ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOMS055

A MODERNIZED ARCHITECTURE FOR THE POST MORTEM SYSTEM AT CERN

J.F. Barth, F. Bogyai, J.C. Garnier, M.L. Majewski, A. Mnich, M.P. Pocwierz, T.M. Ribeiro,
R. Selvek, R. Simpson, A. Stanisz, D. Wollmann, M. Zerlauth
CERN, Geneva, Switzerland

Abstract

The control system of the accelerators at CERN stores and analyzes more than 200 million dumps of high resolution data recordings every year in the Post Mortem (PM) system. A continuous increase in the complexity of the Large Hadron Collider's (LHC) systems and the desire to collect more accurate data requires continuous improvement of the PM system. Recently, the PM system has been modernized ahead of the third operational Run of the LHC. The upgraded system implements well known data engineering principles such as horizontal scaling, stateless services and readiness for extensions. This paper recalls the purpose of the PM service and its current use cases. It presents its modernized architecture, reviews the current performance and limitations of the system, and draws perspectives for the next steps in its evolution.

INTRODUCTION

The particle beams and the magnet circuits of the LHC store unprecedented amounts of energy. An uncontrolled release of this energy would cause significant damage to the accelerator complex, requiring extensive repairs which considerably reduce the available time of the accelerator to produce physics [1].

Therefore, it is imperative to verify the correct behavior of the accelerator's many control and protection systems after each beam dump. Furthermore, it is important to understand the origin of the termination of a physics fill (beam dump) and the conditions under which the event occurred to decide whether the next beam injection is safe or whether a device is behaving faulty. Since 2008, the PM system has stored and used data recorded at the moment of a beam dump by thousands of devices installed in the LHC to reliably provide diagnostic assistance, verify device behavior, explain unexpected events, and ultimately guarantee the safe and efficient operation of the LHC [2]. In addition to providing machine protection, PM also aides with monitoring and tuning of the Super-Proton-Synchroton (SPS) for performance optimisation.

During CERN's recent Long Shutdown 2 (LS2) from 2018 to 2022, the PM system underwent substantial upgrades in areas of data collection, storage, and access, to address the drawbacks identified in the previous architecture [3]. The first section of this paper explains the current use cases of the system. The second section outlines the modernized architecture, giving detailed explanation of data collection, data storage, data access, event building, data analysis, and the data model. Finally, the limitations of the new system are highlighted and the vision for the future is discussed.

USE CASES

The PM system needs to be able to reliably collect, store, analyze, expose and persist large amounts of data arriving as a discontinuous stream with load peaks both for storing and retrieving data during events between quiet hours. Table 1 summarises the different PM use cases.

Deterministic Time Constraints

For deterministic time constraint use cases, the data sent to the PM system must be analyzed within an exact time frame to provide operators with critical in-time information. During SPS Quality Checks (SPSQC), operators need to monitor and tune certain accelerator parameters for each successive cycle of the accelerator to improve the overall performance. A cycle has a minimum length of 10 seconds in which the the beam is injected, accelerated, and extracted. This means that PM must reliably collect, store, and analyze a high volume of data within 10 seconds. Other CERN applications of deterministic time constraint use cases are the Injection Quality Checks (IQC) for checking the quality of beam injections into the LHC [4], and Extraction Post Operational Checks (XPOC) for checking that a beam extraction occurred under nominal conditions [5].

Non-Deterministic Time Constraints

For non-deterministic time constraints, the data collection, storage, and analysis do not have to be complete in a strict time frame. There are two main use cases that are not bound by deterministic time constraints. Global Event Analysis for analysing events affecting all device domains in the LHC and Global Powering Analysis for analysing events affecting magnet powering devices such as Quench Protection Systems (QPS) [6] and power converters in the LHC.

Table 1: PM Use Cases

Use Case	Type
Global Event Analysis	Non-deterministic
Powering Event Analysis	Non-deterministic
SPSQC	Deterministic
IQC	Deterministic
XPOC	Deterministic

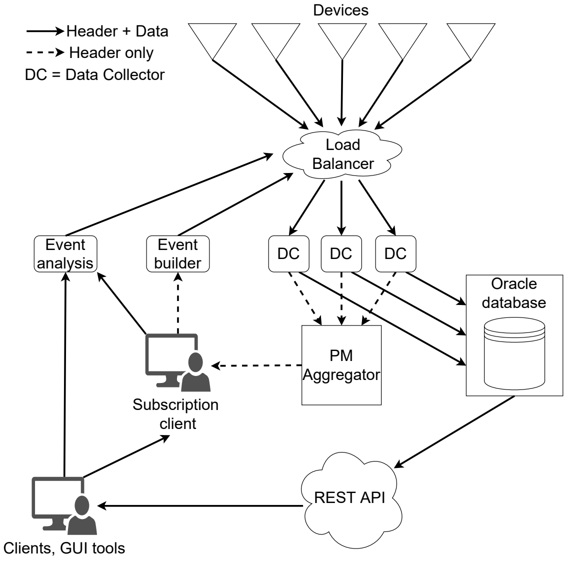

Figure 1: PM Architecture. Arrows represent the data flow inside the PM system.

ARCHITECTURE

The starting point of data in the PM system is the devices monitoring various domains of the accelerator. For instance, Beam Loss Monitors (BLM) [7] monitor the amount and location of particles lost around the accelerator. Devices have an internal, high-frequency, rolling buffer with data which is frozen and sent to a data collector when an event occurs. An event is for example a beam dump, where the circulating beam is extracted from the accelerator. Since storing data for analysis is crucial, data sending is a blocking operation. Devices also have fallback strategies such as retrying the send operation and sending to a dedicated fallback data collector. Figure 1 shows the architecture of the PM system with the flow of data and header information.

Data Collection

The data collector is a thin, yet important layer that is responsible for receiving data from the devices and forwarding it to permanent storage. It also sends data to the aggregator service which combines data from all data collectors into a single stream (see Event Building section). It acts as an abstraction layer between devices and storage, preventing devices from accessing storage directly and making them unaware of changes to the underlying storage technology and structure. Devices and data collectors communicate via the CERN Controls Middleware (CMW) Remote Device Access 3 (RDA3) protocol [8].

Devices are data collector agnostic, which has the advantage that devices and their targets are loosely coupled which allows for greater maintainability and easier scalability of the overall system. This gives PM the flexibility to integrate additional and novel use cases for accelerators other than the LHC. Furthermore, by allowing any device to send data to any data collector it is possible to distribute the load of the data collection process more evenly, benefiting performance.

Following an event, devices are assigned a data collector with the help of a load balancer and a Eureka discovery service. Devices request a data collector and receive its address as determined by a round robin algorithm. To avoid a bottleneck at the service discovery layer, multiple Eureka instances serve available data collectors to devices. Devices use a single Eureka URL which is resolved to a specific Eureka instance via a Domain Name System (DNS) server. Upon a DNS refresh, devices will receive a different Eureka instance from which they request data collectors.

Data Storage

Data storage is an important aspect of the PM system. The storage needs to have high availability to serve numerous clients, high capacity for the growing use cases and data complexity, and high read and write throughput for fast data availability. It also needs to be fault tolerant and to allow concurrent access. To satisfy these requirements, PM uses a Relational Database Management System (RDBMS) by Oracle [9] since 2019. Due to the heterogeneity of data recorded by devices, it is stored in the DB as compressed JSON [10] strings. Storing data as JSON has the added benefit of permitting devices to change their schema without forcing DB schema changes as well as allowing for advanced queries using the Spark API. Following the post-LS2 test beam and powering tests campaign [11], the DB of the LHC PM system reached a size of 2TB. Because of the special requirements of the SPSQC use case which sees a high volume and high velocity of data being sent to and analyzed by PM (~190 TB in 2021), SPSQC data is stored in a dedicated, separate database with a data retention of 2-4 weeks.

Data Access

Users of PM data need to have a consistent and reliable access to the data saved in storage. An important aspect is to prevent direct access to storage to avoid unintentional or malicious data corruption. Additionally, any changes to underlying storage technology and/or structure should not force users to make adaptations, as this would make modifications to the overall system more costly. To meet these requirements, PM clients access data via a read-only Representational State Transfer (REST) [12] API that utilises caching for fast access and serves multiple language technologies. With the REST API, users are decoupled from the storage and the system provides data consistency and integrity, and minimises the impact of storage changes on users. In addition, the storage agnosticism of the REST API adds more flexibility and further minimises the effect of a storage technology change, setting PM up for the future. The REST API exposes endpoints for all PM data types and returns data in JSON [10] format.

Event Building

To allow for an efficient analysis, the device data needs to be first grouped into events. The grouping is done by the event builder service, using the header information of the device data. To decouple the event builder from the data collectors, it does not directly subscribe to all data collectors that received data, but instead uses a subscription client to receive header information of the device data from an intermediary aggregator service. The aggregator is responsible for combining header information of data streams coming from the various data collectors into a single data stream that can then be used to build an event or read by other interested clients. Data collectors write to the aggregator and as soon as the event builder receives data from the aggregator, it will start a collection time window where all data arriving within that window will be associated with a single event. Using the patterns of the data dumps, the event builder will create an event that functions as a label for all of the received data. For example, if data only came from QPS devices and power converters, the event will be characterised as a powering event. Following creation, the event is sent to a data collector and into storage, where it is accessible to clients via the REST API for analysis.

Data Analysis

The analysis of an event provides information to operators and equipment experts whether all accelerator systems have been functioning correctly and, in the case of the LHC, whether the operation can be continued safely. As soon as an event is built, it can be analyzed by the event analysis framework. For each analyzed event, the analysis framework creates an analysis session, which contains the overall analysis result. To analyze an event, the analysis framework fetches all data that is associated with the event from the REST API and passes it to the various analysis modules. The analysis consists of multiple domain specific analyzes performed by separate modules. Each module produces its own analysis result and is triggered automatically, e.g. to give immediate feedback after a beam dump. For further analysis, they can also be triggered manually by an operator or system expert. The result of an analysis module can be fed into other analysis modules. Finally, all analysis results are combined into the overall analysis result, which indicates success or failure. Like the device data and the events, all analyzes results are persisted into storage via the data collection process, after which they are available through the REST API.

Data Model

There are four different types of PM data: raw data, event data, analysis results and analysis sessions. Raw data is gathered by devices in the accelerator and may only contain primitive values and arrays, complex objects are not supported. Each instance of raw data can be linked to a single event or stand alone. Event data is always referencing multiple raw data instances which were likely sent by devices following the same event such as a beam dump. The same event can be analysed multiple times and therefore produce multiple analysis sessions. For instance, an event analysis may first be triggered automatically and then later manually by an equipment expert. An analysis result is part of an analysis session which points to all module analysis results produced as part of that session.

OUTLOOK

Rest API Scaling

Currently a single REST API instance is serving all clients, which may cause data availability and service performance problems in the future, due to new demands and use cases. Therefore, the goal is to provide horizontal scaling as well for the REST API to increase the availability and overall robustness of the system in the long-term.

Data Collection

The current round robin algorithm for the load balancing does not take into account the actual load of each data collector when assigning them to devices. In the future, the goal is to implement fair load balancing such that devices are assigned to data collectors based on the load they experience.

Data Storage

In the long term the relational Oracle database shall be replaced by using the NextGen Common Accelerator Logging Service (NXCALS). NXCALS is a CERN logging system based on Hadoop Big Data technologies using cluster computing power [13]. This would allow PM clients to leverage the Spark API for more complex queries on top of the storage and to directly retrieve the data they are interested in instead of first fetching data through the REST API and then distilling it down into useful information. Furthermore, as users often analyze PM and logging data together, they will be able to use the same API for their analysis. Currently, NXCALS does not fully meet the performance criteria of the PM use cases. This is especially the case with the low latency requirements for data availability and analysis, and limitations of the size of data dumps.

CONCLUSION

Post Mortem is a reliable system for collecting, storing, and analysing millions of high resolution data recordings originating from thousands of control and protection systems in CERN's accelerators. The upgrades implemented during LS2 have transformed PM into a system with scalability and maintainability at its core. Through horizontally scalable and device-domain agnostic data collectors, the data collection process has become more robust and fault-tolerant. A RDBMS for data storage and REST API for data access offer stronger guarantees for data consistency and availability and increases robustness to future changes of storage technology. All of this means that PM is able to meet the growing set of use cases with and without deterministic time constraints.

REFERENCES

[1] R. Schmidt *et al.*, "LHC Machine Protection", in *Proc. PAC'07*, Albuquerque, NM, USA, Jun. 2007, paper TUZAC03, pp. 878–882.

[2] M. Zerlauth *et al.*, "The LHC Post Mortem Analysis Framework", in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper TUP021, pp. 131–133.

[3] J. C. Garnier *et al.*, "Smooth Migration of CERN Post Mortem Service to a Horizontally Scalable Service", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 806–809. doi:10.18429/JACoW-ICALEPCS2015-WEPGF047

[4] N. Magnin, E. Carlier, B. Goddard, V. Mertens, and J. A. Uythoven, "Internal Post Operation Check System for Kicker Magnet Current Waveforms Surveillance", in *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper MOPPC029, pp. 131–134.

[5] N. Magnin *et al.*, "External Post-Operational Checks for the LHC Beam Dumping System", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper WEPMU023, pp. 1111–1114.

[6] L. Coull, D. Hagedorn, V. Remondino and F. Rodriguez-Mateos, "LHC magnet quench protection system", *IEEE Trans. Magn.*, vol. 30, issue 4, pp. 1742-1745, Jul. 1994. doi:10.1109/20.305593

[7] B. Dehning *et al.*, "The LHC Beam Loss Measurement System", in *Proc. PAC'07*, Albuquerque, NM, USA, Jun. 2007, paper FRPMN071, pp. 4192–4194.

[8] A. Dworak, P. Charrue, F. Ehm, W. Sliwinski, and M. Sobczak, "Middleware Trends and Market Leaders 2011", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper FRBHMULT05, pp. 1334–1337.

[9] Oracle, "Introduction to Oracle Database", https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/introduction-to-oracle-database.html

[10] JSON, http://www.json.org/

[11] A. Apollonio *et al.*, "Summary of the Post-Long Shutdown 2 LHC and Hardware Commissioning Campaign", presented at IPAC'22, Bangkok, Thailand, June 2022, paper MOPOPT040, this conference.

[12] R. Fielding, "Architectural Styles and the Design of Network-based Software Architectures", University of California, Irvine, 2000.

[13] J. P. Wozniak and C. Roderick, "NXCALS - Architecture and Challenges of the Next CERN Accelerator Logging Service", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1465–1469. doi:10.18429/JACoW-ICALEPCS2019-WEPHA163