

WLCG Dashboards with Unified Monitoring

Pedro Andrade¹, Alberto Aimar¹, Simone Brundu¹, Borja Garrido Bear¹, Gonzalo Menendez Borge¹, Luca Magnoni¹, Diogo Lima Nicolau¹, and Nikolay Tsvetkov¹

¹CERN, 1211 Geneva 23, Switzerland

Abstract. Monitoring of the CERN Data Centres and the WLCG infrastructure is now largely based on the new monitoring infrastructure provided by CERN IT. This is the result of the migration from several old in-house developed monitoring tools into a common monitoring infrastructure based on open source technologies such as Collectd, Flume, Kafka, Spark, InfluxDB, Grafana and others. This new infrastructure relies on CERN IT services (OpenStack, Puppet, Gitlab, DBOD, etc) and covers the full range of monitoring tasks: metrics and logs collection, alarms generation, data validation and transport, data enrichment and aggregation (where applicable), dashboards visualisation, reports generation, etc. This contribution will present the different services offered by the infrastructure today, highlight the main monitoring use cases from the CERN Data Centres and WLCG, and analyse the last years experience of moving from legacy well-established custom monitoring tools into a common open source-based infrastructure.

1 Introduction

The objective of the CERN IT monitoring team is to provide a common *Monitoring as a Service* solution for the CERN Data Centre and the WLCG collaboration. To make this possible, the project operates a common monitoring infrastructure, called MONIT [1], used by different producers and users of monitoring data. This infrastructure was first deployed in 2017 and has been growing since then. Many of the use cases supported by the MONIT infrastructure today are the result of migrating legacy custom monitoring solutions, such as the Dashboards infrastructure [2], used by CERN/IT and WLCG over the past 10 years.

This paper is organised in two main sections. A first section to present the MONIT architecture, describe the architecture key concepts, explain the main technologies used, and provide statistics on the infrastructure scale and growth. Followed by a second section to explain the migration process and status of the main WLCG dashboards from the legacy Dashboards infrastructure to the new MONIT infrastructure.

2 MONIT Architecture

2.1 Key Concepts

The MONIT infrastructure has been designed from a set of well defined key concepts. Even if the infrastructure has been evolving over time, mostly to support new use cases and new data producers, these core ideas have been kept and are the infrastructure foundation:

- Modular architecture with separate components that can be easily replaced.
- Based on open source tools widely-adopted by other external communities.
- Easy to integrate new producers via several ingestion endpoints.
- Reliable workflows with decoupled producers and consumers of data.
- Built-in support for stream processing of monitoring events.
- Multiple storage backends with different data retention and SLAs.

2.2 Architecture and Tools

Following the key concepts presented in Sect. 2.1, the MONIT architecture has been designed around five main workflow steps: data sources, data transport, data processing, data storage, and data access. The technologies selected to implement this architecture are all open source tools widely adopted by other major software vendors and communities. A high-level architecture diagram and the main technologies used is depicted in Fig. 1.

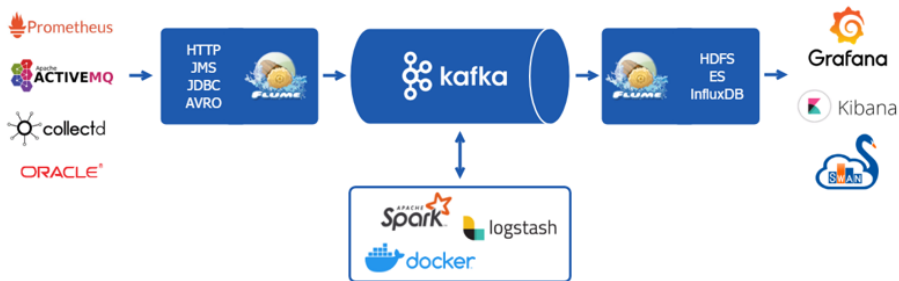


Figure 1: MONIT architecture

The data source layer is responsible for the generation of monitoring events (metrics, alarms, and logs) and their integration in the infrastructure following push or pull models. The main tools used are Collectd [3] and Prometheus [4]. The data transport layer is used to reliably transport the monitoring data from the data sources to the different storage solutions, and keep a persistent 72 hours buffer of the data. The tools used are Kafka [5], Flume [6], and Kafka Connect [7]. The data processing layer is optionally used in data streaming or batch processing modes for data enrichment or data aggregation. The main tool used is Spark [8]. At the data storage layer, data is kept in different storage backends to serve different use cases with different data retention policies. The tools used are HDFS [9], InfluxDB [10], and ElasticSearch [11]. Finally, the data access layer offers different methods to access the data, such as dashboards, reports, and alarms. The tools used are Grafana [12], Kibana [13], and SWAN [14].

2.3 Evolution

Since its initial deployment as a production service in 2017 the MONIT infrastructure has been continuously growing. Such growth has been driven by a regular increase in the number of integrated data producers. As demonstrated in Fig. 2, the number of data producers grew from less than 100 producers two years ago to close to 200 producers today. These different data sets are being accessed mostly from Grafana by more than 1500 registered active users (active users are users that connected at least once in the last 30 days).

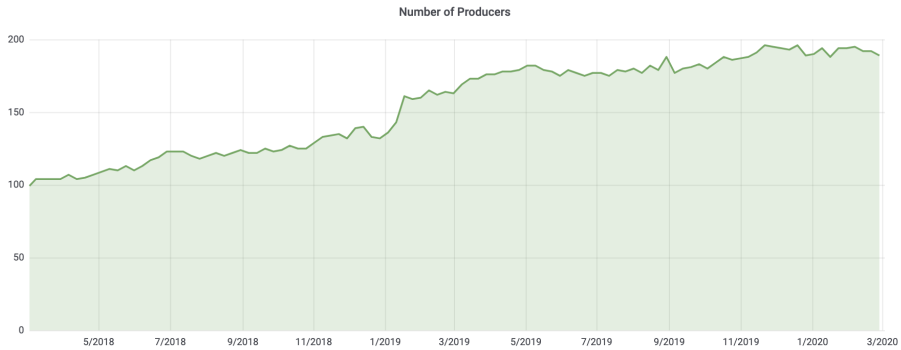


Figure 2: Number of data producers integrated in the MONIT infrastructure

Another important aspect to mention is the data rate and volume. In a typical day, the infrastructure handles around 100k documents per second and around 30MB per second. This corresponds to a daily data rate of 8.5M documents and 3.2TB of data. Fig. 3 shows a representative one hour plot. Some data sets are very "large", such as Collectd the main monitoring agent gathering operating system and service metrics from all CERN data centre nodes (many samples per minute), while others are "smaller", such as the producer of service availabilities for CERN/IT services sampling metrics with a very low sampling frequency (one sample per 15 minutes).

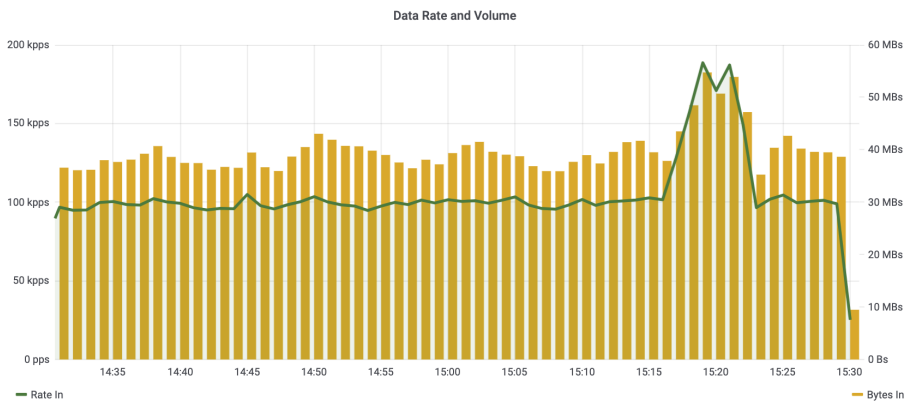


Figure 3: Example of data rate and data volume handled by the MONIT infrastructure

3 Dashboards Migration

3.1 Dashboards Overview

The dashboards migrated from the old Dashboards infrastructure to MONIT are:

- **WLCG Transfers:** shows FTS and XRootD transfer metrics between WLCG sites, used by WLCG management and sites to monitor data transfers between sites;

- **WLCG Site Monitoring:** shows aggregated availability/reliability metrics for WLCG sites, used by WLCG management and sites to monitor sites health;
- **WLCG Site Status Board:** shows experiment specific metrics for WLCG sites, used by WLCG sites to monitor sites health;
- **ATLAS DDM Transfers:** shows Rucio [15] events and traces data for ATLAS, used by ATLAS shifters to monitor data transfers between ATLAS sites;
- **ATLAS DDM Accounting:** shows Rucio aggregated data for ATLAS, used by ATLAS management and sites for accounting of data transfers;
- **ATLAS Job Accounting:** shows aggregated metrics for ATLAS workloads, used by ATLAS management and sites for accounting of job workloads
- **CMS Job/Task Monitoring:** shows aggregated and raw metrics for CMS workloads, used by CMS management and sites for monitoring of job workloads

An example of a standard Grafana dashboard is the FTS Transfers dashboard. This dashboard is available in the Grafana WLCG organisation and is shown in Fig. 4.

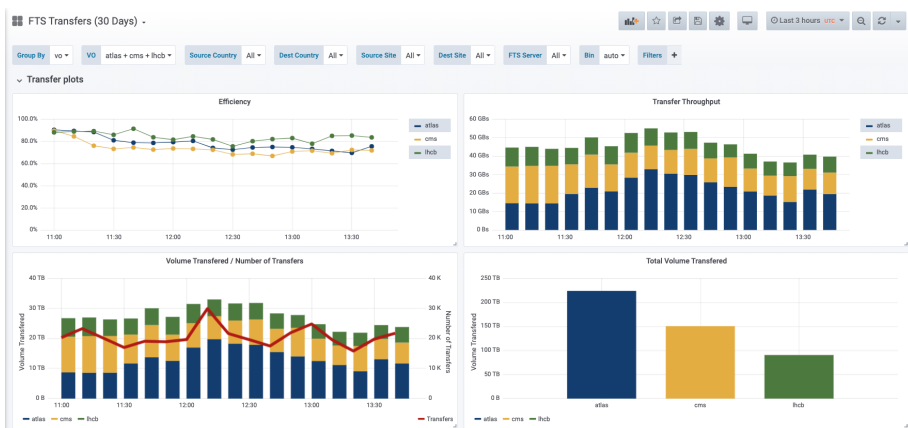


Figure 4: Grafana dashboard example - FTS Transfers dashboard

3.2 Migration Strategy

As introduced in Sect. 1, the dashboards to migrate were running for many years in the Dashboards infrastructure. Given the relevance and daily usage of these dashboards a clear migration strategy was defined and communicated in advance to the different dashboard users.

The most important aspect of this migration process was to provide in the new dashboards the same functionality as available in the old dashboards. In many cases, and profiting from the large features set of Grafana, the exact same feature was available. In few other cases the same functionality had to be reimplemented from a different visualisation perspective.

Profiting from another Grafana capability, the possibility to have several editors for each dashboard, the new dashboards were built in close collaboration with different people from WLCG and the experiments. This allowed to create a community of Grafana experts building dashboards and following the upstream progresses of the Grafana project.

Once the new dashboards were approved by the different responsible entities, the MONIT team imported the historical data from the old Oracle databases into the MONIT infrastructure, more in particular into the InfluxDB and HDFS storage. From the moment the new

dashboards were approved and the historical data was made available, a migration date was set. On the day of the migration, the old dashboards were stopped, but the old infrastructure behind was kept running as a preventive measure. Two months after the migration date, the old infrastructure was retired.

3.3 Migration Status

Five dashboards were fully migrated and the associated legacy infrastructure decommissioned. The migration of the remaining two dashboards is scheduled to happen during the second quarter of 2020:

- **Sep 2015**: WLCG Transfers
- **May 2019**: ATLAS DDM Transfers
- **May 2019**: ATLAS DDM Accounting
- **July 2019**: ATLAS Job Accounting
- **July 2019**: CMS Job/Task Monitoring
- **Q2 2020**: WLCG Site Monitoring
- **Q2 2020**: WLCG Site Status Board

3.4 Dashboards Implementation

Following the dashboards introduction done in Sect. 3.1, some details on the dashboards implementation in MONIT are now presented. Each dashboard is explained by describing the data sets used, how the data is ingested, if and how the raw data is enriched, and finally the Grafana organisation providing the dashboard.

3.4.1 WLCG Transfers

- *Sources*: FTS transfers and XRootD transfers raw documents
- *Ingestion*: from AMQ brokers using Flume AMQ source
- *Enrichment*: using topology information from VOFeeds
- *Dashboards*: Grafana WLCG organisation

3.4.2 ATLAS DDM Transfers

- *Sources*: Rucio events and Rucio traces raw documents
- *Ingestion*: from AMQ brokers using Flume AMQ source
- *Enrichment*: using topology information from ATLAS AGIS
- *Dashboards*: Grafana ATLAS organisation

3.4.3 ATLAS DDM Accounting

- *Sources*: Rucio summaries and Storage accounting documents
- *Ingestion*: from AMQ brokers using Flume AMQ source
- *Enrichment*: using topology information from ATLAS AGIS
- *Dashboards*: Grafana ATLAS organisation

3.4.4 ATLAS Job Accounting

- *Sources*: Panda Oracle database records
- *Ingestion*: from Oracle using Flume JDBC source
- *Enrichment*: using topology information from ATLAS AGIS
- *Dashboards*: Grafana ATLAS organisation

3.4.5 CMS Job/Task Monitoring

- *Sources*: raw documents generated from queries to Condor
- *Ingestion*: from AMQ brokers using Flume AMQ source
- *Enrichment*: no topology enrichment required
- *Dashboards*: Grafana CMS organisation

4 Conclusions

The process of retiring old services and supporting users to move to a new set of dashboards is always a complex task. The migration of several WLCG and experiment dashboards from the Dashboards infrastructure into the MONIT infrastructure, more in particular to Grafana based dashboards, was a successful process built on two main factors. First, the fact that MONIT is a stable production infrastructure, available since 2017, and supporting many other production use cases (e.g. data centre monitoring using Collectd). And second, a very close interaction with users, exposing them in advance to the new dashboard prototypes, getting early feedback, and working together on solutions. Following the migration to the new MONIT/Grafana dashboards the old dashboards were stopped and all legacy nodes and services decommissioned.

References

- [1] A. Aimar et al, MONIT Monitoring the CERN Data Centres and the WLCG Infrastructure, EPJ Web of Conferences **214**, 08031 (2019)
- [2] J. Andreeva et al, Experiment Dashboard for Monitoring Computing Activities of the LHC Virtual Organizations, Journal of Grid Computing **8**, 323-339 (2010)
- [3] Collectd, <https://collectd.org>
- [4] Prometheus, <https://prometheus.io>
- [5] Apache Kafka, <http://kafka.apache.org>
- [6] Apache Flume, <https://flume.apache.org>
- [7] Kafka Connect, <https://www.confluent.io>
- [8] Apache Spark, <http://spark.apache.org>
- [9] Apache Hadoop, <http://hadoop.apache.org>
- [10] InfluxDB, <https://www.influxdata.com/time-series-platform/influxdb>
- [11] Elasticsearch, <http://elastic.co>
- [12] Grafana, <http://grafana.org>
- [13] Kibana, <https://www.elastic.co/products/kibana>
- [14] SWAN, <https://swan.cern.ch>
- [15] Rucio, <https://rucio.github.io>