# How to Run Scientific Applications with DIRAC in Federated Hybrid Clouds

Víctor Méndez Muñoz
LHC Tier-1 Computing Production.
Port d'Informació Científica (PIC).
Universitat Autònoma de
Barcelona (UAB).
Bellaterra, Spain.
Email: vmendez@pic.es

Adrian Casajús Ramo and
Ricardo Graciani Diaz
Department of Structure and
Constituents of Matter.
Universitat de Barcelona (UB).
Barcelona, Spain.

Víctor Fernández Albor
Particle Physics Department.
Universidade de Santiago de
Compostela (USC).
Santiago de Compostela, Spain.

***Abstract*---Nowadays, the eScience big issue in Cloud Computing is how to leverage on-demand computing in scientific research. For this purpose, the specifics requirements of the complex scientific applications have been addressed with DIRAC, which has the motto *the interware*, because it is a proven scientific community solution, currently providing transparent access and interoperability between different distributed infrastructures, such as European Grid Infrastructure (EGI), Open Science Grid (OSG), computing clusters, standalone hosts, and cloud infrastructures. In this context, Federated Hybrid Clouds are emerging as a model of coordinated service access and delivery to multiple Infrastructure as a Service (IaaS) providers. The term hybrid comes from the integration of community clouds and commercial clouds in a federated manner, which also requires the use of additional services, such as federated authentication, accounting or monitoring. This paper explains how DIRAC is providing Software as a Service (SaaS) for generic scientific computational purposes. The cloud extension of DIRAC (VMDIRAC) is used to instantiate, monitor and manage Virtual Machines (VMs) in multiple IaaS aggregations of Amazon EC2, OpenNebula, OpenStack and CloudStack. Furthermore, DIRAC and VMDIRAC extension can provide SaaS of any scientific application through a contextualization management of few *golden* VM images, which automates the necessary context to run transparently in multiple IaaS providers, as well as the required software and tools for any eScience application.**

***Keywords*---*Cloud Computing; Federated Hybrid Cloud; on-demand Cloud Computing models***

## I. INTRODUCTION

VMDIRAC is the chosen tool in Cloud Computing matters for the scientific computing of LHCb[1] and Belle [2] in high energy physics (HEP) and the different Virtual Organizations (VOs) of the France Grilles[3] using the DIRAC portal, with an important community of life science. There is work in progress for the adoption of a DIRAC portal including the VMDIRAC extension for Federated Clouds in the context of National Grid Initiatives (NGIs) of the European Grid Infrastructure (EGI), as well as DIRAC portals of some scientific communities, like BES[4] and CTA [5] in astrophysics, or ILC[6] (HEP).

The proposed solution represents a big step forward in terms of scope. Any adopter is able to target cloud computing resources transparently, while the storage is supported by third-party solutions. Moreover, this allows the users to take advantage of the virtualization assets, mainly the VM encapsulation opening an opportunity window in the multi-core running [7], to exploits the latest many cores hardware without dependency on the platform, which becomes user specific. Additional opportunities for user requirements in High Performance Computing (HPC) are also addressable by HPC Cloud providers [8], [9], [10].

The term VMDIRAC is used for the specific features of the federated cloud extension, while the term DIRAC is used for the general features or components.

This paper is focused on the IaaS provider and user communities requirements to deploy Federated Hybrid Clouds with DIRAC. It is organized as follows. Section 2 defines how to aggregate an IaaS provider to a DIRAC Federated Hybrid Cloud. Section 3 is focused on the DIRAC setup to run a generic eScience application using the available IaaS providers. Section 4 describes the Web interface for cloud management. The paper finishes with a conclusion section.

## II. HOW TO AGGREGATE AN IAAS PROVIDER TO DIRAC

The first step is to have a DIRAC portal with a VMDIRAC extension installed, or request the VMDIRAC installation to the portal administrators. The installation, configuration and operational maintenance of a DIRAC portal is not a trivial matter, so the easy way would be to ask to your NGI for this purpose. Currently, some NGIs have their own DIRAC portals: France Grilles DIRAC portal[11] and also Ibergrid portal[12], which is the Spanish and Portuguese common infrastructure. Other NGIs are considering to deploy they own DIRAC portal, therefore NGI would be the first place to ask for support. For medium and big eScience communities interested in having a DIRAC portal, the official DIRAC webpage[13] may provide instructions for further support.

Once there is a DIRAC portal with a VMDIRAC extension installed, there are few IaaS provider requirements. This section describes the minimal requirements to deploy a Federated Hybrid Cloud using DIRAC, then, the supported IaaS cloud managers and some basic specifications and recommendations to the IaaS providers.

### *A. MINIMAL REQUIREMENTS TO DEPLOY A FEDERATED HYBRID CLOUD WITH DIRAC*

Federated hybrid cloud computing model [14] is considering commercial and community cloud end-points as resources. An overall federated cloud model is including federated services which are necessary in a scientific community. Such federated services definition is usually related with Metadata

 ISBN: 978-1-61208-290-5

repository of images, Information System, Accounting and Monitoring, as well as third-party services, which are offshore of the Federated Hybrid Cloud infrastructure, for example authentication and authorization or external storage.

To facilitate the aggregation of IaaS providers in a federated manner, VMDIRAC is able to deal with minimal requirements that do not require external services, but include manually the necessary information in the DIRAC Configuration Server. This is a compromise solution to allow a fast deployment of the Federated Hybrid Cloud having only the IaaS providers end-points, then VMDIRAC is able to extend and automate specific implementations of such external federated services.

The minimal requirements for a Federated Hybrid Cloud can be deployed in DIRAC as follows:

- Including a metadata repository of images, with the necessary contextualization. Any third-party automated image distribution can be used or manually uploaded to the different IaaS providers, the corresponding metadata of these images is described in the DIRAC Configuration Server.
- The IaaS providers information about the end-points is defined in the DIRAC Configuration Server. This information is usually published in the Information System.
- VM Monitoring for the eScience community users by the VM Browsing of the Web interface, to monitor VM history logs and plots related with transfers, jobs and CPU loads. See Fig. 1.
- External IaaS provider monitoring of the VM running on their site can be included using VM contextualization to deploy the monitoring client. For example, monitoring and notification based on Nagios alarms[15] or VM statistics monitoring with Ganglia[16]. This is an IaaS provider requirement, which is optional to VMDIRAC.

The mentioned third-party services are transparent from DIRAC point of view, because the interaction with them is not directly assumed by VMDIRAC. Thus, scientific applications can use external storage resources. The current approaches are using Grid and Cloud Storage, such as standards gridftp[17] and Cloud Data Management Interface (CDMI[18]), a pay-per-use storage like Amazon S3[19] or external storage resource interfaces [20] [21]. Some of these storage resources can be supported as part of the IaaS resources. Authorization and authentication methods associated to different user access to the end-points, requires from DIRAC a minor support because Configuration Server has the information related to the user, X509 proxy or other user identifications. At the same time, VMDIRAC is not interacting with the third-party authentication service but with the IaaS provider, which transparently supports the authentication and authorization. In general terms, the IaaS *auth* is managed on VO basis, the VM are created and owned by the VO, not by a particular user. Once the VM is created DIRAC takes control supporting DIRAC user policies. VM can run multiple jobs and each one is corresponding to a particular DIRAC user. This job *auth* management is fully supported by DIRAC, which eventually may contact external services for proxy or token authorization and it is decoupled from the IaaS *auth* management.

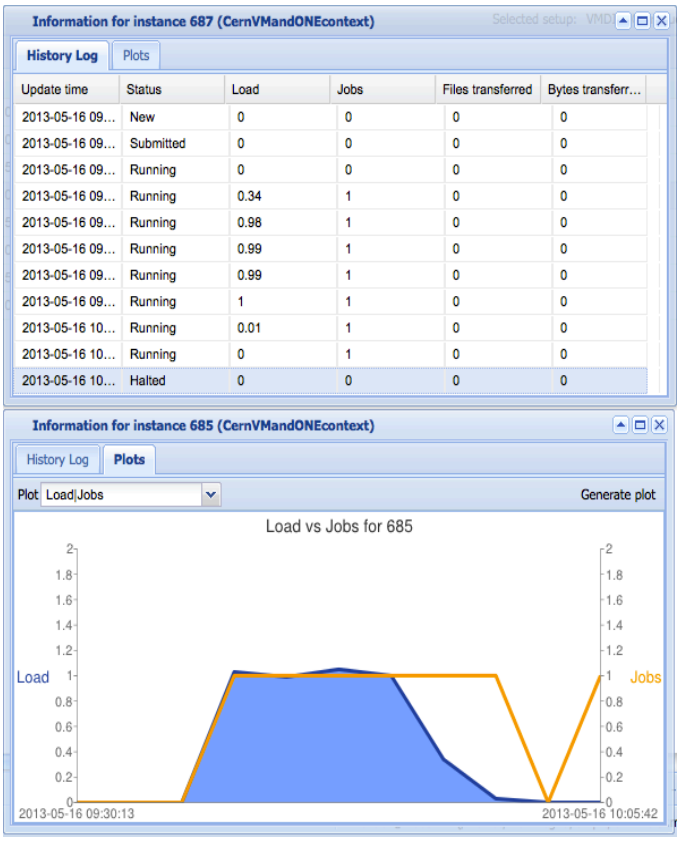


Fig. 1: DIRAC VM Monitoring for the eScience community

## B. SUPPORTING IAAS PROVIDERS IN DIRAC

Current VMDIRAC release supports the following cloud managers APIs for commercial and private clouds:

- Amazon EC2
- OpenNebula OCCI 0.8
- OpenStack Nova 1.1
- CloudStack 2

It is necessary to have at least one of such end-point deployed in the cloud manager server. The VM needs to have out-bound connectivity to the VMDIRAC server.

From the scaling test in some IaaS providers [22] [23] [24], there are some known lessons that should be considered:

- To scale up in the IaaS site it is necessary to have a snap-shot image management, just to accelerate the instance creation in the host hypervisors. This has been particularly tested with OpenNebula, which can use *qcow2* and *NFS* for the image distribution and creation in the host hypervisors.
- The opportunistic use of the hypervisor memory among the VMs it is highly discouraged. This has been particularly tested with KVM hypervisor. The result

 ISBN: 978-1-61208-290-5 

shows the average performances are not improved, while the dispersion on the performances is increased.

- The hypervisor kernel flags should be compatible with the software to run. In particular, KVM hypervisor is using *lib viewer*, which is working transparently on *Intel* platform for the wide range of scientific software tested, however, *AMD* processor architecture uses a different subset of kernel flags, which currently are not implemented on *lib viewer*. The platform compatibility of *lib viewer* roadmap should be checked before the IaaS deployment.
- The optimum VM network interface to work with VMDIRAC is an authomatic network configuration with private IP and out-bound connectivity.
- VMDIRAC is able to deal with static network configuration. This can be used for testing purposes, but the maintenance of such approach in production level is problematic, any minor change in the IaaS network configuration, should have the corresponding adjustment in DIRAC Configuration.
- OpenStack floating IP assignment to a previously created VM is not recommendable because there is a gap in the responding time between the time the VM is booted and the time the network interface is available and routable [14].

Moreover, there are different ways to provide the scientific software and tools to be deployed at the VM. This part is related to the contextualization of the next section, but at the same time is a matter of the IaaS provider. For image maintenance reasons and also for performance reasons it is recommended to allocate the required software and tools in a *cvmfs* repository [25] and to setup a site http proxy for the VMs use of the *cvmfs* repository.

## III. DIRAC SETUP TO RUN A GENERIC SCIENTIFIC APPLICATION AMONG THE IAAS PROVIDERS

Two main topics are related to DIRAC setup for the VMDIRAC extension: the image and contextualization setup, and the VM horizontal auto-scaling setup, which are the possible policies to create and stop VMs. Once the setup is ready, VMDIRAC is able to create and stop VMs among the IaaS providers and to contextualize those VMs in a transparent manner to provide computing resources to run user jobs. Such jobs can be any scientific software which is able to run decoupled jobs in distributed resources.

### A. IMAGE AND CONTEXTUALIZATION SETUP

There are three steps to setup at DIRAC Configuration Server: Running Pods, Images and End-points. VMDIRAC defines the Running Pod as a logical abstraction of a particular running conditions. A Running Pod is matching an Image with the corresponding cloud end-point list to run VMs of such Image. VMDIRAC concept of an Image, is including a *boot image* and, optionally, the contextualization of such image. This approach can deal with *ad-hoc* image ready to run without further contextualization, this image has to be prepared to run in a specific endpoint and a particular DIRAC configuration. Additionally, VMDIRAC can manage context images that use a *golden image* and the necessary information for a particular contextualization method. The use of a *golden image* allows to simplify the image management [26], because all the specifics of the endpoint and scientific application environment is in the contextualization part, while the *golden* image can be distributed to the different IaaS providers without modification. On the latter contextualization, VM is deployed for a particular scientific application environment. HEP has a contextualization approach, namely HEPiX, using CernVM images and contextualization methods supported by OpenStack and OpenNebula. This CernVM approach can also be used with other scientific applications. Instead of a *golden* image depending on the CernVM platform, VMDIRAC also supports a generic *golden* image, which can be configured using a *ssh* contextualization, if an in-bound connectivity is available in the VM for *ssh* and *sftp* operations. A DIRAC image setup can be an *ad-hoc* image or the following contextualized images:

- HEPiX - OpenNebula: DIRAC image context is included in an ISO context image, which has to be previously upload to the IaaS provider to be mounted by the CernVM init process. The end-point context is passed to the VM at submission time. VMDIRAC gets the parameters from the corresponding end-point section and set this environment using the OpenNebula context section, which creates an on-the-fly ISO image, then CernVM mounts it and loads the end-point context.
- HEPiX - OpenStack: DIRAC image context is provided by *amiconfig* tools, sending the scripts in nova 1.1 *userdata*. End-point context is provided through nova 1.1 *metadata*, which is specific for each OpenStack IaaS end-point and selected on submission time from the DIRAC Configuration Server.
- Generic contextualization, using any platform for *golden* image with a *ssh* demon listening in a port with in-bound connectivity in the VM. The VM boots, the VMDIRAC polls the active *sshd* port, runs the DIRAC and the end-point configuration using *sftp* and *ssh* connections.

In this manner, a VM *golden* image is contextualized to deploy DIRAC on multiple IaaS providers, following methods of the industry and research image contextualization, and also considering a generic contextualization valid for any VM images with *ssh* connectivity.

### B. VM HORIZONTAL AUTO-SCALING SETUP

VMDIRAC can be configured with different policies for the creation and stoppage of the VMs. Each end-point has associated a VM allocation policy and a VM stoppage policy.

The VM allocation policy can be *elastic* or *static*. The *static* VM allocation is used when a IaaS provider defines a constant number of VM slots that can be accessed. The *elastic* allocation is used to create new VMs when there are jobs queued in DIRAC. For this purposed the Running Pod configuration section has the *CPUPerInstance* option, which defines the minimal overall CPU of the DIRAC jobs waiting in the task queued to submit a new VM. The parameter is used for the tuning of the VM delivery elasticity. Therefore,

 ISBN: 978-1-61208-290-5 

a *CPUPerInstance* can be set to a longer time to use the available resources in a more efficient manner, saving creation overheads, and to a shorter time to setup an exhaustive use of the available resources aiming to finish the production in a shorter total wall time, but with higher resource costs due to additional overhead. In regular basis, *CPUPerInstance* references, from shorter to longer values, could be defined to:

- *Zero* to submit a new VM with no minimal CPU in the jobs of the tasks queue.
- A longer value could be the average required CPU of the jobs as a compromise solution between VM efficiency and total wall time.
- A very large value to maximize the efficiency in terms of VM creation overhead, for the cases where the production total wall time is not a constrain.

The VM stoppage policy can be setup to *elastic* or *never*. Elastic policy stops the VM if there are no more jobs running in the last *VM halting margin time*, which is an option to be setup. Anyway, VMs can be stopped by the VMDIRAC admin or by the HEPiX stoppage in the CernVM images (responsibility of each IaaS provider). If a running VM is required to be stopped, then the VM orderly stops, declaring the running job stopped in DIRAC (which can be resubmitted), then halting the VM.

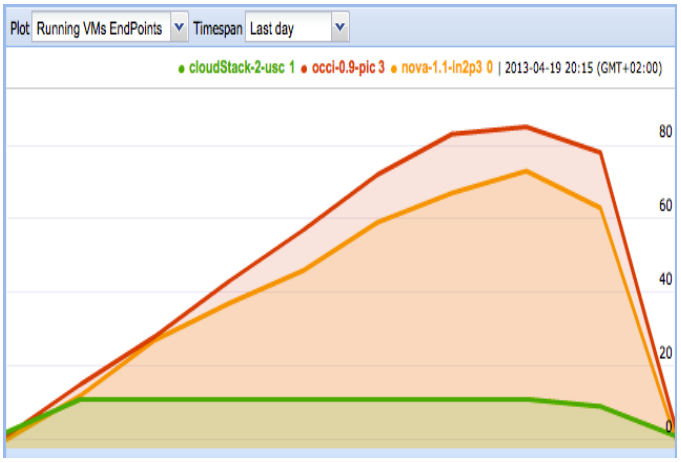


Fig. 2: Running VMs by End-point

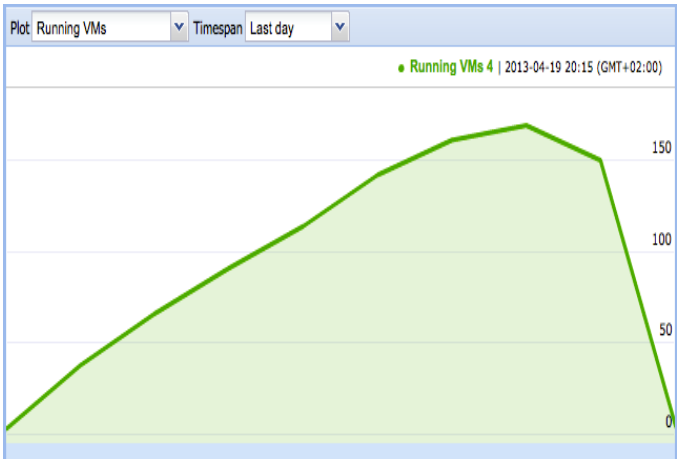


Fig. 3: Overall Running VMs

## IV. DIRAC WEB INTERFACE FOR CLOUD MANAGEMENT

DIRAC supports the job management that can be setup to run in Cloud or other distributed resources. This includes a wide number of tools to submit jobs, design workflows for eScience productions, manage execution, plots, accounting and all the necessary for eScience communities [27]. The Configuration Server Web interface allows to setup the Federated Hybrid Cloud as well as different Running Pods, Images and End-points to use transparently such IaaS infrastructure.

VMDIRAC, the DIRAC cloud extension, is also providing a Web interface for the management of the Cloud. There is a VM Browsing to monitor the VMs and take history logs and plots of each VM as it was shown above in Fig. 1. Furthermore, there is a VM Overview to plot the main statistics: running VM by end-point in Fig. 2, overall running VMs in Fig. 3, started jobs, average load, transfer data and transfer files.

Fig. 2 and 3 are plots of the same run. The running application is LHCb simulation of proton-proton collision. The IaaS providers are USC with CloudStack, CC.IN2P3 is supported by OpenStack and PIC by OpenNebula. The VM allocation policy is *elastic* for the three IaaS providers. Contextualization is an *ad-hoc* image with Centos, while PIC and CC.IN2P3 are using CernVM *golden* image and the corresponding contextualization. The VM stoppage policy is *elastic* in the three IaaS providers.

DIRAC configuration has a maximum number of 10 VMs for USC, once the threshold is reached there is a plateau until the end of the workload (green line in Fig. 2). For the case of CC.IN2P3 and PIC IaaS providers the maximum number of VMs has not been reached.

Fig. 3 shows the overall aggregation of the three IaaS providers. VM allocation and stoppage policy is corresponding to a scale-up when there are jobs pending in the DIRAC task queue, and a scale-down when there are no more jobs in the task queue and the VMs workload is finished.

Some of the operations, like the Configuration Server management or the manual VM stoppage, are only authorized

 ISBN: 978-1-61208-290-5 

for the VM administrator, who can also monitor all the jobs. A general DIRAC user is authorized to monitor his or her jobs, and the VM Web interface without operate with the VMs.

## V. CONCLUSION

This paper has described how to run eScience applications in Federated Hybrid Clouds using DIRAC. This includes instructions and recommendations to the IaaS providers to be aggregated in a federated manner. It also has been defined how to setup the image and context management to run scientific applications, also considering the VM deployment of the scientific software and tools. At the same time, it has been shown the necessity of two roles: administrator of the configuration system and VMs, and the scientific user, who uses DIRAC submitting jobs, which transparently run in Federated Hybrid Clouds.

The deployment of DIRAC portals including the cloud extension VMDIRAC, is a proved tool to aggregate IaaS providers in the level of NGIs supporting multiple VOs, and also in medium and big scientific communities.This schema provides solutions to the deployment and management of SaaS of a wide range of scientific communities, from small communities which can be federated to face the DIRAC portal operations, to big communities who may exploit their own DIRAC portal or integrates in a multiple VO portal. Thus, the proposed strategy is addressing the sustainability through *industrial concentration* of the management of SaaS in Federated Hybrid Clouds, and at the same time allowing *local development* by the aggregation of distributed IaaS resources.

## ACKNOWLEDGMENT

PIC is maintained through a collaboration between the Generalitat de Catalunya, CIEMAT, IFAE and the Universistat Autónoma de Barcelona. This work was supported in part by grants of the Ministerio de Educación y Ciencia, Spain: FPA2007-66152-C02-01/02 and FPA2010-21816-C02-01/02, assigned to PIC. Additional support was provided by the EU 7th Framework Programme INFRA-2007-1.2.3: e-Science Grid infrastructures Grant Agreement Number 222667, Enabling Grids for e-Science (EGEE) project and INFRA-2010-1.2.1: Distributed computing infrastructure Contract Number RI-261323 (EGI-InSPIRE).

This work was also supported by projects FPA2007-66437-C02-01/02 and FPA2010-21885-C02-01/02, assigned to UB and USC.

We are greatly in debt with France Federated Cloud, in particular with the *Centre de Calcul* of the IN2P3 for providing part of the VMs used in the presented test results.

## References

[1] Lhcb computing with dirac . [Online]. Available: http://lhcb-comp.web.cern.ch/lhcb-comp/DIRAC/

[2] R. Graciani Diaz, A. Casajus Ramo, A. Carmona Aguero, T. Fifield, and M. Sevior, ``Belle-dirac setup for using amazon elastic compute cloud," *Journal of Grid Computing*, vol. 9, pp. 65--79, 2011, 10.1007/s10723-010-9175-7. [Online]. Available: http://dx.doi.org/10.1007/s10723-010-9175-7

[3] France grilles et du cloud . [Online]. Available: http://www.france-grilles.fr/-Presentation-

[4] Bes collaboration . [Online]. Available: http://bes.ihep.ac.cn/

[5] Cta observatory . [Online]. Available: http://www.cta-observatory.org

[6] Linear collider collaboration . [Online]. Available: http://www.linearcollider.org

[7] D. Wentzlaff, C. Gruenwald, III, N. Beckmann, K. Modzelewski, A. Belay, L. Youseff, J. Miller, and A. Agarwal, ``An operating system for multicore and clouds: mechanisms and implementation," in *Proceedings of the 1st ACM symposium on Cloud computing*, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 3--14. [Online]. Available: http://doi.acm.org/10.1145/1807128.1807132

[8] A. Gupta, D. Milojicic, and L. V. Kalé, ``Optimizing vm placement for hpc in the cloud," in *Proceedings of the 2012 workshop on Cloud services, federation, and the 8th open cirrus summit*, ser. FederatedClouds '12. New York, NY, USA: ACM, 2012, pp. 1--6. [Online]. Available: http://doi.acm.org/10.1145/2378975.2378977

[9] G. Mateescu, W. Gentzsch, and C. J. Ribbens, ``Hybrid computing - where hpc meets grid and cloud computing." *Future Generation Comp. Syst.*, vol. 27, no. 5, pp. 440--453, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/fgcs/fgcs27.html#MateescuGR11

[10] B. Kocoloski, J. Ouyang, and J. Lange, ``A case for dual stack virtualization: consolidating hpc and commodity applications in the cloud," in *Proceedings of the Third ACM Symposium on Cloud Computing*, ser. SoCC '12. New York, NY, USA: ACM, 2012, pp. 23:1--23:7. [Online]. Available: http://doi.acm.org/10.1145/2391229.2391252

[11] France grilles dirac portal . [Online]. Available: http://dirac.france-grilles.fr/DIRAC/

[12] Ibergrid dirac portal . [Online]. Available: http://dirac.ub.es/DIRAC/

[13] Official dirac webpage . [Online]. Available: http://diracgrid.org/

[14] V. Méndez, A. Casajus, V. Fernández, R. Graciani, and G. Merino, ``Rafhyc: An architecture for constructing resilient services on federated hybrid clouds," *Journal of Grid Computing*, 2013.

[15] Nagios infrastructure monitoring . [Online]. Available: http://nagios.org

[16] Ganglia monitoring system . [Online]. Available: http://ganglia.info

[17] Globus gridftp . [Online]. Available: http://www.globus.org/toolkit/docs/latest-stable/gridftp/

[18] Cloud data management interface (cdmi) . [Online]. Available: http://www.snia.org/cdmi

[19] Amazon simple sotorage service (s3) . [Online]. Available: http://aws.amazon.com/es/s3/

[20] A. Alvarez, A. Beche, F. Furano, M. Hellmich, O. Keeble, and R. Rocha, ``Dpm: Future proof storage," in *Computing in High Energy and Nuclear Physics 2012*, 2012. [Online]. Available: http://cdsweb.cern.ch/record/1458022

[21] F. Furano, P. Fuhrmann, R. B. da Rocha, A. Devresse, O. Keeble, and A. A. Ayllon, ``Dynamic federations: storage aggregation using open tools and protocols," in *EGI Technical Forum Book of Abstracts*, 2012. [Online]. Available: https://indico.egi.eu/indico/conferenceDisplay.py/abstractBook?confId=1019

[22] V. Mendez, A. Casajus, R. Graciani, and G. Merino, ``Use case: Running monte carlo lhcb simulations using dirac with egi federated cloud," in *EGI Tehcnical Forum Book of Abstracts*, 2012. [Online]. Available: https://indico.egi.eu/indico/conferenceDisplay.py/abstractBook?confId=1019

[23] V. Méndez, V. Fernández, R. Graciani, A. Casajus, T. Fernández, G. Merino, and J. J. Saborido, ``The integration of cloudstack and occi/opennebula with dirac," *Journal of Physics Conference Serires*, 2013. [Online]. Available: http://iopscience.iop.org/1742-6596/396/3/032075

[24] V. F. Albor, J. J. S. Silva, F. Gómez-Folgar, J. López-Cacheiro, and R. G. Diaz, ``Dirac integration with cloudstack," in *Proceedings of 3rd IEEE International Conference on Cloud Computing Technology and Science (IEEE CloudCom 2011)*, 2011, pp. 537--541.

 ISBN: 978-1-61208-290-5 

[25] V. F. Albor, V. Mendez, J. López-Cacheiro, R. G. Diaz, J. J. S. Silva, and T. F. P. na, ``User access to cvmfs software repositories on ibergrid,'' in *IBERGRID, 6th Iberian Grid Infrastructure Conference Proceedings*, 2012, pp. 39--49.

[26] P. Bunic, C. Aguado-Sanches, J. Blomer, and A. Harutynunyan, ``Cernvm: Minimal maintenance approach to the virtualization,'' *Journal of Physics Conference Serires*, 2011. [Online]. Available: http://iopscience.iop.org/1742-6596/331/5/052004

[27] A. Tsaregorodtsev, M. Bargiotti, N. Brook, A. Casajus Ramo, G. Castellani, P. Charpentier, C. Cioffi, J. Closier, R. Graciani Diaz, G. Kuznetsov, Y. Y. Li, R. Nandakumar, S. Paterson, R. Santinelli, A. C. Smith, M. S. Miguelez, and S. G. Jimenez, ``Dirac: a community grid solution,'' *Journal of Physics: Conference Series*, vol. 119, no. 6, p. 062048, 2008. [Online]. Available: http://stacks.iop.org/1742-6596/119/i=6/a=062048

 ISBN: 978-1-61208-290-5 