Monitoring Evolution at CERN

2015 J. Phys.: Conf. Ser. 664 052002

(http://iopscience.iop.org/1742-6596/664/5/052002)

Journal of Physics: Conference Series **664** (2015) 052002 doi:10.1088/1742-6596/664/5/052002

Monitoring Evolution at CERN

P. Andrade[1], **B. Fiorini**[1], **S. Murphy**[1], **L. Pigueiras**[1], **M. Santos**[1]
[1] European Organisation for Nuclear Research CERN, CH-1211 Geneva, Switzerland

Abstract. Over the past two years, the operation of the CERN Data Centres went through significant changes with the introduction of new mechanisms for hardware procurement, new services for cloud provisioning and configuration management, among other improvements. These changes resulted in an increase of resources being operated in a more dynamic environment. Today, the CERN Data Centres provide over 11000 multi-core processor servers, 130 PB disk servers, 100 PB tape robots, and 150 high performance tape drives. To cope with these developments, an evolution of the data centre monitoring tools was also required. This modernisation was based on a number of guiding rules: sustain the increase of resources, adapt to the new dynamic nature of the data centres, make monitoring data easier to share, give more flexibility to Service Managers on how they publish and consume monitoring metrics and logs, establish a common repository of monitoring data, optimise the handling of monitoring notifications, and replace the previous toolset by new open source technologies with large adoption and community support. This contribution describes how these improvements were delivered, present the architecture and technologies of the new monitoring tools, and review the experience of its production deployment.

1. Introduction

The deployment of new tools and workflows to manage CERN Data Centres in the areas of procurement, installation, provisioning, and configuration lead to a significant increase in the number of resources to manage. Moreover these resources are now operated under a more dynamic environment. Table 1 provides an overview of the current scale of CERN Data Centres. In addition to this continuous increase of resources, the demand for better tools to monitor these resources and extract more knowledge out of monitoring data also increased. E.g. automatically generate dashboards combining host data with service site, quickly generate a 10-years report for a given service metric, etc.

Resources	Total	Geneva	Budapest
Number of Servers	13,552	10,843	2,709
Number of Cores	152,639	109,359	43,280
Total Memory Capacity (TB)	588	416	172
Total Disk Space (TB)	189,070	117,345	71,725

Table 1. CERN Data Centres resources

When bringing these requests together it became clear that the old monitoring tools (Lemon[1] and SLS[2]) should be replaced: they could not scale to the current needs, the code was old and difficult to maintain, and provided limited functionality. Section 2 introduces how we classified

the new monitoring solutions into different areas, Section 3 explains the architecture defined to address the requirements of each monitoring area, and Section 4 presents the technologies selected to implement the monitoring architecture. The current status and future work of the monitoring infrastructure is explained in Section 5 followed by a last section with conclusions.

2. Monitoring Areas

To design the most appropriate architecture and identify suitable technologies for better monitoring solutions, four major monitoring areas were identified. Knowing in advance how we want to use the monitoring data is key to set up a better system. Each monitoring area tackles different user needs:

Alerts
to get notified of problems affecting nodes/services

Archives
to create a data lake of monitoring events for offline analysis

Displays
to plot relevant monitoring data in realtime

Streaming
to programmatically process monitoring data

3. Monitoring Architecture

The new monitoring architecture is based on the lambda architecture[3] data processing principles. A lambda architecture aims at providing a scalable, robust, fault-tolerant system capable of serving streaming and batch workloads. In addition to these use cases, we have extended the lambda architecture to include a third data processing layer to handle monitoring alerts. Overall the architecture should sustain large quantities of timestamped events (either metrics or logs) and be easily applied on any existing monitoring producer. Figure 1 depicts all layers that compose the monitoring architecture. For each architecture layer we can highlight the following characteristics:

- Sampling Layer
  - Continuous sampling of monitoring data supporting metrics and logs
  - Easy integration with existing monitoring producers
- Transport Layer
  - Scalable transport mechanism for all monitoring data sets
  - Easy integration with different providers and consumers
- Batch Layer
  - Common repository where all monitoring data is retained (data lake)
  - Used as base for the serving layer, disaster recovery, data replay, etc.
- Serving Layer
  - Batch views for offline data processing (e.g. data filtering, data curation, etc.)
- Speed Layer
  - Real time views on recent monitoring data
  - Support for programatic access to data streams and online human querying
- Alerts Layer
  - Quick and reliable delivery of alarms to Service Managers and System Administrators
  - Possibility to delivery alarms on multiple channels (email, ticketing system, etc.)

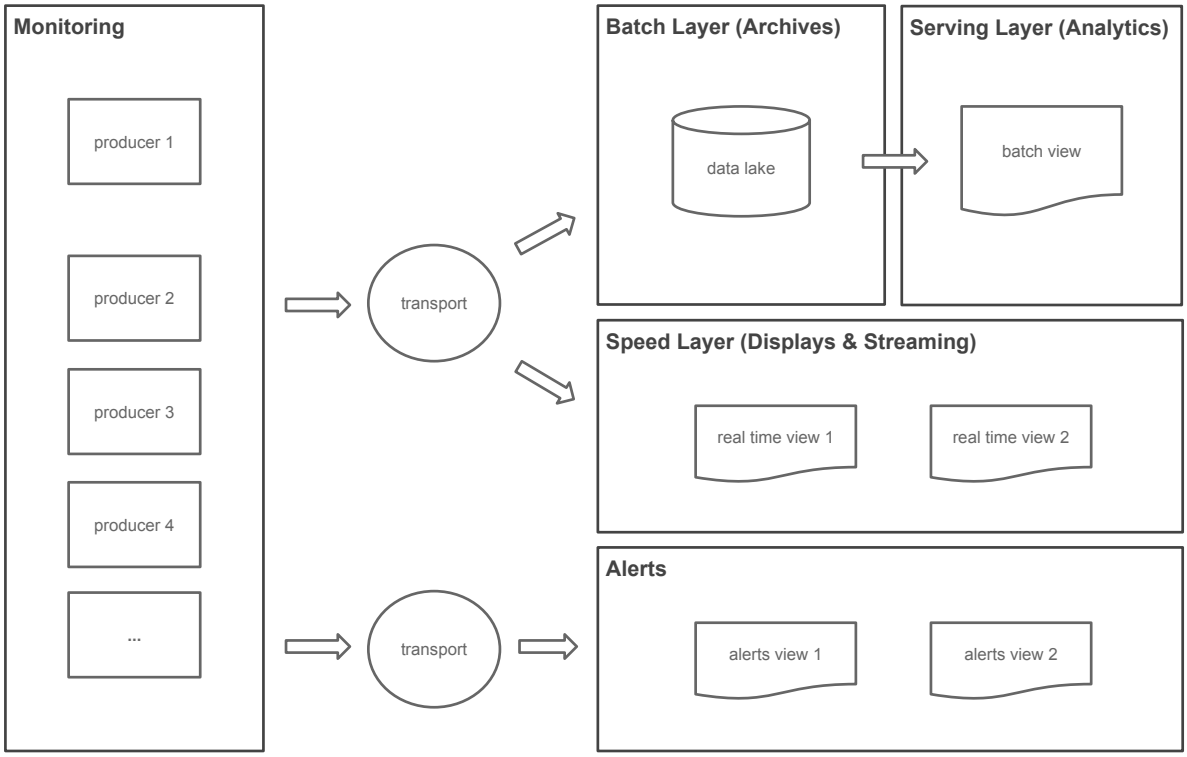

Figure 1. Monitoring lambda architecture

4. Current Implementation

To deliver concrete solutions for the monitoring architecture we based our work on existing open source technologies. These tools were selected following these conditions: functionality, large adoption outside CERN, strong community support, easy to adopt and test, modular solution easy to be replaced in case new (and better) solutions are found. The following technologies were tested and integrated in a common monitoring workflow as shown in Figure 2.

4.1. Flume

For the transport layer we selected Flume [4]. Flume is a distributed service for collecting large amounts of data. It is robust and fault tolerant and can horizontally scale due to its multi-tier deployment. It provides many ready-to-use input and output plugins, such as Avro, Thrift, JMS, Syslog, HTTP, ElasticSearch, HDFS, and allows the implementation of custom ones. We have one Flume agent running in each data centre node producing monitoring samples (metrics and logs) that are sent to a second tier of aggregation agents. Finally, a third tier of agents get the aggregated data and write it into HDFS and ElasticSearch.

4.2. HDFS

For the batch layer we selected Hadoop HDFS[5]. Hadoop HDFS is a distributed fault-tolerant filesystem for low cost hardware, suitable for applications with large data sets. Data is written by Flume HDFS sinks and is organised per producer, per hostgroup, and per hour. To avoid keeping too many smaller files in HDFS a day aggregation/deduplication MapReduce job is executed once per day segmenting the date in 1GB files. For the time being there is no data expiration policy.

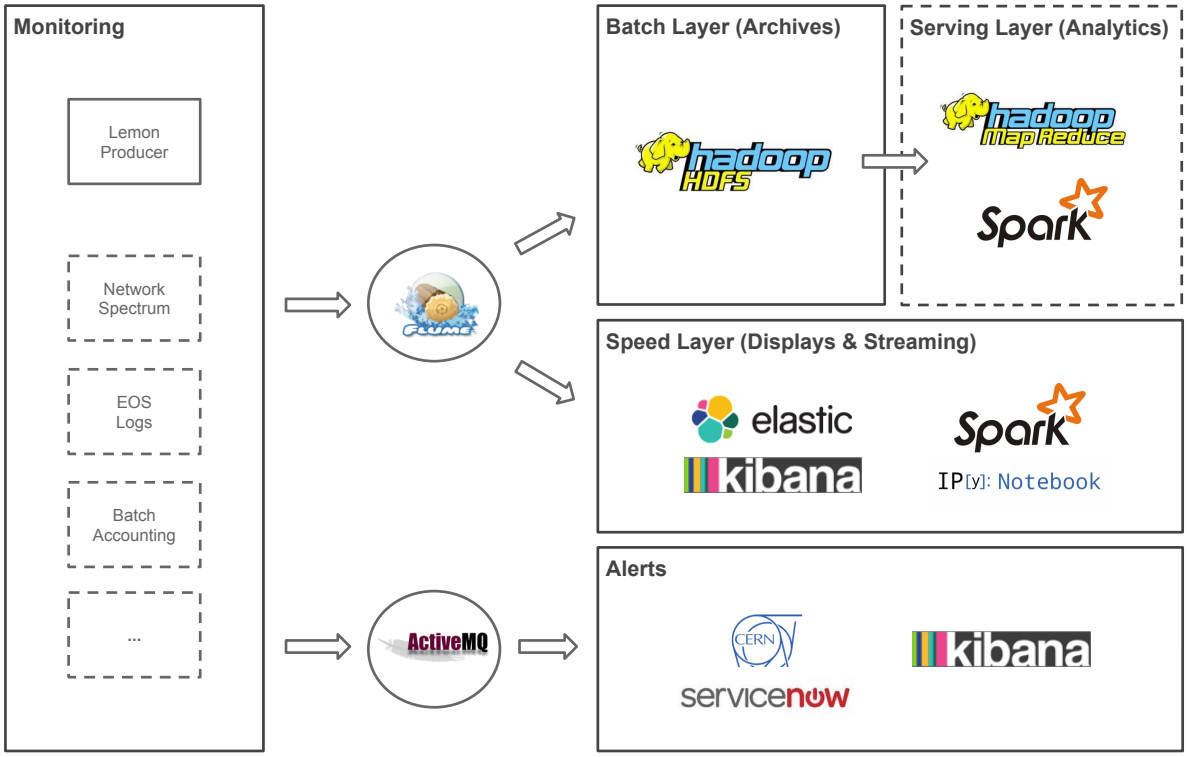

Figure 2. Monitoring technologies

4.3. ElasticSearch
For the speed layer we selected ElasticSearch [6]. ElasticSearch is a distributed RESTful search and analytics engine. It provides real time acquisition capabilities and all data is indexed in real time. ElasticSearch offers built-in features for automatic sharding and replication. It is schema-free and document-oriented (JSON). To visualise the data stored in ElasticSearch we use Kibana [7]. Kibana allows the dynamic creation of dashboards with simple point and click. It is designed to query and analyse logs but also works with time-series data. Raw data stored in ElasticSearch is aggregated in multiple bins with different granularity to provide up to 1 year of metrics and logs history.

4.4. Spark
For the speed layer we are starting to evaluate Spark [9]. Spark is a distributed large-scale data processing engine, with support for streaming, SQL like queries, machine learning, and graph processing. We are starting to test it with the Spark streaming API on a number of well identified use cases (e.g. data centre temperatures and power, detection of nodes no contact, etc.) to understand how to ingest data and process it.

4.5. GNI
For the alerts layer we have developed the General Notification Infrastructure (GNI). This is a collection of tools to handle alarms triggered in the data centre nodes. Different from the other tools, GNI relies on a dedicated transport layer based on messaging brokers running ActiveMQ. Different producers publish alarms to the messaging infrastructure, which are consumed by three components: one consumer creates tickets for non-masked nodes in the CERN central ticketing

system (based on ServiceNow [8]), a second consumer writes the alarms into ElasticSearch to allow the creation of dashboards in Kibana, and finally a third consumer can deliver alarms via email or SMS.

5. Current State and Future Work

Using the technologies presented in the previous section, the IT monitoring infrastructure has been deployed in production. This infrastructure runs almost entirely on virtual machines provided by CERN cloud service based on OpenStack. All nodes and services are managed through CERN configuration service based on Puppet. The monitoring infrastructure nodes represent less then 1% of the total number of data centre nodes. The most relevant user facing services are:

- Speed Layer: https://timber.cern.ch (displays for syslog data)
- Speed Layer: https://meter.cern.ch (displays for host and service metrics)
- Alarms Layer: https://gni.cern.ch (displays for host and service alarms)

Table 2 provides more details about the infrastructure major components. On aspect to note is the different data volumes between the batch layer (HDFS) and the speed layer (ElasticSearch). This is explained by the different data retention and data resolution policies: HDFS keeps raw data for several years while ElasticSearch keeps aggregated data for 1 year.

Component	Production Date	Current Release	Number of Nodes	Data Volume
Flume	Sep 2013	1.5.0	72	-
HDFS	Sep 2013	5.3.0	24	32 TB
ElasticSearch	June 2014	1.4.4	43	1,8 TB
GNI	Apr 2013	-	16	-

Table 2. Monitoring infrastructure

In addition to this infrastructure, other dedicated deployments of Flume and ElasticSearch were performed by individual teams for dedicated monitoring solutions. To facilitate this type of deployments we are working on providing Platform-as-a-Service (PaaS) solutions using OpenStack Heat[10]. Another area of work is the deployment in production of Apache Spark to provide a online streaming analytics solution. Finally, we are also working on a Business Continuity Plan studying different deployment strategies to make sure the monitoring infrastructure runs continuously. This activity explores the benefit of having two data centres geographically distributed between Geneva and Budapest.

6. Conclusions

A new set of technologies are now in use to monitor CERN Data Centres replacing the old toolset. These tools can handle the evolution of CERN Data Centres in terms of number of resources and new workflows. The clear identification of the key monitoring areas, the definition of a monitoring architecture based on the lambda architecture, and the selection of existing open source technologies were key factors in the successful deployment of an improved monitoring infrastructure.

7. References

[1] http://lemon.cern.ch
[2] http://sls.cern.ch
[3] http://lambda-architecture.net

[4] https://flume.apache.org
[5] http://hadoop.apache.org
[6] https://www.elastic.co/products/elasticsearch
[7] https://www.elastic.co/products/kibana
[8] http://www.servicenow.com
[9] https://spark.apache.org
[10] http://docs.openstack.org/developer/heat/