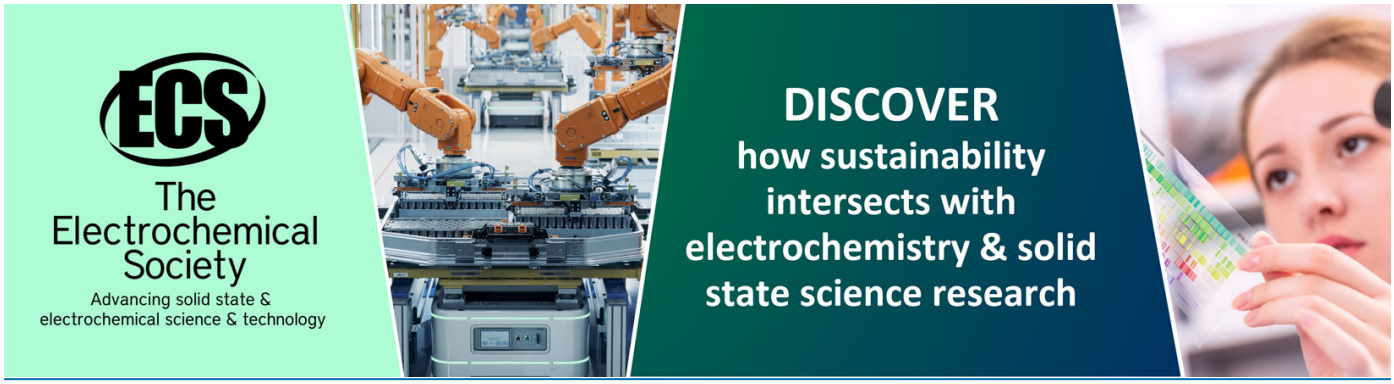
Utilizing clouds for Belle II



Journal of Physics: Conference Series **664** (2015) 022037 doi:10.1088/1742-6596/664/2/022037

Utilizing clouds for Belle II

R.J. Sobie (on behalf of the Belle II Computing Group)
Institute of Particle Physics, University of Victoria, Victoria, Canada
E-mail: rsobie@uvic.ca

Abstract. This paper describes the use of cloud computing resources for the Belle II experiment. A number of different methods are used to exploit the private and opportunistic clouds. Clouds are making significant contributions to the generation of Belle II MC data samples and it is expected that their impact will continue to grow over the coming years.

1. Introduction

Belle II is a next generation B-factory experiment at the SuperKEKB accelerator that is currently under construction at the KEK laboratory in Japan [1] This facility will produce more than 30 times the combined amount of data recorded by BABAR [2] and Belle [3] experiments. The commissioning of the accelerator and detector is scheduled for 2016 with the first physics data in 2017. The goal is collect a data sample of 50 ab^{-1} by 2022.

Belle II has adopted a distributed computing model [4], similar to the structure of the Worldwide LHC Computing Grid (WLCG) [5]. The requirements of Belle II for storage, processing power, and network bandwidth are comparable to an LHC experiment. The raw data will be recorded and processed at KEK, and a second copy of the raw data will be distributed to a set of regional centers around the world. The processed data is distributed to the Belle II centers for physics analyses.

Belle II uses the DIRAC Workload Management system [6]. DIRAC was developed to provide a solution for using the distributed computing resources of the LHCb experiment [7]. A DIRAC master server runs at KEK, which controls the submission of pilot jobs and keeps track of the running jobs.

Currently the fraction of Belle II MC generated on cloud resources is greater than 15% of the total sample. The resources include private clouds that are operated on behalf of the Belle II collaboration, or opportunistic private and commercial clouds. In this paper we will describe the different methods used for exploiting the clouds and highlight some of the issues and challenges of utilizing cloud resources.

2. Cloud use in Belle II

An overview of the Belle II job management system is shown in fig. 1. As mentioned, Belle II uses DIRAC for job submission and work load management. The majority of resources are WLCG facilities, and in many cases, they are also used by LHC experiments.

There are two strategies for using clouds in Belle II. The first uses the virtual machine (VM) provisioning services in DIRAC, called VMDIRAC [8]. The use of VMDIRAC in Belle II is the discussed in a separate contribution to this conference [9]. The second strategy is to construct a

site that appears as a typical grid site to DIRAC. A local VM provisioning service monitors the job queues and manages to VM's on the clouds. The separation of the work load management and VM provisioning services is beneficial when the system needs to run applications from multiple projects or projects outside of HEP. For example, some of the systems we describe, are also used by the ATLAS experiment.

There are three centers of cloud activity (not including the VMDIRAC effort) located at the University of Melbourne in Australia, University of Victoria in Canada and the Pacific Northwest National Laboratory in the US. These sites follow a similar design strategy for utilizing clouds, and have developed services that make efficient use of their existing facilities. We briefly describe the systems used at each center.

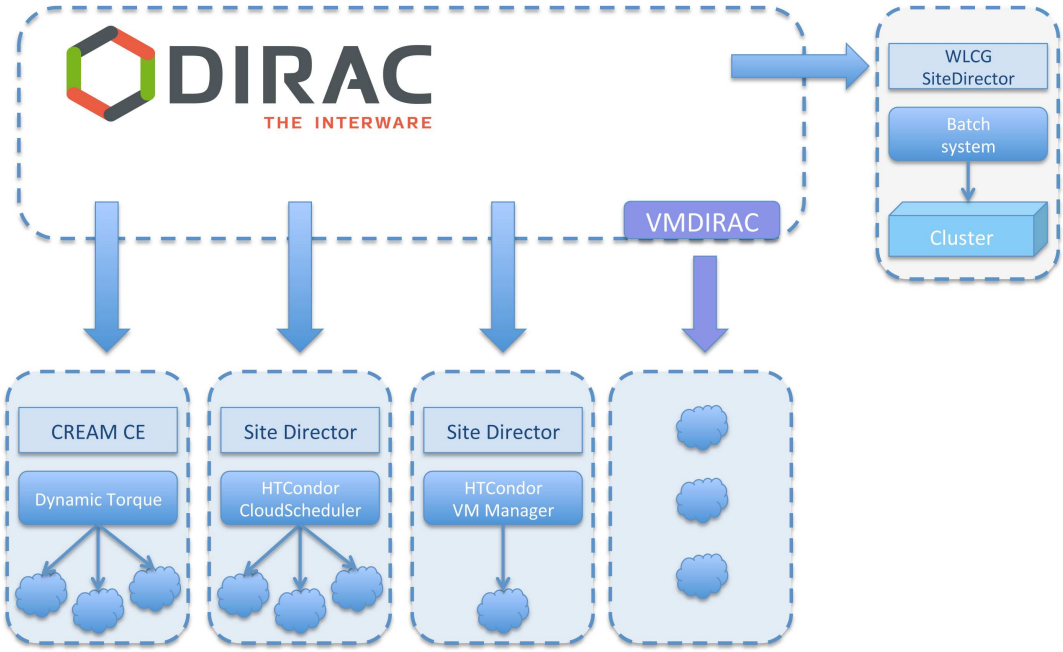


Figure 1. An overview of the Belle II distributed computing system. DIRAC is linked to the traditional (non-cloud) WLCG centers around the world. Most of the cloud resources appear as another grid center to DIRAC. The local centers use Dynamic-Torque (NeCTAR, Melbourne), HTCondor/CloudScheduler (Victoria) and HTCondor with a static cloud (PNNL) for internal job scheduling and VM provisioning. In addition, VMDIRAC is used by the Krakow group.

2.1. University of Melbourne site

The Belle II group at the University of Melbourne has been active in cloud computing for many years [10]. The group uses the resources provided by Australian National eResearch Collaboration Tools and Resources (NeCTAR) project [11], which has built a federated system of OpenStack clouds for Australian researchers.

NecTAR uses the TORQUE/Maui queue manager and job scheduler for its resources. Although there exist other batch systems with cloud utilization features, the Australian team felt it was important to leverage the existing expertise with TORQUE/Maui and integrate the services with OpenStack. As a result, the Melbourne team developed an external component, called Dynamic-TORQUE [12], that runs alongside TORQUE/Maui and communicates with the OpenStack API to manage the virtual machines. This approach has two major benefits: first, it is transparent to existing users as they continue to submit jobs to the TORQUE queue in the same way (without the need to know about clouds) and second, if Dynamic-TORQUE crashes, then TORQUE will not be affected and can continue to process jobs.

Dynamic-TORQUE works by periodically querying the TORQUE/Maui job queue. If there are waiting jobs and free resources, then Dynamic-TORQUE will launch new VMs in the cloud based on the job priorities gathered from Maui and the resource requirements of these jobs. The new VM's are contextualized from a standard SL6 image with a customized Puppet script [13]. When a VM is ready, Dynamic-TORQUE adds it to the TORQUE server as a normal worker node. TORQUE/Maui can then distribute idle jobs to the new worker node.

A high-level diagram of the use of the Australian clouds for Belle II is shown in fig. 2. The Belle II application software is obtained from CVMFS and input/output data files are stored in the Australian Storage Element.

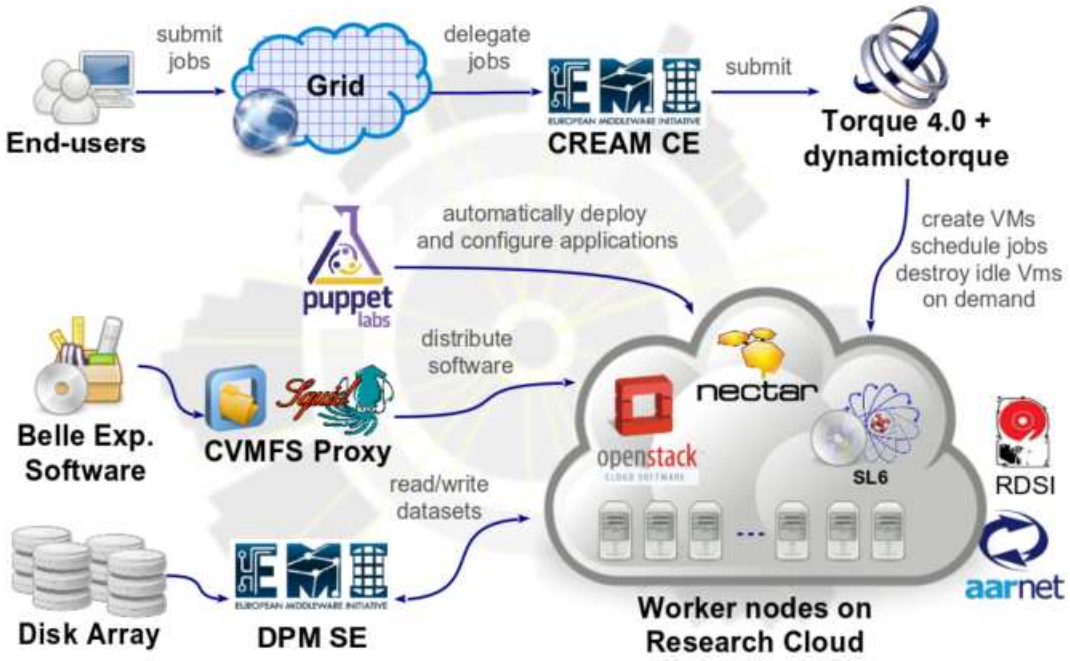


Figure 2. An overview of the Australian cloud system used by the Belle II experiment.

2.2. University of Victoria site

In fig. 3, we present a high-level view of the distributed cloud used by the Victoria group for Belle II (and ATLAS) [14, 15]. The scheduling of batch jobs is performed by HTCondor [16] and the deployment of VM images is done by CloudScheduler [17]. HTCondor was designed as a cycle scavenger, making it an ideal job scheduler for a dynamic cloud environment where VM instances appear and disappear based on demand. CloudScheduler periodically reviews the

requirements of the jobs in the HTCondor job queue and makes requests to boot user-specific VM images on one of the IaaS clouds. Once the VM image is booted, it attaches itself to the HTCondor resource pool and is ready to accept jobs. The instances are shut down when there are no jobs in the HTCondor queue.

The system uses μ-CernVM images [18] where both the operating system and the application software are stored in CVMFS [19]. Squid HTTP web caches are used to cache the software from CVMFS. Our reliance on a single Squid cache for the cloud resources was found to be a bottleneck as the system expanded to more sites. Since there are many Squid caches for the HEP community distributed around the globe, we can distribute the load and minimize the long distance transfers by having the VM instances use the nearest Squid. There are no services that provide this functionality, and, as a result, our group has developed Shoal, which is a dynamic web cache publishing service [20].

Shoal provides a reliable, scalable solution for a distributed cloud environment and can manage large numbers of requests. Shoal builds and maintains a list of Squid caches that advertise their existence to the central shoal-server using a shoal-agent. The clients (VM instances) contact the server for a list of Squids that is ordered on the relative location to the client and the load of the Squid server.

With the emergence of clouds that require images to be stored locally before instantiation (such as OpenStack), an image propagation component is needed to automatically deploy images to remote clouds. We are developing a service, called *Glint*, for managing images stored on multiple clouds, concentrating on OpenStack clouds and the Glance repository, but with a plug-able architecture to allow support for different cloud types [21]. Through a web browser interface, the user will identify clouds and be able to control the distribution of images.

Glint provides site management, image registration, credential management and image deployment. By design, Glint has no pre-established list of clouds. Instead the user must specify and provide their credentials for each cloud. The images to be distributed are uploaded by the user to Glint who then selects the target clouds for each image. Image transfers are multithreaded to reduce upload times and the user is notified once the transfers are completed.

The HTCondor-CloudScheduler system is currently used for Belle II MC production on six different clouds with more than 2500 running jobs including up to 480 jobs on Amazon EC2.

2.3. Pacific Northwest National Laboratory site

The Pacific Northwest National Laboratory (PNNL) is expected to be one of the larger centers for Belle II computing. Currently, PNNL has configured a static cloud (manually booted VM instances with an infinite lifetime) that are attached to the HTCondor pool. PNNL is investigating other solutions (including those described in this paper) for dynamically managing its resources and sharing them between different user groups.

3. Summary

Cloud computing centres are producing a significant fraction of the Belle II MC sample (see fig. 4). The number of clouds used is increasing with many of them being opportunistic private and commercial clouds. It is expected that the fraction of computing in Belle II using clouds will continue to grow as the technology matures and new sites migrate from traditional computing centers to clouds. In the short term, Belle II will start running multicore applications that will increase the utilization of clouds by reducing the memory footprint and making effective use of the software caches. In the future, the use of data federations and 100 gigabit/second networks will enable Belle II to run analysis applications on the clouds.

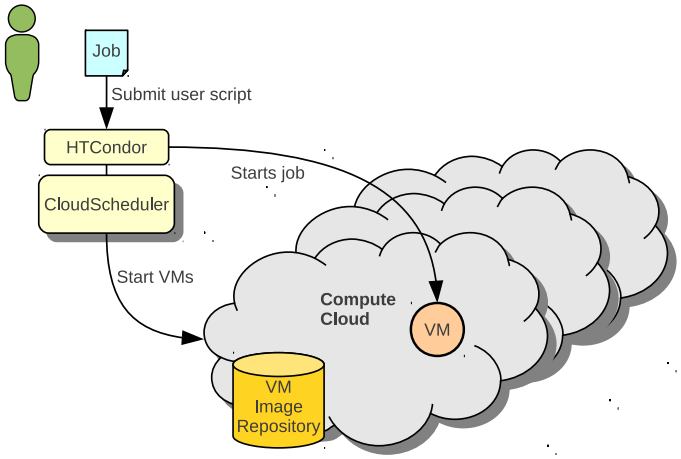


Figure 3. An overview of the architecture used for the system. A user prepares their VM image and a job script. The job script is submitted to the HTCondor job scheduler. CloudScheduler reads the job queue and makes a request to boot the user VM on one of the available clouds. Once the VM is booted, it attaches itself to the HTCondor pool and HTCondor assigns jobs to the VM image. When there are no more user jobs requiring that VM type, CloudScheduler makes a request to the proper cloud to shutdown the user VM.

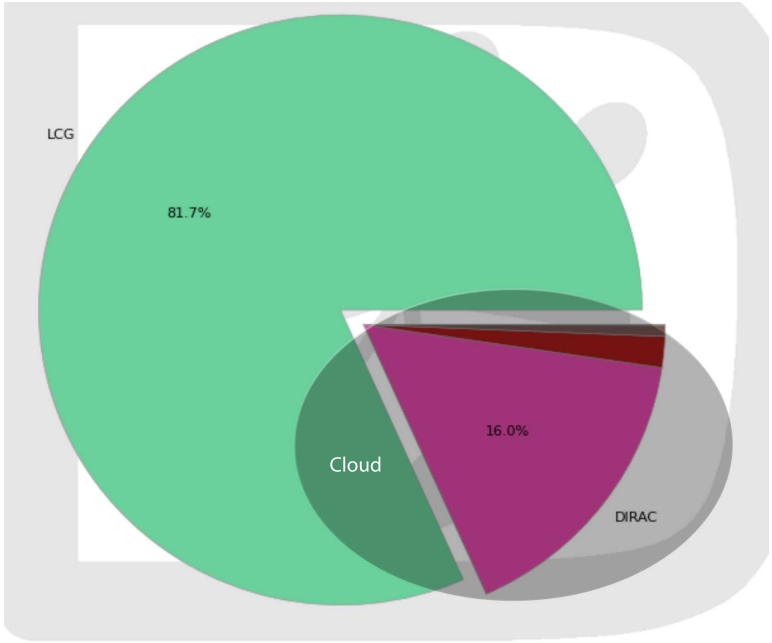


Figure 4. Pie chart showing the numbers of Belle II MC production jobs in March-April 2015. The LCG (WLCG) sites are traditional grid computing centers (green). Approximately 95% of the DIRAC jobs are generated on clouds (light red). The darker red slice are cloud jobs using VMDIRAC.

References

[1] Belle2 Collaboration. http://belle2.kek.jp
[2] BaBar Collaboration, Aubert B. *et. al.*. Nucl. Instrum. Meth. A479 (2002) 6.
[3] Belle Collaboration, Abashian A. *et. al.*, Nucl. Instrum. Meth. A479 (2002) 117.
[4] Hara T., *The Belle II Computing System*. To be published in the Proceedings of CHEP 2015.
[5] The World-wide LHC Computing Grid. http://wlcg.web.cern.ch.
[6] Casajus A. *et. al.* J. Phys. Conf. Ser. 219 (2010) 062049. doi:10.1088/1742-6596/219/6/062049.
[7] LHCb Collaboration. Alves A., *et. al.* JINST 3 (2008) S08005.
[8] Tsaregorodtsev A. J. Phys. Conf. Ser. 513 (2014) 032096. doi:10.1088/1742-6596/513/3/032096.
[9] Grzymkowski R. *Using VMDIRAC in Belle II*. To be published in the Proceedings of CHEP 2015.
[10] Sevior M. *et. al.* J. Phys. Conf. Ser. 219 (2009) 012003.
[11] Limosani A. *et. al.* J. Phys Conf. Ser. 513 (2014) 032058. doi:10.1088/1742-6596/513/3/032058
[12] Zhang S. *et. al.* J. Phys. Conf. Ser. 513 (2014) 032107. doi:10.1088/1742-6596/513/3/032107
[13] Pupper for IT Automation. http://info.puppetlabs.com/.
[14] Sobie R.J., *Distributed cloud computing in high energy physics* 2014 ACM SIGCOMM workshop on Distributed cloud computing.
[15] Gable I. *et. al. HEP cloud production using the CloudScheduler/HTCondor Architecture*. To be published in the Proceedings of CHEP 2015.
[16] Thain D., Tannenbaum T., and Livny M. Concurr. Comput. Pract. Exper. 17 (2005) 323.
[17] Armstrong P. *et. al. CloudScheduler: a resource manager for a distributed compute cloud*. arXiv:1007.0050v1 [cs.DC].
[18] CernVM software appliance. http://cernvm.cern.ch/portal/.
[19] Blomer J., Buncic P., and Fuhrmann T. *CernVM-FS: delivering scientific software to globally distributed computing resources.* Proceedings of the First International Workshop on Network-aware Data Management. 2011 doi:10.1145/2110217.2110225.
[20] Gable I. *et. al.* J. Phys.: Conf. Ser. 513 (2014) 032035. doi:10.1088/1742-6596/513/3/032035
[21] Berghaus F. *et. al. Glint: VM image distribution in a multi-cloud environment*. To be published in the Proceedings of CHEP 2015.