

HEP@home : A Volunteer Computing Project to Run Fast Simulation with Delphes for CEPC

Ran Du^{1,*}, Jingyan Shi^{1,**}, and Gang Li^{1,***}

¹Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China, 100049

Abstract. Delphes is a C++ framework to perform a fast multipurpose detector response simulation. The Circular Electron Positron Collider (CEPC) experiment runs fast simulation with a modified Delphes based on its own scientific objectives. The CEPC fast simulation with Delphes is a High Throughput Computing (HTC) application with small input and output files. Besides, to compile and run Delphes, only ROOT software is necessary. Therefore, all these features make it appropriate to run CEPC fast simulation as a Volunteer Computing application. As a result, a BOINC project named HEP@home is developed to run fast simulation with Delphes for CEPC. This paper describes the internal structure of the project, pre and post data operations, and its development status.

1 Introduction

Delphes is a C++ framework to perform a fast multipurpose detector response simulation. The Circular Electron Positron Collider (CEPC) experiment runs fast simulation with a modified Delphes based on its own scientific objectives. To get more computing resources and make CEPC experiment better known by the public, a BOINC project HEP@home is developed to run fast simulation with Delphes as its first application. This paper is organized in the following sections. Section 2 presents related works. The design and implementation details of the project components can be found in section 3. Section 4 and section 5 describe the current status of the project and conclude our work respectively. And our acknowledgement comes in section 6.

2 Related Works

BOINC[1][2] is a middleware system for Volunteer Computing(VC). To run applications of multiple discipline on volunteer computers, BOINC provides a framework and components for VC projects, and these components should be customized based on requirements of target applications. The history of VC projects based on BOINC is longer than 20 years. As a result, multiple research topics on the framework and components of BOINC are developed including workunit scheduling algorithms[3][4], data distribution[5][6], host and volunteer study[7][8][9], validator study[10][10] and integration with grid or other systems[11][12][13]. Besides, another important research topic is how to make a project

*e-mail: duran@ihep.ac.cn

**e-mail: shijy@ihep.ac.cn

***e-mail: ligang@ihep.ac.cn

based on a specific application. There are tens of BOINC projects set up on different area, and here we list a few representative and influential projects like the SETI@home project to analyze radio signals from space[14], the Einstein@Home to search for periodic gravitational waves[15], the Rosetta@home project for protein structure prediction[16], the LHC@home project for massive numerical simulations of accelerator and high energy physics events[17][18], and the Wilelife@home project to analyze avian nesting videos[19]. Compared with the mentioned projects, our project HEP@home is set up to run fast simulation with Delphes software for CEPC. Because VC projects are set up based on BOINC, these projects normally have a similar system framework, but different implementation details of project components.

3 Design and Implementation

According to the system design concept of BOINC and the running mechanism of Delphes App, five components are developed for the HEP@home project. These five parts are running in the BOINC server and volunteer computer separately. Among which only the Delphes App is running in the volunteer computer. And the rest four components including work generator, validator, assimilator and project web site are located in the BOINC server. The architecture of the HEP@home project is shown in Figure 1. We will describe the design and implementation details in the following subsections.

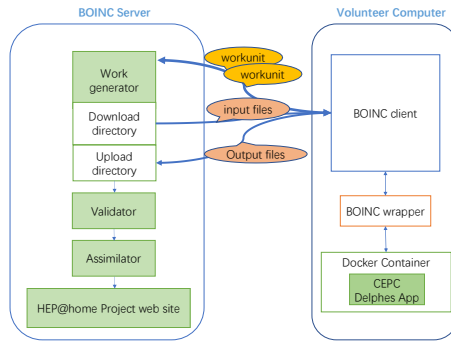


Figure 1. Architecture of the HEP@home project.

3.1 Delphes App

The Delphes App is used to run CEPC fast simulation which is a High Throughput Computing (HTC) application with small input and output files. Besides, to compile and run Delphes, only ROOT software is dependent, which makes it appropriate to run Delphes as a volunteer computing application.

To run Delphes App on Windows, a customized Docker image is built and uploaded to the Docker Hub as shown in Figure 2. Besides, to make high availability to campus and domestic volunteers, the same docker image is also uploaded into two Docker registries of IHEP and Alibaba. As long as any Docker registry of the mentioned is available, Delphes App could be run in the volunteer computers.

The docker image of Delphes App contains Delphes and ROOT software. The BOINC wrapper is used to call a .bat script and a PowerShell script in order, and the input files will be

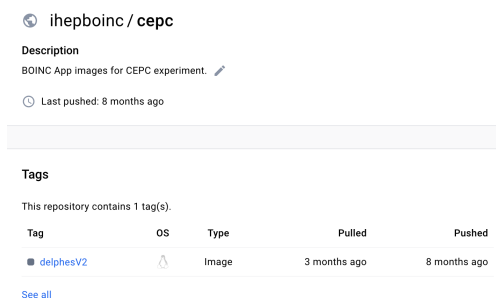


Figure 2. Docker image of the Delphes App to run on Windows volunteer computers.

passed as parameters to start a Delphes container with the image. If everything goes well, a ROOT file will be generated and copied back to the host from the container, and the container will be stopped when copy finished. As a result, The BOINC client will upload the generated ROOT file back to the BOINC server for validation and assimilation.

3.2 Work Generator

The Work generator is developed to submit workunits in batches. Each workunit of Delphes App has one .stdhep[20] file as input. All the .stdhep input files are located in a Lustre filesystem with three level directory hierarchy. To make it easier to understand, these three level directories are called task, subtask and input from top to bottom. To make workunit submission efficient, a MariaDB database, delphes_task_db, is designed to save relationships among tasks, subtasks and input files, the database diagram is provided in Figure 3.

Two status flags are maintained by work generator, one is file status, the other is workunit completion status. File status indicates whether the input file is available, and is updated during the task-subtask-input data insertion. While the completion status indicates if the workunit which takes the file as input is completed, so that work generator will choose input files available and uncompleted by default in the next run.

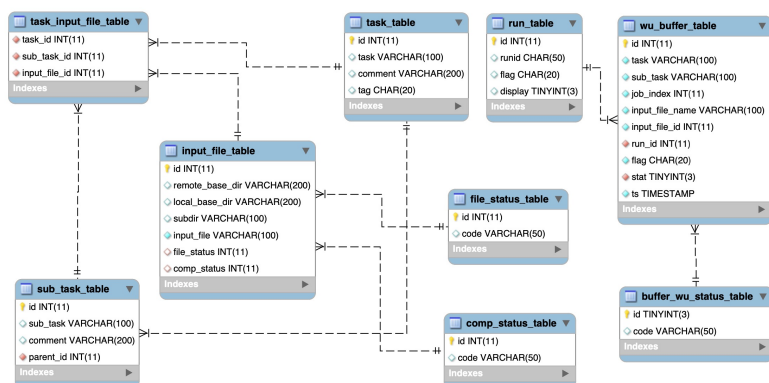


Figure 3. Diagram of delphes_task_db for the work generator.

During workunits submission, there will be a lot of input files copied to the download directory, meantime, many output files will be uploaded to the BOINC server. If too many workunits are submitted, it will cause heavy read/write load to the local disk. To make the BOINC server stable, workunit submission will be made in control, and if available disk space reached the predefined threshold, workunit submission requests will be stopped and stored in the `wu_buffer_table`. Next time when local disk available, submission will be recovered from the `wu_buffer_table`. The components of the work generator are shown in Figure 4.

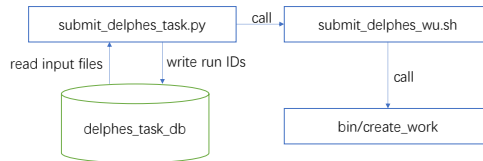


Figure 4. Components of the work generator.

3.3 Validator

The validator is used to check if the output files uploaded are valid based on the requirements of Delphes App. Validating metrics are number of events, number of particles, momentum resolution and energy resolution. The output files are valid only when all the metric tests passed.

We use the `script_validator` tool, provided by BOINC, to validate the ROOT output files. The `script_validator` will call a python and a shell script in sequence, and three C programs will be called to validate all the mentioned metrics. The components of the validator are shown in Figure 5.

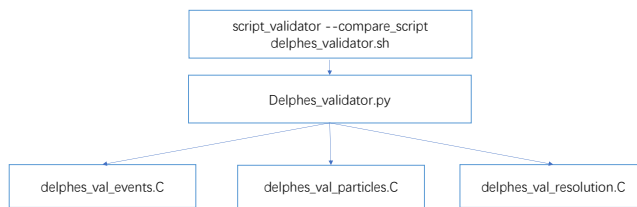


Figure 5. Components of the validator.

3.4 Assimilator

If output ROOT files are valid, the assimilator will be started to finish post operations. Possible post operations could be another procedure to handle these validated files, or save them to another location, etc. In our case, the assimilator for the Delphes App will analyze the ROOT files and generate image files both in EPS and PNG format. Later, generated image files will be saved in a database named `hep_assimi_db`. These images in PNG format could be displayed later on the project website.

Besides, the ROOT files uploaded from volunteer computers will be saved in a Lustre file system, and the metadata about the ROOT files will be saved in the `hep_assimi_db` as well.

So that physicists of CEPC can get the ROOT files if they want to do more analysis. When the assimilation done, the completion status of input files in the delphes_task_db will be updated accordingly. The components of the assimilator are shown in Figure 6.

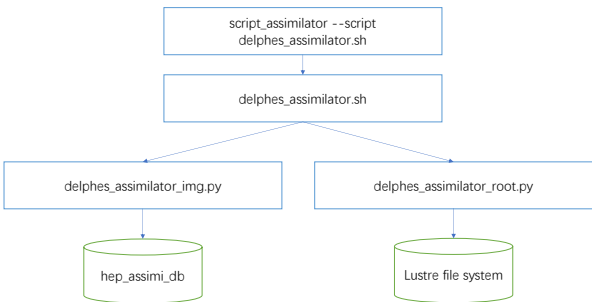


Figure 6. Components of the assimilator.

3.5 Project website

BOINC provides a web site framework for each project in PHP. The project web site is used to attract volunteers and display project status. The master URL of HEP@home project is: <https://ihepboinc.ihep.ac.cn/hep/>.

We customized the web site based on our requirements. There are two major modifications made. One is the document, the other is the output display. Because we use Docker not a virtual machine to run Delphes App, we provide documents about how to install and run Docker on Windows step by step. As to the output display, we mentioned earlier that the assimilator will generate image files in PNG format and save in a database, and these PNG images are used to display output results on our web site as shown in Figure 7.

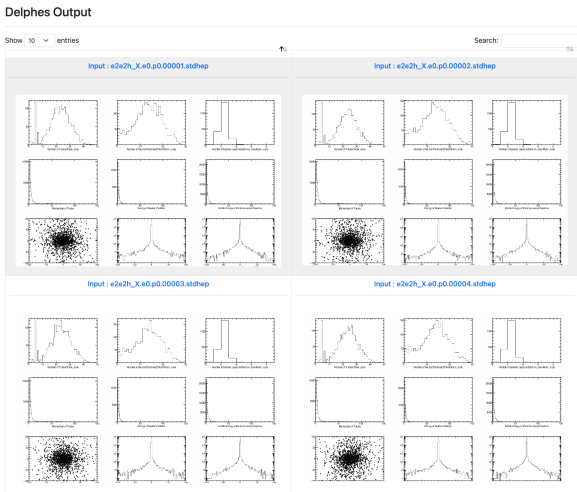


Figure 7. Output images on the project web site of HEP@home.

4 Project Status

The HEP@home project is young. Before we started to submit production workunits, we set up an experimental project for debug. All the development and test including image file test, work generator, validator and assimilator are done in the debug project first, and then published in the HEP@home project.

Considering many volunteers are adopting Windows as their host operating system, we made a lot of efforts to make Delphes App run on Windows. By far, we have tens of Windows volunteers who have about 27000 workunits done on their computers.

Later, we will publish new Delphes App to run on Linux and MacOS systems. Again, these apps will be tested and debugged in our experimental project before production. Besides, we are applying to join the BOINC official project list, so that more volunteers will be attracted and join our HEP@home project.

5 Conclusion

The volunteer computing project HEP@home is created to run fast simulation with Delphes for the CEPC experiment. To run Delphes within HEP@home project, five components are developed based on BOINC. We present the design and implementation details about these five components which are the Delphes App, the work generator, the validator, the assimilator and the project web site. By far, about 27000 workunits of the Delphes App are run successfully on volunteer Windows computers, and we are going to publish more App versions for the platform of Linux and MacOS, so that more computing power will be contributed to CEPC.

6 Acknowledgement

HEP@home project is developed with the support of the following projects: Chinese National Natural Science Foundation(No. 12105303, No. 12175255) and Youth Innovation Promotion Association of Chinese Academy of Sciences(No. 2020017).

References

- [1] D. Anderson, *BOINC: a system for public-resource computing and storage*, in *Fifth IEEE/ACM International Workshop on Grid Computing* (2004), pp. 4–10
- [2] D.P. Anderson, *Journal of Grid Computing* **18**, 99 (2020)
- [3] D.P. Anderson, *Emulating volunteer computing scheduling policies*, in *2011 IEEE international symposium on parallel and distributed processing workshops and Phd forum* (IEEE, 2011), pp. 1839–1846
- [4] A.L. Bazinet, M.P. Cummings, *Journal of Grid Computing* **14**, 429 (2016)
- [5] B. Javadi, B. Zhang, M. Taufer, *Bandwidth modeling in large distributed systems for big data applications*, in *2014 15th International Conference on Parallel and Distributed Computing, Applications and Technologies* (IEEE, 2014), pp. 21–27
- [6] S. Alonso-Monsalve, F. García-Carballeira, A. Calderón, *Concurrency and Computation: Practice and Experience* **29**, e4198 (2017)
- [7] D. Kondo, A. Andrzejak, D.P. Anderson, *On correlated availability in internet-distributed systems*, in *2008 9th IEEE/ACM International Conference on Grid Computing* (IEEE, 2008), pp. 276–283

- [8] E.M. Heien, D. Kondo, D.P. Anderson, *Correlated resource models of internet end hosts*, in *2011 31st International Conference on Distributed Computing Systems* (IEEE, 2011), pp. 278–287
- [9] O. Nov, O. Arazy, D. Anderson, *ScientistsHome and in the Backyard: Understanding the Motivations of Contributors to Digital Citizen Science* (SSRN, 2014)
- [10] M. Taufer, D. Anderson, P. Cicotti, C. Brooks, *Homogeneous redundancy: a technique to ensure integrity of molecular simulation results using public computing*, in *19th IEEE International Parallel and Distributed Processing Symposium* (IEEE, 2005), pp. 9–pp
- [11] D.S. Myers, A.L. Bazinet, M.P. Cummings, *Expanding the reach of grid computing: combining globus-and boinc-based systems*. (2007)
- [12] J.A. López-Pérez, Ph.D. thesis, Valencia U. (2008)
- [13] A. Marosi, J. Kovács, P. Kacsuk, *Future Generation Computer Systems* **29**, 1442 (2013)
- [14] D.P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, D. Werthimer, *Communications of the ACM* **45**, 56 (2002)
- [15] B.P. Abbott, R. Abbott, R. Adhikari, P. Ajith, B. Allen, G. Allen, R. Amin, S. Anderson, W. Anderson, M. Arain et al., *Physical review d* **80**, 042003 (2009)
- [16] R. Das, B. Qian, S. Raman, R. Vernon, J. Thompson, P. Bradley, S. Khare, M.D. Tyka, D. Bhat, D. Chivian et al., *Proteins: Structure, Function, and Bioinformatics* **69**, 118 (2007)
- [17] M. Giovannozzi, P. Skands, I. Zacharov, P. Jones, A. Harutyunyan, N. Høimyr, E. McIntosh, B. Segal, F. Grey, L. Rivkin et al., *LHC@ HOME: A volunteer computing system for massive numerical simulations of beam dynamics and high energy physics events*, in *Conf. Proc.* (2012), Vol. 1205201, p. MOPPD061
- [18] J. Barranco, Y. Cai, D. Cameron, M. Crouch, R.D. Maria, L. Field, M. Giovannozzi, P. Hermes, N. Høimyr, D. Kaltchev et al., *Open Engineering* **7**, 379 (2017)
- [19] T. Desell, R. Bergman, K. Goehner, R. Marsh, R. VanderClute, S. Ellis-Felege, *Wildlife@ Home: Combining crowd sourcing and volunteer computing to analyze avian nesting video*, in *2013 IEEE 9th International Conference on e-Science* (IEEE, 2013), pp. 107–115
- [20] M. Selvaggi, *J. Phys. Conf. Ser.* **523**, 012033 (2014)