Distributed data management on Belle II

Siarhei Padolski[1,*], *Hironori* Ito[1], *Paul* Laycock[1], *Ruslan* Mashinistov[1], *Hideki* Miyake[2] *and Ikuo* Ueda[2]

[1]Brookhaven National Laboratory, Upton, NY, USA
[2]High Energy Accelerator Research Organization (KEK), Japan

Abstract. The Belle II experiment started taking physics data in April 2018 with an estimated total volume of all files including raw events, Monte-Carlo and skim statistics of 340 petabytes expected by the end of operations in the late-2020s. Originally designed as a fully integrated component of the BelleDIRAC production system, the Belle II distributed data management (DDM) software needs to manage data across about 29 storage elements worldwide for a collaboration of nearly 1000 physicists. By late 2018, this software required significant performance improvements to meet the requirements of physics data taking and was seriously lacking in automation. Rucio, the DDM solution created by ATLAS, was an obvious alternative but required tight integration with BelleDIRAC and a seamless yet non-trivial migration. This contribution describes the work done on both DDM options, the current status of the software running successfully in production and the problems associated with trying to balance long-term operations cost against short term risk.

1 Introduction

The Belle II [1] experiment on the SuperKEKB[2] accelerator at KEK (Tsukuba, Japan) aims to find evidence of New Physics in the flavour sector, beyond the Standard Model. This is an intensity frontier measurement focused on collecting a large data sample, 50 ab^{-1}, of electron-positron collisions. This challenging research program implies significant requirements on the ability to store, manage and access both experimental and simulated data. The Belle II Distributed Data Management system (DDM) successfully tackles this problem. This paper discusses its current state and further evolution.

2 Current Distributed Data Management system

The Belle II distributed computing system [3,4] overview is presented in Fig. 1. There are four principal components in the diagram: Sites, the DIRAC Interware [5], its extension for Belle II named as BelleDIRAC[4,6,7] and grid services which are third party components such as the LCG File Catalog (LFC) [8], AMGA [9-11] and FTS [12].

* Corresponding author: spadolski@bnl.gov

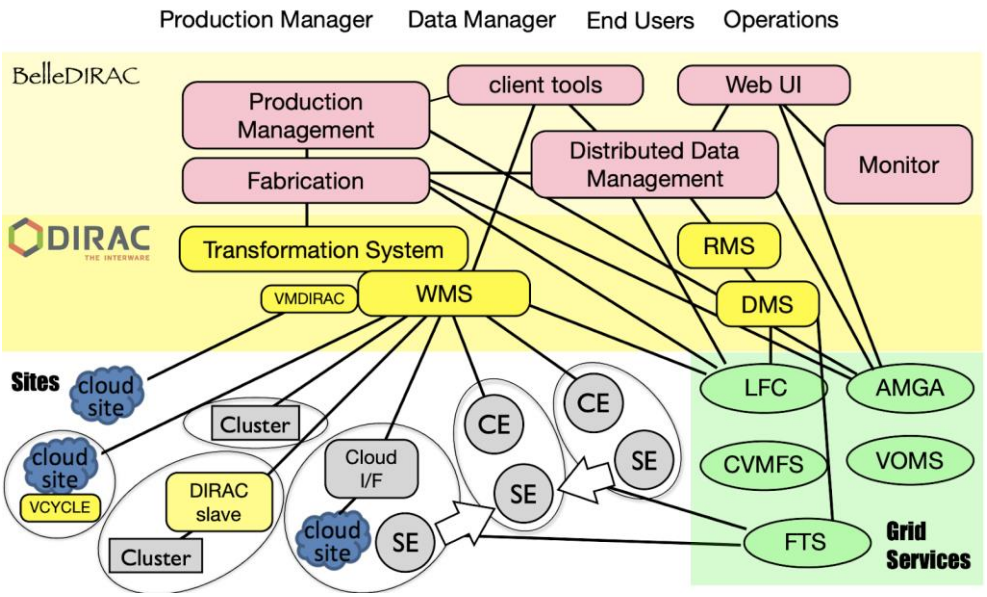

Fig. 1. The Belle II distributed computing system.

Sites provide two basic resources: Computing Elements (CE) and Storage Elements (SE). All direct file transfer operations on 29 registered SEs are executed by the FTS service with a DIRAC component submitting requests to it. Each file located on a Belle II SE is registered in LFC. Information about files is stored in the AMGA metadata catalog. To simplify data handling a moderate number of files is organized into a logical container called a datablock. It is the primary data unit used in DDM operations while DIRAC treats operations on the file level.

A data operation request issued by the Fabrication system, a component of the Belle II production system, is the most common use case for the DDM. The request can also be initiated by a user or another Distributed Computing system component. Both deletion and replication operations are executed by the Belle II Distributed Data Management system (top level) but they have different workflows. File deletion is performed directly by a Belle II DDM component, the DDM deletion agent, with the GFAL2 library [13] which takes care of the interactions with the target SE. Parallelism of deletion operations is provided by multiple instances of the DDM deletion agents, each of them configured for a particular set of SEs. Details of the DDM internal components are presented in Fig.2

The file replication process has a more advanced workflow. The Belle II DDM submits replication operation requests to the DIRAC Request Management System (RMS). Once a request has been submitted to RMS, it schedules the operation for the DIRAC Data Management System which submits corresponding file transfer jobs to FTS. This design allows the use of DIRAC components for communication with FTS and monitoring the progress of transfers. Another advantage of the implemented approach is compliance with the DIRAC configuration paradigm where the FTS backend is considered as a system-wide resource wrapped by the corresponding interware API making it accessible in a distributed fashion.

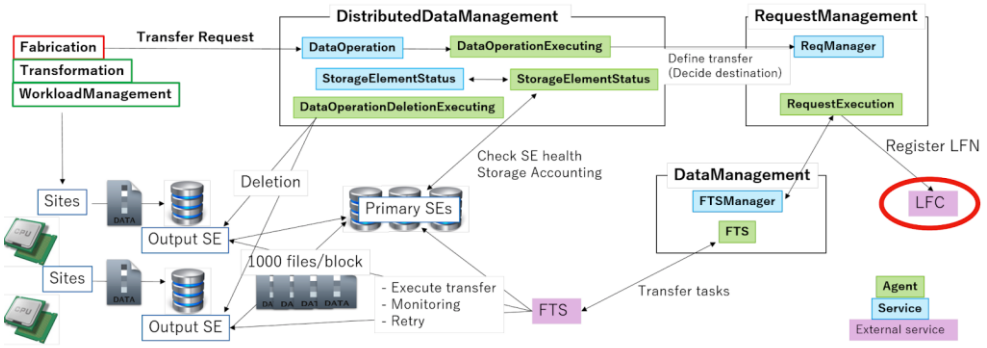

Fig. 2. The Belle II Distributed Data Management System (DDM) and its interaction with the other components, such as the Fabrication system that makes transfer requests to DDM, the DIRAC RequestManagement and DataManagement systems which interact with FTS and LFC, and the SEs hosted at the sites. FTS performs file transfers between SEs, while DDM executes file deletion at the source SEs after transfers.

The implemented architecture also provides tight integration with the DIRAC Interware at the level of the database, resource status management, file catalogue (FC), authentication, component configuration and execution, and remote procedure calls. It also provides high performance deletion and replication scheduling.

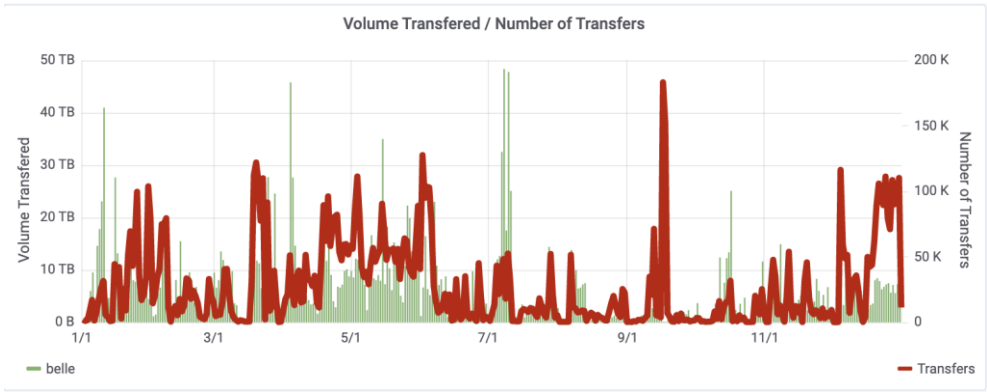

Fig. 3. Daily transfers performed by the Belle II DDMS during 2019 using the FTS service.

As shown in Fig. 3 the maximum daily volume of transferred data was about 50TB with 0.2M files at peak.

3 Further development

The Belle II DDM is a custom solution which suffers from a lack of functionality in comparison to state-of-the-art DDM projects and it would need continuous development effort to fix these problems. In particular, alignment of data across SEs according to the distributed computing plan currently requires a lot of operational effort, such as:

- often needing to investigate data operations that are stuck and affecting the whole system,
- manually triggering data distribution,
- manually choosing datasets to be removed,

- manually fixing and adjusting data distribution.

Another sufficient development necessary for keeping the current DDM updated and matching new BelleDIRAC releases is migration of the code to Python 3. An alternative is to adapt a solution that is driven and supported by a larger community, and Rucio [14] is a strong candidate replacement. Rucio provides advanced functionality not yet available in the Belle II DDM such as improved automation, data subscription, policies, permissions control, a variety of monitoring tools and more. Rucio also provides its own file catalog which could replace the LFC.

The first step of the migration proposal is presented in Fig. 4.

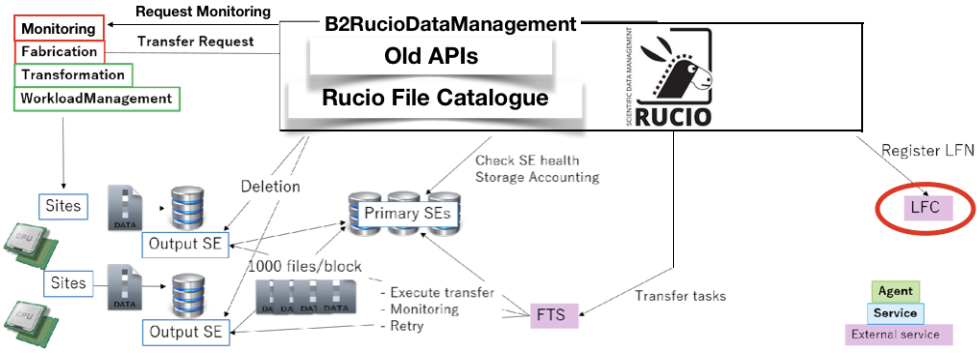

Fig. 4. The Belle II Distributed Data Management System at the initial step of migration to Rucio.

As an initial step, a wrapper layer will be developed which hides the actual Rucio API and provides a set of functions and object states supported by the current DDM. Rucio serves as a data transfer engine with its own FC. A special synchronization functionality will be used to keep the system wide LFC catalogue updated. At the time of CHEP 2019 the performance of the Rucio installation at BNL, using Belle II filename conventions and a PostgreSQL database backend, had been shown to be capable of exceeding the requirements of the experiment. Prototypes of key DDM APIs had been written and were capable of submitting tasks to FTS via Rucio and a series of checkpoints had been defined for completing the stage 1 migration.

Once a production grade Rucio based solution is established a deeper integration is planned (Fig. 5). At this step the BelleDIRAC components, including Monitoring and the Fabrication system, could start using more Rucio interfaces exploring the full power of this system. Another significant implication at this step is to completely retire the LFC catalogue providing a new DIRAC file catalogue plugin to allow the Rucio file catalogue to be used everywhere.

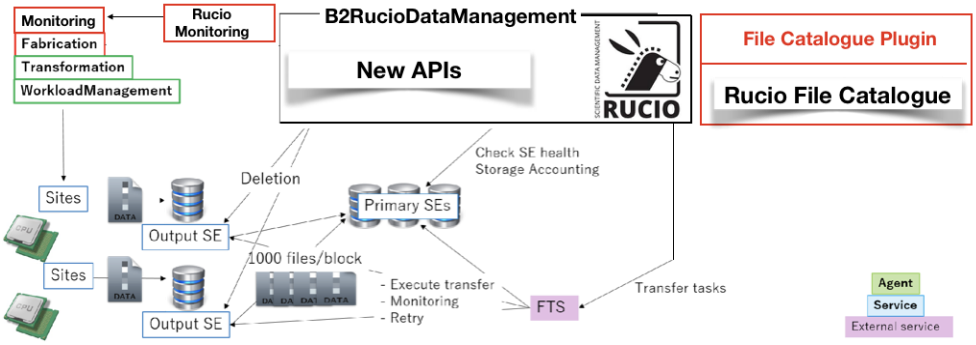

Fig. 5. Belle II Distributed Data Management System at the second step of migration plan.

Successful accomplishment of this migration plan will deliver a full functioning Rucio based DDM solution for Belle II. It also opens further perspectives for the activation of many features which Rucio contains out-of-the-box, such as user quotas or lifetime. This development is foreseen in Step 3.

4 Conclusion

In this paper, the current Belle II DDM system was described. This system successfully provides file management operations for the experiment's data production and processing needs. Due to a lack of features and the need for continuing development, a plan to migrate to a Rucio-based DDM system is being considered. An overview of this migration plan was also presented in this paper.

References

1. T. Abe, et al., KEK-REPORT-2010-1, arXiv:1011.0352 (2010)
2. K. Akai, et al., Nucl. Instrum. Meth. A **907**, 188-199 (2018)
3. T. Hara and Belle II computing group, J. Phys.: Conf. Ser. **664**, 012002 (2015)
4. Y. Kato et al., *PoS*, **KMI2017**, 024 (2017)
5. A. Tsaregorodtsev, V. Garonne and I. Stokes-Rees, *5th IEEE/ACM International Workshop on Grid Computing*, 19-25 (2004)
6. F. Stagni et al., J. Phys. Conf. Ser. **898**, 092020 (2017)
7. H. Miyake et al., J. Phys.: Conf. Ser. **664**, 052028
8. J.-P Baud, J. Casey, S. Lemaitre, C. Nicholson, *IEEE International Symposium on High Performance Distributed Computing*, 91-99 (2005)
9. B. Koblitz, N. Santos, V. Pose. J Grid Computing **6,** 61–76 (2008)
10. Geunchul Park et al., J. Phys.: Conf. Ser. **664**, 042030 (2015)
11. Jae-Hyuck Kwak et al., J. Phys.: Conf. Ser. **664**, 042041 (2015)
12. P. Baldino, P. Z. Kunszt, G. McCance. *Proceedings of the 5th International Conference on Computing In High Energy and Nuclear Physics*, 685-688 (2006)
13. Gfal2 Project. https://dmc.web.cern.ch/projects-tags/gfal-2
14. M. Barisits, T. Beermann, F. Berghaus, et al., Comput Softw Big Sci, **3**, 11 (2019).