ACCOUNTING AND MONITORING INFRASTRUCTURE FOR DISTRIBUTED COMPUTING IN THE ATLAS EXPERIMENT

A. Alekseev[1,a], D. Barberis[2], T. Beermann[3]
on behalf of the ATLAS Computing and Software activity

[1] *Institute for System Programming of the RAS, Russia*

[2] *Università di Genova and INFN, Via Dodecaneso 33, I-16146 Genova, Italy*

[3] *University of Wuppertal, Germany*

E-mail: [a] Aleksandr.Alekseev@cern.ch

The ATLAS experiment uses various tools to monitor and analyze the metadata of the main distributed computing applications. One of the tools is fully based on the Unified Monitoring Infrastructure provided by the CERN-IT MonIT group. The Unified Monitoring Infrastructure uses modern and efficient open-source solutions such as Kafka, InfluxDB, ElasticSearch, Kibana and Grafana to collect, store and visualize metadata produced by data and workflow management systems. This software stack is adapted for the ATLAS experiment and allows the development of dedicated monitoring and accounting dashboards in Grafana visualization environment. The current state of the monitoring infrastructure and overview of core monitoring and accounting dashboards in the ATLAS are presented in this contribution.

Keywords: Scientific computing, BigData, Hadoop, InfluxDB, ElasticSearch, Grafana, Kibana.

1. Introduction

During the LHC Run 1 and Run 2 (2009-2018), the ATLAS Collaboration [1] used custom frameworks to collect and display the monitoring and accounting data related to the worldwide operation of its distributed computing system [1]. These frameworks and dashboards served well during that time, but they started to show their age in several areas; in particular, the data retrieval was slow because of the large amount of data stored in Oracle databases. Since 2016 CERN-IT MonIT group started developing a new unified infrastructure and environment for monitoring and accounting applications based on a modern and efficient open-source software stack [2]. This stack is adapted for the ATLAS experiment and allows the development of dedicated monitoring and accounting applications in the Grafana [3] and Kibana [4] visualization environments [5].

2. The CERN MonIT Unified Monitoring Infrastructure

The CERN MonIT Unified Monitoring Infrastructure (UMI) can receive data from many sources of information, through appropriate interfaces based on Apache Flume [6]. It feeds the data through a buffer and pipeline in Kafka [7]; this allows the execution of Spark [9] jobs to aggregate and/or enrich the data, and stores each type of data in the appropriate technology: ElasticSearch [9], the time-oriented database InfluxDB [10] for aggregated data, or directly in the Hadoop file system HDFS [11] for web-based analyses and long-term archival. Several data display frameworks can then be used to create different representations and dashboards: Kibana for the data in ElasticSearch, Grafana for the data in InfluxDB or in ElasticSearch, and Swan [12] for interactive web-based analyses and displays of the data in HDFS. Figure 1 gives a graphical representation of the data flow through the UMI.

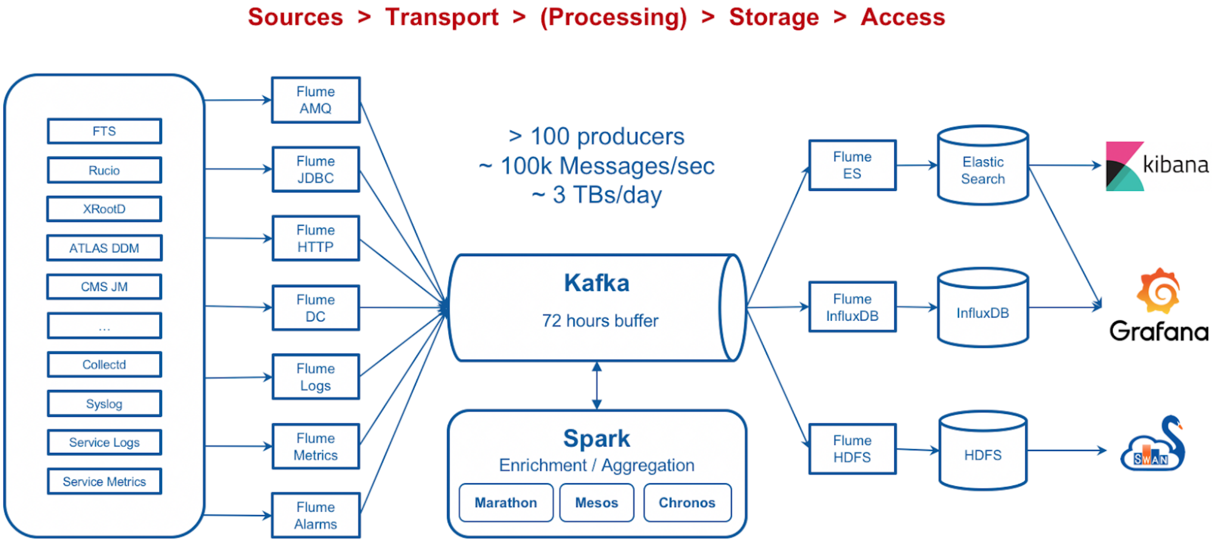


Figure 1. Graphical representation of the data flow through the CERN MonIT Unified Monitoring Infrastructure

The ATLAS Distributed Computing (ADC) activities use the UMI to feed several different dashboards, some of which take input information from different data sources and provide correlated information to the users. The users are normally shifters who monitor the performance of the ATLAS distributed computing components, activity managers who need numbers and figures for resource accounting purposes, and Grid site managers who wish to check the performance of their sites for ATLAS computing activities.

There are three major groups of ATLAS dashboards:

1. Job monitoring and accounting dashboards, showing the number of jobs pending, running or completed at each Grid site and their status, possible errors, CPU and wall-click time etc.
2. Data management monitoring and accounting dashboards, showing file transfer status, transfer data volumes by origin and destination, transfer results, data volumes at each site etc.
3. Site status dashboards, showing the status of the Grid services at each site, the number and efficiency of the running or completed jobs, the ongoing data transfers, the data storage volume for each data category etc.

Next sections describe some of these dashboards in more detail.

3. Job monitoring and accounting dashboards

For the job dashboards, Flume JDBC extracts information from jobs tables in the database of the PanDA [13] workload management system every ten minutes, while pledge information is taken from the WLCG REBUS [14] database and site topology information is taken from the ATLAS Grid information system CRIC [15]. Then Spark jobs process the data executing the following steps: first additional information is computed for each job (ADC activities, error messages, execution time), then the data is enriched with topology information from CRIC, and finally the processed data is written back to Kafka. Kafka keeps aggregated data in five separate indices for different jobs statuses (submitted, pending, running, finalizing and completed jobs) in the ElasticSearch store. Aggregated information about jobs is also kept in long-term HDFS storage. Figure 2 shows the data flow related to the jobs dashboards in UMI.

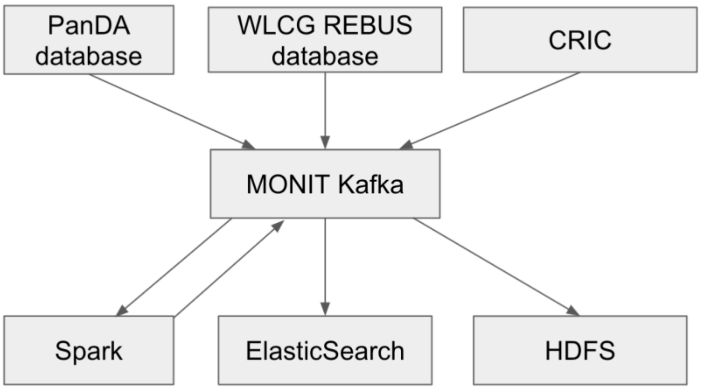
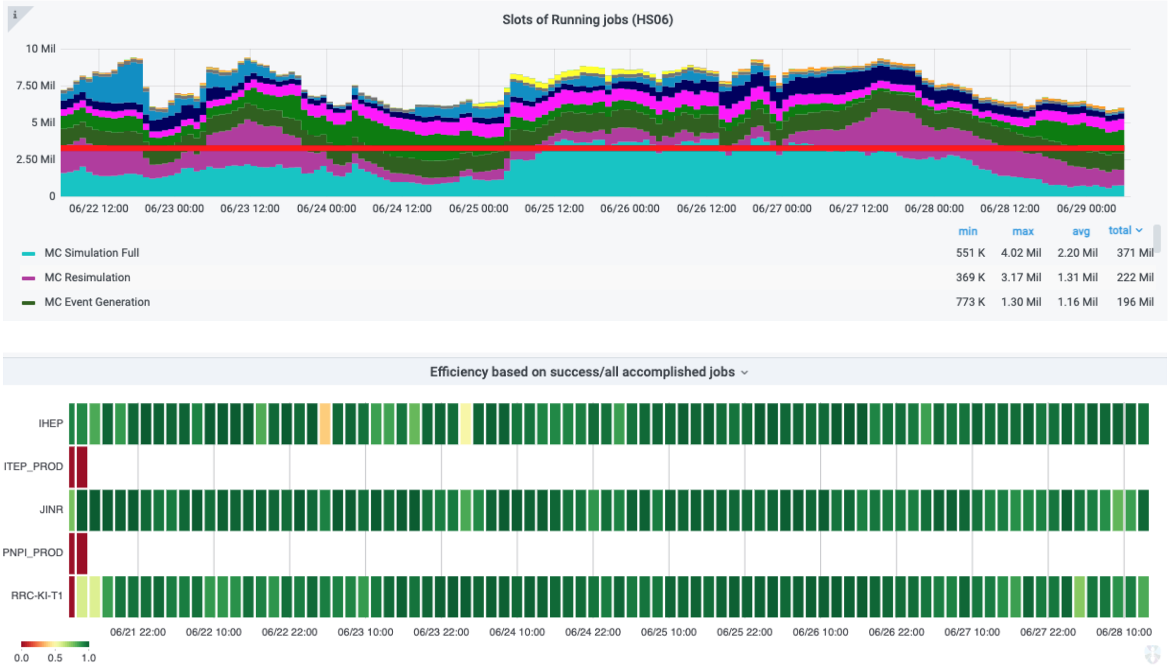


Figure 2. Graphical representation of the data flow related to the jobs dashboards

The Job Accounting dashboard shows aggregated information about Grid and Tier-0 jobs run since the start of data-taking in 2010. Eighty-seven histograms and other charts can be displayed for selected time ranges, with additional selections based on job types, execution sites, input data types, status etc. It is used by shifters, experts and management to spot problems with the operation of the workflow management system and the Grid sites. Different display styles can be selected, and the binning can be set to one hour (the default aggregation) or one, seven or thirty days.

The Job Monitoring dashboard provides extended information about completed jobs for 2 months. It is useful to monitor separate jobs, tasks or requests. The data are processed in the same way as for the job accounting dashboard; Kafka keeps aggregated data in a dedicated index for completed jobs in the ElasticSearch storage. The binning can be set to 10 minutes, 30 minutes, one, six or 12 hours, one, seven, 14 or 30 days.

The HS06 Report dashboard is used by management to generate resource usage reports, where the CPU consumption is shown in HepSpec2006 (HS06) normalized units [16]. The dashboard uses aggregated data from ElasticSeach index for completed jobs (as in the job accounting dashboard), splitting information into separate tables for computing sites, federations and tiers. The data from the tables can be exported to CSV reports using the Grafana API and python scripts. Figure 3 shows some examples of plots from the Job Accounting dashboard: the number of CPU cores used by running jobs in HS06 units, split by ADC activity, and the CPU usage efficiency based on successful and all completed jobs grouped by site.



	min	max	avg	total
MC Simulation Full	551 K	4.02 Mil	2.20 Mil	371 Mil
MC Resimulation	369 K	3.17 Mil	1.31 Mil	222 Mil
MC Event Generation	773 K	1.30 Mil	1.16 Mil	196 Mil

Figure 3. Examples of data representations provided by the job monitoring and accounting dashboards

4. Data monitoring and accounting dashboards

The MonIT infrastructure collects events and traces sent by the data management system Rucio [17] and injects them into a Kafka pipeline; furthermore, the infrastructure also regularly reads topology information from CRIC and adds it to Kafka.

The Spark jobs that process the data consist of the following steps: first the Spark job formats the events and traces so they fit in a similar structure, then they are enriched with topology information from CRIC, and finally everything is aggregated in one-minute bins by source/destination, activity, cloud, federation, country, tier, etc. and the number of failed and successful transfers are counted. These data are written back to Kafka and then stored in ElasticSearch, InfluxDB and/or HDFS. Figure 4 shows the data flow related to the DDM dashboards in UMI.

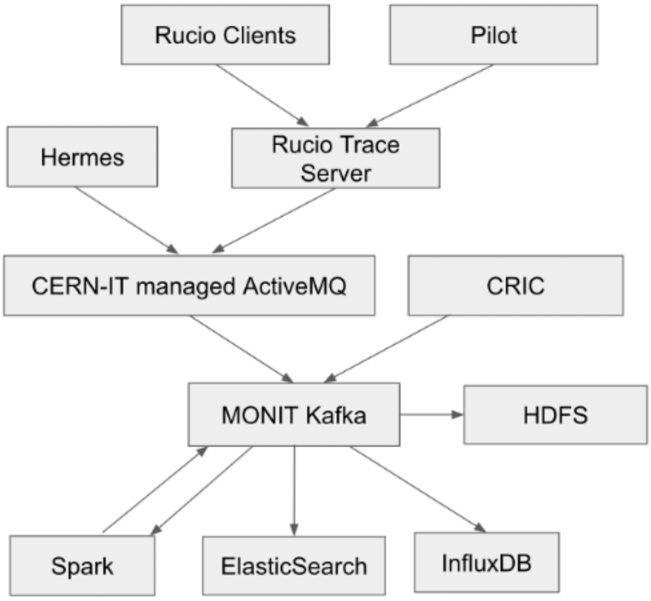


Figure 4. Graphical representation of the data flow related to the Data Management (DDM) dashboards

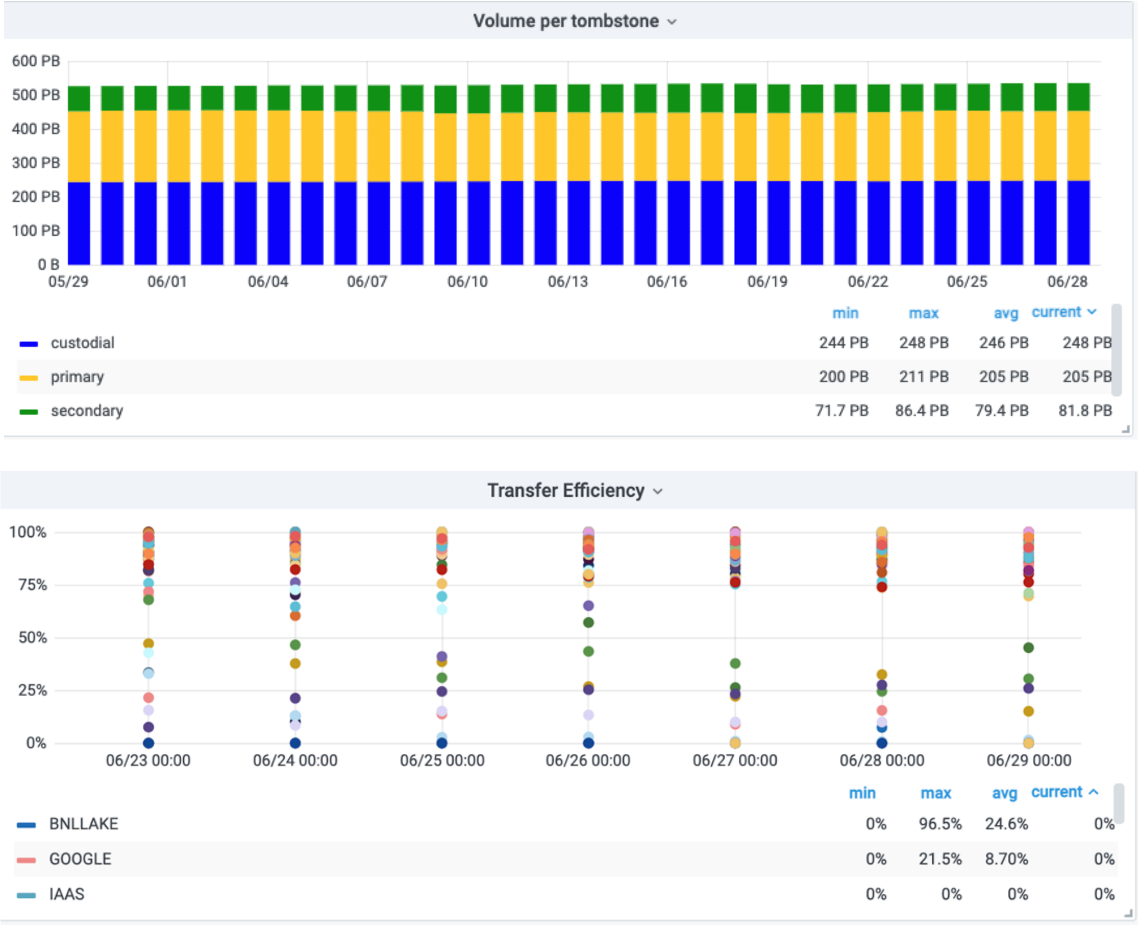
There are two Global Accounting dashboards. The “historical” dashboard is based on datasets. For each dataset, it counts how many file replicas are available split by primary, secondary, tier, disk/tape and dataset metadata, and shows the evolution of replicas over time. It includes 13 plots, 6 filters, 5 options to group the data.

The “snapshot” dashboard includes the same data as the historical dashboard but displays the numbers for the last week.

The Site Accounting dashboard is based on information provided by the Rucio Storage Elements (RSEs) and displays information useful for central operations and for local site administrators. It counts the files and datasets for each RSE and can display the numbers of files or datasets and their volume according to selections set by the users.

The Data Transfer dashboards are mainly used by shifters and experts to monitor the worldwide transfers on the Grid and spot problems. The main dashboard has a granularity of 1 minute for the last 30 days; it uses both InfluxDB and ElasticSearch at the same time, with time-series information in the plots coming from InfluxDB and the transfer details coming from ElasticSearch. It includes 35 plots with 18 filters and 17 options to group data. The binning can be set to 10 minutes, 1 hour, 6 hours, 12 hours or 1 day.

The Historical Transfer dashboard has a minimum granularity of 1 hour for up to 5 years, and displays data in 39 plots with 16 filters and 23 options to group data. Some examples of data representations from data management dashboards are shown in Figure 5: the data volume per data class plot from the Data Accounting dashboard and the data transfer efficiency by site from the Data Transfer dashboard.



Volume per tombstone

	min	max	avg	current
custodial	244 PB	248 PB	246 PB	248 PB
primary	200 PB	211 PB	205 PB	205 PB
secondary	71.7 PB	86.4 PB	79.4 PB	81.8 PB

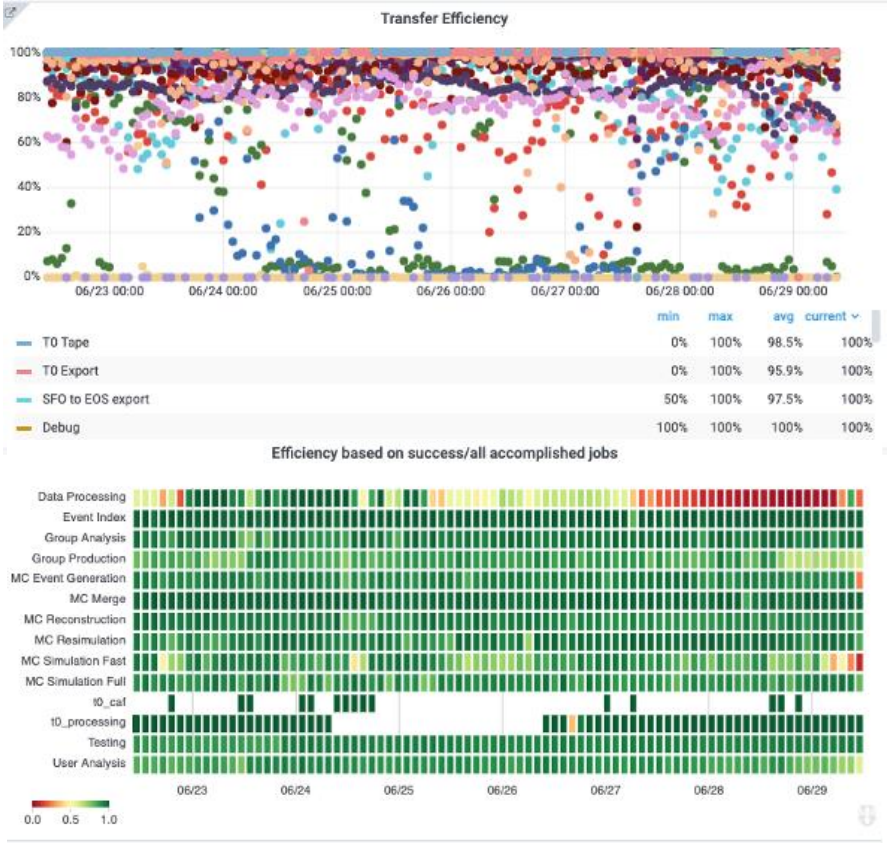
Transfer Efficiency

	min	max	avg	current
BNLLAKE	0%	96.5%	24.6%	0%
GOOGLE	0%	21.5%	8.70%	0%
IAAS	0%	0%	0%	0%

Figure 5. Examples of data representations provided by DDM monitoring and accounting dashboards

5. Site monitoring dashboards

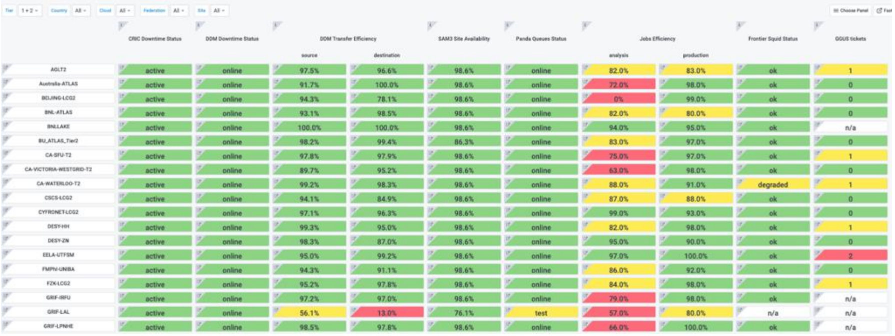
Other dashboards have been developed to provide comprehensive and coherent views of each site status to local system administrators and managers, as well as to ATLAS central operators and shifters. The Site-oriented dashboard combines information from the Job Accounting and Data Transfer dashboards to allow monitoring and analysis of the efficiency of the computing sites; it contains 8 plots (6 from the Job and 2 from the Data dashboards), 5 filters and 55 options to group data. Figure 6 shows some examples of the plots from this dashboard: the transfer efficiency by activity and the efficiency based on successful and all completed jobs grouped by activity.



	min	max	avg	current
T0 Tape	0%	100%	98.5%	100%
T0 Export	0%	100%	95.9%	100%
SFO to EOS export	50%	100%	97.5%	100%
Debug	100%	100%	100%	100%

Figure 6. Some of the plots provided by the Site-oriented dashboard

The Site Status Board groups information provided by ATLAS tools (PanDA and Rucio) as well as by tests run regularly by the WLCG team to check the status and response times of the Grid sites for typical unit tests of simple remote commands [18]. It includes several dashboards displaying different data types, all containing links to the primary source of information. Figure 7 shows the information from the Site Status Board dashboard.



Tier 1 + 2 · Country All · Cloud All · Federation All · Site All

	CRIC Downtime Status	DDM Downtime Status	DDM Transfer Efficiency		SAM3 Site Availability	Panda Queues Status	Jobs Efficiency		Frontier Squid Status	GGUS tickets
			source	destination			analysis	production		
AGLT2	active	online	97.5%	96.6%	98.6%	online	82.0%	83.0%	ok	1
Australia-ATLAS	active	online	91.7%	100.0%	98.6%	online	72.0%	98.0%	ok	0
BEIJING-LCG2	active	online	94.3%	78.1%	98.6%	online	0%	99.0%	ok	0
BNL-ATLAS	active	online	93.1%	98.5%	98.6%	online	82.0%	80.0%	ok	0
BNLLAKE	active	online	100.0%	100.0%	98.6%	online	94.0%	95.0%	ok	n/a
BU_ATLAS_Tier2	active	online	98.2%	99.4%	86.3%	online	83.0%	97.0%	ok	0
CA-SFU-T2	active	online	97.8%	97.9%	98.6%	online	75.0%	97.0%	ok	1
CA-VICTORIA-WESTGRID-T2	active	online	89.7%	95.2%	98.6%	online	63.0%	98.0%	ok	0
CA-WATERLOO-T2	active	online	99.2%	98.3%	98.6%	online	88.0%	91.0%	degraded	1
CSCS-LCG2	active	online	94.1%	84.9%	98.6%	online	87.0%	88.0%	ok	0
CYFRONET-LCG2	active	online	97.1%	96.3%	98.6%	online	99.0%	93.0%	ok	0
DESY-HH	active	online	99.3%	95.0%	98.6%	online	82.0%	98.0%	ok	1
DESY-ZN	active	online	98.3%	87.0%	98.6%	online	95.0%	90.0%	ok	0
EELA-UTFSM	active	online	95.0%	99.2%	98.6%	online	97.0%	100.0%	ok	2
FMPhI-UNIBA	active	online	94.3%	91.1%	98.6%	online	86.0%	92.0%	ok	0
FZK-LCG2	active	online	95.2%	97.8%	98.6%	online	84.0%	98.0%	ok	1
GRIF-IRFU	active	online	97.2%	97.0%	98.6%	online	79.0%	98.0%	ok	n/a
GRIF-LAL	active	online	56.1%	13.0%	76.1%	test	57.0%	80.0%	n/a	n/a
GRIF-LPNHE	active	online	98.5%	97.8%	98.6%	online	66.0%	100.0%	ok	n/a

Figure 7. The entry page of the Site Status Board

6. Conclusions

Since 2016 the CERN-IT MonIT group has been developing a new Unified Monitoring Infrastructure (UMI). The infrastructure uses modern and efficient open-source solutions such as Kafka, InfluxDB, ElasticSearch, Kibana and Grafana to collect, store and visualize metadata produced by data (Rucio) and workflow management (PanDA) systems. This software stack is adapted for the ATLAS experiment and allows the development of dedicated monitoring and accounting dashboards in the Grafana visualization environment. ATLAS currently has over a dozen production dashboards for Jobs, DDM and Sites monitoring; other dashboards relate to the status of central computing servers. All parts of the monitoring based on UMI are constantly being improved and adapted to the rapidly-evolving technologies.

7. Acknowledgement

This work was partially funded by the Russian Science Foundation under contract No.19-71-30008 (the research is conducted in the Plekhanov Russian University of Economics).

References

[1] ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 S08003 doi:10.1088/1748-0221/3/08/S08003

[2] Sargsyan L et al. 2012 ATLAS job monitoring in the Dashboard Framework, J.Phys.Conf.Ser. 396 (2012) 032094 doi:10.1088/1742-6596/396/3/032094

[3] Aimar A et al. 2019 MONIT: Monitoring the CERN Data Centres and the WLCG Infrastructure, EPJ Web Conf. 214 08031 doi:10.1051/epjconf/201921408031

[4] Grafana: https://grafana.com/

[5] Kibana: https://www.elastic.co/kibana

[6] Beermann T et al. 2020 Implementation of ATLAS Distributed Computing monitoring dashboards using InfluxDB and Grafana, EPJ Web Conf. 245 03031 doi:10.1051/epjconf/202024503031

[7] Apache Flume: https://flume.apache.org/

[8] Apache Kafka: https://kafka.apache.org/

[9] Apache Spark: https://spark.apache.org/

[10] ElasticSearch: https://www.elastic.co/elasticsearch

[11] InfluxDB: https://www.influxdata.com/

[12] Hadoop: http://hadoop.apache.org/

[13] CERN SWAN service: https://swan.web.cern.ch/

[14] Barreiro Megino F et al. 2016 PanDA: Exascale Federation of Resources for the ATLAS Experiment at the LHC, EPJ Web Conf. 108 01001 doi:10.1051/epjconf/201610801001

[15] WLCG REBUS: http://wlcg-rebus.cern.ch/

[16] ATLAS CRIC: https://atlas-cric.cern.ch/

[17] HepSpec2006: http://www.spec.org/cpu2006

[18] Barisits M et al. 2019 Rucio: Scientific Data Management, Computing and Software for Big Science 3:11 doi:10.1007/s41781-019-0026-3

[19] WLCG ETF: http://etf.cern.ch/docs/latest/user/overview.html