BelleII@home: Integrate volunteer computing resources into DIRAC in a secure way

Wenjing Wu¹, Takanori Hara², Hideki Miyake², Ikuo Ueda², Wenxiao Kan¹ and Phillip Urquijo³

¹IHEP, 19B Yuquan Road, Beijing, 100049 China
²KEK, 1-1 Oho, Tsukuba, Ibaraki, 305-0801 Japan
³University of Melbourne, Parkville VIC 3010, Australia

E-mail: wuwj@ihep.ac.cn

Abstract. The exploitation of volunteer computing resources has become a popular practice in the HEP computing community as the huge amount of potential computing power it provides. In the recent HEP experiments, the grid middleware has been used to organize the services and the resources, however it relies heavily on the X.509 authentication, which is contradictory to the untrusted feature of volunteer computing resources, therefore one big challenge to utilize the volunteer computing resources is how to integrate them into the grid middleware in a secure way. The DIRAC interware which is commonly used as the major component of the grid computing infrastructure for several HEP experiments proposes an even bigger challenge to this paradox as its pilot is more closely coupled with operations requiring the X.509 authentication compared to the implementations of pilot in its peer grid interware. The Belle II experiment is a B-factory experiment at KEK, and it uses DIRAC for its distributed computing. In the project of BelleII@home, in order to integrate the volunteer computing resources into the Belle II distributed computing platform in a secure way, we adopted a new approach which detaches the payload running from the Belle II DIRAC pilot which is a customized pilot pulling and processing jobs from the Belle II distributed computing platform, so that the payload can run on volunteer computers without requiring any X.509 authentication. In this approach we developed a gateway service running on a trusted server which handles all the operations requiring the X.509 authentication. So far, we have developed and deployed the prototype of BelleII@home, and tested its full workflow which proves the feasibility of this approach. This approach can also be applied on HPC systems whose work nodes do not have outbound connectivity to interact with the DIRAC system in general.

1. Introduction
The Belle II [1] is the next-generation flavor factory experiment at the SuperKEKB accelerator in Japan, aiming to collect a 50 times amount of data than Belle by the end of 2024. The first physics run will take place in early 2018. Such a huge amount of data allows us to explore the new physics possibilities through a large variety of analyses with statistical accuracy and to deepen understanding of nature.

The Belle II is a worldwide collaboration of about 700 researchers working in 23 countries and region. Therefore, we adopted a distributed computing model, in which the sites are categorized based on its functionality, such like Raw Data Center, Regional Data Center and MC Production site. We
also utilizes existing technologies. We chose DIRAC [2][3][4] as a workload and data management system and AMGA as a metadata service. For the file replica catalog, we use LFC (LCG File Catalog). In particular, DIRAC provides us an interoperability of heterogeneous computing systems such as grids with different middleware, academic/commercial clouds and local clusters. In this paper, we will report a possibility to use the volunteer computing resources in the Belle II distributed computing.

2. Similar projects and Services

2.1. Other HEP volunteer computing projects

Applying the volunteer computing resources to the computing of High Energy Physics has been a long term effort, and as easy as 2004, CERN launched the first volunteer computing project LHC@home [5][6][7] which runs simulation jobs for theoretical physics simulation. Afterwards there have been a few volunteer computing projects being applied to the computing of the big HEP experiments and some of them are in production, such as ATLAS@home[8]. Some of the projects such as CMS@home[9], Beauty@home, LHCb@home are still in their development phase, and are not completely released to the public yet. One of common efforts of these projects is to integrate the untrustable volunteer computing resources into the X.509 authentication based grid computing platform for these experiments, so the challenges vary depending on the implementation of the grid services.

For LHCb@home, as it uses DIRAC interway, and in the DIRAC pilot implementation, the payload running is closely coupled with all the other operations which require the X.509 authentication. So in the prototype of LHCb@home, a use proxy was dispatched with the job to the volunteer computer for the pilot running. This approach is considered to be risky, so the project was only run inside the collaboration and not released to the public.

For ATLAS@home, as its PanDA [10] system is already integrated with the ACT (ARC Control Tower) service [11] and the ARC CE service [12], and the combination of the ATLAS pilot [13] implementation and the ARC CE features allow the separation of the payload running from all other pilot functions, so the solution is pretty straightforward: using the ARC CE as the bridge which caches the jobs from PanDA and forwards the payload running to the BOINC [14] server, and handles all the data operations of this job.

2.2. ARC CE

ARC CE is the computing element of the NorduGrid [15] middleware, and it includes a set of services and modules: authorization/access control, job handling, job files uploading/downloading, information handling and accounting.

As shown in Figure 1, the major advantage of ARC CE compared to its peer services is its data caching mechanism which enables itself for data intensive jobs. Unlike other CE services which usually forward the entire job to its LRMS (Local Resource Management Service), ARC CE could be configured to download and cache the input data of the job from the grid SE (Storage Element) service with the X.509 credential, and then copy the input data to the LRMS with local protocols or tools such as scp and cp. And this approach also applies to the output data of the job, in which case the output data will be firstly copied to the ARC CE server via local protocols or tools, then from ARC CE the output data can be uploaded to the SE server with the X.509 credentials. With this approach, all the authentication can be done on the ARC CE server, and it requires no credential being placed on both the LRMS and its computer nodes. In this scenario, the LRMS and its computer nodes only handle the payload running, and the data transfer are all operated with local tools or protocols. However, in order to support this feature, the pilot needs to support the separation of payload running and data handling, also another service ACT (Arc Control Tower) is used to bridge the ARC CE and the Grid Job Portal such as PanDA in the case of ATLAS experiment. And currently the implementation of ACT is very ATLAS PanDA specific, for example, it fetches information from the AGIS server (ATLAS specific
information system), and talks to the PanDA server with its specific APIs. It is still complicated to make the ACT into a generic service.

![Figure 1 Architecture of ARC CE](image)

3. BelleII@home project

3.1. System architecture

The goal of BelleII@home is to harness the very considerable amount of volunteer computing resources from the public and run the CPU intensive BelleII simulation jobs on the platform. As the BelleII distributed computing platform was already built upon the DIRAC system, and has been used as the major source for BelleII computing tasks, so in order to make an unified interface for the operation of BelleII computing tasks, the BelleII@home platform needs to be integrated into the BelleII distributed computing platform, more specifically, the DIRAC interware.

As shown in Figure 2, in the DIRAC system, a BOINC site directory — a DIRAC plugin which can communicate with the backend batch system (such as PBS and Condor) or cloud computing interface (such as OpenStack and EC2) for job submission and other operations if needed — is needed for forwarding the BelleII jobs to the BOINC server, then the BOINC server wraps the BelleII job, usually includes packing the job with the BOINC vboxwrapper (a program which controls the operations of virtual machines on the volunteer hosts) and its configuration files.

3.2. BelleII DIRAC Pilot

The BelleII pilot is a customized version of the DIRAC pilot, and it is being dispatched to and executed on the computer node. The BelleII pilot handles operations such as fetching one or more jobs from the DIRAC server, downloading and uploading files, starting watchdog services which monitor the job status, and updating the job status to the DIRAC server. As shown in Figure 3, the BelleII job pilot (DIRAC_pilotwrapper.py) includes several layers of wrappers and scripts, and some of the scripts involve functions requiring the X.509 authentication: check waiting jobs, download input sandbox files, upload output sandbox/output files, and start watchdog and job-report. As shown in the diagram, the inner script is invoked and wrapped by the outer script. In the X.509 authentication, a valid user proxy with an associated lifetime needs to be placed on the work node which runs the job.
The user proxy serves as the authenticator of the user who submitted the jobs, and can be used to authenticate itself to interact with all the grid services.

3.3. Challenges and solutions
As the volunteer computers are in the control of volunteers’ hands and not trustable, hence the user proxy which is required by the X.509 authentication can’t be stored on the volunteer computers in any forms because malicious users can abuse the proxy with operations such as emptying the job queues on the DIRAC server and uploading junk data to and flooding the Grid Storage Element. With this constraint, all the X.509 authentication required operations need to be token out from the BelleII pilot code and executed on a secure machine which can store the user proxy. In other words, only the payload running can be executed on the volunteer computers.

4. The Gateway Service
4.1. Components and workflow
As discussed in section 2, in order to integrate the BelleII@home resources into the BelleII DIRAC system in a secure way, the payload running needs to be separated from the BelleII pilot code, and all the operations which requires the X.509 authentication need to be placed on a secure server which can store the user proxy. Therefore all the authentication related operations are taken out from the BelleII DIRAC code, and regrouped into different daemons, running as a gateway service on a secure server. As shown in Figure 4, three daemons are running on the gateway server: Job Generator, Job Submitter and Job Assimilator; Job Generator does the “job caching”, including fetches jobs from the DIRAC BOINC site, downloads the input sandbox and input files of the job, install and configure the DIRAC client, and setup the job running environment; Job Submitter submits the job prepared by the Job Generator to the BOINC server; Job Assimilator takes the valid result of the BOINC job, uploads the output files and output sandbox, and updates the job status to the DIRAC server. The workflow marked in Figure 4 includes:
1. The Job Generator checks the DIRAC BOINC site for jobs in “waiting” status.
2. The Job Generator fetches and caches the job from the DIRAC BOINC site.
3. The Job Submitter submits the job to the BOINC Job Queue.
4. The BOINC client requests and gets a job from the BOINC server, starts a virtual machine to run the job payload (no credential required), upon finishing the payload successfully, it compresses the
5. The compressed work directory is transferred back to the BOINC Server.
6. The Validator validates the results, and marks a successful result.
7. The Assimilator processes the successful result:
   a) Upload the root files to the Grid Storage Element (SE).
   b) Upload output sandbox and update job status to the DIRAC Server.

The BelleII pilot code is decoupled, and reorganized to implement the daemons on the gateway server.

4.2. Decoupled BelleII DIRAC pilot structure
The original BelleII DIRAC pilot functions are split into the Job Generator, Job Assimilator and
payload. Both the Job Generator and Job Assimilator require the X.509 credentials, they serve as the gateway between the BOINC server and the DIRAC server, and the gateway server runs on a trusted machine with a valid user proxy stored on it.

As shown in Figure 5, all the scripts marked in red indicate they are modified from the original scripts compared to Figure 3. The modified `DIRAC_pilot.py` which runs on the gateway server will install and configure the DIRAC pilot code, download the input sandbox files and input files, update the job status to the DIRAC server, create a few job wrapper scripts and compress all the input files of the job, then exit.

The job wrapper scripts include:

- S1: Download the output sandbox files
- S2: setup the runtime environment on the virtual machine
- S3: start the job payload on the virtual machine
- S4: upload the output files to the grid storage element
- S5: upload the output sandbox files to the DIRAC server
- S6: update the job status to the DIRAC server and finalize the job

These scripts are placed into different daemons, and run on either the gateway server or the virtual machine.

![Decoupled BelleII pilot structure](image)

**Figure 5** Decoupled BelleII pilot structure

5. Identified performance issues

The prototype is built upon the above system architecture design, and test jobs are sent to test the feasibility of the prototype. Some bottlenecks of the performance are found with the test jobs. For example, on the gateway server, each job needs 2 times of compression and decompression operations for its input and output files—one compression of the input files and one decompression for the output files—and usually this size is around 40MB, so it increases the IO load on the server when it handles a large scale of job submission. Also currently the implementation of the Job Generator is based on the DIRAC site director, but use the DIRAC client commands to check the “waiting” jobs in the queue, then pull the jobs, and the daemon scans periodically to create the jobs and forward them to the
BOINC server. Ideally, this service should be implemented as all the other site director services such as PBS and Condor on the DIRAC server. Also, in order to create the pure payload job for the virtual machine, part of the pilot code is run on the gateway server which includes functions of downloading and installing all the DIRAC client tools, configuration, downloading jobs from the DIRAC server, this process takes around 2 minutes for each job and creates a compressed file of around 40MB in size, so the preprocessing of each BOINC job is pretty heavy. As most of the functions of the preprocessing of each job is standard, it is possible to avoid the repetitive operations for each job, which should significantly reduce the preprocessing time.

6. Conclusion and summary
The project of BelleII@home aims at harnessing the widely available volunteer computing resources and integrate them into its current distributed computing system which is built upon the DIRAC interway. However, due to the implementation of the DIRAC pilot, there is no straightforward solution to decouple all the operations requiring the X.509 authentication from the BelleII job payload running inside the pilot code, and placing a valid user proxy on any volunteer computer is considered to be risky for undermining the DIRAC server and other grid services. In order to separate the payload running from the BelleII DIRAC pilot, we tailored and modified the BelleII DIRAC pilot, taking out all the functions which requires the X.509 authentication, and running them as daemons on a trusted gateway server. The prototype uses a simple implementation as a proof of concept, and it proves the feasibility of this approach, however several performance issues such as long preprocessing time and heavy IO load for each job are also identified and they should be the focus of future optimizations.

7. Acknowledgements
This project is supported and sponsored by the KEK short term fellow program and also by the Chinese NSF grants "Research on fine grained Event Service for the BESIII offline software and its scheduling mechanism (No.11675201)" and "Research on BESIII offline software and scheduling mechanism on desktop grid (No.11405195)". During the design of the system, we also get valuable comments and suggestions from both the DIRAC development team and the LHCb@home developers.

References
[1] The BelleII experiment: https://www.belle2.org/
[2] Tsaregorodtsev A. DIRAC distributed computing services[C]//Journal of Physics: Conference Series. IOP Publishing, 2014, 513(3): 032096. (DIRAC)
[3] Van Herwijnen E, Closier J, Frank M, et al. Dirac-distributed infrastructure with remote agent control[C]/2003 Conference for Computing in High-Energy and Nuclear Physics (CHEP 03). 2003. (DIRAC)
[4] Fifield T, Carmona A, Casajús A, et al. Integration of cloud, grid and local cluster resources with DIRAC[C]//Journal of Physics: Conference Series. IOP Publishing, 2011, 331(6): 062009. (DIRAC)
[5] Sanchez C A, Blomer J, Buncic P, et al. Volunteer clouds and citizen cyberscience for LHC physics[C]//Journal of Physics: Conference Series. IOP Publishing, 2011, 331(6): 062022. (LHC@home)
[6] Herr W, McIntosh E, Schmidt F and Kalchev D 2006 Conf.Proc. C060626 526-528 (LHC@home)
[7] Høimyr N, Marquina M, Asp T, et al. Towards a Production Volunteer Computing Infrastructure for HEP[C]//Journal of Physics: Conference Series. IOP Publishing, 2015, 664(2): 022023. (LHC@home)
[8] Cameron D. ATLAS@ Home: Harnessing Volunteer Computing for HEP[R]. ATL-COM-SOFT-2015-018, 2015. (ATLAS@home)
[9] Field L, Borras H, Spiga D, et al. CMS@ home: Enabling Volunteer Computing Usage for CMS[C]//Journal of Physics: Conference Series. IOP Publishing, 2015, 664(2): 022017.
(CMS@home)

[10] Maeno T 2008 PanDA: distributed production and distributed analysis system for ATLAS Journal of Physics: Conference Series vol 119 (IOP Publishing) p 062036 (PanDA)

[11] Filipicic A 2011 arcControlTower: the System for Atlas Production and Analysis on ARC J. Phys.: Conf. Ser. vol 331 p 072013

[12] Ellert M, Grnager M, Konstantinov A et al. 2007 Future Gener. Comput. Syst. 23 219-240 ISSN 0167-739X (ARC CE)

[13] Nilsson P, Caballero J, De K, et al. The ATLAS PanDA pilot in operation[C]//Journal of Physics: Conference Series. IOP Publishing, 2011, 331(6): 062040.(ATLAS pilot)

[14] Anderson D 2004 BOINC: a system for public-resource computing and storage Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, GRID 04 (IEEE Computer Society) pp 4-10 (BOINC)

[15] The NorduGrid Collaboration web site URL http://www.nordugrid.org