Unified Monitoring Architecture for IT and Grid Services

Unified Monitoring Architecture for IT and Grid Services

A Aimar[1], A Aguado Corman[1], P Andrade[1], S Belov[2], J Delgado Fernandez[1], B Garrido Bear[1], M Georgiou[1], E Karavakis[1], L Magnoni[1], R Rama Ballesteros[1], H Riahi[1], J Rodriguez Martinez[1], P Saiz[1] and D Zolnai[1]

[1] CERN, European Organization for Nuclear Research

[2] JINR, Joint Institute for Nuclear Research

Email: Alberto.Aimar@cern.ch, Pedro.Andrade@cern.ch, Edward.Karavakis@cern.ch, Luca.Magnoni@cern.ch, Pablo.Saiz@cern.ch

Abstract. This paper provides a detailed overview of the Unified Monitoring Architecture (UMA) that aims at merging the monitoring of the CERN IT data centres and the WLCG monitoring using common and widely-adopted open source technologies such as Flume, Elasticsearch, Hadoop, Spark, Kibana, Grafana and Zeppelin. It provides insights and details on the lessons learned, explaining the work performed in order to monitor the CERN IT data centres and the WLCG computing activities such as the job processing, data access and transfers, and the status of sites and services.

1. Introduction

For over a decade, the Large Hadron Collider (LHC) [1] experiments have been relying on advanced and specialised Worldwide LHC Computing Grid (WLCG) [2] dashboards [3] for monitoring, visualising and reporting the computing activities of the LHC across its distributed grid and cloud resources.

In the recent years, in order to cope with the increase of volume and variety of the resources, the WLCG monitoring started to evolve [4][5] towards data analytics technologies such as Elasticsearch [6], Apache Hadoop [7] and Apache Spark [8]. Therefore, at the end of 2015, it was agreed to merge these WLCG monitoring services, resources and technologies with the internal CERN IT data centres monitoring [9] services that were also based on similar solutions.

The overall mandate was to migrate, in concertation with representatives of the users of the LHC experiments, the WLCG monitoring to the same technologies used for the IT monitoring while supporting the existing solutions for the experiments. It started by merging the two small IT and WLCG monitoring teams, in order to join forces to review, rethink and optimise the IT and WLCG monitoring and dashboards within a single common architecture, using the same technologies and workflows used by the CERN IT monitoring services.

This paper covers the work performed, starting from early 2016, that resulted in the definition, the development and the deployment of a Unified Monitoring Architecture (UMA) aiming at satisfying the requirements to collect, transport, store, search, process and visualise both IT and WLCG monitoring data.

2. Architecture

The architecture of UMA can be seen below in figure 1. The workflow of UMA is based on the following major components:

- Data sources: to interface with a large variety of data sources, collect and validate the incoming monitoring data sets;
- Transport: to reliably gather those data sets from the original sources to the system that will also be used as a buffer in case of a downtime of any of the following layers;
- Processing: to perform data processing on real-time data or on historical data;
- Storage and Search: to store the data offering different retention policies according to the use of the monitoring data and based on different technologies;
- Data Access: to offer to the users several well-known visualisation, reporting and data analytics technologies.

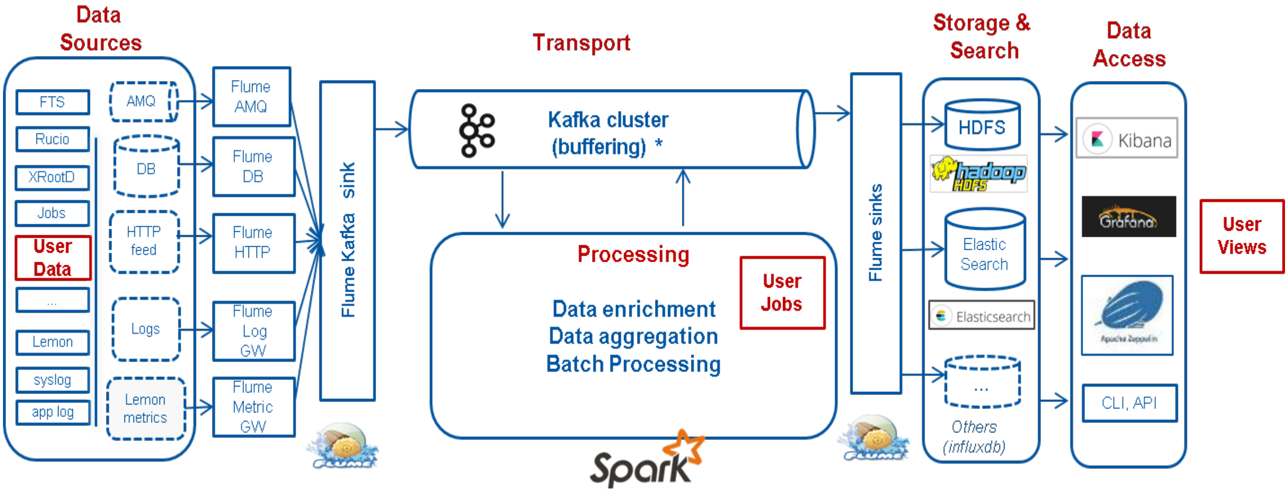


Figure 1. Unified Monitoring Architecture.

The system relies on Virtual Machines (VMs) compliant to the CERN IT Agile Infrastructure (AI) standards and tools [10] utilising OpenStack, Puppet, and several widely-used open source technologies. The selected technology stacks are loosely coupled to each other, thus, adding flexibility to the system by allowing to replace them without having to redesign the defined architecture.

2.1. Data Sources

The majority of the monitoring data sources are owned and controlled by the LHC experiments and have different ways of publishing data. For this, a set of standard ways to gather data were identified using different solutions such as with Apache ActiveMQ [11], a direct connection to a database, loading an HTTP feed, accepting log files and accepting data centre metrics (LEMON) [12]. Data is channeled via Apache Flume [13], validated, and modified if necessary. Adding new data sources to the system is documented [14] and is straightforward. A set of Apache Flume nodes is used to spread the load of the different sources. There is a second instance used for QA with a smaller sized set of Apache Flume nodes.

2.2. Transport

All the data are channeled to an Apache Kafka cluster [15] via Flume sinks in order to have a unified, high-throughput and low-latency platform for handling the data in real-time. Data are published in different domain-specific topics that are partitioned and distributed throughout the cluster with 3 replicas in total. For this, a cluster of 20 Kafka servers is used for production and a smaller one for QA data.

Servers are hosted on VMs with network-attached volumes that are used as a storage spool. Since the primary use of this component is to serve fresh data, topics have been partitioned to be fully spread across the cluster in order to maximise consumer's throughput. The data are currently buffered for a retention period of 12 hours with a goal to increase the retention period to 48 hours.

2.3. Processing

The system supports two different ways of processing the data; either with stream processing of real-time data or with batch processing of historical data.

Stream processing is used to serve multiple needs. First, for the data enrichment when a data source is joined and enriched with information from several other sources such as the WLCG topology meta-information. Second, for the data aggregation over time such as the creation of summary statistics over a time bin, or over other dimensions, i.e. the computation of a cumulative metric for a set of machines hosting the same service. Finally, stream processing can also be used to correlate data and to perform advanced alarming to detect anomalies and failures originating from multiple sources, i.e. data centre topology-aware alarms.

Batch processing is used to reprocess and recalculate historical data, compress old historical data and for the creation of various high-level reports that are used by multiple WLCG bodies and within CERN.

Figure 2 shows the platform used for the processing of the monitoring data collected. The platform relies on Apache Spark for a reliable and scalable processing framework, Apache Mesos [16] Marathon and Chronos for the job orchestration and scheduling and on Docker [17] for an isolated environment and a lightweight deployment of the processing jobs. Furthermore, an integration with GitLabCI [18] allows to automatise the creation of new Docker images when new code is pushed to the Git repositories.

The Monitoring processing platform is also available to teams other than the monitoring one, either in the CERN IT Department or to the various LHC Experiments user-communities, in order to analyse the monitoring metrics and logs collected. An externally-managed Git repository can be given to the monitoring team with the code of the Spark job and then the monitoring team will guarantee for its successful execution inside the Mesos cluster.

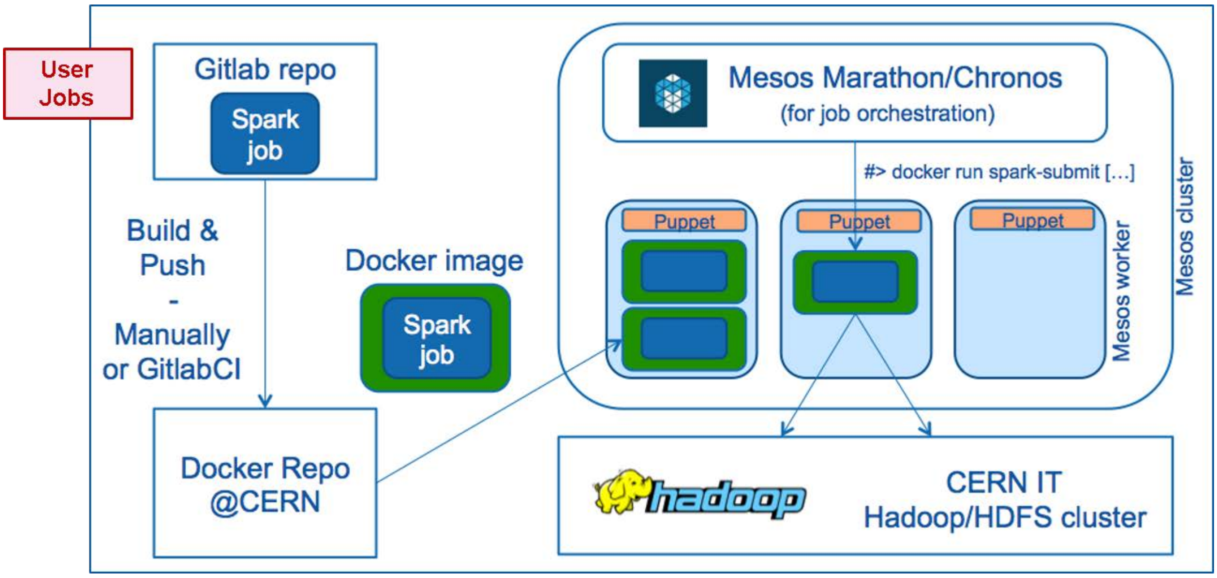


Figure 2. Monitoring Processing Platform.

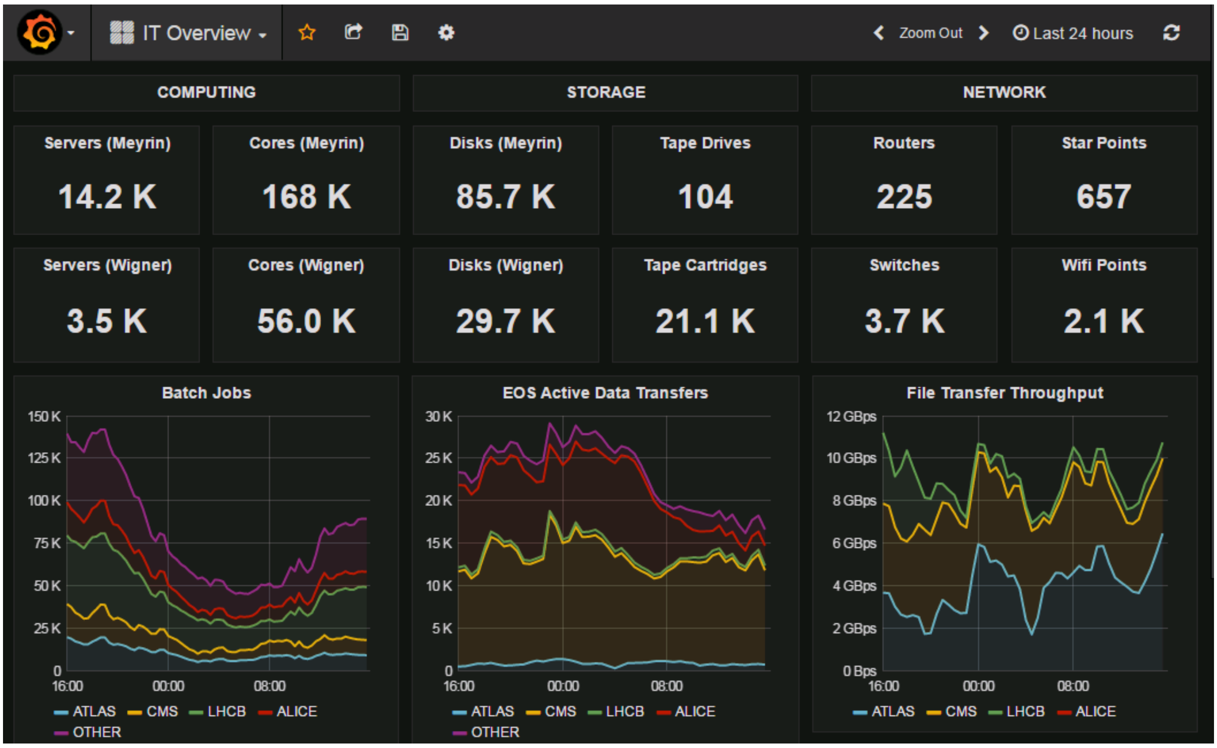
2.4. Storage and Search

A set of Flume sinks read data from Kafka and write to different storage endpoints; HDFS for long-term data archival and offline analytics, Elasticsearch for short-term storage and indexing of the data, and in the near future to other popular time-series storage solutions such as InfluxDB [19] for the medium and long-term storage of time-series data either in a raw or in an aggregated format. The Hadoop/HDFS cluster used by the project is shared by different projects and is provided and supported by the CERN IT Department. The shared cluster consists of approximately 40 nodes and currently around 210 TBs of monitoring data were stored by UMA throughout 2016. The Elasticsearch cluster used is a dedicated one provided by the CERN IT Department and it consists of 33 nodes, 25 of them being the data nodes. 6 Flume nodes are used for the HDFS sinks for the production environment and 4 nodes for the QA environment. In addition, 6 and 4 Flume nodes are used for the Elasticsearch sinks for the production and QA environment respectively.

2.5. Data Access

Users can access the monitoring data using well-known visualisation and data analytics technologies. Firstly, Kibana is offered for the visualisation of time-series metrics and logs, and it is being connected to the Elasticsearch backend. This allows the users to create dashboards with full search/filtering capabilities, and to explore and discover live the monitoring data. Secondly, Grafana [20] is offered for the visualization of time-series data serving data both for the Elasticsearch and the InfluxDB endpoints.

A set of official monitoring dashboards were created by the monitoring team and are offered to the users in Kibana and Grafana in order access metrics and logs stored in UMA. An example of such dashboards can be seen in figure 3 that is implemented in Grafana and in figure 4 that is implemented in Kibana. These official dashboards are used as a basis for the creation of customised user dashboards as they can be copied and modified directly by the users.



COMPUTING

Servers (Meyrin) 14.2 K
Cores (Meyrin) 168 K
Servers (Wigner) 3.5 K
Cores (Wigner) 56.0 K

STORAGE

Disks (Meyrin) 85.7 K
Tape Drives 104
Disks (Wigner) 29.7 K
Tape Cartridges 21.1 K

NETWORK

Routers 225
Star Points 657
Switches 3.7 K
Wifi Points 2.1 K

Figure 3. Data Centres overview in Grafana.

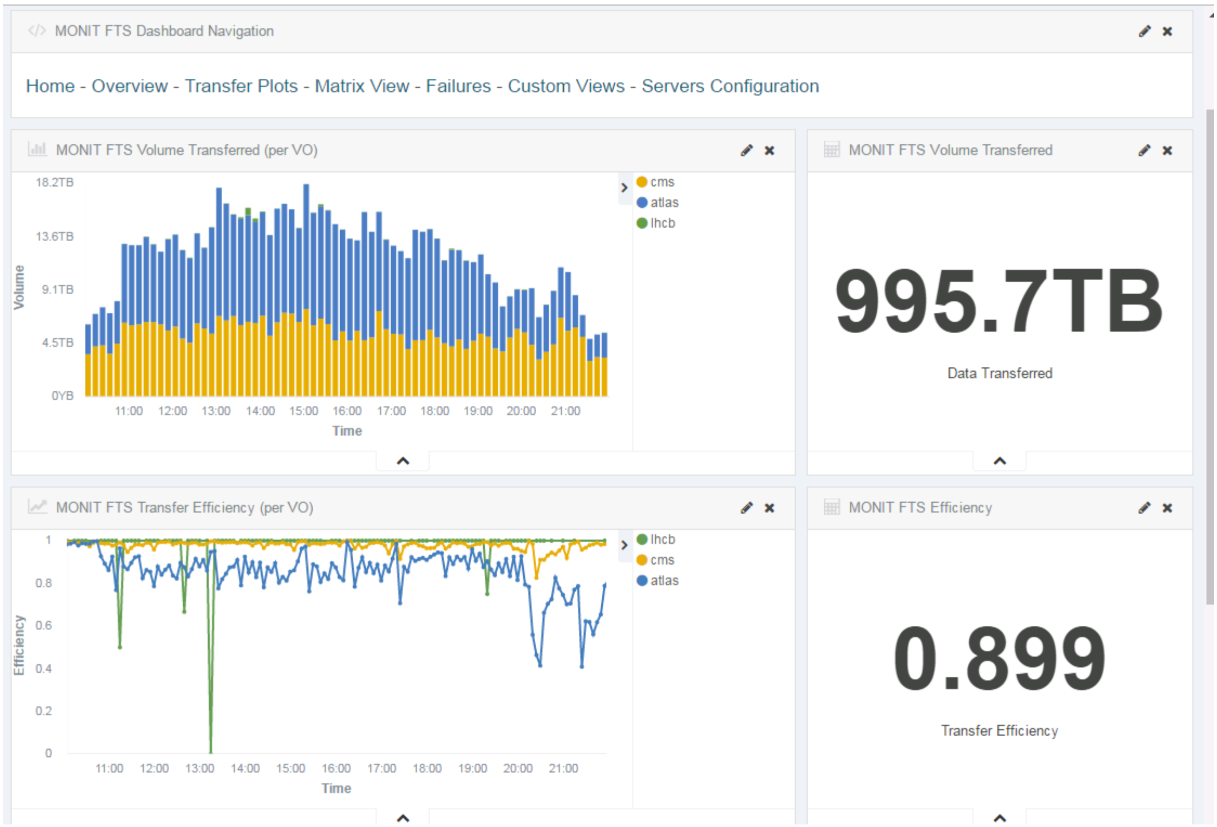
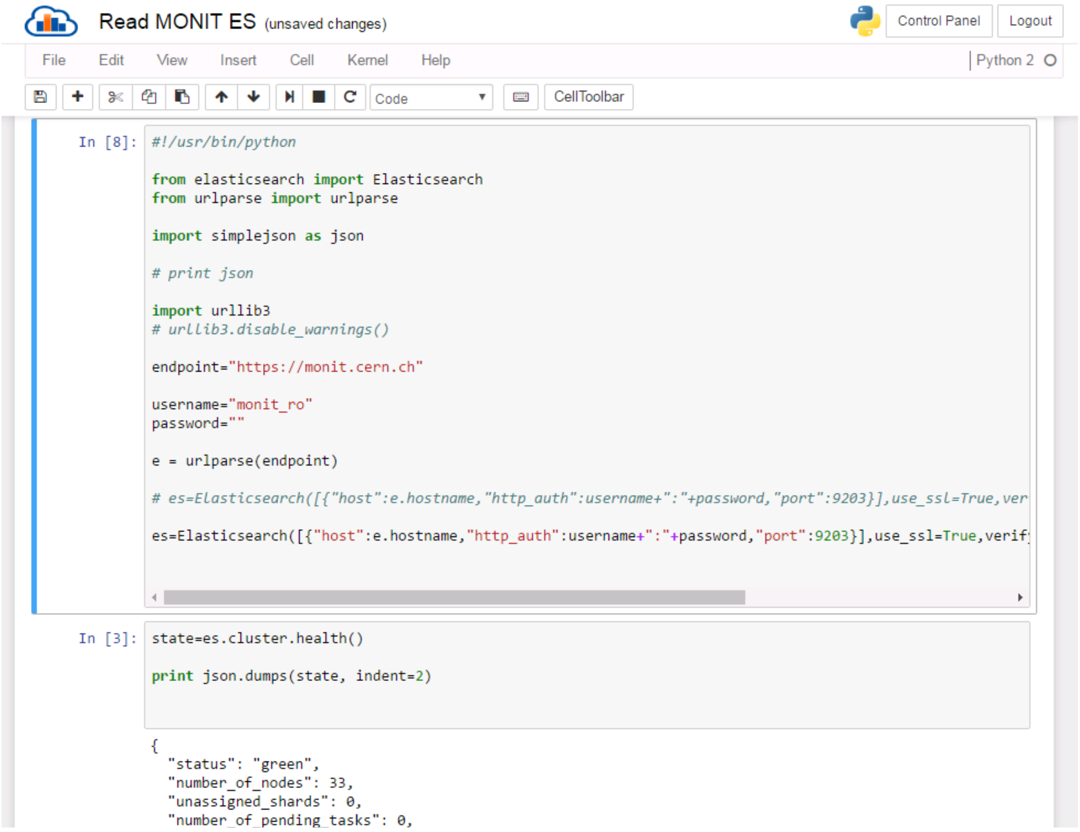


Figure 4. File Transfer Service monitoring in Kibana.

In addition to real-time dashboards, various interactive notebook technologies are offered to the users in order to perform exploratory data analysis and to share results with the user community. These interactive notebooks are collaborative web-based environments for data exploration and visualization that can also be used for “data storytelling”, explaining and documenting the way that a user goes from raw data to valuable information and insights.

Two such technologies are supported by UMA: Apache Zeppelin [21] and Swan [22], a CERN project that is based on Jupyter [23]. In Apache Zeppelin, users can interact with the live streaming data stored in Kafka and/or with data stored in HDFS in order to create interactive notebooks for data analysis or advanced reports. In Swan, users can access data stored in HDFS, Elasticsearch and/or InfluxDB, directly benefiting from the integration with multiple widely-used High Energy Physics (HEP) tools such as ROOT [24], CERNBOX [25] and CVMFS [26] in order to create interactive notebooks for data analysis or advanced reports as seen in figure 5.

Finally, the monitoring data can also be accessed by external third-party systems by using a set of well-defined CLIs and APIs.



```
#!/usr/bin/python

from elasticsearch import Elasticsearch
from urlparse import urlparse

import simplejson as json

# print json

import urllib3
# urllib3.disable_warnings()

endpoint="https://monit.cern.ch"

username="monit_ro"
password=""

e = urlparse(endpoint)

# es=Elasticsearch([{"host":e.hostname,"http_auth":username+":"+password,"port":9203}],use_ssl=True,ver

es=Elasticsearch([{"host":e.hostname,"http_auth":username+":"+password,"port":9203}],use_ssl=True,verif
```

```
state=es.cluster.health()

print json.dumps(state, indent=2)
```

```
{
  "status": "green",
  "number_of_nodes": 33,
  "unassigned_shards": 0,
  "number_of_pending_tasks": 0,
```

Figure 5. Accessing data stored in UMA with Swan.

3. Current Status and Future Work

By the first quarter of 2017, the majority of the CERN data centres and WLCG data sources were integrated into the new system and by the end of 2017, all the remaining data sources are expected to be integrated. Since the implemented solution is generic and provides well-defined ways of injecting data into the workflow, it is not only used to handle metrics from the WLCG and the CERN IT data centres but also to provide a central service for storing and accessing syslog data and service/application logs from the CERN IT Department.

The system handles approximately 250 GBs on a daily basis, which translates into more than 250 million documents per day with spikes of data at 20 KHz. The new system in its first production deployment is designed to handle up to 500 GBs per day (for a retention period of 48 hours) and should be able to handle considerably more load because the technologies based upon are horizontally scalable.

In addition to the previously described work, the monitoring team steadily advances in the replacement of the in-house implemented system for the monitoring of the data centre to Collectd [27], a widely-used open source system, which collects system and application performance metrics periodically from various metrics such as the operating system, applications, log files and external devices.

4. Conclusions

The CERN IT monitoring infrastructure handles millions of events every day coming from the CERN data centres and the computing activities of the WLCG project. In order to cope with the increasing volume and variety of monitoring data, the UMA project was born aiming at merging the two different monitoring projects together.

The newly-developed architecture, relying on state-of-the-art open source technologies and on open data formats, provides solutions for visualisation and reporting that can be extended or modified directly by the users according to their needs and their roles. For instance, it is possible to create new dashboards for the shifters and new reports for the managers, or implement additional notifications and new data aggregations directly by the service managers without any specific modification or development in the monitoring service.

The CERN UMA service is storing uninterruptedly and successfully metrics and logs from several services running on the CERN IT and the WLCG resources and will progressively replace the previous infrastructure during 2017.

References

[1] Evans L and Bryant P, "*LHC Machine*", 2008 *JINST* **3** S08001 doi:10.1088/1748-0221/3/08/S08001
[2] Bird I, "*Computing for the Large Hadron Collider*", 2011 *Annual Review of Nuclear and Particle Science*, **61**:99–118 doi:10.1146/annurev-nucl-102010-130059
[3] Andreeva J et al, "*Experiment Dashboard for monitoring computing activities of the LHC virtual organizations*", 2010 J. Grid Comput. **8** 323-339 doi:10.1007/s10723-010-9148-x
[4] Karavakis E et al, "*Processing of the WLCG monitoring data using NoSQL*", 2014 *J. Phys.: Conf. Ser.* 513 032048 doi:10.1088/1742-6596/513/3/032048
[5] Magnoni L et al, "*Monitoring WLCG with lambda-architecture: a new scalable data store and analytics platform for monitoring at petabyte scale*", 2015 *J. Phys.: Conf. Ser.* 664 052023 doi: 10.1088/1742-6596/664/5/052023
[6] Elasticsearch, http://elastic.co Retrieved: Jan, 2017
[7] Apache Hadoop, http://hadoop.apache.org Retrieved: Jan, 2017
[8] Apache Spark, http://spark.apache.org Retrieved: Jan, 2017
[9] Andrade P et al, "*Monitoring Evolution at CERN*", 2015 *J. Phys.: Conf. Ser.* 664 052002 doi: 10.1088/1742-6596/664/5/052002
[10] Bell T et al, "*Review of CERN Data Centre Infrastructure*", 2012 *J. Phys.: Conf Ser.* 396 042002 doi:10.1088/1742-6596/396/4/042002
[11] Apache ActiveMQ, http://activemq.apache.org Retrieved: Jan, 2017
[12] Babik M et al, "*LEMON - LHC Era Monitoring for Large-Scale Infrastructures*" , 2011 *J. Phys.: Conf. Ser.* 331 052025 doi:10.1088/1742-6596/331/5/052025
[13] Apache Flume, http://flume.apache.org Retrieved: Jan, 2017
[14] CERN IT Monitoring team public documentation, http://monitdocs.web.cern.ch/monitdocs Retrieved: Jan, 2017
[15] Apache Kafka, http://kafka.apache.org Retrieved: Jan, 2017
[16] Apache Mesos, http://mesos.apache.org Retrieved: Jan, 2017
[17] Docker, http://docker.com Retrieved: Jan, 2017
[18] GitLab Continuous Integration, https://about.gitlab.com/gitlab-ci/ Retrieved: Jan, 2017
[19] InfluxDB, https://www.influxdata.com/time-series-platform/influxdb/ Retrieved: Jan, 2017
[20] Grafana, http://grafana.org Retrieved: Jan, 2017
[21] Apache Zeppelin, http://zeppelin.apache.org Retrieved: Jan, 2017
[22] Piparo D et al, "*SWAN: A service for interactive analysis in the cloud*", 2016 *Future Generation Computer Systems* doi:10.1016/j.future.2016.11.035
[23] Jupyter Notebook, http://jupyter.org Retrieved: Jan, 2017
[24] Brun R and Rademakers F, "*ROOT: an object oriented data analysis framework*", 1997 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389(1-2):81-6 doi:10.1016/S0168-9002(97)00048-X
[25] Mościcki JT and Lamanna M, "*Prototyping a file sharing and synchronization service with Owncloud*", 2014 *J. Phys.: Conf. Ser.* 513 042034 doi:10.1088/1742-6596/513/4/042034
[26] Buncic P et al, "*CernVM – a virtual software appliance for LHC applications*", 2010 *J. Phys.: Conf. Ser.* 219 042003 doi:10.1088/1742-6596/219/4/042003
[27] Collectd, https://collectd.org Retrieved: Jan, 2017