Journal of Physics: Conference Series **664** (2015) 062023 doi:10.1088/1742-6596/664/6/062023

Job monitoring on DIRAC for Belle II distributed computing

Yuji Kato[1], Kiyoshi Hayasaka[1], Takanori Hara[2], Hideki Miyake[2], Ikuo Ueda[2,3] on behalf of the Belle II computing group[4]

[1]Kobayashi-Maskawa Institute for the Origin of Particles and the Universe, Nagoya University, Chikusa-ku Furo-cho, Nagoya, Japan
[2]High Energy Accelerator Research Organization, 1-1, Oho, Tsukuba, Japan
[3]International Center for Elementary Particle Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

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

E-mail: kato@hepl.phys.nagoya-u.ac.jp

Abstract. We developed a monitoring system for Belle II distributed computing, which consists of active and passive methods. In this paper we describe the passive monitoring system, where information stored in the DIRAC database is processed and visualized. We divide the DIRAC workload management flow into steps and store characteristic variables which indicate issues. These variables are chosen carefully based on our experiences, then visualized. As a result, we are able to effectively detect issues. Finally, we discuss the future development for automating log analysis, notification of issues, and disabling problematic sites.

1. Introduction

Belle II is a next generation B-factory experiment at KEK in Japan. It will start physics running (without vertex detector) in 2017 and collect a data sample with an integrated luminosity of 50ab^{-1} in order to search for physics beyond the Standard Model. We will eventually need 1 MHS06 of CPU resources and 100 PB of storage for one set of raw data and 100 PB of disk storage for Monte Carlo and analysis data. In order to utilize these huge resources, the distributed computing model is a natural solution. We use DIRAC (Distributed Infrastructure with Remote Agent Control), which can handle heterogeneous computing resources such as grid, cloud and local cluster resources [1] for the software framework of our distributed computing system. The detail of the Belle II computing model and the development of the production system using DIRAC can be found in [2, 3].

The Monitoring system is important in order to maximize the availability of huge resources. We are developing a monitoring system for the Belle II computing system in two approaches: A passive method, where information stored in the DIRAC database is extracted and processed in order to detect issues easily; and an active method, where information is actively acquired, e.g., by submitting test jobs to each site. In this paper, the passive monitoring system is described, while the details of the active monitoring system are described in [4]. We use HappyFace [5],

as the platform for our monitoring system. It has a modular structure, development of plug-ins is rather simple, and several functionaries such as making history plots have already been prepared.

2. Workload management flow in DIRAC

Figure 1 shows the schematic view of the workload management flow in DIRAC. DIRAC utilizes a pilot job framework, which has several advantages in comparison to classical payload pushing scheduling mechanisms [6]. After a user or production system submits a payload job on DIRAC, payload jobs are stored in the task queue and following steps are performed:

(i) DIRAC asks how many slots exist in each computing element (CE).
(ii) If empty slots exist, DIRAC submits pilot jobs to the CE.
(iii) CE submits pilot jobs on the local batch system.
(iv) At the beginning of the pilot job a DIRAC client is installed on the worker node. This communicates with the DIRAC server and performs the sanity check.
(v) After the sanity check the worker node requests the DIRAC server to send a payload job and it is executed.

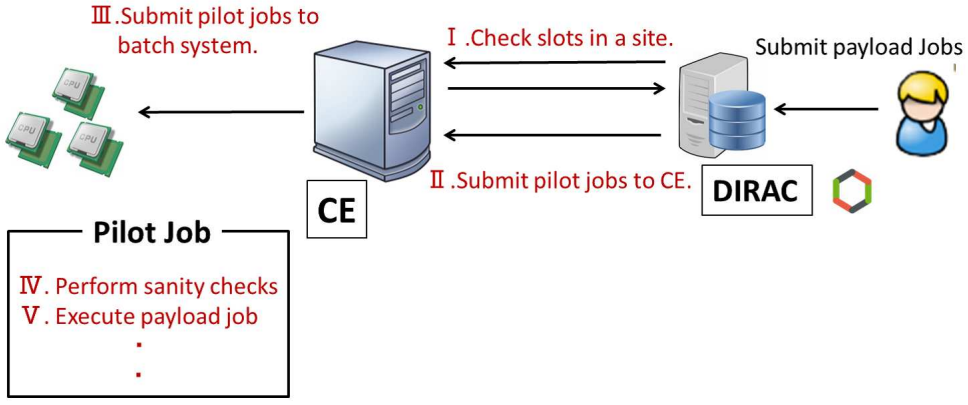

Figure 1. Schematic view of the workload management flow in DIRAC.

We developed a monitoring system which can detect issues in each step based on experiences during the Monte Carlo data production campaigns [2]. They are described in the following section.

3. Monitoring system

In order to detect issues during each step, plots are made which characterize the associated issues as follows:

(i) DIRAC sometimes counts the wrong number of pilot jobs running at a site. For example, a completed pilot job is reported as still running, preventing further jobs from being sent. Such 'ghost' pilot jobs can be characterized by a long sleeping time, defined as the time interval since the last status update. Figure 2 shows the sleeping time distribution together with the maximum allowed sleeping time for normal jobs (indicated by the red line) and its time dependence. CREAM CE often communicates an incorrect number of pilot jobs to DIRAC which results in these 'ghost' pilot jobs.

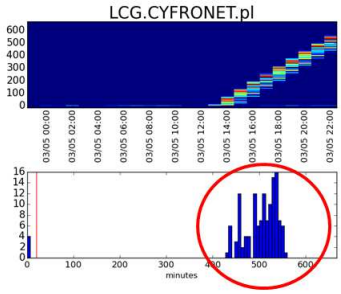

Figure 2. Distribution of pilot job's sleeping time. The top plot shows the time dependence of sleeping time, and the bottom plot shows its projection at the current time. The red line on the bottom plot indicates the maximum allowed sleeping time for normal pilot jobs. The red circle indicates the problematic pilot jobs.

(ii) The pilot job submission is performed by a DIRAC agent 'SiteDirector'. As the activity of the SiteDirector is not stored in the database, a new agent was developed to analyze the log file of the SiteDirector and store the information in the database, which is then visualized as plots. Figure 3 shows the time dependence of the pilot submission status for both one whole site and each individual CE at a site. A possible cause of the submission failures shown is a malfunctioning CE or the use of outdated CRLs.

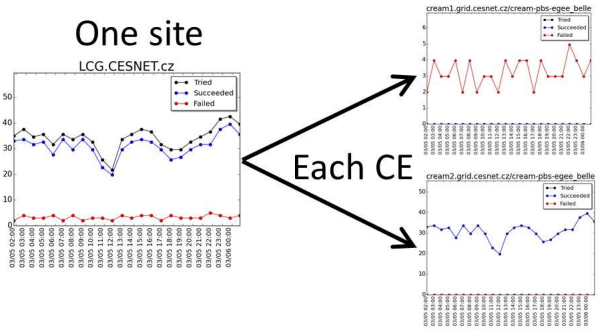

Figure 3. Time dependence of the pilot job submission status. Blue and red points show succeeded and failed submissions respectively. Left plot is for one site and right plots are for each CE.

(iii) Final status of the pilot job registers as 'aborted' when submission to the local batch system fails after the pilot job is submitted to the CE. Figure 4 shows the time dependence of the statuses of pilot jobs. The aborted pilot jobs are usually caused by malfunctioning local batch systems.

(iv) When the sanity check of the worker node fails, the pilot job finishes immediately without receiving payload jobs. This issue can be characterized by a 'short life time' of the pilot jobs. Figure 5 shows the pilot life time distribution with the minimum allowed pilot life time indicated by the red line. Data points below the red line indicate that the sanity check has failed. Possible causes for this include lack of disk space or failure to mount CVMFS (CernVM-FS) [7].

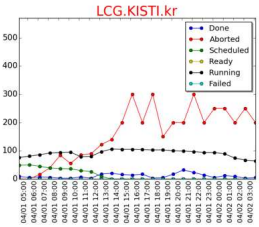

Figure 4. Time dependence of the status of pilot jobs. Blue points are normally finished ones and red points are abnormally finished (aborted) ones

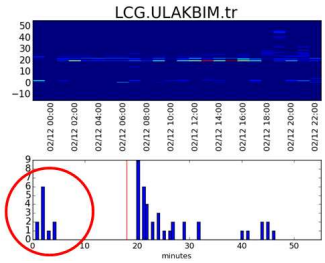

Figure 5. Distribution of the life time of the pilot jobs. Top plot shows the time dependence of the life time and bottom plot shows its projection. The red line indicates the minimum allowed pilot lifetime for normal pilot jobs and red circle indicates problematic pilot jobs.

(v) Finally, execution of the payload jobs can sometimes fail, which is characterized by payload jobs with a final status of "Failed". Figure 6 shows the job efficiency defined by the number of normally finished jobs divided by the total number of finished jobs. Low efficiencies for all the sites in the same time period indicates an issue with a central service such as DIRAC or AMGA, which is a meta data service. Low efficiency for only one site means the site has problems such as failure to download input/output data due to the bad network connection.

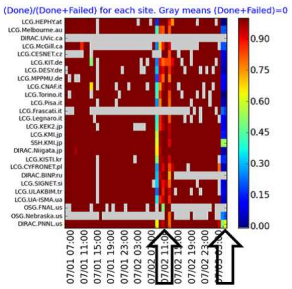

Figure 6. Payload job efficiency as function of time for each site. Arrows indicate the time period for low efficiencies for all the sites.

4. Automatize actions

After identifying the problem, we need to perform the following actions:

(i) Analyze the log file in order to identify the reason for the problem.

(ii) Notify the sites of issues via an issue tracking system such as GGUS [8].

(iii) Disable job submission for problematic sites or re-enable after the issue is fixed using DIRAC Resource Status System (RSS) [9].

In order to make these actions more efficient, we are working on automating the process. A separate procedure must be created for each step where issues can occur, as the characteristic variables and log file locations differ based on the issue. Figure 7 shows an example of the procedure when the sanity check on a worker node fails.

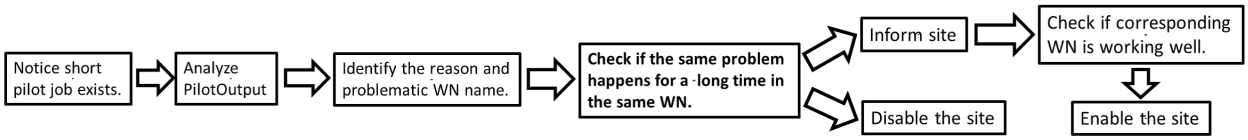

Figure 7. Procedure for action when the sanity check on a worker node fails.

5. Conclusion

In this paper, we present the passive monitoring system for the Belle II computing system. We divide the workload management flow into steps and visualize the characteristic variables to detect issues. These characteristic variables such as "sleeping time", "life time" and "status" of pilot jobs and "status" of payload jobs are carefully selected based on our experiences during the Monte Carlo data production campaigns. Also, a new DIRAC agent to store the submission information has been developed. In order to maximize the availability of huge resources, we are now working on the automation of the responses to detected issues: analyzing log file, notifying sites, and enabling or disabling problematic sites.

6. Acknowledgments

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.

References

[1] http://diracgrid.org/
[2] T. Hara et al, Computing at the Belle-II experiment, Proceedings of the CHEP 2015 conference.
[3] H. Miyake et al, Belle II production system, Proceedings of the CHEP 2015 conference.
[4] K. Hayasaka et al, Monitoring system for the Belle II distributed computing, Proceedings of the CHEP 2015 conference.
[5] https://ekptrac.physik.uni-karlsruhe.de/trac/HappyFace.
[6] A. Casajus, DIRAC Pilot Framework and the DIRAC Workload Management System, Proceedings of the CHEP 2009 conference.
[7] http://cernvm.cern.ch/portal/.
[8] https://ggus.eu/.
[9] F. Stagni et al, A policy system for Grid Management and Monitoring, Proceedings of the CHEP 2010 conference.