Jupyter-based service for JUNO physics analysis

Integrating Big Data techniques with ROOT-based analysis

Tao Lin, Weidong Li, Yan Liu, Wenxing Fang, Jiaheng Zou
(on behalf of the JUNO collaboration)
Institute of High Energy Physics (IHEP), CAS
lintao@ihep.ac.cn and liwd@ihep.ac.cn

1 Introduction

The JUNO [1, 2] experiment, located in southern China, aims to determine the neutrino mass hierarchy and observe neutrinos from terrestrial and extra-terrestrial sources, including the supernova burst neutrinos, diffuse supernova neutrinos, geo-neutrinos, atmospheric neutrinos and solar neutrinos. It is about 53 km away from the Yangjiang and Taishan nuclear power plants.

Figure 1: Schematic view of the JUNO detector

Figure 1 shows the schematic view of the JUNO detector, where the innermost part is the central detector which is surrounded by a water Cherenkov detector. Not only the radioactivity backgrounds, but also neutrons introduced by cosmic-ray muons in the rock are heavily suppressed by layer of water. Instrumented with PMTs, the water Cherenkov detector can detect cosmic-ray muons. There is a 3-layer plastic scintillator top tracker above the Cherenkov detector, which provides precise and independent (cosmic-ray) muon track information.

2 Motivation

- Raw data volume: 2 PB/year. However, the neutrino signals is rare.
- Huge background levels ($\mathcal{O}(10)$ Hz) compared to the neutrino signals (about 60/day).
- Accessing dataset many times may cause I/O performance.
- Meanwhile, the time correlation is still important. No events could be discarded.

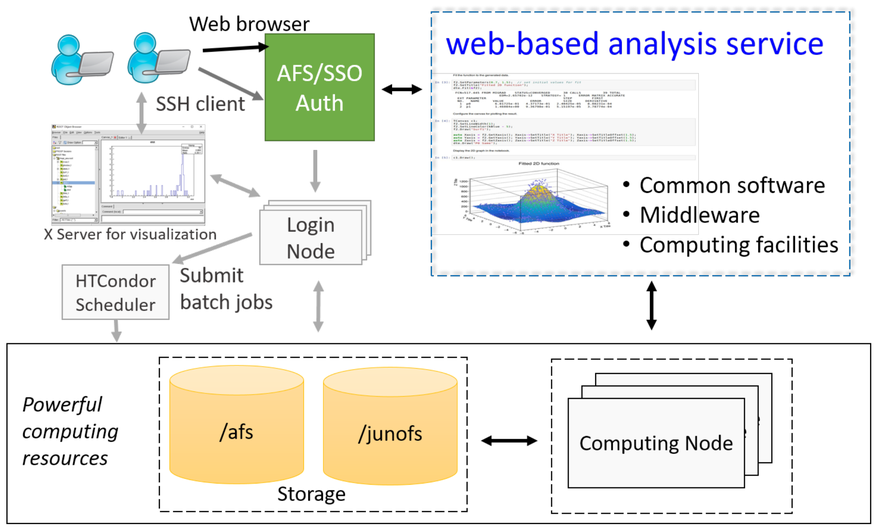

Figure 2: Schematic view of integration of ROOT and interactive analysis

In order to improve the efficiency to search for rare events in the huge background level, the big data techniques such as in-memory analysis are under investigation. Apach Spark is chosen for its simplicity and easiness to use. The physics quantities are loaded into memory once, and can be processed many times. After getting a subset of events, the SNiPER [3] framework is used to analyze the events with time correlation. Figure 2 shows the integration of ROOT and interactive analysis. The underlying computing resources are shared between them. Users could use both the SSH login with X server and the web browser to access the resources. In the web based analysis service, a middle-ware will be developed.

3 Design

3.1 Infrastructures

The Jupyter-based service adopts Jupyter Notebook and its next generation, JupyterLab, as the web-based user interfaces. The JupyterHub is adopted to provide groups of users to access their own dedicated computational environments and resources. The computational resources are then managed by Kubernetes. The requests of resources from Jupyter will be implemented by the Kubernetes.

Figure 3: The packages used in the service

The key features of the service:

- User friendly. Users don't need to install the software themselves.
- Dedicated resources. Users could use the resources allocated for them.
- Scalable. Benefit from Kubernetes, users could request more resources.

In order to enhance the user experience, JupyterHub is customized including the Docker images, memory and CPU quota, and the mounted file systems. The Docker image is based on a distribution from IHEP. CernVM File System (CVMFS) is configured by default in the image, so users could access the JUNO software on-demand.

3.2 Unified Event Data Store

In order to support the Jupyter Notebook, the normal cluster and the Spark cluster, a unified event data store is under development, as shown in Figure 4. The access to an event could be from the direct access in the event data, or from the index files to redirect to the event data.

In the index files, there are two major fields: the file index and the entry of the event in the event data file. The other fields could be extended and used for the event reduction with big data techniques. All these physical quantity are extracted from the event data files, hence the size of each record is reduced. The file formats for these index files could be different according to the big data techniques.

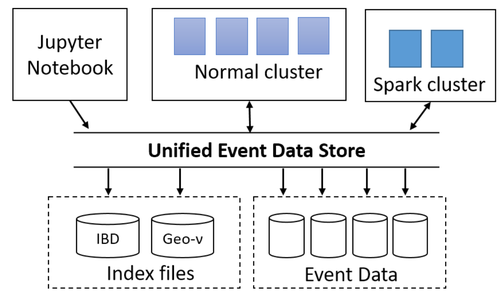

Figure 4: The unified event data store

3.3 Analysis procedures

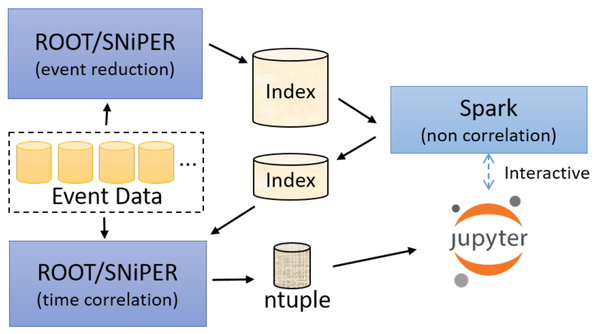

Figure 5: Analysis with Spark and SNiPER

As shown in Figure 5, the procedures are:

1. Using ROOT/SNiPER to produce the primary index files from the event data.
2. Using the Spark to analyze the index files interactively. All the index files are cached in the memory. The final results are then stored into the user index files.
3. Using ROOT/SNiPER to load the user index files and access the selected events directly. Finally, the plots and nutples can be shown in Jupyter.

4 Test-bed and Results

The test-bed is setup in two rack-mounted servers with total 40 cores (Intel Xeon Silver 4114), which are managed by the Kubernetes. The JupyterHub is then started as a container inside the Kubernetes. In order to avoid the resource competition between Jupyter and Spark due to the limited resources, the Spark cluster is not setup inside the Kubernetes. The Spark cluster is setup using 8 nodes with Intel Xeon E5-2630L v2 (12 cores, 24 threads).

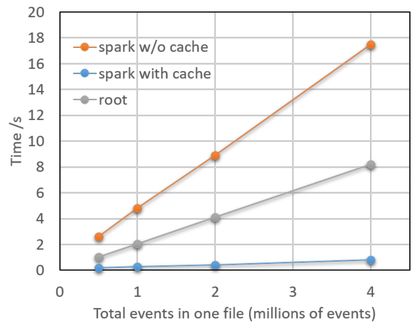

Figure 6: The example of measured performance in the test-bed

Figure 6 shows the data processing time measured in the test-bed with different methods. In this example, ROOT files are generated with different number of events. Then they are processed with a ROOT script respectively. The data files are processed using spark with spark-root [4], without cache and with cache respectively. The cache option will let spark to cache all data in the memory. As shown in the figure, there is about 10x speedup using spark with cache compared with ROOT.

5 Conclusions

- The challenge for JUNO analysis is to efficiently search for rare neutrinos in huge background levels.
- The big data techniques such as Spark's in-memory analysis is investigated to speed up the data processing.
- The time correlation analysis is supported by the index files, which point to the full event data.
- The index files make it possible to skip the backgrounds in the event data files.
- A test-bed is setup at IHEP and will be used for the further studies of big data techniques for JUNO.

6 Forthcoming Research

The next work focuses on the application of big data techniques. One possible research is using the HDF5 file format for index files. Another interesting research is using the PyRDF, which enables the Spark back-end automatically. As the c++ is still used in the SNiPER framework, a MPI based solution could be used to speed up the data processing.

References

[1] F. An *et al.*, "Neutrino Physics with JUNO," *J. Phys.*, vol. G43, no. 3, p. 030401, 2016.

[2] Z. Djurcic *et al.*, "JUNO Conceptual Design Report," 2015, arXiv:1508.07166.

[3] J. H. Zou, X. T. Huang, W. D. Li, T. Lin, T. Li, K. Zhang, Z. Y. Deng, and G. F. Cao, "SNiPER: an offline software framework for non-collider physics experiments," *J. Phys. Conf. Ser.*, vol. 664, no. 7, p. 072053, 2015.

[4] V. Khristenko and J. Pivarski, "diana-hep/spark-root: Release 0.1.14," Oct. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1034230

Acknowledgements

This work is supported by National Natural Science Foundation of China (11805223), Joint Large-Scale Scientific Facility Funds of the NSFC and CAS (U1532258), the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA10010900.