

Development of a Scout Job Framework for Improving Efficiency of Analysis Jobs in Belle II Distributed Computing System

Thursday, March 24, 2022 1:30 PM (30 minutes)

The Belle II experiment is a next generation B-factory experiment using an electron-positron collider, SuperKEKB. Our goal is to broadly advance our understanding of particle physics, e.g. search for physics beyond the Standard Model, precise measurements of electroweak interaction parameters, and exploring properties of the strong interaction. We started collecting collision data in 2019, aiming to acquire 50 times more data than the predecessor experiment, Belle. At the end of the data taking, we need several hundred PB disk storage and tens of thousands of CPU cores are required. To store and process the massive data with the resources, a worldwide distributed computing system is utilized. Its interconnection among end-users and heterogeneous computing resources is done by the DIRAC interware [1] and its extension Belle DIRAC to meet our requirements.

For the physics analysis, an end-user prepares analysis scripts and can submit a cluster of jobs to the system with a single command including input data sets. The system submits jobs to computing sites where input data is hosted. For more efficient analysis, it is important to use computer resources efficiently. However, about ten percent of executed jobs failed in 2019. The reason is not problems of the system but can be broadly categorized as problems in the analysis script or improper settings of the job parameters specified by the end-user. They reduce the job execution efficiency in two points. First, in our system, any jobs spend a few minutes on a worker node in downloading input files, authentication, and so on. Therefore, worker nodes are unnecessarily occupied for failed jobs. Second, when many jobs are submitted at once and they fail quickly, access to the central system is concentrated for a short time. It often triggers system trouble and solving the trouble becomes a load on the operation side.

Therefore, we have developed two features to suppress the failed jobs. For problems originating from analysis scripts, we add a syntax checker in the job submission command. This detects syntax errors at the language level of analysis scripts and notifies the end-user before the job submission. However, this is not enough to detect complicated syntax errors or the improper settings of job parameters. Therefore, we also develop a scout job framework, which submits a small number of test jobs (henceforth referred to as “scout jobs”) with the same analysis script to process a small number of events prior to massive job submissions. Then, only if the scout job succeeds, main jobs are submitted. Otherwise, all submissions of the main jobs are canceled. With these two features, we could reduce the operation load and waste of computational resources. Furthermore, this is beneficial also for end-users because the pre-test is streamlined. In the future, we aim to improve further by adding a function to automatically correct the problematic job parameters and by implementing this framework into the system for automatically generating simulation samples.

In this presentation, an overview of physical analysis in distributed computing systems, the design and operational results of the developed features, and prospects will be given.

[1] DIRAC, <https://github.com/DIRACGrid/DIRAC>

Primary authors: HIRATA, Hikari (Nagoya University); THE BELLE II COMPUTING GROUP

Presenter: HIRATA, Hikari (Nagoya University)

Session Classification: Physics & Engineering

Track Classification: Track 1: Physics (including HEP) and Engineering Applications