EPJ Web of Conferences **214**, 04058 (2019) https://doi.org/10.1051/epjconf/201921404058
CHEP 2018

Evolution of the Hadoop Platform and Ecosystem for High Energy Physics

Zbigniew Baranowski[1,*], *Emil* Kleszcz[1,**], *Prasanth* Kothuri[1,***], *Luca* Canali[1,****], *Riccardo* Castellotti[1,†], *Manuel* Martin Marquez[1,‡], *Nuno Guilherme* Matos de Barros[1,§], *Evangelos* Motesnitsalis[1,¶], *Piotr* Mrowczynski[1,‖], and *Jose Carlos* Luna Duran[1,**]

[1]CERN, Geneva, Switzerland

Abstract. The interest in using scalable data processing solutions based on Apache Hadoop ecosystem is constantly growing in the High Energy Physics (HEP) community. This drives the need for increased reliability and availability of the central Hadoop service and underlying infrastructure provided to the community by the CERN IT department. This paper reports on the overall status of the Hadoop platform and related Hadoop and Spark service at CERN, detailing recent enhancements and features introduced in many areas including the service configuration, availability, alerting, monitoring and data protection, in order to meet the new requirements posed by the users' community.

1 Introduction

Apache Hadoop [1] is an open source industry-standard solution for big data storing and processing. It consists of many components that can cooperate with each other building a distributed multi-purpose platform. The key feature of Hadoop is horizontal scalability, as all the Hadoop components have been designed from the bottom up to perform distributed parallel data processing by complying to shared-nothing architecture paradigms.
Hadoop has two main components: a distributed filesystem, HDFS, and a cluster manager, YARN. This provides a storage system and cluster with multiple data processing interfaces that can operate at scale by design (shared nothing). Typically Hadoop is deployed on clusters of commodity-type servers. It offers many solutions for data analytics and data warehousing as well as stream processing and machine learning.
At CERN, Hadoop has become a common replacement for database-like systems that collect terabytes of data, and for workloads that are hard to scale with traditional relational database

*e-mail: zbigniew.baranowski@cern.ch
**e-mail: emil.kleszcz@cern.ch
***e-mail: prasanth.kothuri@cern.ch
****e-mail: luca.canali@cern.ch
†e-mail: riccardo.castellotti@cern.ch
‡e-mail: manuel.martin.marquez@cern.ch
§e-mail: nuno.guilherme.barros@cern.ch
¶e-mail: vaggelis.motesnitsalis@cern.ch
‖e-mail: piotr.mrowczynski@cern.ch
**e-mail: jose.carlos.luna@cern.ch
[1]https://hadoop.apache.org

systems. [1]. Growing demand for highly scalable data warehouses has led in 2013 to start up a Hadoop service provided by the IT department at CERN.

2 Central Service

2.1 Principals

The central CERN IT Hadoop service was formed to provide to the user community a central service: deploying software and hardware infrastructure, together with the required administration expertise on the Hadoop ecosystem. This project includes an application of best practices in the field of installation, configuration, maintenance and security of the platform. It also involves integration with the rest of the services provided by the IT department. All these objectives would be difficult to achieve in decentralized deployment model. Furthermore, support of the user community, consultancy, and knowledge sharing are the key aspects behind the service offering.

2.2 From a pilot to production

A prototype service has been started in early 2013, in response to the request from the ATLAS [2] experiment, where a new central storage system for all the events produced by the experiment was about to be built [3]. ATLAS wanted to use Hadoop ecosystem as the main data storage and access backend. The initial Hadoop service setup consisted of a single cluster containing few commodity machines. From the very beginning, the Cloudera packages have been used for software installation. During the first 3 years of running the service, a number of new use cases and sub-projects relying on the service have emerged [4]. Notably, CERN IT and WLCG[3] monitoring project [5] decided to create a data lake based on the Hadoop solutions. This led to the creation of a second cluster dedicated to the project in 2014. A significant year in the evolution of the service was 2016 when an LHC critical project, CERN Accelerator Logging System (CALS) [2], has taken the decision to develop their future storage and data access platform using the Hadoop ecosystem. This decision triggered multiple activities aiming at strengthening the service, notably introducing HDFS backups, high availability and monitoring features that led to the production-ready service in 2017.
By following the evolution in hardware trends (high capacity drives, more RAM, and SSD) and technology trends such as vitalization of computing resources, the service continued to evolve. As a consequence, new, more powerful servers have been introduced. Moreover, more specialized components such as Apache Kafka used for real-time streaming workloads or Kubernetes as a solution for dynamic scaling of Apache Spark workloads on the CERN private cloud infrastructure have been taken on board. A detailed timeline of the service evolution has been illustrated in Figure 1. Furthermore, in the table 1 the service is presented by relevant numbers, among others the number of clusters, servers, configured capacity as well as the number of computing resources.

3 State of the art

3.1 Installation and configuration

Most of the clusters are installed on physical servers that are evenly configured. As of 2018, 4 flavors of hardware are used in the service (see Table 2). Differences in specifications profit

[2]http://atlas.cern
[3]http://wlcg.web.cern.ch/

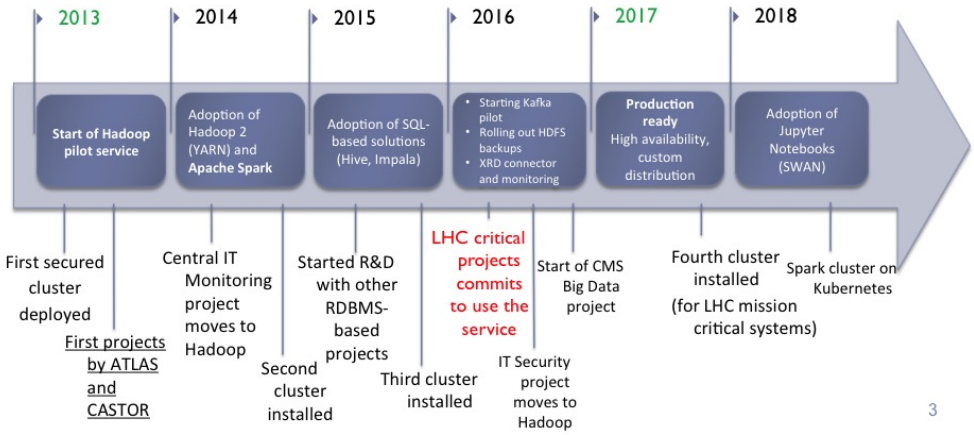

Figure 1. The timeline of the Hadoop central service at CERN (as of July 2018)

Table 1. CERN Hadoop service in numbers

Number of cluster	6	Physical servers	100+
Production clusters	3	Virtual servers	40+
Test clusters	1	Physical cores	1500+
Development clusters	2	Data growth	4 TB per day
Overall capacity	14+ PB		

Table 2. Specification of the servers used by CERN Hadoop infrastructure

	Flavor 1	Flavor 2	Flavor 3	Flavor 4
CPU physical cores	16	16	20	24
Memory	64GB	64GB	128GB	512GB
Local disks	2 x 2TB HDD	2 x 2TB HDD	2x 1TB SSD	3 x 1TB SSD
SAS controllers	1 x 6Gb/s	2 x 6Gb/s	2 x 6Gb/s	2 x 6Gb/s
Number of SAS drives	1 x 24	2 x 24	2 x 24	2 x 24
SAS drive capacity	3TB	4TB	6TB	6TB
SAS drive rotation speed	7200 rpm	5900 rpm	7200 rpm	7200 rpm

from market evolution and piggy back on CERN tender for hardware. The memory and SSD configuration in Flavor 4 (see table 2) of the Hadoop servers deserve a clarification. Internal tests and tests performed in collaboration between the CERN Hadoop service and Intel (in the context of the CERN openlab collaboration) have shown that the possibility to feed the Apache Hadoop and Apache Spark JVM processes with large amounts of memory greatly improves the performance of many workloads. Therefore the Hadoop service currently procures servers with at least 16 GB of RAM per core. Similarly, the use of fast disks (SSD) has proved to be highly beneficial for many workloads. This is explained by the speed-up that fast SSDs can help providing for data processing operations related to sorting and/or shuffling large amounts of data. Because of this insight, Hadoop servers currently have at least 3 TB of local SSD space. The actual amounts of RAM, SSD, storage capacity and a number of cores are also motivated by striking a balance between performance and cost. In order to

guarantee service resiliency to network components failures or rack power distribution units (PDU) failures, it is recommended to dislocate group of machines under different racks and switches. Therefore most of the Hadoop servers are organized in distributed islands and are physically located in various locations of the CERN Data Center [4]. With exception to the mission-critical clusters (e.g. the one for the next generation CALS project), that are backed up by redundant switches with link aggregation enabled, only one 10GigE network interface is used for network communication. All the servers in the clusters are installed with CentOS 7 and configured with Puppet [5] - CERN IT standard configuration management system. This guarantees equality of the configuration on all the machines and allows to streamline installation of the new servers and reconfiguration operations. Moreover, Puppet provides simplified integration with other services offered by the department (like LDAP, AFS, Kerberos, DNS load balancer, etc.), through a rich-ecosystem of Puppet modules implemented by various groups at CERN. CERN IT Hadoop Service has contributed to this environment by developing two Puppet modules, the first one allows to install and configure Hadoop servers, starting from OS, through mounting and formatting attached data drives, ending up on installation and configuration of each component from the Hadoop ecosystem. The second module is dedicated to client machines, and it installs only the relevant binaries and configuration for clients to allow them to interact remotely with the Hadoop clusters.

3.2 Software

From the beginning of the service, Cloudera, one of the most popular Hadoop distributors, has been chosen as the software stack vendor. Since the project always relied on the community edition, this choice was not associated with additional costs. During the lifetime of the service and on-boarding multiple projects, it turned out that holding full control of the software and bundled features are of great importance for service stability. Therefore starting from 2017 it was decided to gradually move towards standard Apache Software Foundation distribution (more details about this change can be found in chapter 4.2).

3.3 Components

The Hadoop ecosystem is growing very fast and provides solutions for a wide range of uses and needs, like batch and streaming processing, query engines, analytics frameworks, NoSQL databases and more. The list below incorporates some of the most popular Hadoop technologies available in the clusters offered by CERN Hadoop Service:

- Hadoop Distributed File System (HDFS): the core component for storing petabytes of data in a stripped, distributed and fault tolerant manner.
- YARN: a central computing resource coordinator for a Hadoop cluster.
- MapReduce: a computing model for performing distributed data processing by spawning multiple Java Virtual Machines around the cluster to execute custom business logic typically implemented in Java.
- Apache Spark[6]: general purpose engine for large-scale data processing, considered as MapReduce successor. Provides a rich framework for a wide range of applications, including ETL and data preparation, SQL processing, machine learning and stream processing.

[4] http://information-technology.web.cern.ch/about/computer-centre
[5] https://puppet.com
[6] https://spark.apache.org

- Apache HBase[7]: distributed and scalable columnar database built on top of Hadoop for random and real-time read/write access to large amounts of data.
- Apache Kafka[8]: a high-throughput, distributed, publish-subscribe messaging system.
- Apache Hive[9]: provides SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.
- Apache Impala[10]: SQL query engine optimized for low latency and high concurrency queries thanks to direct access to data and RAM memory.
- Apache Pig[11]: a platform for analyzing and processing large data sets with using a high-level scripting language.
- Apache Sqoop[12]: a tool for transferring data between Apache Hadoop and relational databases.
- SWAN (Service for Web-based ANalysis) [6] is a platform to perform interactive data analysis in the cloud. It provides user-friendly interfaces with hosted Jupyter notebooks to run PySpark workloads at scale.

3.4 Security

Since the beginning of the service, all the clusters have been running in a secured mode - all the components required authentication via CERN centrally hosted Kerberos service. In 2018 Hadoop user groups have been integrated with e-groups, so-called CERN service for grouping computing accounts. This allowed introducing fine-grained authorization policies. Each cluster can be now accessed only by accounts belonging to a certain e-group. This also allowed to use e-groups by end users for defining ACLs on their resources like HDFS directories or HBase tables. The service administrators can also use e-groups to create advanced policies for YARN queue placement, where members of a certain group can have a different amount of computing resources available than others.

3.5 High availability

In a default configuration, the Hadoop platform has single points of failure as critical master components are not redundant. Notably, HDFS namenode, the component that keeps an image of the entire file system namespace, and YARN resource manager, responsible for job submissions on a cluster and for resource scheduling, are lacking redundancy by default. In such a situation, any hardware or software failure of a server that hosts those services leads to unavailability of the entire cluster. Moreover, most of the service level configurations are not dynamic and require a restart of daemon processes. Also, in this scenario service reconfiguration causes full downtime of the cluster.
Such service downtime events are not desired by mission-critical systems hosted in Hadoop clusters, that should be operable 24/7. Therefore since 2017, all the production clusters have been reconfigured with a redundant master service, notably HDFS, YARN, and HBase have standby services on an additional server. Since then, in case of an important failure such as hardware, software, processes becoming unresponsive or just because of a maintenance

[7]https://hbase.apache.org
[8]https://kafka.apache.org
[9]https://hive.apache.org
[10]https://impala.apache.org
[11]https://pig.apache.org
[12]https://sqoop.apache.org

restart, the affected master services will be automatically transferred to standby machines. Furthermore, all the client RPC requests will be redirected to the new master service locations, without any additional action from an administrator. This new configuration provides much smoother user experience where platform failures are in most cases transparent for the higher stacks of the applications using Hadoop components.

3.6 Monitoring

Service monitoring consists of multiple layers. Basic host metrics and operating system health are collected via a central IT Monitoring system [5]. This framework also provides alerting about a large range of hardware failures (failed disk, memory module, motherboard, PSU etc.).
Since 2017, service level metrics are sent form all Hadoop daemons to a central Elasticsearch[13] instance. For this purpose, a special connector has been developed integrating Hadoop native metric system (metrics2) with the Elastic Search HTTP endpoint.
Starting from 2016, an overall cluster health check is performed via custom probes (shell scripts) that test major functionality of a Hadoop cluster like the ability to create and modify HDFS files, submitting test YARN jobs, executing Hive and HBase queries. On top of that, the scripts also check the availability of all the components daemons, whether they are up and running and reporting to coordinator machines. In case any of the probes fails, an email or SMS report is sent to the service administrators.

3.7 HDFS backup

HDFS backups and recoveries have been introduced in 2016. The backup functionality takes regular snapshots of selected HDFS directories, containing critical data and sends them to the CERN advanced storage and tapes system (CASTOR) [7]. In order to avoid duplication of backed-up data, two modes of taking backups have been implemented by the service team. Level 0, that performs a full copy of the files in the root directory and its sub-directories, and incremental level 1, which copies files that have been added or modified since the last backup run. Level 0 typically runs once or twice a year, and the incremental is happening every single day.
The backup process, no matter in which mode, invokes a map-reduce job that is executed on each server and produces a consolidated backup file containing all the local blocks from the files chosen to be backed-up. At the end of the process, each server is sending its own consolidated backup file to the CASTOR storage via the XRootD protocol [14]. In order to allow a map-reduce job to communicate with XRootD protocol a special connector, implementing an external file system, has been implemented (more details in section 4.1). All metadata information about files and data blocks being backed-up are maintained in a relational database, designed and deployed for this purpose.
In order to recover all or a portion of data (files and directories), a recovery map-reduce job has to be invoked. Based on the information stored in a database the job recreates the original content of files and directory structures by stitching blocks from multiple backup pieces into integral files.

[13]https://www.elastic.co/
[14]http://www.xrootd.org

4 Further evolution

In this chapter, we will cover new features that are constantly developed in order to bring better value from the service platforms as well as resources available in the CERN Data Centre.

4.1 XRootD connector

Hadoop XRootD connector is a library developed by CERN IT Hadoop service that binds Hadoop-based file system API with XRootD native client. This project had originally started to provide a solution to send backup data to CASTOR storage system (see 3.7). However, through its further development and evolution, notably with physics data processing for CMS Big Data project [9], it became a generic solution for communication between Hadoop stack and any external XRootD protocol-based storage, notably CASTOR and EOS [8]. Because the connector integrates with XRootD protocol on a files system abstraction layer (via Java Native Interface calls, JNI), the most components of the Hadoop stack (including Spark, MapReduce, Hive etc.) can transparently read and write the data to EOS or CASTOR. For example, the CMS Big Data project started to utilize Apache Spark for physics analysis with input data read directly from the EOS instance of the CMS experiment. The authentication to the end storage can be performed by client-provided Kerberos Ticket Granting Ticket (TGT) or by a Grid Certificate. Thus, one has to ensure that any of these is properly propagated to all the executors inside the Hadoop cluster.

4.2 Towards Apache Hadoop

In order to gain more control of the core Hadoop software stack and its configuration, in 2017 the CERN IT Hadoop team decided to replace Cloudera distribution with a vanilla Apache Hadoop. This has provided full control over software versions, features, and patches to be deployed on the test and production systems. With its price of extra configuration efforts for infrastructure and for streamlining building and packing software, such as HDFS, YARN, Spark, HBase, and Hive. Consequently, this change allowed rapid resolutions of some lately discovered critical software bugs and a possibility of quick deployments of potential fixes. In such a workflow, numerous customizations could be incorporated into the software, as well as contributed upstream. All software maintain by the CERN IT Hadoop team is available via Maven Central Repository. This approach offers an easy integration of client development environments with a proper version of the software supported by the service. Furthermore, this ensures its compatibility and also opens the way for collaboration with other teams in the HEP community and beyond, interested in making use of the team's effort for producing and maintaining this distribution.

4.3 SWAN - User Web Interface for Analytics

Since 2017, we have been actively collaborating with the SWAN (Service for web-based analysis [6]) team in order to integrate the SWAN service with Hadoop clusters. SWAN project is an interactive platform that combines code, equations, text, and visualizations based on Jupyter notebooks, initially developed for physics data analysis. It offers a very powerful Python-based toolkit for data exploration and visualization that brings great value to data-oriented platforms like Hadoop ecosystem. The goal of the integration with the SWAN service was to allow users to run on a Hadoop cluster any Spark code written in a SWAN notebook This objective has been accomplished and became generally available to all service users in early 2018.

4.4 Spark on Kubernetes

The growing popularity of Apache Spark has led the team to evaluate new ways to deploy Spark jobs across available computing resources at CERN. The most interesting recent development by industry is to integrate Spark with Kubernetes [15]. This feature is available in Spark since version 2.3, while the implementation in the Spark project is still maturing, its popularity grows quickly, in particular for users interested in running Spark on cloud resources. Such an approach appears to be a good solution when data locality is not needed, for example for CPU and memory intensive rather than IO-intensive workloads. Another scenario would be when processing data from non-HDFS storage, like EOS (this is possible thanks to Hadoop XRootD connector) or Kafka streams.
The biggest advantage of using Kubernetes clusters for Spark job deployment is their ability to offload computing to cloud resources and utilize the elasticity of such environments. When using CERN Kubernetes clusters, for example, users can start their Spark jobs on resources provided by CERN OpenStack (private cloud). This offers the possibility to use resources in an opportunistic way, which can result in profiting of a larger pool of computation resources than what would have been possible in Hadoop clusters, where resources are statically allocated.
So Spark on Kubernetes can become a great extension for employing more computing resources when and where they are needed by utilizing cloud resources.

5 The Analytics Platform

By combining multiple infrastructures and systems developed at CERN and already reported in this paper, notably Hadoop platform, with SWAN service, Kubernetes and EOS storage we achieved a modern, powerful and scalable platform for data analysis and analytics. The full architecture and interaction between the platform components are illustrated in Figure 2. One of the key features of the platform is usage simplicity - by facing Jupyter Notebooks (via SWAN service), users can easily develop, execute and share their code. All the submitted workloads are automatically distributed by Spark to the multiple machines or containers of a Hadoop or Kubernetes cluster. User depending on required computing resources can specify to use either of them. We expect that for frequent, IO and computing intensive production workloads with a need for deterministic time to complete users will continue to profit of Hadoop clusters and HDFS storage. For ad-hoc use cases or heavy compute-intensive workloads (like physics data reduction), that cannot be satisfied by the resource available in Hadoop service using Spark on Kubernetes can provide a better option. No matter which cluster manager is used for execution of a Spark job, it can access data on EOS or HDFS or both at the same time.
cite.

6 Summary

The CERN central service for data analytics offers solutions in the domain of "big data" and open source for data processing at scale. In particular, a service based on the Apache Hadoop ecosystem has been built to support the users' community in dealing with big data workloads. Through the years of running the platform, related services have significantly evolved to meet the requirements and the expected quality of production service. Notably, the service has expanded to its maturity by strengthening high availability, redundancy, security, data backup

[15] https://kubernetes.io

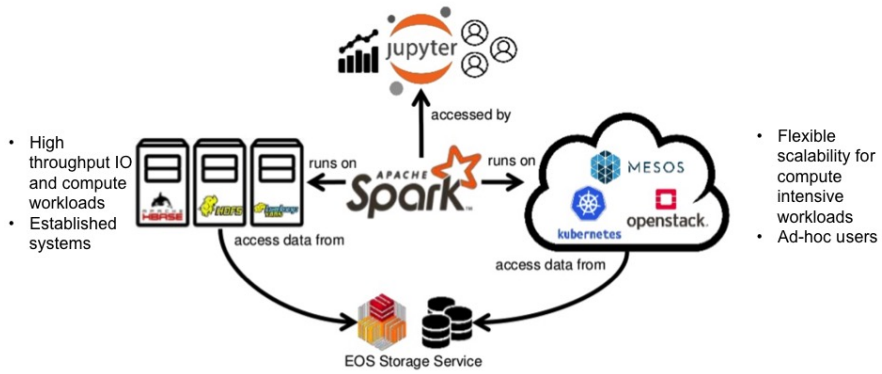

Figure 2. The analytics platform outlook

and recovery, usage of cloud computing and storage resources, as well as, integrating with frontend solutions for web hosted notebooks.
The service has been successfully adopted by multiple groups at CERN and in HEP, including WLCG, computer monitoring, security, accelerator logging, and physics analysis teams.

Acknowledgements

The authors would like to express their gratitude to former team members that made a significant contribution to the CERN IT Hadoop service: Dirk Duellmann, Daniel Lanza, Luca Menichetti, Kacper Surdy and, Rainer Toebbicke. The evolution of the service has profited of the collaboration of Intel in the context of the CERN openlab project and CMS Bigdata project. The collaboration with the SWAN team at CERN has been instrumental to the implementation of the integration of Spark service with the SWAN notebook service. The service could not have grown to the current level without the use cases and collaboration of CERN and HEP users, notably: the collaboration with our users community, at CERN IT department, including monitoring and security, the collaboration with the experiments: ATLAS Eventindex project, ATLAS distributed computing, CMS Bigdata, CMS Spark projects, and collaboration with the beams department for the accelerators and NXCALS project and for controls data analysis.

References

[1] Zbigniew Baranowski et al (2014) Sequential data access with Oracle and Hadoop: a performance comparison J. Phys.: Conf. Ser. 513 042001

[2] Roderic C, Billen R, Gaspar Aparicio R C, Grancher E, Khodabandeh A, Seguera Chinchilla N, 2009, The LHC Logging Service : Handling terabytes of on-line data (CERN-ATS-2009- 099)

[3] Barberis D et al. 2014 The ATLAS EventIndex: an event catalogue for experiments collecting large amounts of data, J. Phys.: Conf. Ser. 513 042002

[4] D Duellmann et al 2017 Hadoop and friends - first experience at CERN with a new platform for high throughput analysis steps J. Phys.: Conf. Ser. 898 072034

[5] A Aimar et al 2017 Unified Monitoring Architecture for IT and Grid Services J. Phys.: Conf. Ser. 898 092033

[6] Piparo, Danilo et al. “SWAN: A service for interactive analysis in the cloud.” Future Generation Comp. Syst. 78 (2018): 1071-1078.

[7] Lo Presti G, Bärring O, Alasdair E,Garcia Rioja R M, Ponce S, Taurelli G, Waldron D,Coelho M (2007) CASTOR: A Distributed Storage Resource Facility for High Performance Data Processing at CERN Proc. of the 24th IEEE Conf. on Mass Storage Systems and Technologies (IEEE Computer Society) p 275-280

[8] Peters A, Sindrilaru E and Adde (2015) EOS as the present and future solution for data storage at CERN J. Phys. Conf. Ser. 664 042042

[9] Oliver Gutsche et al (2018) CMS Analysis and Data Reduction with Apache Spark J. Phys.: Conf. Ser. 1085 042030