BIGDATA TOOLS FOR THE MONITORING OF THE ATLAS EVENTINDEX

Evgeny Alexandrov [1,a], Andrei Kazymov [1], Fedor Prokoshin [2,b],
on behalf of the ATLAS collaboration

[1] Joint Institute for Nuclear Research, Dubna, Russia.

[2] Centro Científico Tecnológico de Valparaíso-CCTVal, Universidad Técnica Federico Santa María.

E-mail: [a] aleksand@jinr.ru, [b] Fedor.Prokoshin@cern.ch

The ATLAS EventIndex collects event information from data at both CERN and Grid sites. It uses the Hadoop system to store the results, and web services to access them. Its successful operation depends on a number of different components, that have to be monitored permanently to ensure continuous operation of the system. Each component has completely different sets of parameters and states and requires a special approach. A scheduler runs monitoring tasks, which gather information by various methods: querying databases, web sites and storage systems, parsing logs and using CERN host monitoring services. Information is then sent to Grafana dashboards via InfluxDB. This platform provides much better performance and flexibility compared to the previously used Kibana system.

Keywords: Monitoring, Hadoop, Grafana, InfluxDB, EventIndex

1. Introduction

The EventIndex [1] is a complete catalogue of all ATLAS [2] events, keeping the references to all files that contain a given event at any processing stage. It takes event information from various data sources, such as CERN and Grid sites. Its successful operation depends on a number of different components. Each component has a completely different set of parameters and states and requires a special approach. The first version of the EventIndex monitoring [3] based on Kibana [4] was developed in late 2014. This version of monitoring had a big problem with the data retrieval speed. For example, it required about 15 seconds to display data from a two days period, about 90 seconds to display data from 7 days, and tens of minutes to display data from longer periods. A new monitoring version based on Grafana [5] with the InfluxDB[6] datasource was produced.

2. Structure of the EventIndex Monitoring

The common structure of the EventIndex monitoring is subdivided into two parts: producer and viewer (see Figure 1). The producer part is responsible for collecting data and transferring it to the database. It consists of the following parts: the scheduler, the Python script and the database. The scheduler uses a cron utility for periodically running jobs (Python script) at fixed times. The Python script collects data from CERN and Grid sites and insert it into the database. The EventIndex consists of 10 different modules: *Open-Ended Production*, *Consumer*, *EI Import Status*, *Hadoop Cluster*, *TriggerDB*, *Web Interface Status*, *Health of EventIndex computers*, *Nightly Builds*, *Event Picking Tests* and *EI Data Volumes*. Only *Consumer* has no monitoring data available yet because it was completely re-created on the new platform. Each of these modules requires a different approach for data collecting and processing, thus every module has its own Python script and scheduler for running it. The viewer part is responsible for the graphical presentation of data.
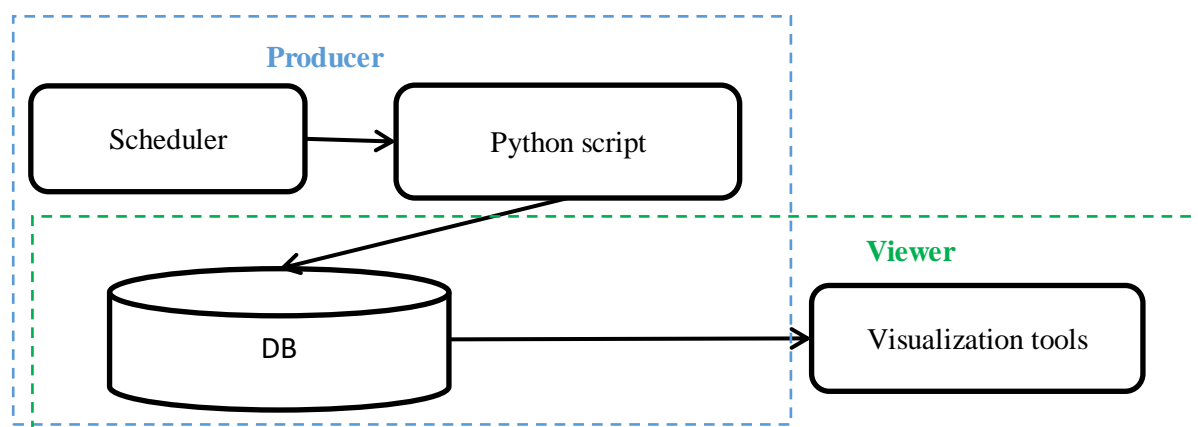

Figure 1. Common structure of the EventIndex Monitoring

The previous version of monitoring used Kibana for data visualization. A Python script prepares data in XML format for a database embedded in Kibana. This version was working correctly but it was perceived as too slow for displaying data covering long time intervals. The EventIndex team decided to change the data visualization service.

Grafana was selected as a new service for visualization data. It supports different databases, such as Graphite, InfluxDB, OpenTSDB and so on. It was decided to use InfluxDB as a frontend database because Grafana + InfluxDB support is provided by the CERN-IT Monitoring group. Although the group policy does not allow to write data directly to InfluxDB, an HTTP endpoint to the middleware that moves data to the database is provided. The input format of middleware is JSON [7]. Data description of this JSON has common and special parts. Common part is the same for all databases which are supported by middleware of CERN-IT Monitoring group. The special part is different for each database. The common part has the following main fields:
- "producer" – dataset name. Only one value allowed,
- "type" – dataset type. Multiple values allowed,

- "timestamp" – time of the event,
- "host" – extra information about the node submitting the data,
- "type_prefix" – categories of metrics. Possible values are raw|agg|enr,
- "_id" – your own ID. By default one random id is assigned,
- "data" – metric fields in format of JSON dictionary.

Only "producer" and "type" are the mandatory fields for all databases. The special part of the JSON structure for InfluxDB has only two fileds: "idb_tags" and "idb_fields". These fields contain a list of the metric names and are used to determine the type of metric. All data with metric name from "idb_tags" has type "tag" in InfluxDB and all data with metric names from "idb_fields" has type "field" in InfluxDB. The detailed description of the format can be found in [8].

The structure of the new version of the EventIndex Monitoring is presented in Figure 2. A scheduler runs Python scripts collecting data for the selected components. These scripts collect data, convert it to the required format and send it to the middleware of the Monit group using the http protocol. This middleware puts data into InfluxDB. Grafana accesses this data and displays it using dashboards. The time of dashboards update is a few seconds even for long periods.
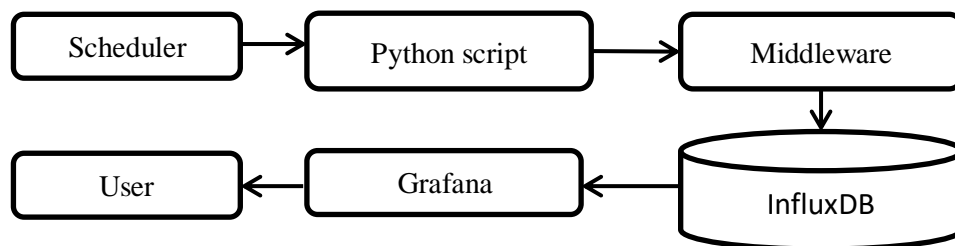

Figure 2. Structure of EventIndex monitoring

The *Health of EventIndex computers* module does not require a special Python script. The producer part of this module is already implemented by the Monit group as a part of the CERN computer system monitoring.

3. View of EventIndex monitoring

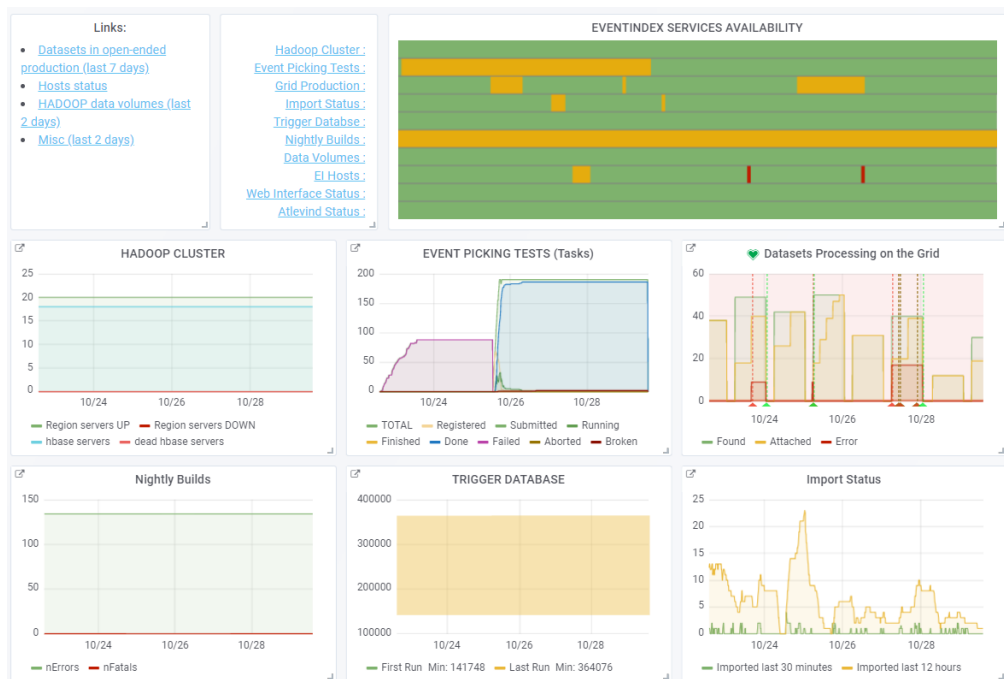Figure 3 presents the main page of the new version.

Figure 3. Main page of EventIndex monitoring

The structure of visualization is similar to the previous version. It has a status dashboard for all modules, dashboards for the most important parameters of each module and links to the module details pages. The current status for each module is calculated using its own algorithm based on the module critical parameters. The status can have one of following values:

- "available" (green color) – module works correctly,

- "degraded" (yellow color) – module has some non-critical problem,

- "unavailable" (red color) – module has critical problem,

- "N/A" (white color) – monitoring data is not available for this module.

The details page of each component usually has more dashboards. Figure 4 shows the details page for the Open-Ended Production component. The main page shows only the number of datasets found, scheduled for processing or skipped, and the number of errors found. The details page shows also information for different projects and data formats.

Figure 4. Detail view of the Open-Ended Production component

4. Conclusion

After half a year of development, renovated and improved monitoring services of the ATLAS EventIndex were successfully implemented, tested and put in production. These services are based on Grafana and InfluxDB. The time required for presenting data is only a few seconds even for long periods. The monitoring system will be evolved and updated following operational experience.

References

[1] Barberis D et al. The ATLAS EventIndex: architecture, design choices, deployment and first operation experience // J. Phys.: Conf. Ser. 664 042003 2015 doi:10.1088/1742-6596/664/4/042003

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

[3] Á Fernández Casaní et al. ATLAS EventIndex general dataflow and monitoring infrastructure // J. Phys. 2017: Conf. Ser.898 062010

[4] Kibana: https://www.elastic.co/products/kibana (accessed 01.11.2018)

[5] Grafana: https://grafana.com/ (accessed 01.11.2018)

[6] InfluxDB: https://www.influxdata.com/ (accessed 01.11.2018)

[7] JSON: https://www.json.org/ (accessed 01.11.2018)

[8] monit-docs: https://monit-docs.web.cern.ch/monit-docs/ingestion/service_metrics.html (accessed 01.11.2018)