Next Generation of Post Mortem Event Storage and Analysis

Serhiy Boychenko, on behalf of TE-MPE-MS

Databases Future Workshop, CERN
29/05/2017

Outlines

- What is Post Mortem?
- Why does it need improvements?
- How the storage can be improved?

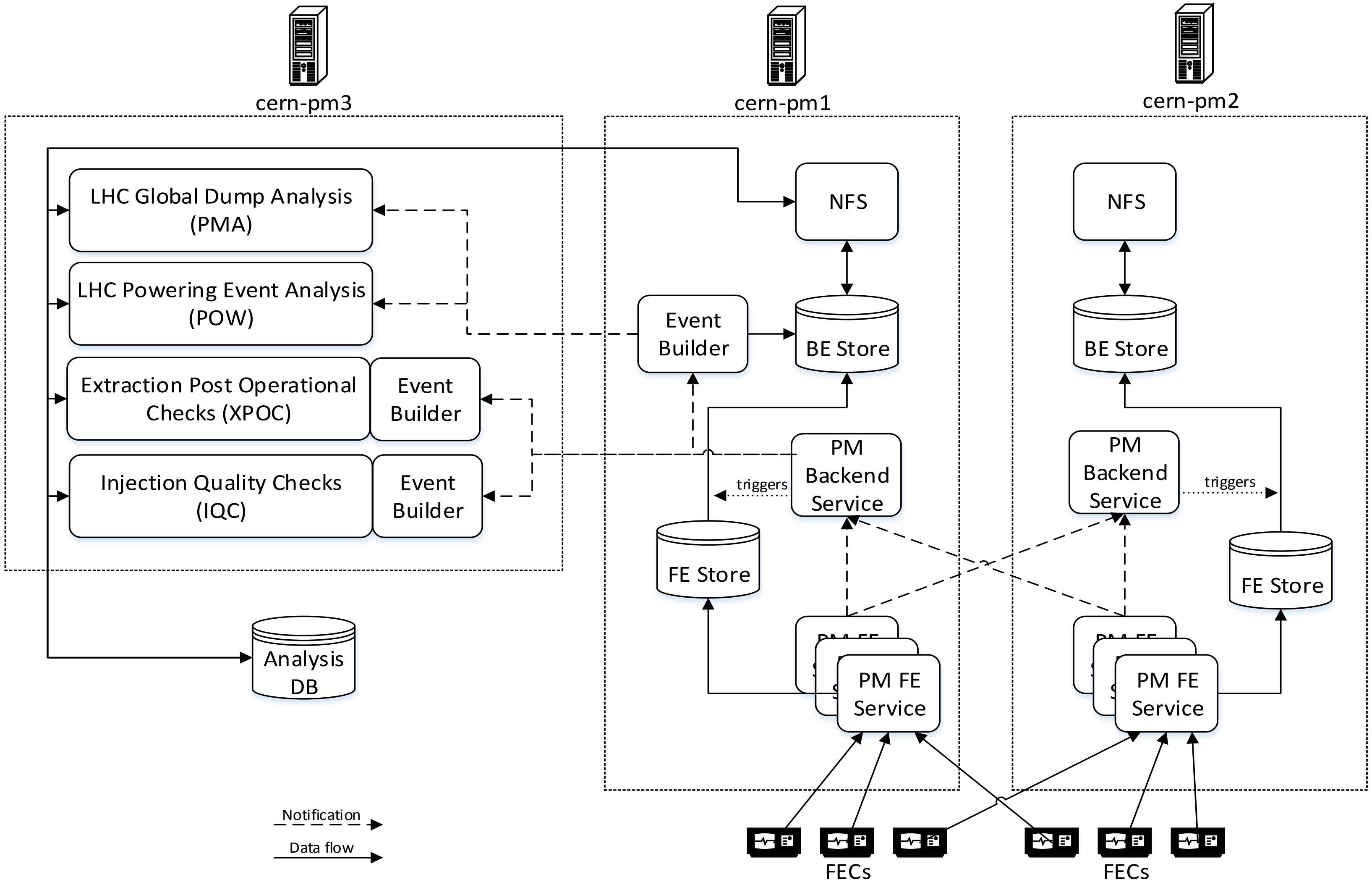
Outlines

- What is Post Mortem?
- Why does it need improvements?
- How the storage can be improved?

Post Mortem for the Accelerators

- The Post Mortem system allows the storage and analysis of transient data recordings from accelerator equipment systems
- Post Mortem data is complementary to Logging data
- Data buffers of shorter length (few seconds to minutes) are acquired with high frequency (KHz to GHz) around relevant events such as beam dumps, injection/extraction or powering events
- Post Mortem data is vital for the analysis and understanding of the performance and protection systems of the machines
- Correct transmission and storage of Post Mortem data has to be guaranteed with high reliability

Post Mortem Architecture



Courtesy:
Matthias Poeschl, Jean-Christophe Garnier

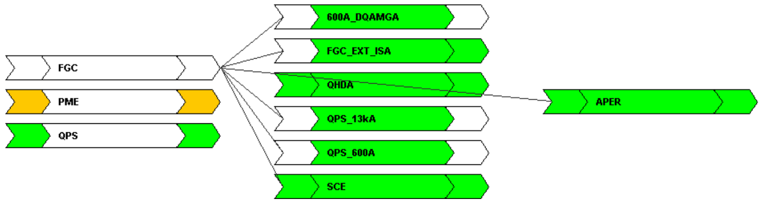
Post Mortem Use Cases

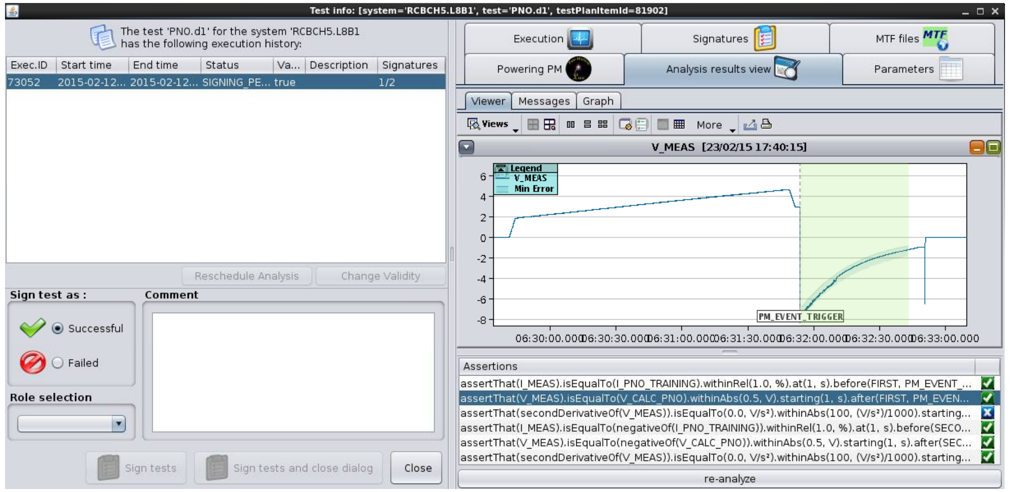
Event Name	Amount of Files	Dump Size	Frequency
Global	thousands of PMD files	hundreds of MB	several times per day
Powering	hundreds of PMD files	dozens of MB	several times per day
XPOC	hundreds of PMD files	couple of MB	several times per hour
IQC	~30 PMD files	couple of MB	several times per hour
SPSQC	~20 PMD files	~250KB	few seconds intervals (every SPS cycle)

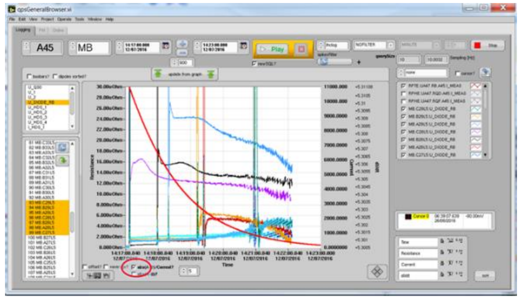
- Variable file size: 1KB-12.7MB (compressed data)
- Variable load: depends on the operation mode of the accelerators

Post Mortem Users

- Continuously used during operation cycles for validation of accelerator safety
- Essential source of the data for analysis orchestrated through the AccTesting framework (mostly during hardware and beam commissioning phases)
- Many users access PM data for ad-hoc queries through LabView or other data analysis tools







Post Mortem in Numbers

- Operating since 2008 (originally proposed in 2002)
- Deployed on 3 nodes (2 storage nodes with RAID1+0, 1 analysis node with 24 GB of RAM for in-memory processing)
- Storage already contains 20+ millions files (raw data, event data and analysis results)
- Total storage size to data is in the order of ~10TB
- Frequent traffic bursts ~1.0MB/second (incoming) and ~12MB/second (outgoing)

Outlines

- What is Post Mortem?
- Why does it need improvements?
- How the storage can be improved?

Shortcomings

- Static load distribution
- Data consistency and integrity
- Limited Write performance in case of simultaneous events
- Direct data access to raw data storage via NFS
- Decentralization (CALS + PM integration)

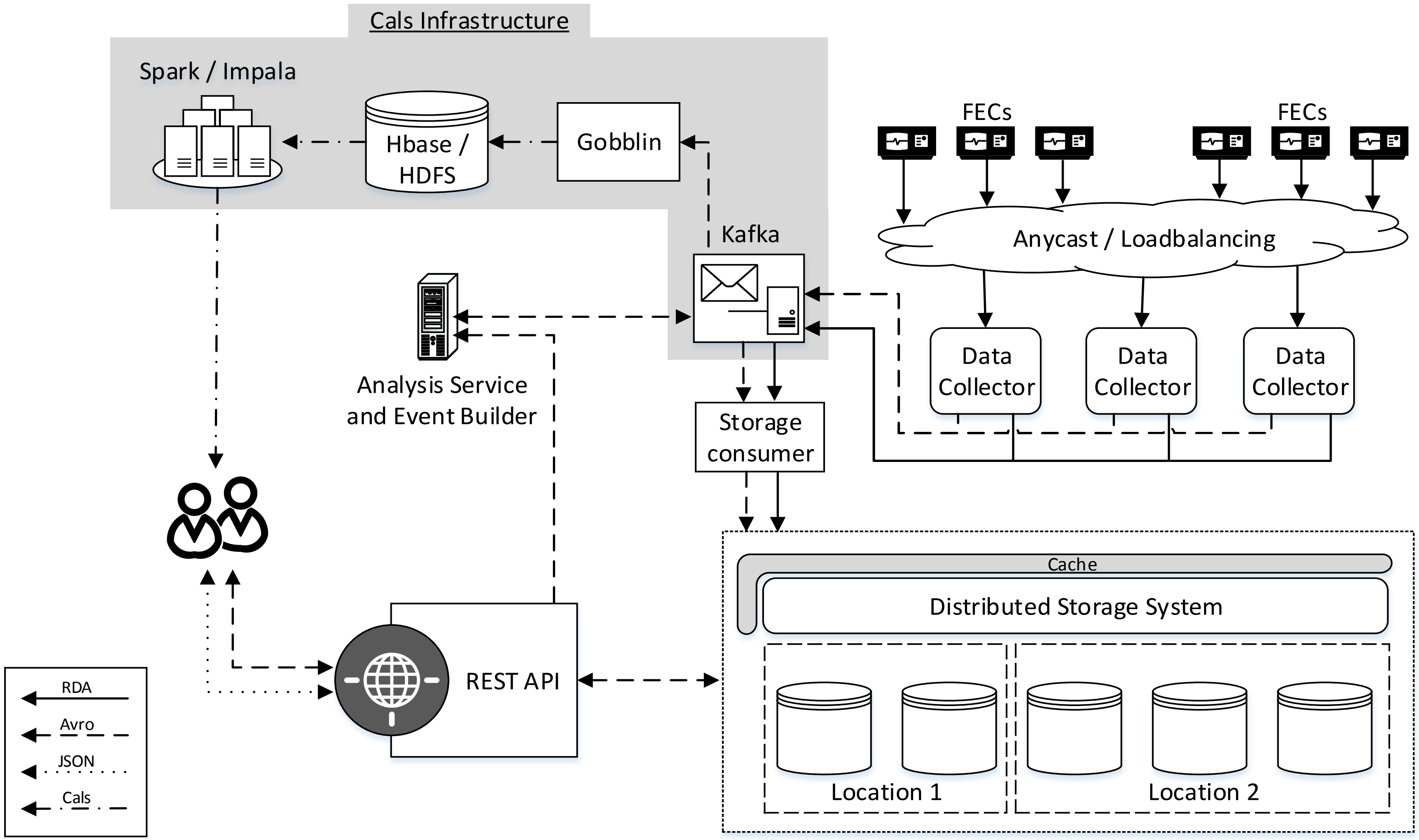
Requirements

- Backward compatibility for all users
- Horizontal scalability
- High level of reliability for data ingestion
- Flexibility for integration of the new the new use cases/extensions to additional accelerators

Outlines

- What is Post Mortem?
- Why does it need improvements?
- How the storage can be improved?

The New Architecture



Courtesy:
Matthias Poeschl, Jean-Christophe Garnier

Data Retrieval Layer - PM REST API

- Abstract storage implementation details from users (potential for common API with CALS)
  - Intend usage of cache for increased performance for low latency use-cases (IQC, XPOC) and to unload data storage layer
- Enable access to the data using standard serialization formats
- Provide advanced data filtering capabilities
- Enhance the data retrieval layer with scaling and load-balancing features
- Detailed monitoring of the Post Mortem system usage

- Already up and running! (http://pm-api-pro.cern.ch/)

Data Collection Layer

- Support dynamic load distribution
- Enable horizontal scalability
- Increase service maintainability and availability
- Provide multiple and pluggable protocols for data collection

- Kafka architecture studied by CALS team is an option (research is ongoing)

Data Storage Layer

- Distributed storage solution
- High availability with advanced fault tolerance
- Consistency ensured by underlying implementation
- Flexibility to support different file formats
- High throughput and low latency

- Research is mostly finished. Developments are being planned.

Data Storage Layer

- Serialization format study
  - Multiple serialization formats have been studied: JSON, BSON, Avro, …
  - Multiple compression techniques evaluated: Deflate, Gzip2, Snappy, xz, …
  - Avro was presenting the best results with Deflate
- Storage solution evaluation
  - Multiple storage systems were compared using representative sets of PM data: CEPH, MongoDB, HDFS, GlusterFS
  - GlusterFS was performing best for separated write/read workloads
  - CEPH was performing best for mixed workloads, especially with small files

NEXTGEN CALS Integration

- Low latency use cases (IQC, XPOC, SPSQC) still prevent the full integration with NEXTGEN CALS infrastructure
- Very small file size (in large quantities) might impact the query execution time significantly
- Remaining use cases, data collection, storage and retrieval API might be shared to provide the user the best possible data

Thank you for attention!
Questions are welcome!

Besides the authors stated in INDICO, additional credit goes to former members of the team:
Mateusz Koza, Roberto Orlandi, Matthias Poeschl.