

Distributed Computing for the Project 8 Experiment

Malachi Schram^{1*}, *Mathew Thomas*¹, *Kevin Fox*¹, *Benjamin LaRoque*¹, *Brent VanDevender*¹, *Noah Oblath*¹, and *David Cowley*¹
(On behalf of the Project 8 Collaboration)

¹Pacific Northwest National Laboratory

Abstract. The Project 8 collaboration aims to measure the absolute neutrino mass or improve on the current limit by measuring the tritium beta decay electron spectrum. We present the current distributed computing model for the Project 8 experiment. Project 8 is in its second phase of data taking with a near continuous data rate of 1Gbps. The current computing model uses DIRAC (Distributed Infrastructure with Remote Agent Control) for its workflow and data management. A detailed meta-data assignment using the DIRAC File Catalog is used to automate raw data transfers and subsequent stages of data processing. The DIRAC system is deployed on containers managed using a Kubernetes cluster to provide a scalable infrastructure. A modified DIRAC Site Director provides the ability to submit jobs using Singularity on opportunistic High-Performance Computing (HPC) sites.

1 Introduction

The goal of the Project 8 experiment [1] is to measure the absolute neutrino mass using tritium beta decays. The approach taken by the collaboration is to make this measurement using a new method of electron spectroscopy, Cyclotron Radiation Emission Spectroscopy (CRES). In order to demonstrate the new method and achieve the overall goal, the experiment is scheduled to have 4 phases each demonstrating the ability to deliver key technical milestones. The current data rates are modest, however, they are expected to increase significantly in the later experimental phase as shown in Table 1.

Table 1. Storage requirements for each phase of the Project 8 experiment

Phase 1 & Phase 2	0.5 Petabytes
Phase 3	≥ 10 Petabytes
Phase 4	≥ 100 Petabytes

In this paper, we will provide a summary of the current Project 8 computing model. Section 2 provides an overview of the computing infrastructure that includes the virtualized service cluster, the use of the DIRAC interware [2], and the integration of the Pacific Northwest National Laboratory (PNNL) heterogeneous computing resources. Section 3 is a summary of the automated data processing pipeline. Finally, in Section 4 we conclude the paper and present some results.

*e-mail: malachi.schram@pnnl.gov

2 Computing model

The Project 8 computing model was developed to provide maximal flexibility and scalability for all 4 phases. The experimental setup is located at the University of Washington in Seattle and produces approximately 200-1000Mbps of continuous data. In order to process this data in a timely manner, the data is automatically transferred to the PNNL HPC cluster for processing and analysis. The computing system was designed to handle a large amount of data and the ability to dynamically integrate a variety of geographically distributed computing resources. To this end, we have leveraged a collection of software stacks to enable a robust, distributed, and scalable solution. Specifically, we are using elements of the existing distributed computing infrastructure developed by the high energy physics community for the Large Hadron Collider experiments. A schematic diagram of the Project 8 computing model is illustrated in Figure 1.

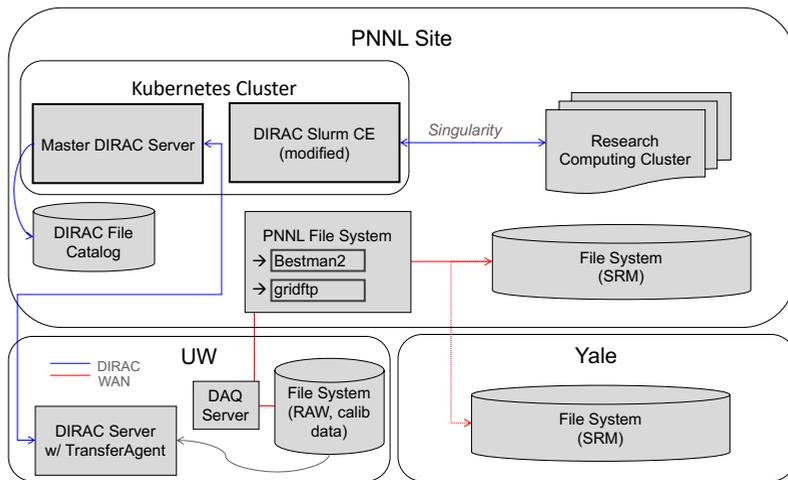


Figure 1. Current Project 8 computing model.

A summary of the current Project 8 computing models is provided in the following subsections.

2.1 Software development and distribution on computing clusters

The Project 8 collaboration uses several Git repositories as their distributed version control system for tracking changes in source code during software development on all essential efforts. Several of these software repositories used for the production pipeline have specific OS and libraries requirements that cannot be satisfied on the PNNL HPC cluster. For example, the base OS on the HPC system is CentOS 6 while the Project 8 production software requires CentOS 7. In order to alleviate these types of conflicts and allow developers more flexibility, we use Docker containers [3] to build new instances of the development and production software and environment stack. The production containers are then converted to Singularity [4] images and integrated into the dedicated DIRAC agent that submits compute jobs using the desired Singularity image. Illustrated in Figure 2 is a schematic diagram of the development to production workflow.

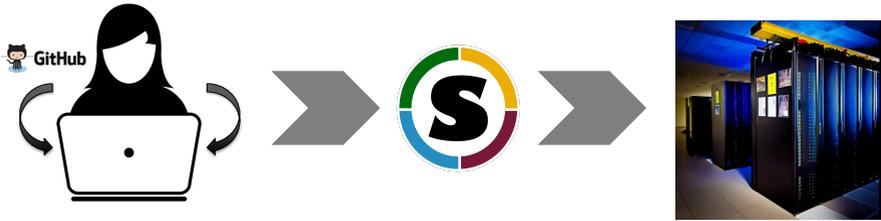


Figure 2. Project 8 development to production workflow.

2.2 Virtualized service cluster

At the heart of the Project 8 computing model is the virtualized service cluster which provides all the central services for the geographically distributed computing infrastructure. This computing infrastructure was developed using containerized components and orchestration. We used an open-source container orchestration system, named Kubernetes, to automate application deployment, scaling, and management [5]. We choose to back our Kubernetes cluster with the containerd container runtime [6], which forms the core runtime engine of the popular Docker tool. Some of the key motivations for using containers and container orchestration was the ability to quickly reinstall machines and workloads. Additionally, it enables high availability by easily and dynamically moving containers off problematic nodes to healthy ones. Kubernetes is an ideal tool to manage compute resource by dynamically adding and/or removing services as required by the current computing load. The use of containers allows to decouple the build, test, and deploy workflows enabling more reliable deployment. They also provide a clean solution for conflicting dependencies when running numerous services on the same node. Kubernetes enables a light weight layer to maintain multiple nodes and many containers.

2.3 DIRAC

The DIRAC framework was selected to provide scalable geographically distributed computing system. DIRAC has a history of satisfying the requirements of a diverse collection of experiments with a vetted file catalog system, workflow execution and data management capabilities. DIRAC was developed for the Large Hadron Collider beauty experiment (LHCb) at European Organization for Nuclear Research (CERN). The DIRAC File Catalog (DFC) and associated services allow collaborations to keep track of all physical copies of existing files and enable users to define file and/or directory level metadata providing efficient and fast access to datasets for specific analysis tasks. The Workload Management System (WMS) orchestrates continuous job execution, simultaneously submitting jobs at various computing sites based on the availability and resource requirements. The WMS follows the task scheduling paradigm using the generic pilot jobs [7]. The pilot job retrieves a compute job payload that matches a job in the task queue. Another critical component, for Project 8, that is provided by DIRAC is the Transformation System (TS). This system allows to automate asynchronous computing workflows that are initiated by newly registered data. Finally, DIRAC's modular structure allows for collaborations, such as Project 8, to easily develop dedicated extensions to satisfy their unique requirements.

2.4 Distributed computing services

As illustrated in Figure 1, Project 8 computing has several components distributed across 3 sites. PNNL hosts all grid computing services, the master DIRAC server, and compute and the primary storage resources. The master DIRAC server at PNNL provides all services, databases, and most of the agents. It is used to manage tasks (jobs, data transfers, etc.) and to automate various meta-data driven asynchronous workflows. Project 8 has a Virtual Organization Membership Service (VOMS) deployed as containerized applications within a Kubernetes cluster. VOMS is used by Project 8 to provide an attribute-based authorization to control access to the DIRAC server and compute resources. The compute element (CE) is responsible for handling job submissions to the compute resources. The CE version at PNNL is a modified version of the DIRAC Slurm [8] agent that uses a defined Singularity image for job execution. The Storage Element (SE) at PNNL and Yale are composed of fetch-crl, gridftp and BeStMan2 containers. A dedicated DIRAC server is deployed at University of Washington with a dedicated agent tasked to replicate and register new files to the PNNL SE. The certificates for the users and DIRAC services are managed using LetsEncrypt [9], a free, automated and open Certificate Authority. The certificates within the Kubernetes cluster are managed by the cert-manager [10].

3 Automating the raw data pipeline using the DFC

In order to automate the Project 8 asynchronous workflows, such as the one illustrated in Figure 3, we assign specific meta-data entries using two levels of metadata for all datasets. The DFC provides two levels of meta-data:

- Directory level: the metadata is at the directory level and will automatically apply for all files/folders within that directory
- File level: the metadata only applies to the specific file

For example, we can define the **RunID**, **DataType**, and **DataLevel** at the directory level and **DataExt** at the file level. Directory level searches can significantly reduce the query time for deep file structures with numerous files. These searchable metadata fields are used to automate each component of the workflow. For example, we can combine specific metadata field values to retrieve the list of files from a specific experiment, acquisition characteristic, and data type. With these metadata labels in place, we developed an automated data-driven workflow in which each component of the workflow responds to new data entries in the metadata catalog.

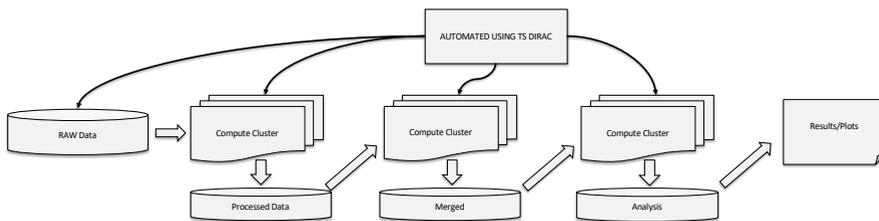


Figure 3. Project 8 automated data processing pipeline.

4 Conclusion

The current Project 8 computing effort has been running for the last year. The data driven automation system using the DFC metadata has significantly reduced operations load and the need for manual intervention. The majority of the computing has been used by the production system to process and analyze the data in near real time. As shown in Figure 4, over 650k jobs have already been executed and nearly 2 Petabytes of data have been transferred in the span of the last 7 months. We expect an increase in computational usage in order to perform simulation studies for Phase 3 and beyond.

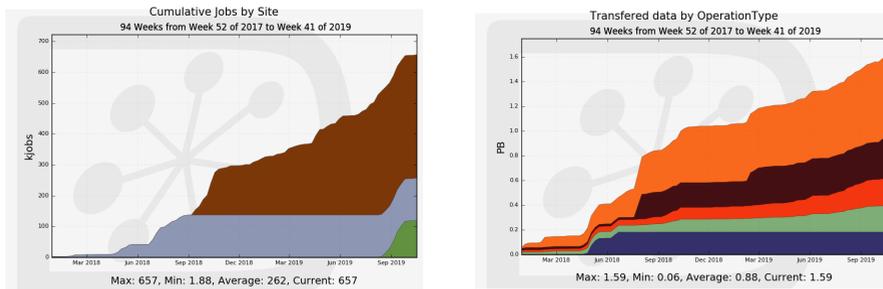


Figure 4. Current Project 8 computing utilizations. The left figure shows the number of cumulative jobs per computing site (top 2 sites are at PNNL). The right figure shows the amount of data transfer per operation type (dominated by transfers between UW and PNNL).

Acknowledgements

This work was supported by the US DOE Office of Nuclear Physics, the US NSF, the PRISMA+ Cluster of Excellence at the University of Mainz, and internal investments at all collaborating institutions. The author recognizes support from the Laboratory Directed Research and Development program at PNNL. A portion of the research was performed using Research Computing at PNNL.

References

- [1] Ashtari Esfahani, Ali and others, textitDetermining the neutrino mass with cyclotron radiation emission spectroscopy—Project 8, *J. Phys.*, G44, (2017)
- [2] A. Tsaregorodtsev, *DIRAC Distributed Computing Services*, *J. Phys.* 513 (3), (2014).
- [3] D. Merkel, *J. Linux* 2 (239), (2014).
- [4] Gregory M. Kurtzer, Vanessa Sochat , Michael W. Bauer, *Singularity: Scientific containers for mobility of compute*, Public Library of Science, 1-20 (2017)
- [5] K. Hightower, B. Burns and J. Beda, *Kubernetes: Up and Running Dive into the Future of Infrastructure, 2nd ed.* (OReilly, Beijing, 2019).
- [6] Containerd (2019). Available at: <https://containerd.io/> (accessed 17 December 2019)
- [7] A. Casajus, R. Graciani, S. Paterson and A. Tsaregorodtsev, *DIRAC pilot framework and the DIRAC Workload Management System*, *J. Phys.* 219 (6), (2010).
- [8] A. B. Yoo, M. A. Jette and M. Grondona, *Job Scheduling Strategies for Parallel Processing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 44-60, (2003).

-
- [9] A. Josh, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-Lopez, J. A. Halderman, J. Hoffman-Andrews, J. Kasten, E. Rescorla, S. Schoen and B. Warren, ACM SIGSAC CCS. 15 (2019).
- [10] Cert-Manager (2019). Available at: <https://docs.cert-manager.io/en/latest/> (accessed 10 September 2019)