1. Trigger and Data Acquisition of the ATLAS experiment at LHC

ATLAS Experiment

A Toroidal LHC Apparatus (ATLAS) is a particle physics experiment at the Large Hadron Collider (LHC) at CERN. The LHC is producing proton-proton head-on collisions with center-of-mass energy equal to 13 TeV (at 45 MHz collision rate). The ATLAS detector comprises more than 140 million electronic channels which deliver raw event data at the rate of order of TBs.

Trigger and Data Acquisition system (TDAQ)

- Manages filtering and transfer of ATLAS experiment data from the detector to large-scale mass-storage, it handles the flow of 1MB events at rate up to 100 kHz.
- A distributed computing system with 40000+ applications running on a cluster of 2300 nodes.
- Non-stop operation 24hrs/day, 7 days/week during LHC Run 1 and Run 2.

2. Motivations for the use of SPLUNK for archiving and visualization

TDAQ applications produce a huge number of operational monitoring data on-line, which is necessary to gather, to archive, to process, to analyze and finally to present to the ATLAS operations crew and to experts to facilitate on-line supervision of the system and also for operational monitoring data on-line, which is presented as a pipeline of Splunk commands. Dashboards are easy to develop.

SPLUNK can receive data from files, TCP sockets or from standard output of user scripts. It is capable to index any type of textual data it receives, and data are immediately available in search queries and in dashboards. To achieve desired search performance, one can provide SPLUNK with some information about the structure of inserted data, e.g. fields which need to be indexed. TDAQ data are well-structured ‘field=value’ events, and for each type of events we developed a ‘properties’ configuration file, that gives SPLUNK an idea which fields need indexing.

A typical event produced by the ERS (Error Reporting Service) application looks like:

```
app=RootController, sev=INFO, text=Transition INITIALIZE  from IGUI is ongoing., context="PACKAGE_NAME: CHIP actions dashboard".
```

Which is indexed by Splunk with the help of this indexing configuration:

```
[ers]
app=RootController, sev=INFO, text=Transition INITIALIZE  from IGUI is ongoing., context="PACKAGE_NAME: CHIP actions dashboard".
```

The process of getting data to and out of Splunk can be presented as a pipeline:

```
(1) define schema - index data
(2) search | process (analyze, transform) | present (charts, reports, dashboards)
```

where the (search | process | present) chain is literally a typical Splunk search string - a pipeline of commands, passing results to each other. A search is usually a selection of interesting fields by value. For processing the search results Splunk offers a great spectrum of functions: statistical, aggregation, transformation etc. Finally, selected and processed data are presented to the user in form of tables, different forms of charts and plots which can be integrated into web pages as ready-to-use dashboards, without exposing any of Splunk. The user is also allowed to work with raw ‘search’ Splunk application, analyzing data with a pipeline of Splunk commands. Dashboards can be developed in simple XML format, also they can be converted to HTML where full power of JavaScript can be used to add more functionality if needed. A very useful feature for operational monitoring is real-time search forms and dashboards, where information is refreshed automatically as soon as fresh data gets indexed.

3. SPLUNK workflow

SPLUNK can receive data from files, TCP sockets or from standard output of user scripts. It is capable to index any type of textual data it receives, and data are immediately available in search queries and in dashboards. To achieve desired search performance, one can provide SPLUNK with some information about the structure of inserted data, e.g. fields which need to be indexed. TDAQ data are well-structured ‘field=value’ events, and for each type of events we developed a ‘properties’ configuration file, that gives SPLUNK an idea which fields need indexing.

A typical event produced by the ERS (Error Reporting Service) application looks like:

```
app=RootController, sev=INFO, text=Transition INITIALIZE  from IGUI is ongoing., context="PACKAGE_NAME: CHIP actions dashboard".
```

Which is indexed by Splunk with the help of this indexing configuration:

```
[ers]
app=RootController, sev=INFO, text=Transition INITIALIZE  from IGUI is ongoing., context="PACKAGE_NAME: CHIP actions dashboard".
```

The process of getting data to and out of Splunk can be presented as a pipeline:

```
(1) define schema - index data
(2) search | process (analyze, transform) | present (charts, reports, dashboards)
```

where the (search | process | present) chain is literally a typical Splunk search string - a pipeline of commands, passing results to each other. A search is usually a selection of interesting fields by value. For processing the search results Splunk offers a great spectrum of functions: statistical, aggregation, transformation etc. Finally, selected and processed data are presented to the user in form of tables, different forms of charts and plots which can be integrated into web pages as ready-to-use dashboards, without exposing any of Splunk. The user is also allowed to work with raw ‘search’ Splunk application, analyzing data with a pipeline of Splunk commands. Dashboards can be developed in simple XML format, also they can be converted to HTML where full power of JavaScript can be used to add more functionality if needed. A very useful feature for operational monitoring is real-time search forms and dashboards, where information is refreshed automatically as soon as fresh data gets indexed.

4. TDAQ applications implemented in SPLUNK

A single Splunk server can host a number of independent applications, thus reducing the deployment and maintenance efforts. The TDAQ Splunk server hosts 4 applications, each handling specific type of data coming from different sources:

- ERS web browser
- Provides access to millions of ERS messages produced by TDAQ application in a web browser. Allows searching the messages based on run number, application name, host, severity, message text etc. Results are available in a table and can be exported. More advanced dashboards with statistics of messages (per application, severity and message type) are available for experts. ERS messages are collected from all running TDAQ applications and stored in files in an intermediate buffer on a shared filesystem before being indexed by Splunk.
- Run efficiency dashboards
- Detailed information about ATLAS data-taking efficiency (fractions of dead-time induced by ATLAS detectors per lumi-block) is stored by Central Trigger in an Oracle database. It is being indexed by Splunk and a number of dashboards are available for experts, allowing to check efficiency evolution by run, to analyze sources of inefficiency by subsystem and by type.
- Access Manager logs browser
- CHIP is a central expert-system like application, aiming to automate different operational and recovery procedures. This Splunk application includes few dashboards which present the history and distribution of actions performed by CHIP.

5. Performance

A single Splunk server running on a virtual machine node (4 cores, 7GB RAM) can handle the events rates from TDAQ data sources as presented on the plot below (maximum and average messages per second, for all data indexed in course of 2017). In total more then 25M of messages were indexed, taking 16 GB of disk space.

Conclusions and Outlook

A simple and powerful solution based on the Splunk framework is developed for archiving and presenting TDAQ operational monitoring data of different origins. Data are archived and indexed by a single Splunk instance and available for ATLAS operators and experts in form of web dashboards and applications. New types of data are easy to add and new dashboards are easy to develop.