

Improving the automated calibration at Belle II

Francis Pham^{1,*}, David Dossett¹, and Martin Seviar¹

¹School of Physics (David Caro Building), The University of Melbourne, VIC 3010, Australia

Abstract. In 2019 a Python plugin package, *b2cal*, based on the Apache Airflow workflow management platform was developed to automate the calibration at Belle II. It uses Directed Acyclic Graphs to describe the ordering of processes and Flask to provide administration and job submission web pages. This system was hosted in Melbourne, Australia and submitted calibration jobs to the High Energy Accelerator Research Organization in Japan. In 2020 the *b2cal* was dockerised and deployed at the Deutsches Elektronen-Synchrotron Laboratory in Germany. Improvements have been implemented that allow jobs to be submitted to multiple calibration centers and for the amount of required human interactions to be greatly reduced. All job submissions and validation of calibration constants now occur as soon as possible. In this paper, we describe the upgrades to the automated calibration at Belle II.

1 Introduction

In March 2019 the Belle II detector [1] began collecting data from the SuperKEKB [2] electron-positron collider at the High Energy Accelerator Research Organization (KEK) in Tsukuba, Japan. Belle II's instantaneous luminosity is continuously increasing and is planned to eventually reach $8 \times 10^{35} \text{ cm}^{-2}\text{s}^{-1}$, forty times the instantaneous luminosity of its predecessor. The resulting data must be calibrated promptly so that it can be properly reconstructed for timely physics analyses.

The calibration procedure at Belle II consists of a large number of interdependent steps that can be extremely inefficient to perform manually. In 2019 a Python plugin package called *b2cal* [3], using the open-source Apache Airflow [4] software, was developed to automate the calibration and provides a common interface for calibration experts. Its main function was to manage the complex calibration workflow and run jobs using the Belle II Analysis Software Framework [5] (*basf2*) at remote sites. In addition, the Calibration Framework [6] (CAF), a set of Python and C++ modules and classes, was also developed in *basf2*. It provides an API to easily parallelize job submission of *basf2* processes on different batch system backends and is used by most calibrations at Belle II.

2 Calibrations at Belle II

Calibration of Belle II data consists of two procedures, prompt calibration and reprocessing. Prompt calibration is the best first attempt at calibrations and produces calibration constants from data collected in a two week period. These calibration constants are called payloads

*e-mail: fpham@student.unimelb.edu.au

and are usually ROOT [7] files containing custom classes used during the reconstruction of events to account for the detector conditions. The prompt datasets are produced continuously and are usable for preliminary results. Reprocessing is performed over a large dataset once or twice per year and aims to produce the best calibration constants possible. These new calibration constants are then used to produce publication datasets.

2.1 Data Processing

During data taking, the High Level Trigger (HLT) flags event passing various selections that would be useful for calibration. Calibration skims are automatically produced from raw data by writing out copies of events based on this HLT flag and transferred to the appropriate calibration site using Rucio [8], a Distributed Data Management system. The calibration skims for each prompt calibration requires approximately 10 TB of storage. These skims are kept for about 6 months before being discarded.

2.2 Calibration overview

A full calibration loop is divided into 5 steps with each step depending on the previous one:

1. **Local calibrations:** Calibrations that are performed manually without `b2cal`.
2. **Pre-tracking calibrations:** Calibrations that run on raw collision data and do not require the most precise track reconstruction.
3. **Alignment:** Requires raw collision data along with local and pre-tracking calibrations.
- (3.5) **Production of cDST files:** Track reconstruction is the most computationally demanding part of the reconstruction and as such files called cDSTs which contain reconstructed track objects are centrally produced from raw files. These cDSTs are the inputs for the post-tracking calibrations.
4. **Post-tracking calibrations:** Calibrations that require precise track reconstruction, and are therefore depending on the alignment.
5. **Analysis-based calibrations:** These calibrations rely on high quality data, and are therefore run last.

Each prompt calibration produces about 13 TB of cDSTs and 350 GB of payloads. Within the calibration steps 2 – 5 there are dependencies between calibrations. To help coordinate all the calibration tasks, the experts are assigned issue tracker tickets in the Jira system [9]. Jira tickets have been successful in managing and monitoring the overall prompt calibration procedure.

3 Improvements to Airflow

Apache Airflow is an open-source Python package created by the company Airbnb to programmatically author, schedule and monitor dynamic workflows as Directed Acyclic Graphs (DAGs) of tasks. A DAG is defined in Python and it contains one or more instances of an *operator* class. Several new DAGs have been created to perform asynchronous processing of tasks at the reprocessing center, Deutsches Elektronen-Synchrotron Laboratory (DESY), and the prompt calibration center, Brookhaven National Laboratory (BNL). The new DAGs include:

- BNL calibration DAGs that are fully automated as shown in Figure 1. When a calibration is finished the next calibration will immediately start. Conversely, at DESY the calibration DAGs require that the experts manually sign-off on the payloads.
- DAGs to monitor the transfer of calibration skims to the calibration centers via queries to a grid metadata catalog system called AMGA [10].
- DAGs that submit a validation job after a calibration is finished. The validation results, usually plots, are attached to the Jira ticket and are used by the expert to check if the payloads are reasonable.

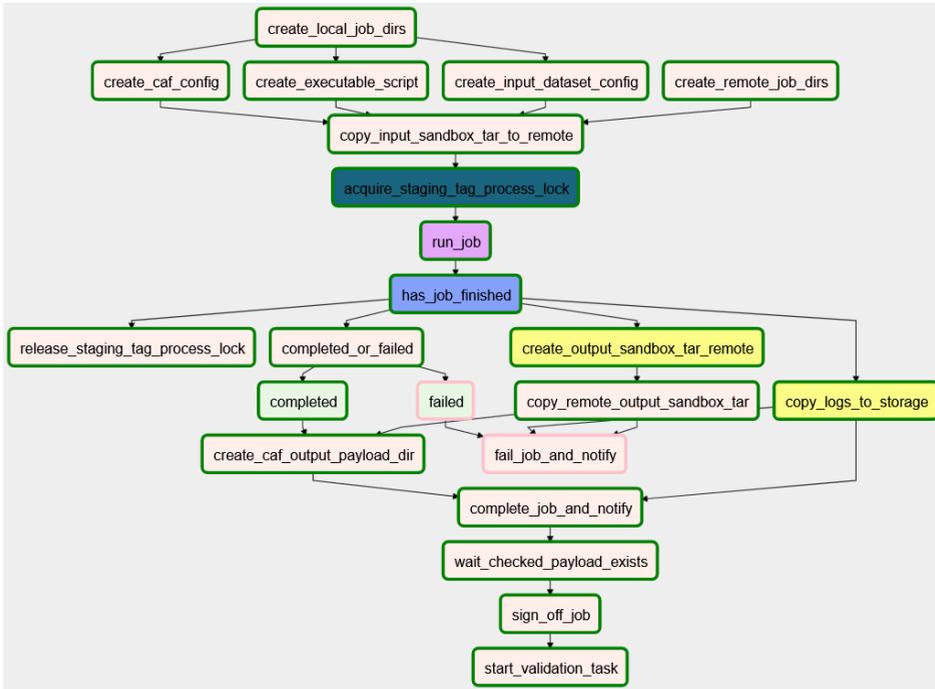


Figure 1. The BNL calibration DAG. A green border is a completed process while a pink border is a skipped process. Dark and light blue processes are sensors that waits for an upstream process to return a specific value. The DAG now signs off by itself and starts a validation job.

The records of any Airflow process such as the triggering of a DAG are stored in a MariaDB SQL database named `airflow`, while the database for the calibrations is stored in one named `calibration`. The `calibration` database is defined using SQLAlchemy [11] which provides an object-oriented way of interacting with databases. SQLAlchemy uses an Object Relational Mapper to create an object-oriented wrapper around the database connection so that database queries can be written using object-oriented Python. To modify/interact with the databases more efficiently, two tools, Alembic and Adminer, have now been incorporated and they are used extensively.

Alembic [12] is a lightweight database migration tool for SQLAlchemy. It is used to change the schema of a database, e.g. modifying a column or adding a table. It uses migration scripts that can be automatically generated or manually created to track any changes made to the Python classes defining the database tables so that a complete history of migrations is available to view and revert to. The migration scripts simplify the process of making any

changes to the database. These migration scripts are stored in the source code making them useful for bookkeeping. If a change is problematic or is no longer required, it is easy to roll back to a previous version. In addition, DAGs have been created to regularly backup the database, so that any data lost during a migration can be restored if necessary. Adminer [13] is a lightweight PHP website for database administration. It is used to directly interact with the MariaDB database as the root user, to change access rights and fix problems without the need to use the command-line or the calibration website. Adminer is only accessible from the localhost by an admin, it is not publicly accessible via the internet.

The Airflow Web Server along with the database was moved from Melbourne to DESY in October 2020. To streamline the process of migration the *b2cal* package was converted to run within a Docker [14] container. The *b2cal* package now has two main components:

1. The scheduler which handles submitting DAG tasks
2. The web-server which runs the website. It is a Flask process run by Gunicorn [15]

These are run as two Docker Compose services in separate containers. Both communicate to the database service but their ports are not accessible from outside Docker directly. They also have the host machine's Docker socket bind-mounted into them. This allows these containers to run Docker containers as if they were running on the host machine e.g. the *basf2* container. Although the *airflow-webserver* service provides the Gunicorn processes to handle incoming HTTPS requests, it does not provide a way to handle public web access to the Flask website in a production environment. NginX [16] provides a production-grade "reverse-proxy" to the Airflow web-server. By doing this we have created a containerized website with a separate task scheduler.

Previously X.509 certificates were used for authentication but this was not ideal as experts could not easily access the calibration website from a different device. To allow convenient and secure access to the calibration website, an internal DESY LDAP server is now incorporated into Airflow. Its inclusion brings the website inline with the rest of Belle II's collaborative services.

In the 2019 calibration loop, when a processing request is started, experts can only submit a calibration when all its dependencies were completed, e.g. pre-tracking calibrations can only begin when all local calibrations are finished. Then the expert would have to wait until their calibration was finished to validate their payloads and manually sign-off, allowing the next calibration to begin. Although this procedure was very well organized, there was often days of inactivity waiting for experts to sign-off or start their jobs. In 2020 when a processing request is started, experts have a given amount of time (e.g. 48 hours) to log in via their DESY accounts to the calibration website and submit their calibration jobs. If they fail to manually submit, their jobs will be submitted automatically. This automated submission uses new default settings called *input_data_filters* that all calibration scripts are now required to have. Experts may want to submit manually so that they can change the default values of their calibration to produce better calibration constants. This is achieved via a new optional *expert_configs* dictionary that can be added to the calibration scripts. The *expert_configs* have default values for the variable that can be overwritten from the command line or via the job configuration files. Experts can easily modify their scripts with a different *expert_configs* input for each calibration during job submission on the calibration website instead of having to create slightly different scripts for every calibration.

The calibration jobs will not start until the time limit is reached. When calibrations are finished the experts are notified on the Jira ticket, validation scripts are run over the calibration payloads, and the validation results are made available to download. The calibration then automatically signs off and allows any downstream tasks to begin. If the job failed, the overall

parent process logs and error outputs are available on the calibration website. At BNL, with valid certificates (X.509), all calibration outputs including individual job logs are accessible via file explore from a browser using BNLBox [17]. All experts have access to DESY, so this is not necessary as they can view the calibration outputs directly. If an expert believes payloads are not acceptable, they can download them from the website, make modifications and then upload the modified payloads to the website.

The cDST files are produced on HTCondor [18] worker nodes at BNL and DESY, and copied onto magnetic tape data storage using the XRootD `xrdcp` command. At BNL these files are registered to DIRAC [19] using Rucio which then automatically copies the cDSTs to DESY. These cDST files can be downloaded by anyone in the Belle II Collaboration with a valid grid certificate.

4 Experience in 2020

Calibration with Airflow was successfully used in 2020 for multiple prompt calibrations at KEK Computing Centre (KEKCC) and BNL along with a subsequent reprocessing at DESY. The calibration website is used by about 20 calibration experts, who use the website to submit, monitor and validate their calibrations. It is accessed approximately 100 times per prompt calibration, and the feedback from the calibration experts has been very positive. The experts find the website easy to use and fully support the automated calibration as they no longer have to manually submit jobs and have significantly more time to check their calibration results. The time taken for each full prompt calibration loop has decreased as experts become more experienced with using the website. The `b2cal` package also now features extensive documentation using Sphinx as shown in Figure 2 that is part of the calibration website. The experts can use this to learn more about `b2cal`, as well as get updated on the latest developments.

The automated calibration has been successful at minimizing the number of human errors, as previous mistakes such as an expert accidentally using the wrong data files in a script is eliminated. This is because `b2cal` enforces that only valid inputs and configurations can be used in each calibration. In addition, the validity of each payload is checked and only payloads that pass this check can be used.

In 2020, job submission has been improved. KEKCC uses the LSF [20] backend while BNL and DESY use HTCondor. A significant refactoring of submission scripts was done in order to be able to submit seamlessly to both backends. After the migration to BNL, job failures sometimes occurred when thousands of jobs were simultaneously attempting to access data via XRootD [21]. To solve this, HTCondor jobs use five times the default values for XRootD timeout and retry attempts, 10 minutes and 25 retries, respectively. Jobs submissions were spaced over a longer period to avoid overloading the system and HTCondor was configured to automatically retry failed jobs after 30 minutes. These changes have led to job failures being eliminated. As a result, the average time for a full calibration loop has decreased from 18 days in 2019 to 9 days in 2020/2021.

The calibration loop is overseen by managers. ‘Management’ web pages have been created, allowing managers greater control over the calibration loop. These web pages let managers change the assigned expert, view the status of all jobs, and prevent certain calibrations, e.g. calibration of the electromagnetic calorimeter, from being included in the calibration loop that was outlined in section 2.2. Additional web pages will need to be created to provide managers with more advanced permission such as restarting jobs.

Eventually, the quantity of incoming raw data will be too large for any one centre to reasonably store and calibrate them. In 2023-2024 the calibration will use distributed computing (grid) to store all the calibration skims and run the calibration jobs. `b2cal` only requires minor

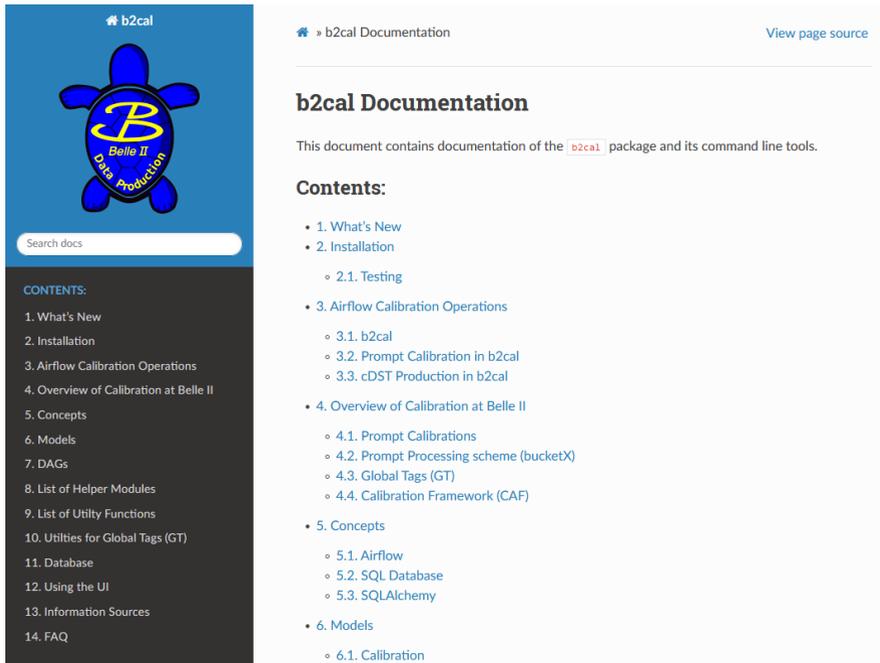


Figure 2. Documentation of the b2cal package via Sphinx. Useful for informing calibration experts of any changes.

changes to accomplish this as the bulk of the work is on the distributed computing group and CAF (Calibration Framework).

5 Summary

The automated calibration of Belle II for prompt calibration and reprocessing has been achieved on 2019-2020 data via b2cal, a Python package based on Apache Airflow. The calibration center has successfully been moved from KEKCC to BNL for prompt calibration and DESY for reprocessing. The CAF backend and *b2cal* have been upgraded to handle HTCondor backends, XRootD file access and copying, along with BNLBox integration for expert monitoring. In addition, Rucio is used to monitor the transfer of raw data files to BNL and automate the replication of cDSTs to DESY. The webserver has been dockerised for streamlined development of new features and migrated to DESY. The DESY internal LDAP servers have been integrated for single sign-on. Alembic has been utilized for developing and migrating database. Details about the operation of *b2cal* have been documented in Sphinx.

The main development for the future will be to include more sophisticated automated validation and monitoring of calibration jobs, the ability to run multiple processings simultaneously and more web pages for managers. In addition, in the near future, as the amount of data collected increases rapidly and the computing resources required become too large for a single site, calibrations will need to shift to a distributed computing environment. Automating the calibration process has proven to be an essential part of the fast, efficient and reliable processing of physics data.

References

- [1] T. Abe *et al.* [Belle-II], [arXiv:1011.0352 [physics.ins-det]].
- [2] K. Akai *et al.* [SuperKEKB], Nucl. Instrum. Meth. A **907**, 188-199 (2018) doi:10.1016/j.nima.2018.08.017 [arXiv:1809.01958 [physics.acc-ph]].
- [3] D. Dossett *et al.* [Belle-II], EPJ Web Conf. **245**, 02016 (2020) doi:10.1051/epjconf/202024502016
- [4] *Airflow*, <https://airflow.apache.org/> (2021), accessed: 2021-05-02
- [5] T. Kuhr *et al.* [Belle-II Framework Software Group], Comput. Softw. Big Sci. **3**, no.1, 1 (2019) doi:10.1007/s41781-018-0017-9 [arXiv:1809.04299 [physics.comp-ph]].
- [6] D. Dossett *et al.* [Belle-II Software Group], “Status of the calibration and alignment framework at the Belle II experiment,” J. Phys. Conf. Ser. **898**, no.3, 032032 (2017) doi:10.1088/1742-6596/898/3/032032
- [7] R. Brun and F. Rademakers, “ROOT: An object oriented data analysis framework,” Nucl. Instrum. Meth. A **389**, 81-86 (1997) doi:10.1016/S0168-9002(97)00048-X
- [8] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, D. Cameron, D. Christidis, D. Ciangottini, G. Dimitrov and M. Elsing, *et al.* “Rucio - Scientific data management,” Comput. Softw. Big Sci. **3**, no.1, 11 (2019) doi:10.1007/s41781-019-0026-3 [arXiv:1902.09857 [cs.DC]].
- [9] *JIRA*, <https://www.atlassian.com/software/jira/> (2021), accessed: 2021-05-03
- [10] A. Tsaregorodtsev *et al.* [DIRAC], “DIRAC file replica and metadata catalog”, J. Phys. Conf. Ser. **396** (2012), 032108 doi: 10.1088/1742-6596/396/3/032108
- [11] *SQLAlchemy*, <https://www.sqlalchemy.org/> (2021), accessed: 2021-05-03
- [12] *Alembic*, <https://alembic.sqlalchemy.org/> (2021), accessed: 2021-05-04
- [13] *Adminer*, <https://www.adminer.org/> (2021), accessed: 2021-05-04
- [14] *Docker*, <https://www.docker.com/> (2021), accessed: 2021-05-04
- [15] *Gunicorn*, <https://gunicorn.org/> (2021), accessed: 2021-05-04
- [16] *NginX*, <https://www.nginx.com/> (2021), accessed: 2021-05-04
- [17] O. Rind, H. Ito, G. Che, T. Chou, R. Hancock, M. Karasawa, Z. Liu, O. Novakov, T. Rao and Y. Wu, *et al.* “The BNLBox Cloud Storage Service,” EPJ Web Conf. **245**, 04011 (2020) doi:10.1051/epjconf/202024504011
- [18] B. Bockelman, T. Cartwright, J. Frey, E. M. Fajardo, B. Lin, M. Selmecci, T. Tannenbaum and M. Zvada, J. Phys. Conf. Ser. **664**, no.6, 062003 (2015) doi:10.1088/1742-6596/664/6/062003
- [19] H. Miyake *et al.* [Belle-II computing group], “Belle II production system,” J. Phys. Conf. Ser. **664** (2015) no.5, 052028 doi: 10.1088/1742-6596/664/5/052028
- [20] *LSF*, <https://www.ibm.com/au-en/marketplace/hpc-workload-management> (2021), accessed: 2021-04-06
- [21] L. A. T. Bauerdick *et al.* [CMS], J. Phys. Conf. Ser. **513**, 042044 (2014) doi:10.1088/1742-6596/513/4/042044