scientific (non-commercial) computing centres (CC) may find it difficult to size themselves so that they may absorb the peak usage of the experiments without generating excessively long queues and therefore compromising...
usage of the computing centre for the other users. In order to cope with the production peaks, CMS [1] - along the lines followed by other LHC experiments as well - is exploring the opportunity to access Cloud resources provided by external partners or commercial providers. Specific use cases have already been explored and successfully exploited during Long Shutdown 1 (LS1) and the first part of Run 2. In this work we present the proof of concept of the elastic extension of a CMS site, specifically the Bologna Tier-3, on an external OpenStack [2] infrastructure. This extension is performed through three logical steps: virtualization of the local resources and dynamic allocation; dynamic extension of a CMS farm in external Openstack resources; direct access to external Openstack resources.

2. Elastic extension of the Bologna Tier-3
In order to cope with the request peaks, INFN - along the lines of other WLCG [3] communities - is exploring the opportunity to access Cloud resources by external partners or commercial providers. In CMS we realised a prototype to extend an existing site’s batch system to external resources. Specifically we focus on the “Cloud Bursting” of a CMS Grid site using a newly designed LSF configuration that allows the dynamic registration of new worker nodes (WNs) to LSF. In this approach, the dynamically added WNs instantiated on Cloud infrastructure are transparently accessed by the LHC Grid tools and at the same time they serve as an extension of the farm for the local usage. The amount of resources allocated thus can be elastically modelled to cope up with the needs of CMS experiment and local users.

The first step performed is the virtualization of the CMS resources of a Tier-3 [4] centre serving both as a CMS Grid site and local Farm. As a general rule, we are keeping the images as lightweight as possible by using remote services instead of local installation. For this reason we are relying on LDAP [5] and Grid Pool Account services for the users mapping; experiment software access through CVMFS [6]; central authentication systems (ARGUS); and LSF access through remote mounting and central configuration. In order to enable the Grid site usage, we provide a custom installation of the GLEXEC [7] package. In order to enable the local farm use case and to be able to perform effective tests, we access the local storage also via nfs export from a dedicated machine. Following these guidelines, we created images for the user interface (UI) and the Worker Nodes (WNs), and we tested the usability of the system through a workflow belonging to the Top Quark mass measurement analysis.

![Figure 1. The VPN used for the Tier-3 Bursting: a configuration server provides the VM at boot with all needed info (configs, addresses,...); a VPN server is then contacted by the VM in order to be registered in the private network; a VPN tunnel connects VMs and a VPN server; a GRE tunnel enable the connection among the VMs and necessary services (LSF server, CE, ...). As soon as the configuration is completed, LSF Master sees the new node and start sending jobs.](image)

We used then a dedicated LSF configuration which allows to resize a standard site by bursting out to third parties (Figure 1). This is achieved by instantiating virtual machines (VM) on
a remote site and making them part of the local farm. Each VM at startup will contact a Configuration Service in the bursting site: after passing an authorization phase, it receives a set of configuration files and commands allowing the VM to establish a connection with a VPN server inside the CC. The VPN server has established a GRE (generic routing encapsulation) tunnel with the machines that must be visible to the VM to allow it to work and adjust the proper routes. The VPN connection will make the VM and the OpenVPN server visible to each other, but it will have no further effect on the network connectivity of the VM. This will ensure that all and only the necessary traffic will reach the CC, therefore limiting the problems that may be caused by an increased network latency due to the geographical distance from the CC. This mechanism allows a CC to accept workloads greatly superior to those it has been created to accept.

The basic functionalities have been tested by executing the aforementioned analysis workflow using one VM as UI and one as WN, and a dedicated LSF master and queue. The WN was initially instantiated in the local Tier-3 cluster. Then the test was repeated with the VMs instantiated in the CNAF Cloud infrastructure, an Infrastructure as a Service (IaaS) based on OpenStack Havana.

The first problem hit is related to the access to the local storage. The reduced performances outside of the CC perimeter of the GPFS (IBM general parallel file system) storage accessed through the VPN server affect the whole storage system and represents a real bottleneck that is a show stopper for the extension of the local farm.

The next step was the actual Cloud Bursting of the Grid site, as the natural evolution of this proof of concept and the solution proposed for the production peaks absorption. The images need a very small tuning (adding cloud-aware packages, removing local storage access) to properly run in Openstack. The Bologna site LSF Master has been reconfigured in order to allow the dynamic extension of the nodes. The VM instantiated were included first in a test queue of the master LSF and tested with direct Grid submission. Right after we added the newly instantiated nodes to the production LSF queue used for Grid submission.

We submitted a Top Quark skimming workflow, accessing remote data through xrootd and copying the results with a standard Grid command to the destination storage. The test setup included five virtual nodes statically allocated with QuadCore CPU with 4 cores and 8 GB RAM.

Since the virtual nodes were inserted in a production queue, the jobs were distributed between standard WN and virtual nodes, with only about 5% to the latter. Over more than 3000 jobs submitted, 172 jobs ran in the virtual nodes. Since the jobs were submitted in bunches, they hit all the variability of a production systems, encountering the concurrency of other users jobs, variation of the load.

We measured the job efficiency as a ratio between the cpu time over the cpu wall clock time for both the jobs running on the Openstack nodes and the jobs running on the normal WNs. The distribution of the efficiencies does not show a peak, as it can be observed in Figure 2, given all the fluctuation affecting a productions system. Nevertheless, a relevant part of the jobs reached more than 95% efficiency while almost their totality has an efficiency greater than 80%. No job failures on the virtual nodes were observed.

3. The Bologna Tier-3 as a pure Cloud site
Finally we tested the possibility to allocate an on-demand, brand new, CMS site as a service in Openstack, decoupled from the existing Grid site. The goal was to access this new site, created in the Openstack infrastructure using the standard CMS Workload Management tools (full description can be found here [10]) as described in Figure 3.

The main difference with the already described prototype is the complete absence of a dedicated batch system and of the Grid infrastructure logical parts such as the Computing
Elements (CE).

Instead of using the standard Grid CE to access the resources, we used directly OpenStack through the EC2 (Amazon Elastic Compute Cloud [11]) interface of the OpenStack-Havana infrastructure. EC2 allows scalable deployment of applications by providing a Web service through which GlideIn-WMS can boot an AMI-type (Amazon Machine Image) image to create a virtual machine. The GlideIn-WMS can thus elastically instantiate virtual machines running the HTCondor [12] startd processes that are able to fetch users’ jobs. Moreover, the GlideIn-WMS can create, launch, and terminate these VMs as needed. The virtual nodes image can be thus even shrunk in terms of required packages.

A first attempt was performed using the old CMS analysis submission tool, Crab2 [13], a private test instance of the GlideInWMS, and relying on an Openstack Havana infrastructure. Once the prototype was tested, we required access to the CERN CMS Integration Test Bed (ITB) GlideInWMS and for the first time used Crab3 [14], the newly redesigned tool for CMS jobs submission. Given the reduced availability of the ITB GlideInWMS, shared with all the other CMS-wide projects, we limited the number of jobs submitted to 800, executing the very same Top Quark skimming workflow used for the bursting case. We did not observe any failure. Being the jobs running in a controlled and protected environment, they all run with an average efficiency of 98% (Figure 4). As a further exercise, we proved that the infrastructure can also accept more memory demanding jobs, using reconstruction workflow in the context of the SuperLHC upgrade simulation, just increasing from 8 GB to 12 GB the ram allocated for the virtual machines.
4. Conclusions
The possibility to dynamically extend the computing resources is already crucial for the high energy physics experiments and is becoming more and more appealing for many eScience fields. While big experiments may require to acquire external resources to absorb production peaks, other fields may totally rely on opportunistic resources, specifically provided through Cloud systems, to cope with occasional needs instead of maintaining proprietary farms. In this paper we discussed several possible approaches as we implemented and tested in the Bologna Computing Centre. In conclusion we proved that a CMS Computing Centre using LSF as a local batch system can be efficiently and dynamically expanded on external Cloud-aware resources without a serious loss of performance. The resources dynamically allocated can thus be used for the normal operation of the site. The Bologna Tier-3 has been a realistic use case for the CNAF OpenStack infrastructure. The LSF extension served as development environment for the tools and setup used for Tier-1 extension over commercial resources. We also proved that Openstack can host an on-demand, brand new, decoupled CMS site “as a service”. After this successful experience, the Bologna Tier-3 is evaluating to become a pure Cloud site in order to reduce maintenance costs and profit from the CNAF Tier-1 infrastructure. We plan to extend in the next future the CMS specific case to a multi-VO environment.

References
[1] The CMS Collaboration, The CMS experiment at the CERN LHC, JINST 3 (2008) S08004.
[2] OpenStack, http://www.openstack.org/
[3] J. D. Shiers, The Worldwide LHC Computing Grid, Comp. Phys. Comm. 177 (2007) 219-223.
[4] D. Bonacorsi, WLCG Service Challenges and Tiered architecture in the LHC era, IFAE, Pavia, April 2006.
[5] C. Donley, LDAP Programming, Management, and Integration, Manning Pubs (2002), ISBN 1-930110-40-5.
[6] P. Buncic, C. Aguado Sanchez, J. Blomer, L. Franco, A. Harutyunian, P. Mato, Y. Yao, CernVM - a virtual software appliance for LHC applications, J. Phys.: Conf. Ser. 219 (2010) 042003.
[7] D. Groep, O. Koeroo, G. Venekamp, gLEexec: gluing grid computing to the Unix world, J. Phys.: Conf. Ser. 119 (2008) 062002.
[8] OpenVPN software, http://openvpn.net/
[9] J. Andreeva et al, Monitoring of large-scale federated data storage: XRootD and beyond, J. Phys.: Conf. Ser. 513 (2014) 032004.
[10] M. Cinquilli, D. Evans, S. Foulkes, D. Hufnagel, M. Mascheroni, M. Norman, Z. Maxa, A. Melo, S. Metson, H. Riahi, S. Ryu, D. Spiga, E. Vaandering, S. Wakefield, R. Wilkinson, The CMS workload management system, J. Phys.: Conf. Ser. 396 (2012) 032113.
[11] Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2/
[12] HTCondor, http://research.cs.wisc.edu/htcondor/
[13] Giuseppe Codispoti et al., CRAB: A CMS Application for Distributed Analysis, IEEE Trans. Nucl. Sci. 56 (2009) 2850-2858. doi:10.1109/TNS.2009.2028076.
[14] M. Mascheroni et al, CMS distributed data analysis with CRAB3, J. Phys.: Conf. Ser. 664 (2015) 062038.