MONITORING AND ACCOUNTING FOR THE DISTRIBUTED COMPUTING SYSTEM OF THE ATLAS EXPERIMENT

D. Barberis[1,a], A. Aimar[2], A. Alekseev[3,4,5], P.M. Rodrigues De Sousa Andrade[2], T.A. Beermann[6], R.W. Gardner[7], B. Garrido Bear[2], T. Korchuganova[3,4,5], L. Magnoni[2], S. Padolski[8], E. Schanet[9], N. Tsvetkov[2], I. Vukotić[7], T. Wenaus[8]

[1] *Dipartimento di Fisica dell'Università di Genova e INFN Sezione di Genova, Via Dodecaneso 33, I - 16146 Genova, Italy*

[2] *CERN, CH - 1211 Genève 23, Switzerland*

[3] *Institute of System Programming, Russian Academy of Science, Moscow, Russia*

[4] *Universidad Andrés Bello, Santiago, Chile*

[5] *Plekhanov Russian University of Economics, Stremyanny Lane 36, RU – 117997, Moscow, Russia*

[6] *Bergische Universitaet Wuppertal, Gaußstr. 20, DE - 42119 Wuppertal, Germany*

[7] *University of Chicago, Enrico Fermi Institute, 5640 S Ellis Ave, Chicago, IL 60637, USA*

[8] *Brookhaven National Laboratory, Upton, NY, USA*

[9] *Fakultät für Physik, Ludwig-Maximilians-Universität München, Am Coulombwall 1, DE - 85748 Garching bei München, Germany*

E-mail: [a] Dario.Barberis@ge.infn.it

Over the years, ATLAS has developed a large number of monitoring and accounting tools for distributed computing applications. In advance of the increased experiment data rates and monitoring data volumes foreseen for LHC Run 3, starting in 2012, a new infrastructure has been provided by the CERN-IT Monit group, based on InfluxDB as the data store and Grafana as the display environment. ATLAS is adapting and further developing its monitoring tools to use this infrastructure for data and workflow management monitoring and accounting dashboards, expanding the range of previous possibilities with the aim of achieving a single, simpler, environment for all monitoring applications. This contribution describes the tools used, the data flows for monitoring and accounting, the problems encountered and the solutions found.

Keywords: ATLAS, distributed computing, monitoring

Dario Barberis, Alberto Aimar, Aleksandr Alekseev,
Pedro Manuel Rodrigues De Sousa Andrade, Thomas A Beermann,
Robert W Gardner, Borja Garrido Bear, Tatiana Korchuganova, Luca Magnoni,
Siarhei Padolski, Eric Schanet, Nikolay Tsvetkov, Ilija Vukotić, Torre Wenaus

1. Introduction

Every experiment that produces and processes large amounts of data needs to monitor the infrastructure and the applications that deal with these data. Monitoring is essential to be able to spot and fix any system failure in a short time and to identify ways to improve the system performance, within the available hardware resources. The ATLAS experiment [1] at the CERN LHC accelerator collected in 2015–2018 ("Run 2") almost 20 billion physics events plus a large amount of detector calibration data and about three times as many simulated events; events are stored in files that are then grouped into datasets. The processing of all these events takes place using the distributed computing infrastructure comprising the World-wide LHC Computing Grid (WLCG), consisting of over 120 sites distributed in all continents, and a few High-Performance Computers (HPCs) that are available to ATLAS.

The Distributed Data Management system Rucio [2] is used to move, store and catalogue all ATLAS data. The processing operations are accomplished by the Workload Management system PanDA [3], which takes all processing requests, transforms them into "tasks" that act on datasets, splits tasks into jobs and finally submits the jobs to the best computing facility depending on CPU availability and input data location. Both Rucio and PanDA operations depend on the availability of central computing clusters where all components of their systems run.

ATLAS used during LHC Run 1 (2009–2012) and Run 2 (2015–2018) a monitoring and accounting infrastructure for the distributed computing applications developed about 10 years ago by CERN-IT together with ATLAS members. These "old dashboards" started showing aging effects in the last few years, visible primarily as an increasing slowness of data retrieval due to the massive amount of data in Oracle databases. Also, the lack of in-depth knowledge for maintenance as most original developers left long ago, led to a lack of flexibility and impossibility to develop new views and/or data correlations across different data sources. This system worked well enough for general monitoring until the end of Run 2 last year but was evidently in need of a good refurbishing.

Since 2016 the CERN-IT MonIT group started developing a new infrastructure and environment for monitoring and accounting applications based on modern Open Source components, and ATLAS started implementing "new" dashboards using this infrastructure, for data and workload accounting and global monitoring.

In the meantime, the BigPandaMon [4] application was developed for user and task-oriented monitoring of the jobs submitted to the ATLAS Grid/Cloud/HPC resources through PanDA; this is now the workhorse of user-level job monitoring.

In recent years many Analytic tools have appeared on the market. They can be used for more detailed investigations and to correlate data from different sources. The Analytics cluster provided by the University of Chicago allows a more interactive use of monitoring data for detailed investigations and correlations between the various distributed computing systems.

2. ATLAS dashboards in the MonIT infrastructure

2.1 The MonIT infrastructure

The CERN-IT MonIT group provides "Monitoring as a Service" for the CERN data centre and the WLCG collaborations. The services consist in providing the infrastructure to collect, transport, process and store the monitoring data, and the dashboards to display all collected information. The diagram in Figure 1 shows the components used and the data flow through them:

- A number of data collector units receive information from the services to be monitored and feed the data pipeline. The relevant collectors for our applications are the messaging system ActiveMQ [5], Collectd [6], Apache Flume [7] and Logstash [8].
- Apache Kafka [9] is the core of the data pipeline. It decouples the producers of information from the consumers, enables stream processing and is resilient, with a data retention time of 72 hours.

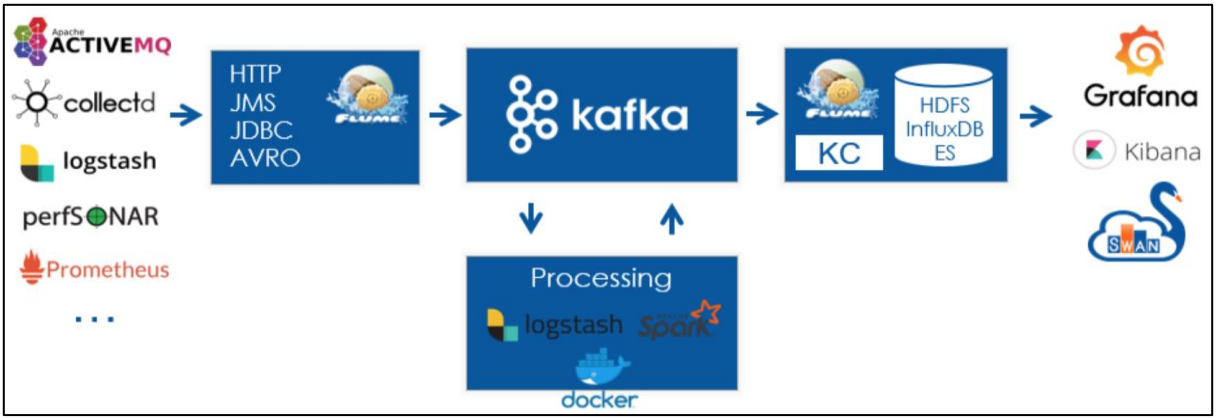

Figure 1. Data flow through the MonIT infrastructure

- Data are stored as time series in the InfluxDB [10] database, which can keep data long-term, with adjustable time bins. All data are also kept in the Hadoop file system HDFS [11] for archival and batch usage. Part of the data can be stored in ElasticSearch [12], which offers more interactive analysis possibilities but has a shorter retention time (one month as default).
- The data can be visualised using dashboards in Grafana [13] or by developing more interactive views in Kibana [14] or using Jupyter notebooks [15] using Swan [16].

There are three groups of dashboards, with different read/write access parameters and frequencies of upgrades: the *Production* dashboards are stable and updated only with tested improvements; the *Development* dashboards are used to test new features or views before implementing them into production; the *Playground* dashboards are used to explore new possibilities or tools.

2.2 The ATLAS dashboards

Several groups of dashboards have been developed to monitor ATLAS Distributed Computing (ADC) applications. For Distributed Data Management, data usage and transfer information is constantly sent by Rucio to the message brokers and then further processed; data storage information is periodically extracted from the Rucio database in Oracle, dumped to HDFS and further processed to be stored in InfluxDB. Many views are available, including historical views going back to the start of Rucio in 2015, current snapshots of data volumes by site or data type, data transfer rates and efficiencies between any two endpoints. Figure 2 shows two examples of DDM dashboards.

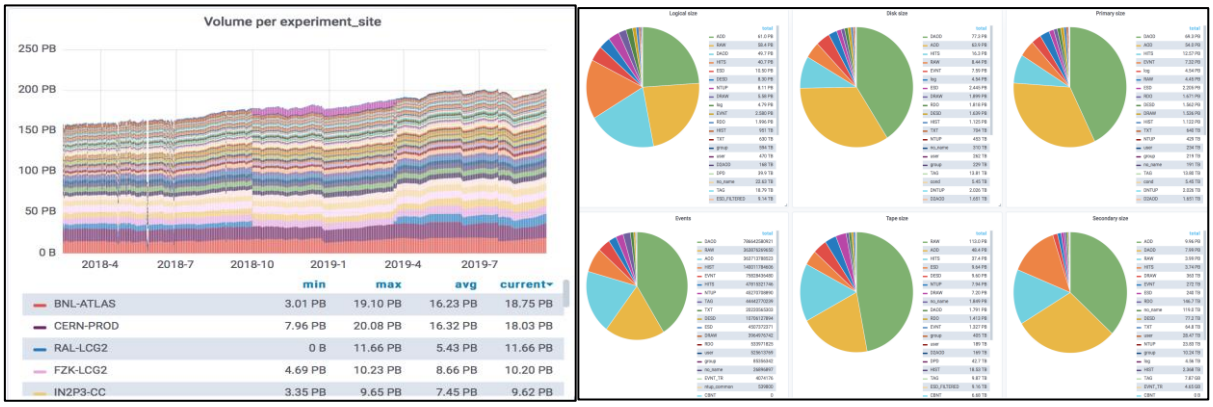

Figure 2. Examples of DDM dashboards: (left) display of the data stored on disk between March 2018 and September 2019, for each storage site; (right) different views of the current data statistics for each data type

The second most important group of dashboards refer to job monitoring and accounting. In this case the information is collected from the PanDA database every ten minutes and stored into 1-hour time bins; 24-hour, 7-day and 30-day bins are calculated automatically. The dashboards display the number of pending, running or finalising jobs, as well as statistics for the completed jobs including

errors, CPU and wall-clock time consumption, and many other job parameters. Information is also imported from other sources, such as the site topology from the ATLAS Grid Information System (AGIS) [17] database, and the pledge information from the WLCG REBUS [18] database; in this way it is possible to group the sites by federation, country or cloud, and display the actual CPU usage against the pledges (see Figure 3). Data stored in the previous generation dashboard, dating back to the start of the LHC in 2009, were also imported in the new system, so it is possible (but not fast) to generate plots for 10 years of data processing operations.

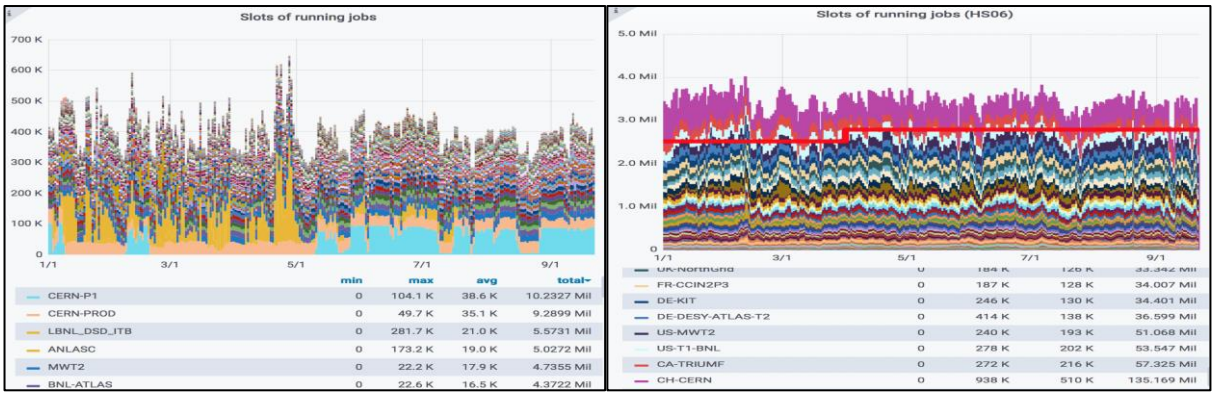

Figure 3. Examples of job accounting dashboards: (left) number of CPU cores used by ATLAS between January and September 2019; (right) used computing power in HepSpec2006 units between January and September 2019, compared to the total pledged resources shown as the overlaid red line

A dashboard for short-term but site-oriented job monitoring was also developed; here the data are aggregated by site, thus reducing the cardinality by a large factor (the number of sites, over 120), and keeping a reduced number of variables. In this way it is possible to display data directly in much smaller time bins and create automatic alarms in case of anomalous conditions in real time. An example of this dashboard is shown in Figure 4.

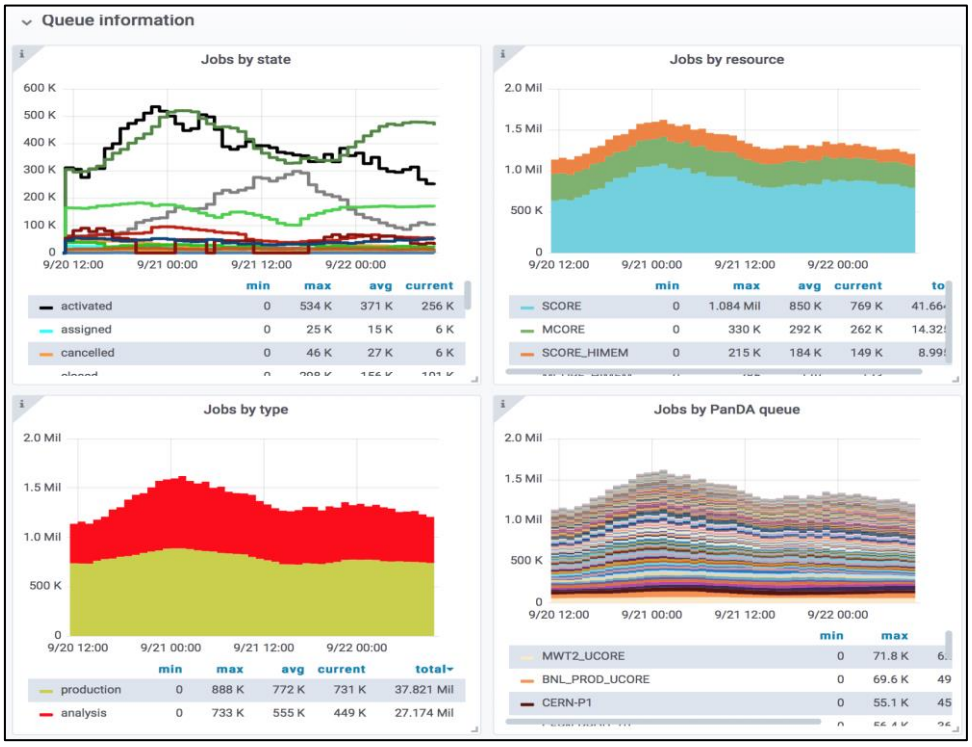

Figure 4. Example of the job monitoring dashboard: (top-left) number of jobs by PanDA status; (top-right) number of jobs by resource type; (bottom-left) number of production vs analysis jobs; (bottom-right) number of jobs by PanDA queue

Several other dashboards have been developed to monitor the status of other central and distributed services and servers. In addition, a summary display of ADC monitoring dashboards was created: the "Live Page". It is a web site with several pages that show some statistics and a few static plots extracted from the various dashboards and refreshed every hour; clicking on each plot leads to the relevant dashboard where more detailed investigations of any problem can be performed (see Figure 5). This is the entry point for shifters and computing managers wishing to have a global view of the current status of the ADC systems.

3. User level job/task monitoring with BigPandaMon

The MonIT dashboards provide a wealth of information but by design they contain only statistical aggregations of jobs and tasks, with their properties, stored as time series. For detailed monitoring of the processing of tasks and job status evolutions, the BigPandaMon application provides real-time access to the PanDA data in the Oracle database.

BigPandaMon is built as an aggregator of information from different sources, of which the PanDA database is the principal, but not the only, one. The retrieved information is cached for 10 minutes, allowing the users to digest it and formulate more detailed queries; if the new query only needs information that is already cached during the main query, the system will respond very fast, otherwise a new query to Oracle will be launched and the response time will depend on the amount of retrieved data – usually a few seconds will suffice. Figure 6 shows the data flow within BigPandaMon.

Figure 5. Examples of the Live Page displays: (left) display of the number of CPU cores used by ATLAS in the last week, grouped by different relevant parameters; (right) display of the data transfer rates and efficiencies in the last week, grouped by different parameters

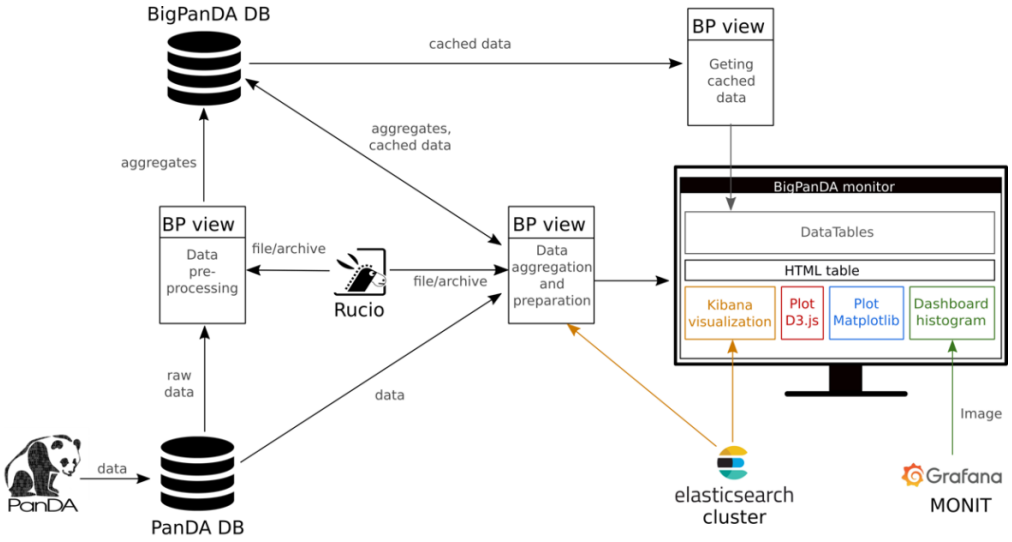

Figure 6. Data flow within BigPandaMon

From the top page, users can select a number of views or search directly for a given task or job, and get all details, including the relations between tasks and jobs, sites where the jobs are running or have run, log files, error conditions and so on. Every displayed piece of information is clickable and leads to the source of this information and additional details. Figure 7 shows examples of the task and jobs tables, as well as the task-level statistics on job execution times, CPU and memory usage, and task completion rates.

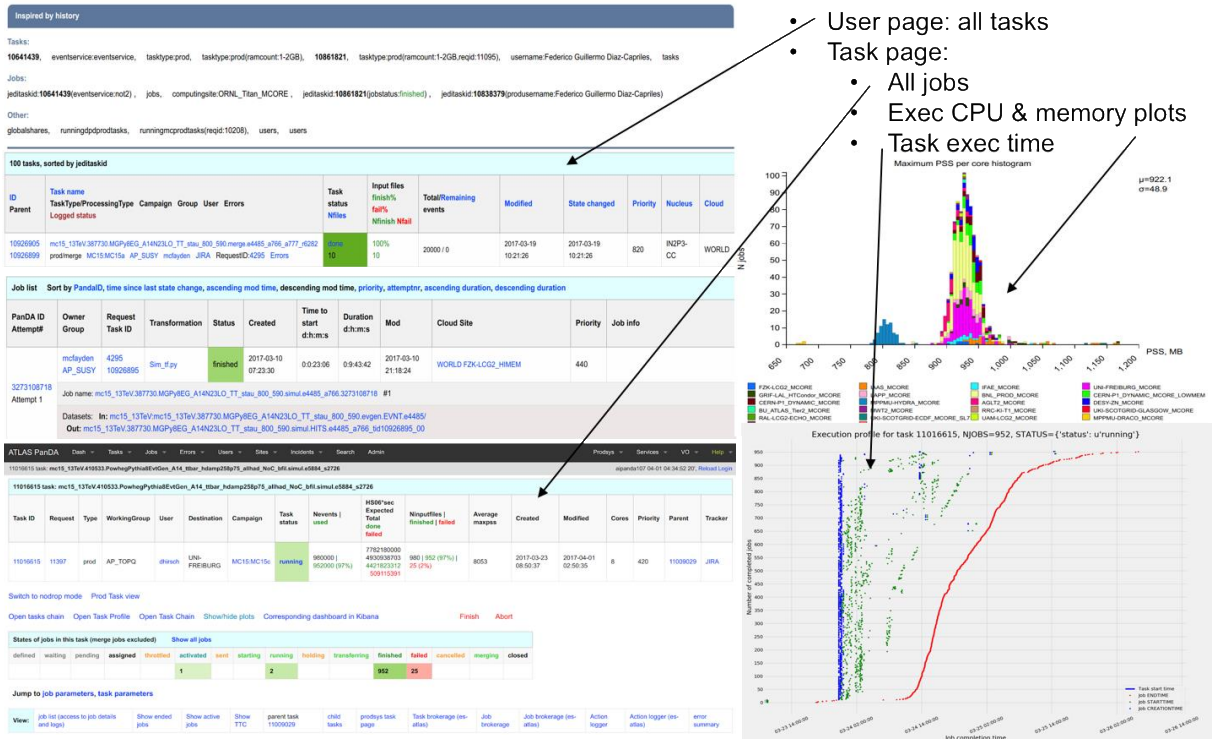

Figure 7. Examples of BigPandaMon displays: (top-left) user page listing all active tasks; (bottom-left) task page listing all jobs in the task and their status; (top-right) memory used by all jobs in a given task vs execution site; (bottom-right) job definition, start and end times for all jobs in a task

4. Analytics cluster at UChicago and its applications

The analytics cluster at the University of Chicago provides an interactive environment to develop additional or alternative dashboards and investigate correlations between several data sources. It is complementary to the monitoring and accounting infrastructure at CERN. In addition to the information imported from the PanDA and Rucio databases at CERN, it collects data from the WLCG File Transfer System (FTS) [19] servers that are distributed in several locations, from the Frontier servers that provide access to the conditions database [20], and from the PerfSonar [21] network testing probes. It has been essential in tracking down wrong and repeated accesses to the conditions database by ATLAS jobs [22] and in identifying malfunctioning network routes that can impact the data transfers and thus the usage of ATLAS resources.

5. Conclusions and Outlook

ATLAS Distributed Computing has a coherent set of monitoring and accounting dashboards and interactive tools. Technologies evolve all the time, so we have to follow them; we try to use Open Source solutions as much as possible, even if at times some home-made parts are inevitable, for example because of the low number of display options available in Grafana.

The future is in the more interactive environments providing the possibility to correlate information from many different sources. ATLAS has already started in this direction and is actively developing new views and new tools to increase the level of automatic error or anomaly detection in view of the start of LHC Run 3 in 2021.

Acknowledgements

This work was performed in the context of the ATLAS Collaboration. It was funded for the part related to workflow management monitoring by the Russian Science Foundation under contract No. 19-71-30008 (research is conducted in Plekhanov Russian University of Economics).

References

[1] ATLAS Collaboration et al., 2008 JINST 3 S08003, doi: https://doi.org/10.1088/1748-0221/3/08/S08003.

[2] Barisits M et al., 2019 Comput Softw Big Sci 3: 11, doi: https://doi.org/10.1007/s41781-019-0026-3.

[3] Maeno T et al., 2017 J. Phys. Conf. Ser. 898 052002, doi: https://iopscience.iop.org/article/10.1088/1742-6596/898/5/052002.

[4] Alekseev A et al., 2018 J. Phys. Conf. Ser. 1085 3, 032043, doi: https://doi.org/10.1088/1742-6596/1085/3/032043.

[5] ActiveMQ: https://activemq.apache.org (accessed 23.11.2019).

[6] Collectd : https://collectd.org (accessed 23.11.2019).

[7] Flume : https://flume.apache.org (accessed 23.11.2019).

[8] Logstash: https://www.elastic.co/products/logstash (accessed 23.11.2019).

[9] Kafka: https://kafka.apache.org (accessed 23.11.2019).

[10] InfluxDB: https://www.influxdata.com (accessed 23.11.2019).

[11] Hadoop and HDFS : https://hadoop.apache.org (accessed 23.11.2019).

[12] ElasticSearch: https://www.elastic.co (accessed 23.11.2019).

[13] Grafana: https://grafana.com (accessed 23.11.2019).

[14] Kibana: https://www.elastic.co/products/kibana (accessed 23.11.2019).

[15] Jupyter notebooks: http://jupyter.org (accessed 23.11.2019).

[16] Swan: https://swan.web.cern.ch (accessed 23.11.2019).

[17] Anisenkov A et al., 2019 EPJ Web Conf. 214 03003, doi: https://doi.org/10.1051/epjconf/201921403003.

[18] WLCG Rebus: https://wlcg-rebus.cern.ch/apps/topology (accessed 23.11.2019).

[19] Ayllon A et al., 2014 J.Phys.Conf.Ser. 513 032081, doi: https://doi.org/10.1088/1742-6596/513/3/032081.

[20] Barberis D et al., 2012 J.Phys.Conf.Ser. 396 052025, doi: https://doi.org/10.1088/1742-6596/396/5/052025.

[21] Perfsonar: https://www.perfsonar.net (accessed 23.11.2019).

[22] Gallas EJ and Ozturk N, 2019 EPJ Web Conf. 214 04017, doi: https://doi.org/10.1051/epjconf/201921404017.