Belle II public and private cloud management in VMDIRAC system.

2015 J. Phys.: Conf. Ser. 664 022021

(http://iopscience.iop.org/1742-6596/664/2/022021)

Belle II public and private cloud management in VMDIRAC system.

Rafa Grzymkowski

Institute of Nuclear Physics PAN, Krakow, Poland

E-mail: rafal.grzymkowski@ifj.edu.pl

Takanori Hara

High Energy Accelerator Research Organization, KEK, Japan

E-mail: takanori.hara@kek.jp

on behalf of the Belle II computing group

https://belle2.cc.kek.jp/ twiki/pub/Public/ComputingPublic/AuthorList4Belle2Computing.tex

Abstract. The role of cloud computing technology in the distributed computing for HEP experiments grows rapidly. Some experiments (Atlas, BES-III, LHCb) already exploit private and public cloud resources for the data processing. Future experiments such as Belle II or upgraded LHC experiments will largely rely on the availability of cloud resources and therefore their computing models have to be adjusted to the specific features of cloud environment, in particular to the on-demand computing paradigm.

Belle II experiment at SuperKEKB will start physics run in 2017. Belle II computing requirements are challenging. The data size at the level of hundred PB is expected after several years of operation, around 2020. The baseline solution selected for distributed processing is the DIRAC system. DIRAC can handle variety of computing resources including Grids, Clouds and independent clusters. Cloud resources can be connected by VMDIRAC module through public interfaces. In particular the mechanism of dynamic activation of new virtual machines with reserved job slots for new tasks in case of an increasing demand for computing resources is introduced.

This work is focused on VMDIRAC interaction with public (Amazon EC2) and private (CC1) cloud. The solution applied by Belle II experiment and the experience from Monte Carlo production campaigns will be presented. Updated computation costs for different use cases will be shown.

1. Introduction

Belle II experiment[1] is able to compute on a variety kinds of resources, i.e. grids, standalone clusters, academic clouds and commercial clouds. Clouds are becoming increasingly important for the Belle II computing model[2].

For now, the Belle II distributed system can handle communication to EC2, OCCI and Nova APIs. It means that we can run virtual machines on Amazon, CC1, OpenStack and other compatible interfaces.

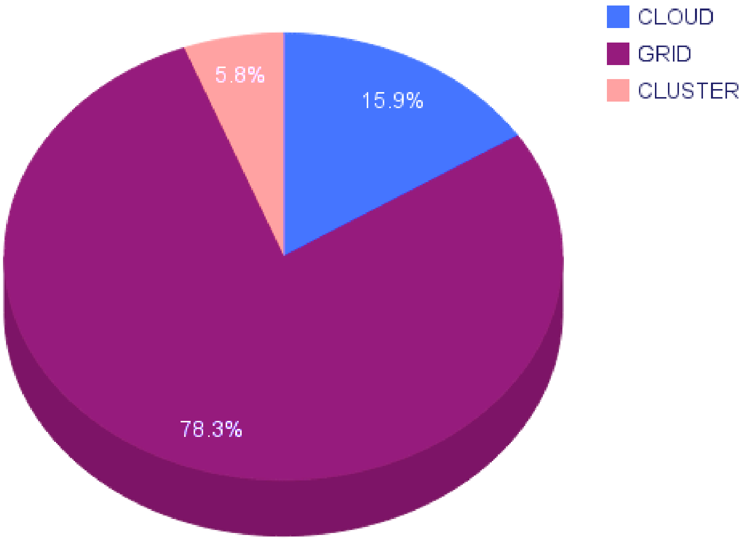


Figure 1. Belle II resources usage in March 2015. Cloud section consist of all kinds of clouds connected by VMDIRAC, Cloud-Scheduler and Dynamic-Torque.

The Belle II software is located on the same CVMFS Stratum-1 mirror servers which support LHC experiments. It gives us more stable and up-to-date access to the experiment's software.

Squid proxy server on each cloud region (Site) is set up for caching the software mounted by the CVMFS[3] to the virtual machine.

Nowadays, Belle II collaborators introduced at least three different ways to handle cloud infrastructure (VMDIRAC, Cloud-Scheduler, Dynamic-Torque). This paper covers VMDIRAC solution only.

2. Commercial and scientific clouds overview

In our work, we used two types of cloud resources, called *Infrastructure-as-a-Service*, public commercial and private scientific ones. An example of commercial cloud is Amazon EC2, which is a part of the Amazon Web Services [4]. Representative of private cloud connected to the Belle II computing model is the CC1 system[5] located in Krakow, Poland. Both types of cloud provide a public API called EC2. In general, private clouds are easier to manage. Often an amount of CPUs is provided by particular organization for free, based on MOU (memorandum of understanding). In case of commercial clouds we need to lead computations in the cheapest possible way, so that we are using non-persistent machines, like Amazon Spot Instances which are at about 70% cheaper than standard On-demand machines. Spot Instance is a type of virtual machine for which the price is constantly changing and often it is much cheaper than other type of instance. The price is different not only in each AWS Region but also in each AWS Zone (part of Region). Moreover, Spot Instance can be rapidly destroyed if the market price reach our maximum price, what is the main issue of such model. Fortunately for the Belle II we can apply such non-persistent model thanks to the Job Scheduler, a part of DIRAC WMS [8] which has a great failover mechanism. Also, Amazon EC2 provides On-demand instances, which are persistent and the cost is stable but much higher than Spot Instances. We are using On-demand machines for the Squid cache servers which need to have very high availability.

3. The DIRAC system and VMDIRAC extension

The DIRAC[6] (Distributed Infrastructure with Remote Agent Control) system is a framework to build a complex distributed environment in a relatively simple way. It is used by large communities like Belle II, LHCb, BES-III, but also smaller groups for example NGI multi-VO portals.

Cloud resources can be connected to the DIRAC by the VMDIRAC[8] module which provides a mechanism of dynamic activation of new virtual machines with reserved job slots for new tasks. All of this activates automatically in case of an increasing demand for computing resources. VMDIRAC is an extension of the DIRAC system. It is used to manage a pool of different kind of clouds by API drivers like EC2, OCCI, rOCCI, Nova.

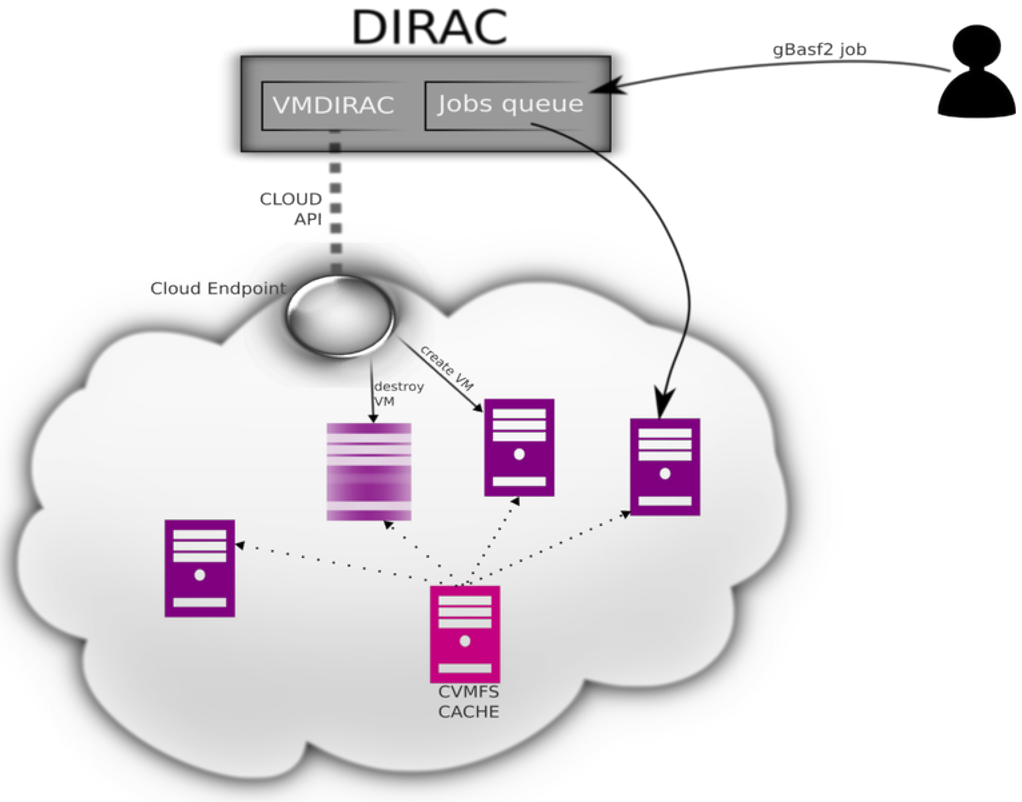


Figure 2. VMDIRAC workflow.

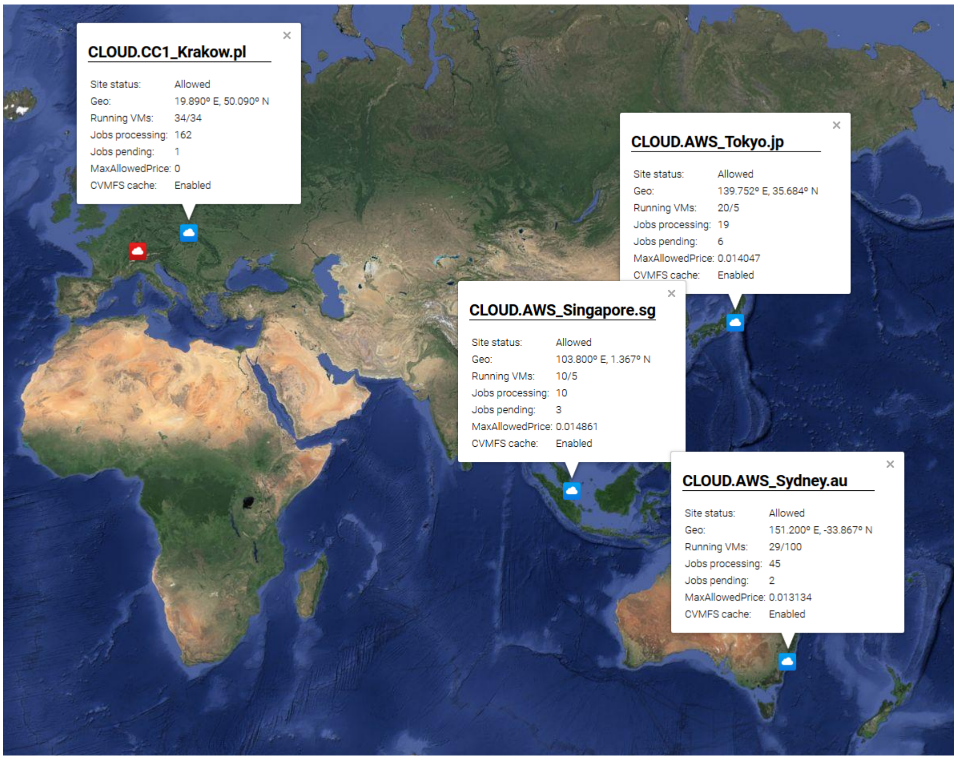
For the VMDIRAC Amazon driver we developed new extensions for a better control of different AWS regions and virtual machines based on Spot Instances.

3.1. VMDIRAC cloud resources map

The variety of cloud resources raises the management needs. For such purpose we prepared a DIRAC extension [9] which represents usage of cloud resources managed by VMDIRAC in the interactive Google Map [fig.3]. Cloud Resources Map is delivered by the BelleDIRAC software.

3.2. EC2UserData Agent

The DIRAC Configuration System is a main knowledge-base about distributed infrastructure, where we can check all available resources and its configuration. EC2UserData agent periodically checks the configuration and generates so called **user_data** script for each cloud endpoint. EC2



CLOUD.CC1_Krakow.pl

Site status:	Allowed
Geo:	19.890º E, 50.090º N
Running VMs:	34/34
Jobs processing:	162
Jobs pending:	1
MaxAllowedPrice:	0
CVMFS cache:	Enabled

CLOUD.AWS_Tokyo.jp

Site status:	Allowed
Geo:	139.752º E, 35.684º N
Running VMs:	20/5
Jobs processing:	19
Jobs pending:	6
MaxAllowedPrice:	0.014047
CVMFS cache:	Enabled

CLOUD.AWS_Singapore.sg

Site status:	Allowed
Geo:	103.800º E, 1.367º N
Running VMs:	10/5
Jobs processing:	10
Jobs pending:	3
MaxAllowedPrice:	0.014861
CVMFS cache:	Enabled

CLOUD.AWS_Sydney.au

Site status:	Allowed
Geo:	151.200º E, -33.867º N
Running VMs:	29/100
Jobs processing:	45
Jobs pending:	2
MaxAllowedPrice:	0.013134
CVMFS cache:	Enabled

Figure 3. VMDIRAC resources monitoring page.

machines are contextualized by the user_data script which starts during the VM boot process. For the private VM images we are using Cloud-init software for this purpose.

3.3. EC2SpotPrice Agent

This is the crucial agent to control EC2 Spot Instances. It calculate an optimal MaxBiddingPrice limit for the Spot Instance requests in each Amazon region and zone. Also, it increases limits of VM amount for the cheapest zone and decreases VM limits for expensive zones. Calculated results are stored in the main DIRAC Configuration. To compare prices on particular AWS Zone we calculate weighted moving average (WMA) indicator from the period of last two hours. It gives us smoother chart of changing price. Final parameter 'MaxBiddingPrice' in our case is always lower than 80% of OnDemand instance price and little bit higher than than WMA indicator [fig.4]. It is our solution to balance between low costs and stability of the system which we can get for such price.

4. Results and Summary

Belle II jobs (gBasf2) can take many hours of CPU, so any interruption is crucial. With the DIRAC system we achieved great failover mechanism of many services, like job submission [8] and even sudden Spot Instances shutdown can be easily handled and job can be rescheduled, thanks to this mechanism the efficiency of whole Belle II distributed system is close to 100%. Processing on Amazon Spot Instances gives us efficiency at about 98% if there are no strong fluctuations of price. Such situation can be observed in one of the chart below. AWS Sydney region became expensive in Australian working hours, so a lot of computing machines have been destroyed [fig.5]. For the long job which can take many hours, the risk of losing generated data and than rescheduling jobs from the beginning is high. For this reason it would be good

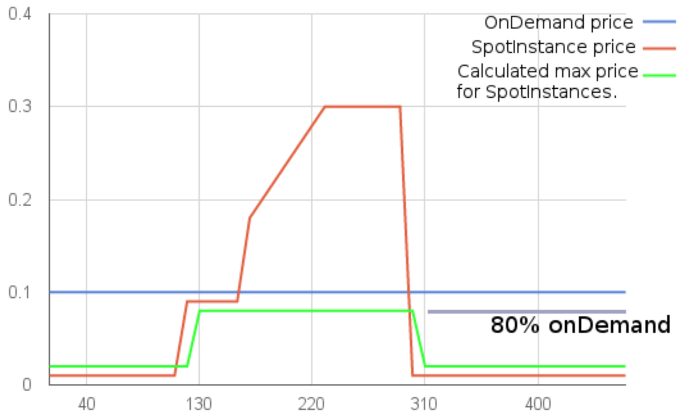


Figure 4. Maximum Spot Instance price based on different factors.

to upload output data periodically or at least to send 'halted' signal in the last minute of the instance life and than resubmit the job. In the current Belle II distributed computing model such halted job need to be resubmit form the scratch, even if the job was processing many hours. For this reason we need to operate Spot Instances carefully and strive for short jobs submission to all temporary resources.

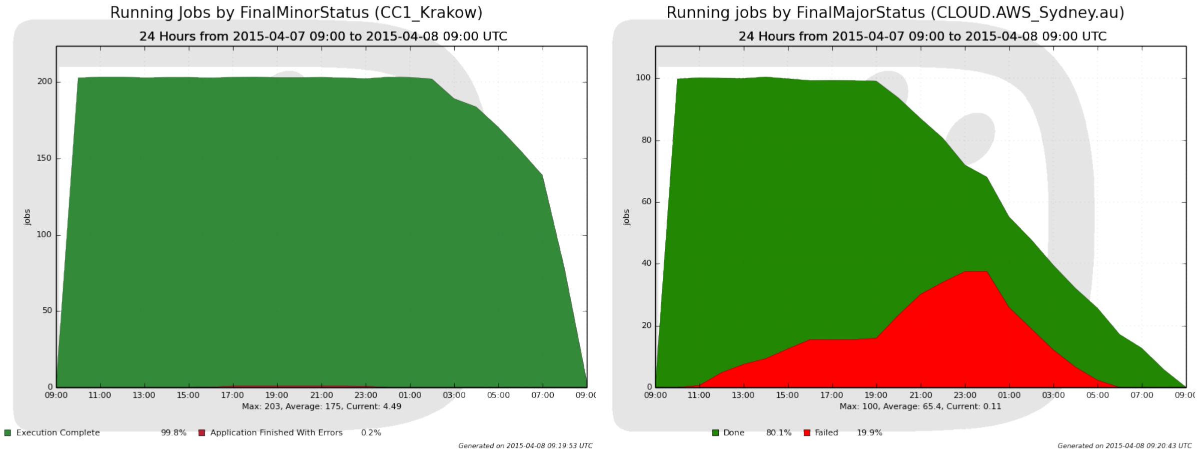


Figure 5. Belle II jobs on private CC1 cloud and Amazon EC2 Sydney region.

In this paper we present working examples of the MC generation on clouds. Commercial clouds comparing to the private, are more challenging for few reasons, one is mutable price for computations and outbound data transfers which need to be handled very carefully. We need to be ready for an unexpected destruction of the virtual machine what implicate a necessity of quick restore of resources so the reaction time is crucial. In some AWS Regions, price for the Spot instances is changing in more determined way, this indicates a possibility to use learning systems to decide where to start next computation machine to have processing more stable.

5. Acknowledgements

We are grateful for the support and the provision of computing resources by CoEPP in Australia, HEPHY in Austria, McGill HPC in Canada, CESNET in the Czech Republic, DESY, GridKa, LRZ/RZG in Germany, INFN-CNAF, INFN-LFN, INFN-LNL, INFN Pisa, INFN Torino, ReCaS (Univ. & INFN) Napoli in Italy, KEK-CRC, KMI in Japan, KISTI GSDC in Korea, Cyfronet, CC1 in Poland, NUSC, SSCC in Russia, SiGNET in Slovenia, ULAKBIM in Turkey, UA-ISMA in Ukraine, and OSG, PNNL in USA. We acknowledge the service provided by CANARIE, Dante, ESnet, GARR, GEANT, and NII. We thank the DIRAC and AMGA teams for their assistance and CERN for the operation of a CVMFS server for Belle II.

Thanks for the Amazon.com, Inc. for getting Education Grant to use AWS services (EC2 cloud in particular).

CC1 cloud installation at Institute of Nuclear Physics PAN Krakow is delivering computational resources for MC production, DIRAC development machines and interfaces for VMDIRAC testbed.

ACK Cyfronet AGH is an academic computer centre which is providing lots of computational and storage resources for Belle II jobs processing. Data generated on the private cloud CC1 are directly stored in Cyfronet's storage element.

The project was supported by the Polish National Science Centre (NCN) and the Polish Ministry of Science and Higher Education, grant reference no. 804/N-JAPONIA/2010/0.

References

[1] Belle II experiment, `http://belle2.kek.jp`

[2] Belle II collaboration, *Belle II Technical Design Report*, arXiv:1011.0352, 2010

[3] CVMFS, `http://cernvm.cern.ch/portal/filesystem`

[4] AWS, `http://en.wikipedia.org/wiki/Amazon_Web_Services`

[5] CC1, `http://cc1.ifj.edu.pl`

[6] DIRAC INTERWARE, `http://diracgrid.org`

[7] *DIRAC pilot framework and the DIRAC Workload Management System*, J. Phys.: Conf. Ser. 219 062049, 2010.

[8] *Cloud flexibility using DIRAC interware*, J. Phys.: Conf. Ser. 513 032031, 2014.

[9] Victor Fernandez Albor et al, *Powering Distributed Applications with DIRAC Engine*, The International Symposium on Grids and Clouds (ISGC) 2014,March 23-28, 2014, Academia Sinica, Taipei, Taiwan