

OPEN ACCESS

Cloud flexibility using DIRAC interware

To cite this article: Víctor Fernández Albor *et al* 2014 *J. Phys.: Conf. Ser.* **513** 032031

View the [article online](#) for updates and enhancements.

Related content

- [The Integration of CloudStack and OCCI/OpenNebula with DIRAC](#)
Víctor Méndez Muñoz, Víctor Fernández Albor, Ricardo Graciani Diaz *et al.*
- [Integration of Cloud resources in the LHCb Distributed Computing](#)
Mario Úbeda García, Víctor Méndez Muñoz, Federico Stagni *et al.*
- [DIRAC Distributed Computing Services](#)
A Tsaregorodtsev and the Dirac Project

Recent citations

- [A modular \(almost\) automatic set-up for elastic multi-tenants cloud \(micro\)infrastructures](#)
A Amoroso *et al*
- [Belle II public and private cloud management in VMDIRAC system.](#)
Rafa Grzymkowski *et al*



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Cloud flexibility using DIRAC interware

Víctor Fernandez Albor¹, Marcos Seco Miguelez¹, Tomas Fernandez Pena²,
Victor Mendez Muñoz³, Juan Jose Saborido Silva¹ and Ricardo Graciani Diaz⁴

¹ Particle Physics Department. Universidad de Santiago de Compostela (USC).
Santiago de Compostela, Spain.

² Centro Singular de Investigación en Tecnoloxías da Información, Universidad de
Santiago de Compostela (USC). Santiago de Compostela, Spain.

³ LHC Tier-1 Computing Production. Port d'Informació Científica (PIC). Campus
Universitat Autònoma de Barcelona (UAB). Bellaterra, Spain.

⁴ Department of Structure and Constituents of Matter. Universitat de Barcelona (UB).
Barcelona, Spain.

E-mail: victormanuel.fernandez@usc.es

Abstract. Communities of different locations are running their computing jobs on dedicated infrastructures without the need to worry about software, hardware or even the site where their programs are going to be executed. Nevertheless, this usually implies that they are restricted to use certain types or versions of an Operating System because either their software needs a definite version of a system library or a specific platform is required by the collaboration to which they belong. On this scenario, if a data center wants to service software to incompatible communities, it has to split its physical resources among those communities. This splitting will inevitably lead to an underuse of resources because the data centers are bound to have periods where one or more of its subclusters are idle. It is, in this situation, where Cloud Computing provides the flexibility and reduction in computational cost that data centers are searching for. This paper describes a set of realistic tests that we ran on one of such implementations. The test comprise software from three different HEP communities (Auger, LHCb and QCD phenomenologists) and the Parsec Benchmark Suite running on one or more of three Linux flavors (SL5, Ubuntu 10.04 and Fedora 13). The implemented infrastructure has, at the cloud level, CloudStack that manages the virtual machines (VM) and the hosts on which they run, and, at the user level, the DIRAC framework along with a VM extension that will submit, monitorize and keep track of the user jobs and also requests CloudStack to start or stop the necessary VM's. In this infrastructure, the community software is distributed via the CernVM-FS, which has been proven to be a reliable and scalable software distribution system. With the resulting infrastructure, users are allowed to send their jobs transparently to the Data Center. The main purpose of this system is the creation of flexible cluster, multiplatform with an scalable method for software distribution for several VOs. Users from different communities do not need to care about the installation of the standard software that is available at the nodes, nor the operating system of the host machine, which is transparent to the user.



1. Introduction

The DIRAC project was initiated in 2003, as a community management software to support scientific computing in the LHC context [1]. DIRAC was created as a Grid middleware to support the user community of LHCb experiment and, at the beginning, with the specific use case of Monte Carlo (MC) simulation. Although developed for the LHCb experiment, it was designed to be a generic system with the LHCb specific features well isolated as plug-in modules. The DIRAC project has all the necessary components to build workload and data management systems of varying complexity. Nowadays, DIRAC is used by LHCb as the management middleware of the community, including not only MC, but all the other computing activities such as the reconstruction of the experiment data, the selection of events by different filters, the reprocessing of that data, the user analysis or the data transfers. Moreover, other scientific communities are using DIRAC such as ILC, Belle II and others. Some of them have extended the DIRAC core functionality to fulfill their requirements [2,3]. These experiences have been useful to validate two important characteristics of the software: the robustness of the framework, and an extensible design, resulting in a solid software stack able to deal with a large variety of use cases in a flexible manner. For these reasons we can define DIRAC as an *interware*, that allows interoperability among different computing resources via different drivers to each of the middleware solutions. In this paper we explain the work aiming to extend the DIRAC interoperability with the flexibility using different software platforms for several user groups. Starting from the previous work with the Belle experiment using Amazon Elastic Compute Cloud (EC2) [2], and the integrations of DIRAC with two well known Cloud management solutions, the specific CloudStack and Open Cloud Computing Interface (OCCI) [4], which is an open standard adopted by many Cloud manager providers like OpenNebula or OpenStack.

This paper is organized as follows: Section 2 describes multi-platform structure using the Virtual Engine created with the multi-endpoint design. Section 3 shows the CloudStack testbed at the Universidad de Santiago de Compostela aiming to demonstrate the flexibility of such type of cloud resources to be used for several Virtual Organization (VO). The software distribution is taking advantage of the Cern VM File System (CernVM-FS) [5] with different deployments depending on the particular purposes. Section 4 presents the results of the testbed. Finally, a brief summary is presented in Section 5 about the Cloud flexibility.

2. DIRAC multi-platform internal structure

The multi-platform engine is based in the multi-endpoint design, but with different DIRAC Configuration setups.

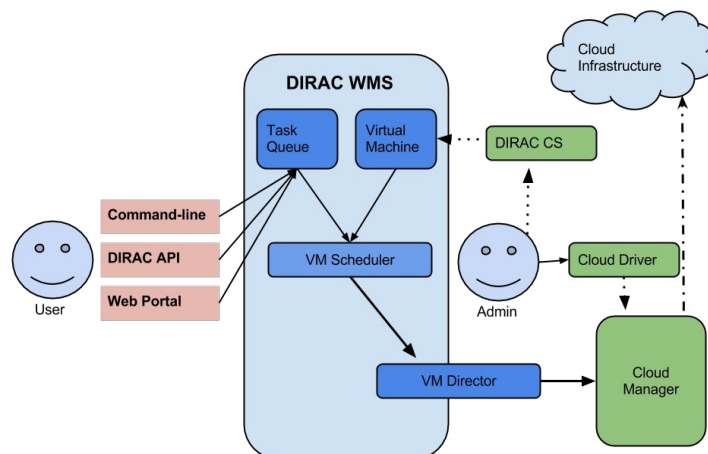


Figure 1: VM Scheduler interaction with other DIRAC components.

Figure 1 shows the general interaction of the VM Scheduler with other components to submit VMs to a CloudStack. The VM Scheduler is the main component in the VM Manager that is the principal service of VMDIRAC. VM Scheduler is the Agent in charge of VM submission to a CloudStack manager. After that, CloudStack deploy the required VMs, which report to the VM Monitor their status.

3. Testbed structure

This section describes the testbed for the multi-platform use case, where the objectives were the proof-of-concept of the multi-endpoint architecture with the CloudStack Cloud manager for Multi-VO.

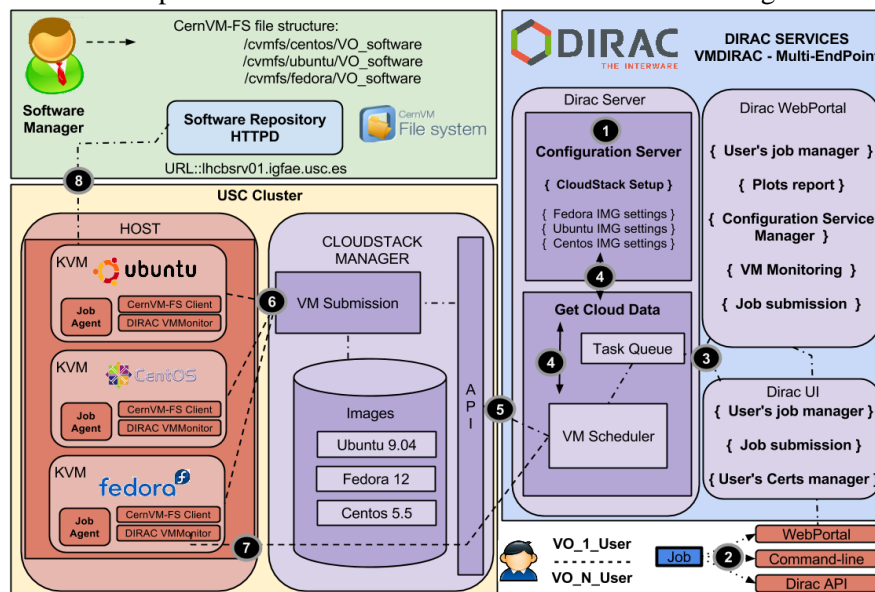


Figure 2: Multi-endpoint architecture: USC CloudStack infrastructure Multi-VO and multi-platform tests.

In the proof-of-concept process, simulations of several groups were executed in different platforms such as Fedora 12, CentOS 5.5 or Ubuntu 9.04, in order to test the multi-platform feature. The HEP-Software of three groups were selected to make the tests: (i) GAUSS which is a specific software developed for the LHCb experiment; (ii) AIRES developed for Auger, an experiment to study cosmic rays; and (iii) a program developed by one of the groups at the *Instituto Galego de Física de Altas Enerxías* (IGFAE) [6,7], who are part of the ALICE experiment and makes use of FastJet. FastJet is software that contains jet finding algorithm used by major experiments like ATLAS, CMS, D0, CDF and others. And as non HEP-Software was selected Parsec Benchmark Suite that was running in Ubuntu 9.04.

The test infrastructure used KVM hypervisors on 4 nodes, and each physical node was equipped with 2 quad-core processors IntelXeon X5355 @ 2.66GHz, and 16 GB of total memory. The jobs were classified in short jobs, with a time execution of approximately 20 minutes, and long jobs, of around 8 hours of execution time. These jobs are similar to MonteCarlo jobs and they were using high CPU power and memory consumption. Figure 2 shows the job submission steps. In this tests, there are several users belonging to different VOs. In the first step the users with DIRAC admin role make an user list depending on the VO. Furthermore, in this first step, the DIRAC admin is in charge of uploading the Cloud settings to the DIRAC Configuration Server (DIRAC CS), taking care of the different preconfigured images of the Cloud manager. The second and third steps are the job submission, with the 3 ways to submit the job, and the job reception by DIRAC. The new job is inserted in the TaskQueue.

The VM Scheduler queries to TaskQueue if there are new jobs in the queue, and in this case, specific Cloud information is obtained from the DIRAC CS according to the user credentials, this is the step 4. At this stage of the process, each user has been assigned to a specific VO, and each VO has been assigned to an unique EndPoint. The VM Scheduler component sends the specific EndPoint command to the CloudStack Server API [8], and the manager submits the specific image, which in this case correspond to Ubuntu, Centos and Fedora, these are the steps 5 and 6. These images are ad-hoc images, with the corresponding OS and the DIRAC software preinstalled. The VMs are not using contextualization and when they are starting up, the VM DIRAC agents are launched. Then, the VM Manager running in the DIRAC gets a started message notification from the Virtual Machine indicating that it is been started, which confirms that the VM is in "Up status" in step 7. Finally, when the VM start up process finished the CernVM-FS client connects to the USC [9] . CernVM-FS repository, which is hosted at the USC Tier-2 and provides the software for step 8. At this point the VM is ready to match an appropriated user job and take care of its execution via the standard DIRAC Job Agent mechanism.

4. Test results

The test consisted in the submission of the simulations described in last section, in order to validate the Multi-VO architecture with different instances, which were launched through one instance of the CloudStack 2.2.3 manager. The number of jobs submitted were around 500 of short jobs, and 50 long jobs. In the short jobs 104 jobs were executed with errors as "identified as stalled job", which means there is no contact with the job agent for a long time, 57 errors with "Application Finished With Errors", where the simulations had some failure in the process of execution, and 3 errors with no space in disk. The error trading for the "stalled jobs" showed that the unresponsiveness of the jobs was caused by a saturation of the host machines, which were only being used for theses tests. The saturation was, in turn, caused by an excessive number of VM running simultaneously on a given physical nodes exceeding its capabilities. Initially, 4 VM were simultaneously per node; this number was later reduced to 3 because the 2GB memory requirement of the VM was the origin of the host saturation regardless of the VM type. Finally, the remaining 443 short jobs and the long ones finished successfully. Figures 3 and 4 show the used CPU for short and long jobs.

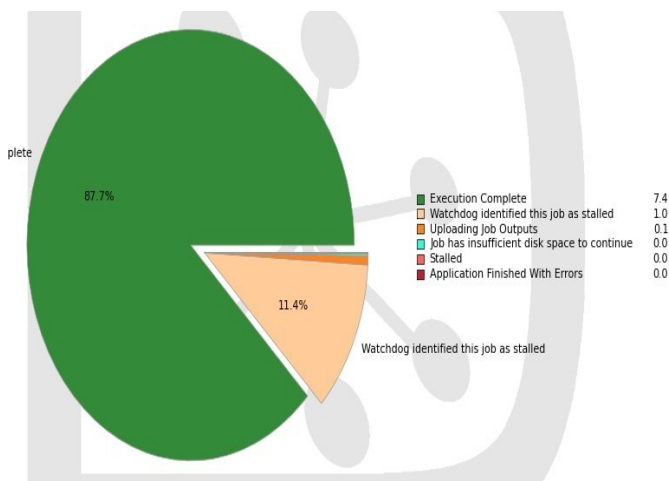


Figure 3: Final status of short jobs for CPU used.

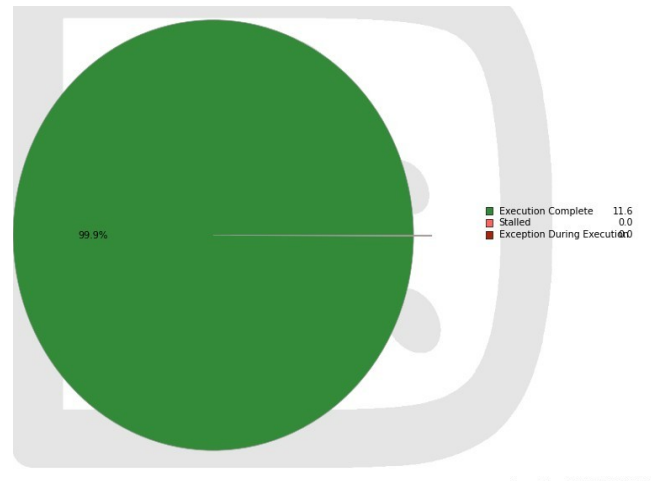


Figure 4: Final status of long jobs for CPU used.

Figure 5 shows a captured of the VM-monitoring web page with a plot representing the number of VMs submitted to Cloudstack for long simulations. It shows the VMs separated by Platform, and show a maximum of 8 Centos-VMs, 4 Fedora-VMs and 3 Ubuntu-VMs submitted at the same time.

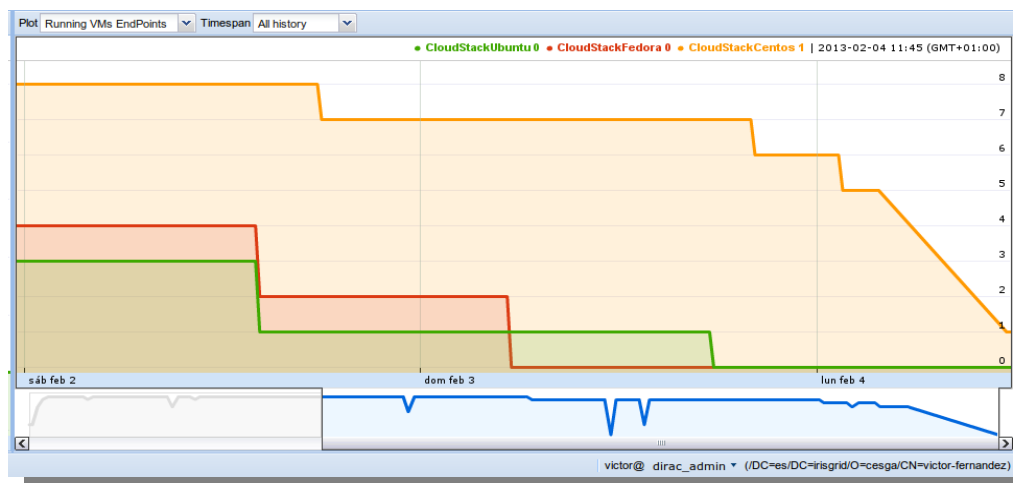


Figure 5: Browser monitoring for multi-platform

5. Conclusions

This paper describes the use of the DIRAC software for multi-platform and Multi-VO. The described approach make use of the flexibility provided by CernVM-FS that was tested in 3 different platforms and with several HEP-Software. In this manner DIRAC is able to take advantage of the strong points of the Cloud technologies, which provides the necessary multi-platform flexibility, which was combined with the Multi-VO functionality of DIRAC.

Acknowledgements

This work was supported by projects FPA2007-66437- C02-01/02 assigned to UB and FPA2010-21885-C02- 01/02 and TIN 2010-17541 USC. FPA2007-66152-C02-01/02 and FPA2010-21816-C02-01/02, assigned to PIC.

References

- [1] Tsaregorodtsev A 2003 DIRAC: distributed infrastructure with remote agent control *Proc. Conference for Computing in High Energy and Nuclear Physics* (La Jolla, California)
- [2] Graciani R, Casajus A, Carmona A, Field T and Seviar M 2011 Belle-DIRAC setup for using amazon elastic compute cloud *Journal of Grid Computing* **9**
- [3] Arrabito L, Cohen-tanugi J, Renaud M, Graciani R, Sapunov M, Tsaregorodtsev A, Rolland A and Zimmer S 2013 DIRAC framework evaluation for the Fermi-LAT and CTA experiments *Conference for Computing in High Energy* (Amsterdam, Holand)
- [4] Fernandez-Albor V, Saborido-Silva J, Mendez-Munoz V, Graciani R, Casajus A, Fernandez-Pena T and Merino-Arevalo G 2012 *The Integration of CloudStack and OCCI/OpenNebula with DIRAC*, Journal of Physics: Conference Series **396**
- [5] Blomer J and Fuhrmann T 2010 A fully decentralized file system cache for the CernVM-FS *Proceedings of 19th International Conference* pp.1-6, 2-5 Aug. 2010, doi: 10.1109/ICCCN.2010.5560054
- [6] Apolinario L, Armesto N and Cunqueiro L 2012 An analysis of the influence of background subtraction and quenching on jet observables in heavy-ion collisions <http://arxiv.org/abs/1211.1161>
- [7] Apolinario L, Armesto N and Cunqueiro L 2012 Background subtraction and jet quenching on jet reconstruction International Conference on Hard and Electromagnetic Probes of High Energy Nuclear Collisions <http://arxiv.org/abs/1207.6587> (Cagliari, Italy)
- [8] CloudStack API - Main Web Page 2010 <http://cloudstack.apache.org/docs/api/>
- [9] Universidade de Santiago de Compostela - Main Web Page 2011 <http://www.usc.es>