JUNO distributed computing system

Xiaomei Zhang[12,*]

[1]Institute of High Energy Physics
[2]on behalf of JUNO collaboration

Abstract. The Jiangmen Underground Neutrino Observatory (JUNO) [1] is a multipurpose neutrino experiment and the determination of the neutrino mass hierarchy is its primary physics goal. JUNO is going to start data taking in 2024 and plans to use distributed computing infrastructure for the data processing and analysis tasks. The JUNO distributed computing system has been designed and built based on DIRAC [2]. Since last year, the official Monte Carlo (MC) production has been running on the system, and petabytes of massive MC data have been shared among JUNO data centers through this system. In this paper, an overview of the JUNO distributed computing system will be presented, including workload management system, data management, and condition data access system. Moreover, the progress of adapting the system to support token-based AAI [3] and HTTP-TPC [4] will be reported. Finally, the paper will mention the preparations for the upcoming JUNO data-taking.

1 Introduction

JUNO is a multipurpose neutrino experiment located in Jiangmen, South China. Its primary physics goal is to determine the neutrino mass hierarchy and neutrino oscillation mixing parameters. To achieve this, JUNO is designed to detect electron antineutrinos from nuclear reactors with unprecedented precision and sensitivity. The experiment is expected to start taking data in 2024, with a projected data volume of 2 petabytes per year. The expected event rate is 1 kHz, and the raw data rate is expected to be 60MB/s.

The JUNO resources are distributed among different geographical locations. To handle such a large amount of data and perform the required simulation, reconstruction, and analysis activities in these worldwide resources, JUNO built a distributed computing system based on DIRAC in 2018. This system allows JUNO to distribute computing tasks to the CPU resources located in remote data centers and share data promptly among data centers. The official MC production started to run in the JUNO distributed computing system in 2020, and massive amounts of MC data have been shared among JUNO data centers through this system.

In recent years, WLCG (Worldwide LHC Computing Grid) is undergoing two large-scale transitions. One is HTTP-TPC which replaced GridFTP as basic protocol of data transfer. The other is token-based Authentication and Authorization Infrastructure (AAI) which is replacing X509-based AAI. Most of JUNO data centers are WLCG sites and the JUNO distributed computing system is using WLCG middleware. In this paper, we will also describe the progress of adapting the system to those two changes.

[*]e-mail: zhangxm@ihep.ac.cn

2 JUNO computing model

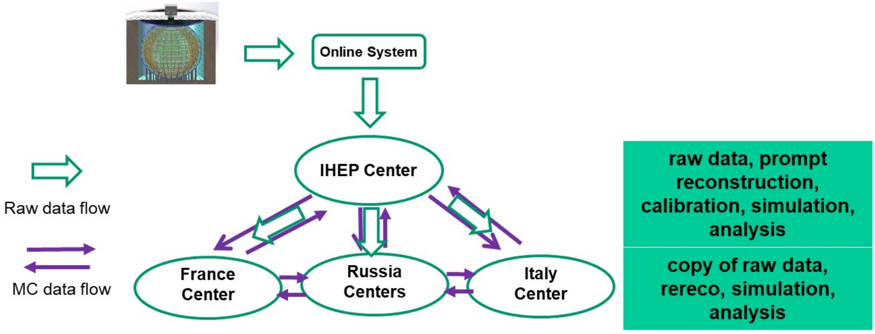

Online System
IHEP Center
Raw data flow
MC data flow
France Center
Russia Centers
Italy Center
raw data, prompt reconstruction, calibration, simulation, analysis
copy of raw data, rereco, simulation, analysis

Figure 1: JUNO computing model

As shown in Figure 1, five main data centers located in China, France, Russia, Italy will take part in the JUNO computing, namely IHEP, CNAF, IN2P3, JINR, MSU. Besides these big data centers, some small cloud and cluster sites are also contributing resources to JUNO. The offline activities in JUNO including simulation, reconstruction, analysis are planned to run in the JUNO distributed computing platform. The JUNO computing model aims to organize JUNO offline activities in JUNO data centers and provides seamless data access in a proper and efficient way. The JUNO computing model is organized in a hierarchical way. The IHEP data center acts as both Tier0 and Tier1, which is responsible for holding all the data including raw data, MC data and reconstructed data, etc. Other big data centers act as Tier1. CNAF and JINR data centers hold a copy of these data each and IN2P3 holds 1/3 of all the raw data. Prompt reconstruction and calibration are performed using a dedicated cluster in IHEP, while other activities are assigned to all the data centers including MC simulation, re-reconstruction, analysis. The simulation and analysis tasks are also assigned to small sites. Re-reconstruction will be running in the data centers where raw data is located. The data will be copied to and stored in data centers after being produced by the above activities. The raw data from the detector in Jiangmen is replicated to the IHEP data center, where these data are transferred to other three data centers without delay. The JUNO distributed computing system is designed according to the need of the JUNO computing model.

3 JUNO distributed computing system (DCS)

As shown in Figure 2, the architecture of DCS is organized in three layers: resource, service, and applications. The service layer provides all the services needed for JUNO distributed computing in a centralized infrastructure, including DIRAC, FTS, VOMS, CVMFS, IAM, etc. Most of central services are running at IHEP, and some redundancy services such as VO Management Service (VOMS) are running at other data centers. The IAM service is hosted at CNAF. DIRAC represents the core framework of this layer, which allows to integrate heterogeneous and remote resources, and provides a unified platform by integrating other necessary services. The implementation of the JUNO distributed computing system benefits from WLCG middleware as much as possible. The data centers provide CPU resources through CE (Computing Element) service and storage resource through SE (Storage Element) service. The lower layer is composed of resources located in data centers including CPU, storage and network. The applications layer provides tools and interfaces to support

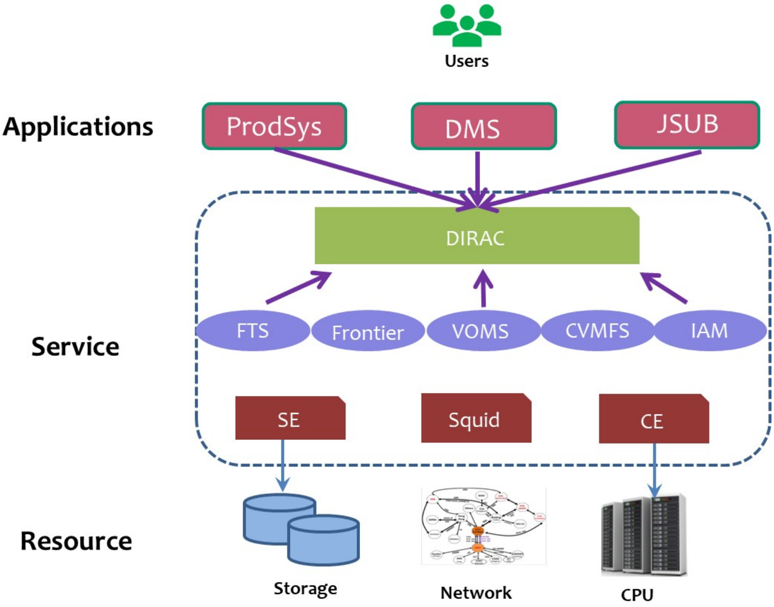


Figure 2: Architecture of the JUNO distributed computing system

data placement and data processing activities, which are specific to JUNO . In this layer, the JUNO production system (ProdSys)[5] for managing MC production and data production, and JSUB [6] for providing submission tool for user analysis, have been developed. From a function point of view, the system consists of four parts: Workload management, Data management, Raw data transfer, and Condition data access. Workload management contains all the necessary components to take care of JUNO job submitting, scheduling and running in DCS environment. Data management provides a way for users to access, query and share JUNO data in grid environment. Raw data transfer takes care of replicating raw data to data centers for permanent backup and reconstruction. Condition data access provides efficient access to the JUNO condition data.

3.1 Workload Management (WM)

The WM is responsible for job submission, scheduling and management. DIRAC provides the framework for WM, which is able to accept and schedule jobs to proper resources, hide complexity of resources from users, and also provide solutions for experiments to handle massive tasks. Each task could contain thousands of jobs.

Based on DIRAC, the ProdSys has been developed to handle bulk Monte Carlo simulation and reconstruction tasks for production groups, and manage workflow and dataflow of the tasks in an automatic way. Each JUNO production task contains several steps, including detector simulation (detsim), electronics simulation (elecsim) , and PMT Reconstruction (cal), event reconstruction (rec), replication of output to destination sites. To chain these steps with data and handle the whole process automatically, the ProdSys shown in Figure 3 is implemented as a data-driven pipeline system based on DIRAC transformation system (TS) [7]. The ProdSys system is capable of supporting the production of multiple particles in events

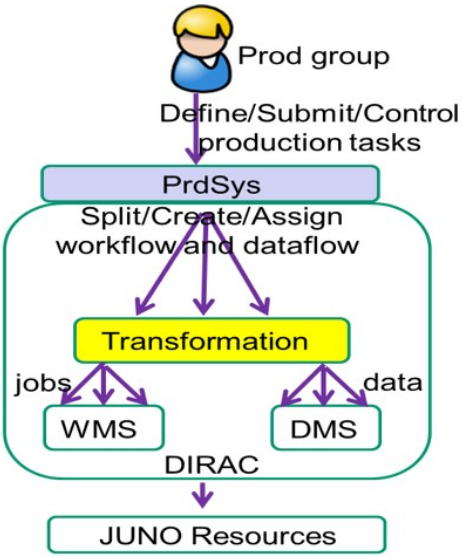

Prod group
Define/Submit/Control production tasks
PrdSys
Split/Create/Assign workflow and dataflow
Transformation
jobs
data
WMS
DMS
DIRAC
JUNO Resources

Figure 3: Architecture of Prodsys

that reach billion-level scale. In this system, all the data in the tasks are supposed to be registered in the Dirac File Catalogue (DFC) [8]. Each step in the chain except detsim is triggered by the availability of data registered in DFC. Users can define tasks through a steering file which allow to define task parameters, eg steps in the chain, destination SE, etc. DIRAC TS provides an architecture which can organize JUNO workflow and dataflow into a pipeline. The requests of job and file transfer generated by the pipeline are submitted to the workload management system and data management system separately. Another component in WM is JSUB, which is a lightweight user job submission tool, developed in Python. JSUB aims to automatically take care of life cycle of user analysis, and ease process of physics analysis for JUNO users in grid environment. The main function packages in JSUB are job splitting and creation, job submission to various backends (DIRAC, HTCondor, etc), job management (status monitoring, failed jobs rescheduling, job logs retrieval, etc), dataset operations (dataset query, create, delete, etc), and user interface. Analysis task parameters can be configured in YAML. JSUB utilizes the DIRAC parametric job submission interface to achieve a high job submission rate. It can submit thousands of jobs to DIRAC in just a few seconds. Additionally, JSUB offers a crucial capability of efficiently splitting tasks into sub-jobs with the inclusion of multiple variables as inputs. Moreover, JSUB is designed to be extensible, enabling its usage across multiple experiments and backends.

3.2 Data management (DM)

The DM is responsible for data placement, data access and data management in grid. The DIRAC data management system acts as the core of DM, which can integrate various storage elements and necessary services. There are three types of storage element (SE) in JUNO data centers: dCache, EOS and StoRM. FTS3 [9] in DM is responsible for file level movement, which is connected to the DIRAC data management system. The DIRAC DFC is adopted to be JUNO metadata and replica catalogue. All JUNO data are expected to register in DFC. DFC provides a global view of JUNO data located in different data centers and the functionality to manage dataset. DIRAC TS provides a way to translate dataset-level operation requests into file-level operation requests, and DIRAC RMS (Request Management System) manages large amounts of file operation requests in queues, and send transfer requests to FTS3. The JUNO-specific DMS tools, which provide an interface for JUNO users to take care of JUNO data, were developed on top of these systems.

The GridFTP protocol was the basis for data transfers between SEs in the past decade. But GridFTP is deprecated since Globus is dropping support of its open source Globus Toolkit (GT), which forms the basis for several FTP client and servers. The HTTP-TPC has been adopted by WLCG to replace GridFTP in the future. The migration was started in JUNO DM at the beginning of 2022 and the related system updated to support HTTP-TPC in 2023 including SEs. To make sure HTTP-TPC working properly in data transfer, HTTP-TPC tests have been carried out. A monitoring dashboard was developed to record and track test results. Three transfer modes (stream, pull, push) are tracked. From the long-term test results, the pull mode is working fine and some random failures were found in the push-mode transfers between StoRM and EOS, which is caused by unsuccessful communications on reading HTTP headers between two SEs.

The migration from X509-based to token-based AAI started last year. The core server of token-based AAI – IAM [10] was installed and maintained by the CNAF data center in 2022. HTCondor CEs in data centers demonstrated their capability to support tokens. All the SEs are able to support macaroon token. Most of SEs can support scitoken except EOS SE which is being updated to EOS5 to support scitoken. The DIRAC instance in the DCS is being updated to the latest version which can support new AAI. The whole system is planned to be ready for testing before end of year 2023.

3.3 Raw data transfer

One of the crucial tasks of the JUNO DCS during data taking is to promptly replicate raw data from IHEP to other data centers. The transfer of data from online to IHEP is managed by a dedicated transfer system called SPADE. Once the data files arrive at the IHEP EOS disk, the raw data transfer system initiates the process. Communication between SPADE and the DCS relies on Kafka, a message queue system. Upon receiving a message from SPADE, the DCS retrieves data file information from the message, registers the raw data in the DFC, replicates them to the disks of data centers in parallel, and eventually copies the data from disk to tape while registering it in the DFC. The data files are organized in blocks classified by data receiving date and data blocks are defined as transfer units. The system consists of four main components: *Data Receiver*, *File Register*, *Block Creator*, and *Block Transfer*, as shown in Figure 3. The *Data Receiver* is implemented as an independent daemon, listening for messages, and triggering registration and transfer process when receiving messages. Other three components are implemented based on the DIRAC Data Management System. The *File Register* is responsible for registering data files in DFC which arrived in EOS. The *Block Creator* takes care of defining blocks with metadata in DFC. The *Block Transfer* creates transfer tasks block by block using DIRAC Transformation System with unique transform id for each transfer task. Each transfer task will be transformed into a list of file transfer requests. The transfer requests will be sent to FTS3 for final execution.

3.4 Condition data access

JUNO stores condition data in MySQL. The central MySQL database is located in IHEP. Same set of condition data will be repeatedly used by thousands jobs in the same period which are spread in several data centers. Caching these condition data close to where the jobs

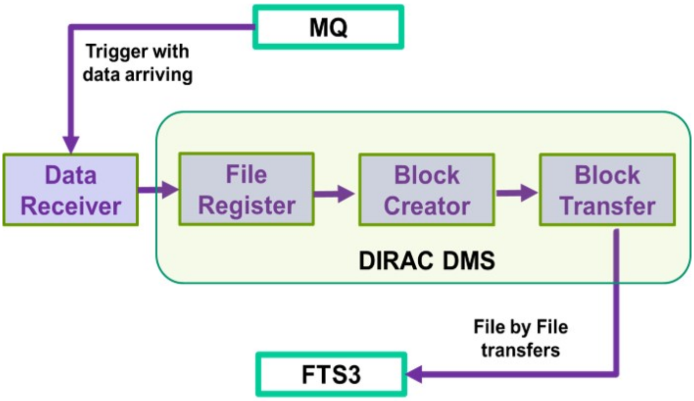


Figure 4: Implementation of raw data transfer

are running provides significant performance gains. Therefore, to avoid high load to central DB and speed up the access to condition data from remote data centers, the FroNTier/Squid infrastructure [11] has been adopted. This infrastructure includes the FroNTier server, the FroNTier client, and the Squid server.

The FroNTier server, which connects to the offline condition database, is responsible for accepting HTTP requests from the FroNTier client. It converts these requests into SQL queries, retrieves the results, and returns them as an HTTP stream back to the client. The FroNTier server is deployed at IHEP, while the client is integrated into the JUNO offline software to be used by JUNO offline data processing. Each data center is equipped with one or more Squid servers.

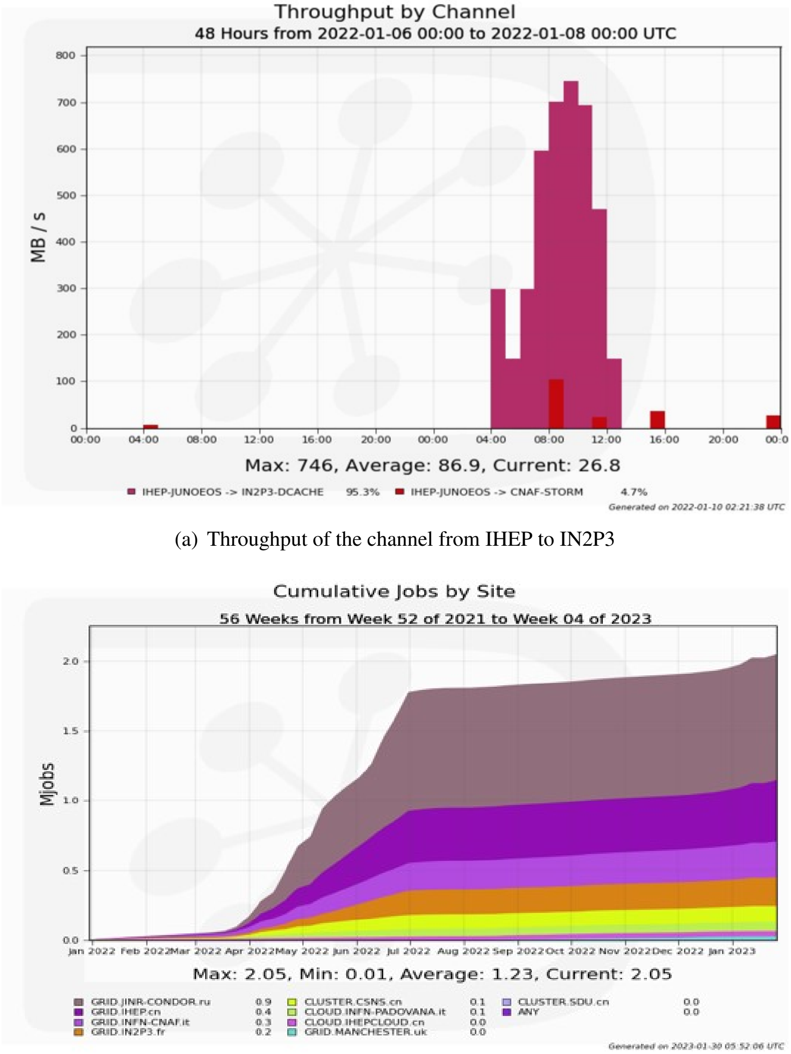
Tests conducted with JUNO jobs have demonstrated more than a 10-fold improvement in performances when accessing the cached data compared to direct database access. Further pressure tests to simulate a production-like environment will be conducted later this year.

4 Production campaigns and data challenges

To evaluate the JUNO distributed computing system and prepare for data taking in 2024, several MC production campaigns have been conducted.

The first data production campaign began in February 2020, generating over 16TB of electron, positron, alpha, and gamma samples with varying energy and position points in a single submission. The workflow step required for this campaign was detsim. Jobs were distributed among all data centers, and the output was uploaded to the closest Storage Elements (SEs) while registering their metadata information in the DFC. An independent transformation process handled the replication of output data to the designated SEs. Remarkably, we achieved a 100% success rate in this campaign.

The second production, focused on muon generated samples, took place in June 2020. Approximately 10% of the muon events required around 8GB of memory and more than two CPU days per event, without specific selection rules from these events with special requirements. The JINR data center provided 16GB of memory per core, while two cloud sites allocated an additional 4GB of memory to each virtual machine running JUNO jobs. Other sites had 2.5-3.5GB of memory per core. In total, one million events were generated, with each job processing five events. Jobs were assigned to all available sites. Ultimately, a 0.05% failure rate was observed, primarily due to memory issues. The failed jobs were manually



(a) Throughput of the channel from IHEP to IN2P3

(b) Total number of jobs running in DCS in 2022

Figure 5: production campaigns and data challenges

resubmitted to sites with high-memory work nodes. Improvements are necessary to reduce the failure rate and address failed jobs during the muon production phase.

The first data challenge was conducted in 2020 to assess the functionality and performance of the data management system and network capabilities against JUNO requirements. The challenge utilized 20TB of data, with each file being 5GB in size. Four sites, namely IHEP, IN2P3, CNAF, and JINR, participated in the test. The protocols employed were HTTP and XRootD.

During the first round of testing, data files were replicated in parallel from IHEP to other data centers (JINR, IN2P3, CNAF), simulating the transfer of raw data. The network bandwidth from IHEP to the other data centers was 10Gb/s, shared with other experiments. The raw data volume amounted to 5.2TB per day, necessitating a minimum transfer speed of

600Mb/s. The tests yielded significant success, with average speeds ranging from 1Gb/s to 3Gb/s and a total speed of approximately 8Gb/s, nearly reaching the full network performance. In the second round of testing, data files were independently replicated from IHEP to other data centers, achieving an average speed of 5Gb/s to 7Gb/s, as depicted in Figure 5(a).

The ProdSys, utilized by the JUNO production group since the inception of the 2022 Mock Data Challenge (MDC), facilitated the submission and execution of over 2 million jobs. Successfully, 1.4PB of output data was transferred to the designated Storage Elements (SEs), as illustrated in Figure 5(b). In the Data File Catalog (DFC), a total of 1PB of data and 4 million files are registered.

5 Conclusions

The JUNO distributed computing system has been meticulously designed and developed to fulfill the demanding requirements of JUNO data processing. In order to ensure its readiness for the data taking period, several data challenges have been conducted. These challenges are allowed to assess the system's performance and identify areas for improvement. Further pressure tests are necessary to comprehensively understand the effects and benefits of system updates and optimizations.

The authors would like to thank the JUNO offline software team for their help with understanding JUNO data processing, the DIRAC team for their help with the DIRAC transformation system, and colleagues at JUNO data centres for their support. This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA10010900

References

[1] A. Abusleme et al, JUNO Collaboration, "JUNO Physics and Detector", Prog. Part. Nucl. Phys. **123**, 103927 (2022)

[2] A.Tsaregorodtsev, M.Bargiotti, N.Brook, A.C.Ramo, G.Castellani, P.Charpentier, C. Cioffi, J.Closier, G.DiazR, G.Kuznetsov, Y.Y.Li, R.Nandakumar, S.Paterson, R.Santinelli, A.C.Smith, M.S.Miguelez and G.JimenezS, J. Phys.: Conf. Ser. **119** 062048 (2008)

[3] A Ceccanti et al J. Phys.: Conf. Ser. (898) 102016 (2017)

[4] Brian Bockelman, Andrea Ceccanti, Fabrizio Furano, Paul Millar, Dmitry Litvintsev, Alessandra Forti, arXiv:2007.03490 [https://doi.org/10.48550/arXiv.2007.03490]

[5] Xiaomei Zhang et al, EPJ Web of Conferences **245** 03007 (2020) [https://doi.org/10.1051/epjconf/202024503007]

[6] Yifan Yang et al 2023 J. Phys.: Conf. Ser. **2438** 012048

[7] F Stagni et al 2012 J. Phys.: Conf. Ser. **368** 012010

[8] Tsaregorodtsev, A, Poss, S, J. Phys.: Conf. Ser. **396** (2012) pp.032108

[9] Kiryanov, etc, FTS3 - A File Transfer Service for Grids, HPCs and Clouds 10.22323/1.239.0028. (2016)

[10] INDIGO Identity and Access Management (IAM), https://indigo-iam.github.io/docs

[11] Blumenfeld B, Dykstra D, Lueking L and Wicklund E 2008 CMS conditions data access using FroNTier J. Phys.: Conf. Ser. **119** 072007 (7pp)