The Belle II Online–Offline Data Operations System

Matthew Barrett¹ · Takanori Hara^{1,2} · Michel Hernández Villanueva³ · Kunxian Huang⁴ · Dhiraj Kalita¹ · Petteri Kettunen¹ · Prashant Shingade⁵

Received: 29 July 2020 / Accepted: 9 November 2020 / Published online: 2 January 2021
© The Author(s) 2021

Abstract

Data collection at the Belle II experiment started in the spring of 2019. During the early stages of the experiment it is important that the raw data are both copied to permanent storage and made available soon after being recorded to allow for the timely commissioning and calibration of the detector. Automated procedures have been developed to transfer the data from the detector in a timely manner; these procedures include fault management, performance monitoring, and quality checks. It is important that the systems put in place will also scale to the much higher data rates expected in the coming years at Belle II. The development, implementation, and operations of the Belle II online–offline data transfer system will be described.

Keywords Belle II · Data management · Data transfer operations · Monitoring

Introduction

Belle II is a particle physics experiment running at the High Energy Accelerator Research Organization, known as KEK, laboratory in Tsukuba, Japan. Belle II is the successor to the Belle experiment, which collected data from 1999 to 2010. The goal of Belle and Belle II is to investigate CP violation and search for new physics by making high precision measurements of the decays of B mesons, along with programmes in charm, tau, and dark sector physics [1]. Detailed descriptions of the design and the physics programme of the experiment can be found in references [2] and [3], respectively.

The Belle II experiment records the collisions of electrons and positrons provided by the SuperKEKB accelerator. To

achieve its physics goals Belle II aims to record data with an integrated luminosity¹ of 50ab^{-1} [4], which is more than 50 times that achieved by its predecessor, Belle. This corresponds to a storage requirement of ~ 60 petabytes for a single copy of all of the raw data; these data are stored using the ROOT [5] format developed by CERN. All of the data have to be transferred from the Belle II detector to permanent storage, from where they can be distributed, in such a way as to be available to Belle II analysts worldwide. An automated data transfer system has been developed to manage the movement of data.

This data transfer system also interacts with the distributed computing system of the Belle II experiment which uses *DIRAC* [6] as its distributed computing framework [7]; this framework is also widely used by many other scientific experiments. Two complete copies of the raw data are maintained at all times: one at KEK, and a second at Brookhaven

✉ Matthew Barrett
mattb@post.kek.jp

¹ Institute of Particle and Nuclear Studies, High Energy Accelerator Research Organization (KEK), 1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan

² School of High Energy Accelerator Science, The Graduate University for Advanced Studies, Shonan Village, Hayama, Kanagawa 240-0193, Japan

³ The University of Mississippi, University, Mississippi 38677, USA

⁴ Department of Physics, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan (R.O.C.)

⁵ Tata Institute of Fundamental Research (TIFR), Mumbai 400005, India

¹ For particle physics experiments the size of a dataset recorded is usually described by an integrated luminosity—this is a quantity dependent only on the operational parameters of the accelerator and is independent of any physics processes being studied. The storage size of the dataset (in, for example, petabytes) is approximately proportional to the integrated luminosity recorded by the experiment. To describe the rate of interactions produced by the accelerator, the closely related quantity instantaneous luminosity is used; this quantity is approximately proportional to the data rate (in, for example, GB/s). Deviations from these approximately proportional relationships are expected during periods when additional calibration or debugging data are collected.

National Laboratory (BNL), Upton, New York, USA [8]. Due to the potential for earthquakes and related risk factors it was decided to keep the second copy outside of Japan, and further decided that holding this second copy would be a shared responsibility between Belle II member countries that have computing centres able to host the raw data. Thus, from the fourth year of Belle II operations, the second copy will be split between BNL and computing sites in Italy, Germany, Canada, and France.

This document describes the details of the Belle II data transfer system which has been developed to meet the needs of the experiment, ensuring a stable, fault tolerant, and maintainable system, and the monitoring tools developed along with the system. Important interactions of the data transfer system with the distributed computing system are described, however details of the distributed computing system are beyond the scope of this document.

"[Belle II Running Phases](#)" describes the commissioning and early data collection of Belle II; "[Belle II Online Offline Data Processing](#)" introduces the data transfer system, while "[Data Pipeline Implementation](#)" details the implementation of the system. "[Raw Data Registration and Replication](#)" describes how the raw data files are registered and made available via the grid; "[Data Transfer Monitoring](#)" and "[Monitoring and Bookkeeping](#)" cover details of the monitoring and bookkeeping tools that have been developed; and DevOps are described in "[DevOps](#)". Future plans and conclusions are covered in "[Future Plans and Outlook](#)" and "[Conclusions](#)" respectively.

Belle II Running Phases

The commissioning and operation of Belle II and SuperKEKB was divided into three phases, with an initial goal to measure the levels of radiation that the components of Belle II would be exposed to during operation of SuperKEKB, and ensure that the experiment would be able to operate in these conditions.

- *Phase 1* took place in 2016 to commission the SuperKEKB accelerator. The Belle II detector was not involved in Phase 1, instead a commissioning detector [9] was used to measure radiation levels in the space that Belle II would occupy.
- *Phase 2* took place in 2018 after the Belle II detector had been moved into its operating location. This phase was used to commission both SuperKEKB and Belle II; data were recorded but the innermost component (and thus the component that will receive the highest dose of radiation) of Belle II, the *vertex tracker*, was not installed in Phase 2.

- *Phase 3*, beginning in early 2019, marked the start of full data taking operations of Belle II including the newly installed vertex tracker. Phase 3 was further subdivided into *Runs*, with the first being the *Spring 2019 Run* followed by the *Autumn 2019 Run*.

The instantaneous luminosity delivered by the SuperKEKB accelerator will increase significantly from that achieved at the start of Phase 3, continuing to increase throughout the years of operation. Information on the recorded and projected luminosity is updated regularly and the latest information can be found on the SuperKEKB web site [10]. The size of the data will scale approximately linearly with the luminosity, with higher luminosities requiring both higher data transfer rates, and placing a greater demand on the data processing pipeline, storage, and post-processing.

During the Spring 2019 Run, Belle II recorded a dataset with an online integrated luminosity of 6.49 fb^{-1} [11], corresponding to 0.013% of the goal of 50 ab^{-1} [4]. The maximum instantaneous luminosity reached during data taking was $6.1 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ [12], or 1.0% of the target instantaneous luminosity of $6.0 \times 10^{35} \text{ cm}^{-2} \text{ s}^{-1}$ [4]. The total storage size for all of these data in ROOT format is 216 TB; due to storing additional events for commissioning and calibrating the detector this is a far higher storage size per unit luminosity than will be observed for future data taking.

The methods used during Phase 2 relied on manual transfers and bookkeeping when transferring data from the detector's data acquisition (DAQ) system to permanent storage. It was already known prior to the start of Belle II Phase 3 operations that these methods would not scale to the higher data rates that are expected during the later stages of operations. This necessitated the development of a new data transfer system to be deployed during the early part of Phase 3.

Belle II Online Offline Data Processing

During operations, the Belle II detector systems accumulate more data than can be recorded. To reduce the data size a trigger system is used that selects data of importance to physics analysis to be stored. The data are recorded as events, with each event corresponding to one or more collisions between the electron and positron beams during a single crossing of the two beams. One event is the smallest unit of data that is recorded.

The Belle II DAQ system is designed to handle a trigger rate of up to 30 kHz, i.e. up to 30,000 events per second, with a raw event size of up to 1 MB [13, 14]—this translates to a data flow rate of 30 GB/s at the hardware based first level trigger system. The experiment uses a pipelined trigger flow control system to select data of interest to be stored, including an Event Builder which passes events to

Fig. 1 Schematic presentation of Belle II online to offline data operations

the software based High Level Trigger (HLT) system which then makes the final decision of whether to store the data from a particular event. Once stored, in ROOT files, the final size of raw events is 60–80 kB/event.

There are two modes of operation used by the HLT: filtering, where only events passing defined criteria are stored; and monitoring, where all events reaching the HLT are stored, and the result of the filtering logic is recorded but not applied. The HLT was run in monitoring mode during the early stages of Phase 3 and before as the additional events were important for calibration and validation—the HLT was used in filtering mode for the latter part of the Spring 2019 Run, and this mode will be used for future Runs, as its use will be essential to maintain a manageable data size.

Selected events are written to HLT storage, which is located in the Tsukuba Experimental Hall, close to the Belle II detector. The data will be written on the storage system at a rate of up to 3 GB/s. At the start of Phase 3 operations, the HLT system had five servers; four additional servers were added, to take the total to nine, with plans to increase this number further to meet the demands of increased capacity when the accelerator is running at its maximum design luminosity.

The data that are recorded are divided into four types:

- *Physics* Data intended for physics analysis.
- *Cosmic* Interactions of cosmic rays within the detector recorded during times when the accelerator is not running. These are used to calibrate detector systems.
- *Beam* Data collected while the parameters and conditions of the particle beams are being modified or studied. These may be used for calibration and beam background studies.
- *Debug* Data collected to debug detector systems; generally useful only in the short term.

The data type is set by the DAQ system during data taking. All types of data are preserved, however each of the different types have different policies that apply to them: for example Physics data must be easily accessible to the Belle II collaboration until the end of data taking and beyond (data analysis may continue for a decade or more after the end of detector operations), whereas after a short time Debug data will be archived to tape only and thereafter will not be quickly available. The majority of data recorded are Physics data and references to data in this document refer to Physics data unless stated otherwise.

Data are collected in runs with each individual run lasting from several minutes to several hours. Running periods of several weeks or more, during which there are no major changes in accelerator or detector conditions, are called experiments. Each run is of a single type (Physics, Cosmic, Beam, or Debug) and has an experiment number and run number associated with it, which together can be used to uniquely identify the run. Data from a run are stored as a number of files, with the longest duration runs having several thousand files.

A schematic overview of the Belle II data flow is shown in Fig. 1. The computing systems are divided into online and offline domains. The online domain covers all of the systems, typically located very close to the detector, that have to run in real-time, or near real-time, to collect and record the data during Belle II operations. The offline domain covers all other systems. When data are transferred from temporary storage located close to the detector to other storage locations, as described in the next paragraph, this marks the transition between the online and offline systems.

Once a run has finished, the files for the run on the HLT storage can be made available to be transferred to the offline operations domain facility located 1.2 km away on the same campus. Offline Worker Node (WN) servers hosted in the

KEK Computing Research Center (KEKRC) pull the raw data from the HLT servers via a dedicated data link. Currently there are nine WNs commissioned matching the number of HLT servers in production. This one-to-one HLT–WN correspondence will be kept as the number of HLT servers increases to guarantee the data transfer and processing capability.

Although WN servers by default process data from one assigned HLT only, they can be configured to process data from any HLT source. This ensures uninterrupted operations even in the case where one or more WNs go out of service.

Each of the HLT servers has storage for raw data with ten partitions of 9 TB capacity each. Data are written to a given partition with the aim that data transfer takes place once a partition is full and writing has moved to a new partition. A file, known as a `list_send` file, is then created on each of the HLT servers, containing a list of all the data files available to be transferred. At the start of Phase 3 a manual procedure was used: an operator would copy the `list_send` file to the WN servers and start the transfer, via `rsync` [15], of all of the listed files.

The raw data are written on the HLT servers in a Belle II defined *Sequential ROOT* (SROOT) format. The average number of events in an SROOT file is $\sim \mathcal{O}(2.5 \times 10^4)$. SROOT files are written in a serialised and uncompressed fashion—this format was adopted to ensure that in the event of any failures during data taking only the current event is lost, previous events are correctly recorded and the SROOT file remains readable.

The SROOT format cannot, however, be used directly for physics analysis. Therefore the files have to be converted to the standard ROOT format. The tools required for the conversion are part of the Belle II Analysis Software Framework *basf2* [16]. As ROOT files incorporate compression by default, the format conversion reduces the file size. During the Spring 2019 Run the size of each SROOT file was 2 GB resulting in ROOT files of around 800 MB. As larger file sizes are more efficient for the distributed computing system it was subsequently decided to increase the SROOT file size to 8 GB, producing ROOT files of about 3 GB.

The conversion from SROOT to ROOT occurs on the WN servers before the data are transferred to permanent storage and from there replicated around the world for physics analysts. SROOT files are archived following the conversion, and once a second copy of the ROOT file has been made, it is strictly enforced that there must always be at least two copies of every raw data file in ROOT format.

Figure 2 shows the total size of raw data files in SROOT format transferred from the HLT servers to the WN servers for a period of 36 days in May and June 2019. Typical daily variation can be seen during this time. During the

Fig. 2 The total size, in terabytes, of raw data SROOT files transferred per day and the integrated luminosity recorded for the period from 2019-05-18 to 2019-06-22, typical daily variation can be seen. The total data size includes all run types, whereas the luminosity only includes physics runs. The time period shown also highlights the effect of HLT filtering mode, which was used from 2019-06-10; the data size per unit of luminosity after this date is close to that expected for future Belle II operations. The different ratio observed between 2019-05-23 and 2019-05-27 arises from accelerator studies undertaken on these days

Fig. 3 A representation of the online–offline data copy system showing the processing of raw files from one HLT at the start of Belle II Phase 3 operations. The input batch is divided to four chunks, which are then handled in four processes concurrently

time period covered, the HLT was switched to running in filtering mode which reduced the data size per unit of luminosity by a factor of ~ 9 . The data size per unit of luminosity after this change is expected to be close to the typical data size that will be realised over the whole data taking period of the Belle II experiment.

Data Pipeline Implementation

At the start of Phase 3, the data transfer worked as follows. After the file is transferred from the online side to the offline side, it is converted from SROOT to ROOT.

Following conversion, the SROOT and ROOT files are archived in permanent storage and a check on the number of events is performed. It is expected that the input SROOT file and output ROOT file should have an identical number of events. This data pipeline is shown in Fig. 3, together with the time taken for each step. It can be seen that the slowest step was the conversion, and this caused a bottleneck in the system. The sequential processing of each individual file, lacking an effective parallelisation mechanism, led to underutilisation of the system resources as roughly only $\sim 25\%$ of the available CPU was being used. To make better use of the system a new process was implemented early in Phase 3, whereby transfers and the conversion of files were no longer handled in a sequential fashion—details of the implementation of this process are given in “[Raw File Transfer and Conversion](#)”.

Hereafter, the re-worked data pipeline software, hardware, and protocols are described in detail. The implementation was guided by the Unix philosophy of writing small programs which do one thing and do it well, work well together, and failures are easy to diagnose and fix. Simplicity of the implementation has been given priority in design decisions. Consequently, the core data pipeline processing programs are almost entirely implemented in Bash [17] and a text stream interface is used extensively. Data processing works autonomously yet the operators can choose to manually execute the processing tasks. For example, when required the re-processing of arbitrary data can be invoked by the human operator.

Offline Processing Infrastructure

The WN servers are Lenovo x3550 M5, with single-socket 14-core 2.60 GHz Intel® Xeon® E5-2697 v3 CPU, 64 GB RAM, and two 300 GB internal SAS disks running Red Hat Enterprise Linux Server release 6.10. They are connected to the DAQ network using dedicated 10GbE—LR network links and to the storage network using 56 Gb/s InfiniBand links. The main storage for the WN servers is a shared IBM Spectrum Scale high-performance clustered file system, known as GPFS [18]. The permanent storage for raw data is a hybrid disk-tape system: An IBM TS3500 tape library with a GPFS cache enabling total throughput up to 50 GB/s [19]. Server, network, and storage are all operated and maintained by KEKCRS staff round-the-clock.

Database

The main function of the offline database is to manage the raw data with file-level granularity. For offline operations, it is essential to have file state transition tracking, bookkeeping, data pipeline monitoring, and reporting. The database

itself is a single instance of *MySQL* (Community Server version 5.7) running on a Lenovo System x3550 M5 two-socket 20-core server, 64 GB RAM, redundant 10GbE LAN and running Scientific Linux (release 6.10). In hardware failure scenarios it can be relocated to a standby node. The recovery time and recovery point objectives are 1 and 24 hours, respectively. Portability is not an issue for this database. However, the schema and procedures have been deliberately kept generic to allow migration to other database platforms.

The database uses a fixed schema. Every raw file is registered in the database with important properties such as experiment number, run number, timestamps, checksums, physical file paths, and processing state to mention a few. For bookkeeping and quality control purposes, run-related data are migrated from the online database periodically. Additionally, performance, fault, and accounting metrics are also stored in this database. A web based application is provided for monitoring the operations data in a user-friendly interface.

The raw data processing is tightly integrated with the database, there are three stages where interaction with the database plays a vital role:

- *Data detection* Detects the presence of new raw files on the HLT storage, registers information provided by the HLT servers, and initiates the transfer process.
- *Processing* Processes the SROOT raw files into ROOT raw files, checks the number of events, and performs quality checks.
- *Replication policy* Establishes that two copies of each (ROOT) raw data file exist, and creates meta-information for each file and run.

Log Management

Message logging is consolidated by using a standard *syslog* [20] daemon running on one of the WN servers on a customized port. It is a singleton service and can be relocated to run on any other node. The standard shell command interface to the *syslog*, *logger* [21], can be used to send messages. Additionally, a Python module *netsyslog* is provided which enables the sending of syslog messages directly from Python [22]. The *syslog* log is rotated daily with persistence for one month. Thus the issue of scattered logs was eliminated and operators can refer to a single location to check the data pipeline logs in detail and find information for troubleshooting.

Batch Processing System

Performance is critical in processing high-volume offline data. Therefore a solution for managing a large number of tasks and maximizing the use of computing resources was

Fig. 4 Scaling of parallel SROOT to ROOT conversion throughput vs. number of parallel processes measured on a single-socket 14-core server. Hyper-threading is disabled and the kernel's frequency scaling is enabled using the on-demand governor. Each data point is an average of converting a batch of 104 SROOT files 10 times (standard deviation $\sigma < 0.012$) processed in the data pipeline queue. Real physics data were used and the average conversion time of a single SROOT file without parallelization was 77.10 seconds (sample size, $N = 1040$; standard deviation, $\sigma = 0.17$), disk I/O and enqueueing included. As reference, perfect linear scaling is plotted as a dotted line. The test was carried on a single server. Five servers running the conversion concurrently resulted in up to a 15% increase in conversion times, presumably due to load and increased disk I/O on the shared storage

required. Our implementation is based on the versatile and lightweight Task Spooler, with which one can queue up tasks from the shell for batch execution [23]. By default, the queues are configured per-user and per-host, but can be modified to work as a per-host system with multi-user capability. The number of queues is technically unlimited and the number of parallel jobs in each queue can be adjusted dynamically. A custom wrapper was created for the Task Spooler, which implements three queues for data processing: one for transfer, one for conversion, and one for copying files to permanent storage. The exit status and output file name from each job are logged in *syslog* to facilitate error checking and troubleshooting.

This solution has proved its robustness and stability. Figure 4 shows test data for parallel SROOT to ROOT conversion. The performance scales up nearly linearly for up to 14 parallel processes, which is the number of CPU cores in each server, and without significant deviation from perfect linear scaling. In production, the system has been verified with raw file batches of up to 3800 files per WN server.

Fault Management

Given the high data volumes and requirements to avoid any data loss, reliable fault management systems and workflows are essential. Our centralised fault management system was built by adapting the alarm model of the ITU-T X.733 [24] recommendation to fit the requirements and use cases of these offline operations. Agentless monitoring is implemented for communications, processing, and quality of service type alarms. Raising and clearing alarms is done by database procedure calls. A command-line application is provided for checking and managing the alarms, as well as a web based application.

Raw File Transfer and Conversion

The online HLT storage servers accumulate data from several runs before initiating a request to transfer raw files to the offline side. Once the run data are ready to be transferred, a `list_send` file is created on the storage system which holds the raw files. A scheduled task on the offline side checks for the existence of a `list_send` on each partition frequently. If new `list_send` files are found, the information about them is updated in the database and the files are then spooled to the offline storage with *rsync* and prepared for processing by the data pipeline. Each `list_send` file specifies a batch of data that is to be processed by a WN server.

Another scheduled job on each WN server starts the data transfer and processing when new data are ready for processing. The processing starts by dispatching a task to the transfer queue for each raw file. Each WN server processes raw files from a single, designated HLT. The WN can poll and transfer the files from each HLT running *rsync* in daemon mode. Each WN server can have up to four concurrent instances of the transfer process.

The transfer rate for each spawned *rsync* daemon is intentionally capped at 230 MB/s in order to safeguard the storage server from overload during transfers which may take place concurrently when new data are arriving from the detector. One of the parameters in the `list_send` file is an Adler-32 [25] checksum, calculated on the HLT servers, which is used to validate the transferred files. Once all checks are cleared, the transfer task dispatches a new task to the conversion queue and then shuts down itself.

While an Adler-32 checksum is used to verify transfers, an MD5 [26] checksum is used for data identification, replication, and registration. Both checksums are stored in the database.

The conversion task takes the transferred SROOT file and converts it to standard ROOT format by using tools from the Belle II Analysis Software Framework, *basf2*. The number of events in the converted ROOT file must match the number

Fig. 5 Dynamic adjustment of WN processing queue parameters. After all raw file transfers from online storage are completed, the number of parallel slots in the conversion queue is increased and the system becomes dedicated to the SROOT-ROOT conversion task

of events in the original SROOT file, with further processing of the associated run blocked and an operator notified in the event of any mismatch. Additionally, Adler-32 and MD5 checksums are calculated and stored for the converted file.

The transfer and conversion are done in parallel by dynamically adjusting the numbers of parallel slots in the queues, as depicted in Fig. 5. This allows for $\sim 85\%$ CPU utilisation, significantly higher than the $\sim 25\%$ seen in the earlier implementation. The next step in the processing commences after all tasks in the queues have finished. Then, logs are summarised and checked for unexpected processing errors, after which files are copied to the permanent disk-tape hybrid storage. This file copying is done in parallel. At the end of the processing of one batch, performance metrics are calculated and stored. A report of the processed data is automatically uploaded to an electronic logbook, ELOG [27], with a summary of metadata, quality metrics, logs, errors, and other relevant information.

Data quality checks take place before raw files are released for further offline processing. For example, checking that the number of events is correct and that the data are readable. Once the quality of the data are verified the data, on a run by run basis, are flagged as good or bad for analysis use.

There is a mechanism to prevent the deleting of files on HLT storage while the SROOT files are being transferred and processed. Once the offline processing has confirmed all files for the runs in a *batch* have been copied from the HLT and processed, with at least two copies of each raw ROOT file secured. Then, the offline side notifies the online side that the data copy is completed. This is done by uploading a file containing the file names and Adler-32 checksums back to the HLT storage. Only after the HLT receives the notification and verifies the checksums, the files will be cleaned up from the HLT storage to free space for new runs.

Fig. 6 Raw data registration and replication. The status of each file is retrieved in order to track the registration and replication performed by BelleRawDIRAC

Fast Lane

Sometimes, the Belle II detector experts need to have access to a limited set of the data quickly after they have been collected, for example to assess the impact of a recent change. To facilitate this in an automated fashion a Fast Lane was created; Belle II experts could request, via a script, that a particular data run be transferred to KEKCR and whether the file should be in SROOT or ROOT format. A frequently running cron job checks for new requests, and will initiate the transfer and conversion of requested files. The transfer may be delayed if transfer jobs are already in progress. Transferred files are automatically deleted from KEKCR after seven days without notice, and the privilege to use the fast lane may be taken away if it is abused to request an excessive number of files.

Raw Data Registration and Replication

To manage the distributed computing resources of the Belle II experiment, DIRAC [6] is used. Belle II specific requirements are handled by an extension to DIRAC called BelleRawDIRAC [7]. Another extension of DIRAC named BelleRawDIRAC [28], independent of the production of data and user analysis activities, is dedicated to the registration and replication of raw data files.

After completion of data quality checks and following data release, an automated helper process is initiated which communicates with the BelleRawDIRAC API. Figure 6 shows the communication between the two systems.

Fig. 7 A screenshot of the data transfer monitor display during live network transfer. The plot shows, for one afternoon, the 1 min average data transfer rate (MB/s) for five active WN servers as a function of time of day. The colours indicate the five different WN servers

Files which are marked as good for analysis in the offline database are submitted to *BelleRawDIRAC* through a Remote Procedure Call (RPC) with a predefined input payload structure which contains information such as a unique key (a string generated from the name of the file which is guaranteed to be unique), checksum, file path, file size, and checksum type. After submitting the payload, the API returns a predefined output structure which contains information including the registration status (Success or Failure) for every raw data file or for every submitted unique-key.

The offline system also keeps track of the replication status (Active, Done, Error, or Cancelled) of every submitted file in *BelleRawDIRAC*, retrieving the status through RPC calls. Figure 6 shows each step of the process; for example, if the file is replicated without any error then it is marked as “Done Replication” in the offline database. This step is a crucial part of the process as it helps us to make sure that all the files are replicated and verified in the permanent raw data storage centres. In addition, the *BelleRawDIRAC* extension makes sure that at least two copies of every raw data file are stored on the KEK and BNL permanent storage systems with at least one copy at each site. Transfer of data from KEK to BNL uses the Belle II distributed data management system [29], which is part of *BelleDIRAC*. As part of the plan for raw data replication development, the Distributed Data Management (DDM) engine in *BelleDIRAC* will be migrated to use *Rucio* [30, 31] in the backend.

Data Transfer Monitoring

As the data taking is a round-the-clock operation, excessive accumulation of files on the HLT storage could cause detector operations to cease if there is no available space for new data. During the Spring 2019 run, it would have taken at

least three weeks to exhaust the HLT storage, extrapolating from the most data recorded in a single day. The storage will be maintained to ensure that it will always have the capacity to store at least seven days worth of data.

Data transfer monitoring has been developed to provide visibility of network service activities and operations. A tailored web based service is provided for the operators with a graphical monitor to inspect the data transfer rates. On each WN server, the number of received bytes from the network interface connected to the HLTs is collected once a minute continuously and saved in a central *PostgreSQL* [32] database. The framework is a web based design, which consists of *Django* [33], *ReactJS* [34], *pandas* [35], and *Plotly* [36]. The network traffic plots of the WN servers are interactive, and the Web browser clients connected to the service request the updated plots automatically every five minutes. The use of a *PostgreSQL* database is due to a legacy implementation for the collection of network metrics which has some search based limitations. To consolidate the design we plan to collect network metrics with our present MySQL instance and plan to introduce a time series based cache database to smooth search functionality. Figure 7 shows a snapshot of the network transfer rates of five active WN servers during a period of data transfers.

The historical data stored in the database enables troubleshooting of network issues, detection of abnormalities, and provides a reference in planning network upgrades. For example, one can easily check the online–offline network status in Phase 3 operation in June 2019 as shown in Fig. 8.

Monitoring and Bookkeeping

To provide a visual representation of the offline processing operations, a simple web based user interface has been designed, with an API model to access the information of the

Fig. 8 Overview of network transfer rates for one of the WN servers in June, 2019 (top). The transfer process for each WN was made up to twice per day. The typical transfer rate for each server is between 300 and 400 MB/s, and a transfer rate of 900 MB/s has been reached with a stable operation—this is just below the maximum value of 920 MB/s (four processes, each capped at 230 MB/s). The bottom plot shows a detailed view of the file transfer on 9 June 2019

data, which are gathered by the data pipeline operation for present and future purposes. This interface provides monitoring of the live status of the following: the processing operations, monitoring of internal metrics, and data mining. Figure 9 shows data pipeline throughput metrics for a period of a month.

In addition, the user interface also provides bookkeeping of every run and a search interface providing information on past and present runs. Figure 10 shows the dashboard for an example run. The dashboard serves as a visual aid for operators to check data quality.

A REST API model has been adopted to provide relevant information to other groups in the Belle II collaboration. For example, the teams that perform the initial physics analysis and calibration tasks use our REST API to detect new runs that have been made available to them, and they can search based on run types and other properties.

The APIs interact with the database and provide information in JSON format which is utilised by *JavaScript* frameworks and then visualised in HTML. The frontend uses open source libraries including *DataTables* [37] and *Plotly*, while the REST server is a *Flask* application [38]. All the API

Fig. 9 The CPU walltime of data conversion activity over a time period of one month. The walltime shown is the daily average of total time to convert a given SROOT file to a ROOT file. The worker nodes are numbered WN01, WN02, etc. The top plot shows a view of monitoring for five worker nodes, while the bottom plot focuses on WN02 to allow for detailed inspection. The average conversion times for physics runs has remained consistent. Days with lower daily averages arise from accelerator study periods with non physics runs, which have a shorter conversion time; during these studies some HLTs may be unused, and the corresponding WNs may process no data on a given day. Monitoring tools can check for significantly higher walltimes, which can be an indicator of unexpected issues with the data or processing environment

endpoints and web interfaces were developed for the Belle II experiment.

DevOps

The development toolchain of the system is the same as one used by the Belle II core software development [16]. The enterprise level services are operated by the Deutsches Elektronen-Synchrotron (DESY) in Hamburg, Germany [39]. All the code and in-house tools used in these offline operations are maintained in a *git* [40] repository, which is hosted in a *Bitbucket* [41] server. Tasks, feature requirements, and fault reports are handled with *Jira* [42] with Agile Development [43] plugins, and documentation is maintained in *Confluence*

Fig. 10 A screenshot of the bookkeeping dashboard showing information for a single run; a run known to have issues has been selected to highlight how the dashboard can be used by operators to easily view information and find details of any issues arising

Details	Exp Number	Run Number	Online Total Events	Online Run Type	Offline Total Events	Offline Run Type	Ratio Online-Offline	Grid Status	Offline Status
10	5694	3457593	physics	1565395	physics	0.4527	Done	Recoverable

Core Computing Info

Info	Online	Offline	Difference
WN01 Events	223649	223649	0
WN02 Events	223569	223569	0
WN03 Events	223680	223680	0
WN04 Events	223759	223759	0
WN05 Events	223482	223482	0
WN06 Events	946493	0	946493
WN07 Events	223742	223742	0
WN08 Events	945705	0	945705
WN09 Events	223514	223514	0
Total Files	91	91	0

Other Info	Value
Online Created	Sun, 08 Dec 2019 05:50:57 GMT
Luminosity	3.4851601123809814
Online Trigger Count	1570864
Online Trigger Run Length	829
Online Trigger Type	gdl

[44]. There is a hook script on the *Bitbucket* server which ensures all commit messages have a valid *Jira* reference, ensuring all code changes are fully traceable and identifiable. These tools provide effective, smooth, and collaboration-wide steering of development and code change review.

We have established a procedure to set up a virtualised environment for the development of our processing pipeline and monitoring tools. Even copies of the raw SROOT files can be migrated from the production environment to this virtualised environment. Thus all the processing can be functionally verified by using the actual raw data before deployment for production use.

Future Plans and Outlook

Currently, the offline processing system has capacity for processing up to 36 files in a minute, end-to-end (transfer, conversion, copy to permanent storage), for ROOT files having a size of about 800 MB. Based on conservative estimates this translates to a 45% performance margin. Scaling the throughput up can be achieved simply by adding more worker nodes. The WN servers and storage are operated on a leased infrastructure, the lease renewed every four years. Consequently the offline processing system hardware is renewed regularly.

The raw data conversion to the standard ROOT format is the most computationally expensive step. Plans have been developed for the online system to produce ROOT files, which would significantly streamline the data processing and reduce the operational expenses.

Currently, all raw files are replicated to BNL, which is the major site for data analysis. This replication importantly provides an off-site backup of the data. Beginning in the fourth year of operations, the second copy of the raw data will be distributed over several raw data centre sites worldwide, including BNL.

Our web based monitoring applications use colour plots extensively to visualise data. However, the Belle II collaboration is aware that colour blindness prevalence is not negligible and making visualisations accessible to all collaborators must be addressed. Therefore, the modification of our applications to use colour blind friendly palettes is in progress.

Conclusions

In the Belle II experiment, the luminosity has been steadily increasing since the start of physics data taking. Consequently, the offline data processing has been re-worked to meet the demands of handling the increasing data volumes produced by the detector. An important milestone was met in June 2019: offline data processing reached a fully automated mode of operation. In addition, efficient monitoring tools and workflows have been implemented together with modernised interfaces to external systems to share data. The Belle II offline operations system is now ready for data taking with large datasets, to facilitate the production of future physics results.

Acknowledgements We thank Koichi Murakami and Soh Suzuki of KEKRC for their assistance in clarifying the technical details on server, storage, and network infrastructure. We thank Ikuo Ueda, Paul Jackson, and Yuji Kato for their helpful comments and discussion, and critical reading of the manuscript.

We are grateful for the support and the provision of computing resources by KEK in Japan and BNL in the USA. We acknowledge the network services provided by SINET5 in Japan and ESnet in the USA.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bevan AJ et al (2014) The physics of the B factories. *Eur Phys J C* 74(11):3026. <https://doi.org/10.1140/epjc/s10052-014-3026-9>
2. Abe T et al. (2010) Belle II technical design report. Technical report, Belle II collaboration. [arXiv:1011.0352](https://arxiv.org/abs/1011.0352)
3. Kou E et al (2019) The Belle II Physics Book. *Prog Theor Exp Phys* 2019:123C01. <https://doi.org/10.1093/ptep/ptz106>
4. Taniguchi N (2020) SuperKEKB/Belle II. In: Presented at KEK roadmap open symposium. <https://kds.kek.jp/indico/event/34739/>. Accessed 6 July 2020
5. Brun R, Rademakers F (1997) Root—an object oriented data analysis framework. *Nucl Instrum Methods* 389:81. [https://doi.org/10.1016/S0168-9002\(97\)00048-X](https://doi.org/10.1016/S0168-9002(97)00048-X)
6. DIRAC | The Interware. <http://diracgrid.org/>. Accessed 05 May 2020
7. Miyake H, Grzymkowski R, Ludacka R, Schram M (2015) Belle II production system. *J Phys Conf Ser* 664(5):052028. <https://doi.org/10.1088/1742-6596/664/5/052028>
8. Brookhaven National Laboratory (BNL), P.O. Box 5000, Upton, NY 11973-5000, USA. <http://www.bnl.gov/>. Accessed 11 Oct 2019
9. Lewis P et al (2019) Nuclear instruments and methods in physics research section A: accelerators. *Spectrom Detect Assoc Equip* 914:69. <https://doi.org/10.1016/j.nima.2018.05.071>
10. SuperKEKB Project. <http://www-superkekb.kek.jp>. Accessed 22 Jan 2020
11. Belle II Luminosity. <https://confluence.desy.de/display/BI/Belle+II+Luminosity>. Accessed 7 July 2020
12. Forti F BELLE II and flavor physics in e+e-. In: Presented at European physical society conference on high energy physics (EPS-HEP2019). <https://indico.cern.ch/event/577856/contributions/3396819/>. Accessed 7 July 2020
13. Yamada S et al (2015) Data acquisition system for the Belle II experiment. *IEEE Trans Nuclear Sci* 62(3):1175. <https://doi.org/10.1109/TNS.2015.2424717>
14. Suzuki SY et al (2015) The three-level event building system for the Belle II experiment. *IEEE Trans Nuclear Sci* 62(3):1162. <https://doi.org/10.1109/TNS.2015.2422376>
15. Rsync. <https://rsync.samba.org/>. Accessed 19 Apr 2019
16. Kuhr T et al (2018) The Belle II core software. *Comput Softw Big Sci* 3(1):1. <https://doi.org/10.1007/s41781-018-0017-9>
17. GNU Bash. <http://www.gnu.org/software/bash/>. Accessed 13 May 2019
18. IBM Spectrum Scale. https://www.ibm.com/support/knowledgecenter/en/SSFKCN/gpfs_welcome.html. Accessed 28 Nov 2020
19. KEKCC HSM System. <https://kekcc.kek.jp/service/kekcc/html/Eng/HSM20System.html>. Accessed 21 Jan 2020
20. Gerhards R (2009) The Syslog protocol, RFC 5424. <https://doi.org/10.17487/RFC5424>
21. Logger. <https://www.linux.org/docs/man1/logger.html>. Accessed 15 May 2019
22. Netsyslog. <http://hacksaw.sourceforge.net/netsyslog/>. Accessed 15 May 2019
23. Task Spooler. <http://viric.name/soft/ts/>. Accessed 13 May 2019
24. ITU-T Recommendation X.733 (02/92): information technology—open systems interconnection—systems management: alarm reporting function. <https://www.itu.int/rec/T-REC-X.733-199202-I>. Accessed 6 Aug 2019
25. Deutsch P, Gailly, JL (1996) ZLIB Compressed Data Format Specification version 3.3, RFC 1950. <https://doi.org/10.17487/RFC1950>
26. Rivest R (1992) The MD5 message—digest algorithm. RFC 1321. <https://doi.org/10.17487/RFC1321>

27. ELOG. <https://elog.psi.ch/elog/>. Accessed 1 Aug 2019
28. Villanueva M, Ueda I (2020) The Belle II raw data management system. EPJ Web Conf. 245:04005. <https://doi.org/10.1051/epjconf/202024504005>
29. Padolski S et al (2020) The Belle II raw data management system. EPJ Web Conf. 245:04007. <https://doi.org/10.1051/epjconf/202024504007>
30. Barisits M et al (2019) Rucio: scientific data management. Comput Softw Big Sci 3:11. <https://doi.org/10.1007/s41781-019-0026-3>
31. Lassnig M et al (2020) Rucio beyond ATLAS: experiences from Belle II, CMS, DUNE, EISCAT3D, LIGO/VIRGO, SKA, Xenon. EPJ Web Conf. 245:11006. <https://doi.org/10.1051/epjconf/202024511006>
32. PostgreSQL. <https://www.postgresql.org/>. Accessed 19 Aug 2019
33. Django. <https://www.djangoproject.com>. Accessed 14 Sep 2019
34. React. <https://reactjs.org/>. Accessed 14 Sep 2019
35. Pandas: python data analysis library. <https://pandas.pydata.org/>. Accessed 19 July 2019
36. Plotly. <https://plot.ly/python/>. Accessed 19 Sep 2019
37. DataTables. <https://datatables.net/>. Accessed 22 Jan 2020
38. Flaskapp. <https://pypi.org/project/flaskapp/>. Accessed 22 Jan 2020
39. Deutsches Elektronen-Synchrotron (DESY), Notkestraße 85, D-22607 Hamburg, Germany. <https://www.desy.de/>. Accessed 11 Oct 2019
40. Git. <https://git-scm.com/>. Accessed 22 Jan 2020
41. Bitbucket. <https://bitbucket.org>. Accessed 22 Jan 2020
42. Jira—issue and project tracking software. <https://www.atlassian.com/software/jira>. Accessed 22 Jan 2020
43. Atlassian Agile. <https://www.atlassian.com/agile>. Accessed 22 Jan 2020
44. Confluence—team collaboration software. <https://www.atlassian.com/software/confluence>. Accessed 22 Jan 2020

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.