EXPERIENCE WITH SPLUNK FOR ARCHIVING AND VISUALISATION OF OPERATIONAL DATA IN ATLAS TDAQ SYSTEM

A. Kazarov[1,2], G. Avolio[1], A.Chitan[3], M.Mineev[4]
[1] CERN, Geneva, Switzerland
[2] NRC Kurchatovsky Institute, PNPI, St. Petersburg, Russian Federation
[3] National Institute for Physics and Nuclear Engineering, Bukharest, Romania
[4] JINR, Dubna, Russian Federation

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 13TeV at 40 MHz collision rate. The ATLAS detector comprises more than 140 million electronic channels which deliver raw event data at the rate of order of TB/s.

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 100kHz
- 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
- Includes applications and frameworks for Configuration, Control and Monitoring the overall ATLAS data taking activity

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 postmortem analysis. This includes, in particular, a large amount of operational messages (at the order of O(10^4) messages per second), and other types of data like statistics of the busy fraction induced by ATLAS detectors per luminosity-block intervals.

We have selected SPLUNK, a commercial solution by Splunk Inc, as a **all-in-one** solution for storing different types of operational data in an indexed database, and a web-based framework for searching, analyzing and presenting the data and for **rapid development** of user-oriented and task-oriented **dashboards** accessible in a web browser.

SPLUNK is developed as a system for archiving and processing of applications logs produced by processes running on large-scale clusters, which nicely can be extended to any type of textual operational data present in the TDAQ system.

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 abou thet 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:

```
t=1386776531, rn=224190, part=ATLAS, uname=crrc, msgID=rc::OngoingTransition, host=pc-tdq-onl-77,
app=RootController, sev=INFO, text="Transition INITIALIZE  from IGUI is ongoing.", context="PACKAGE_NAME:
RunController. FILE_NAME: ../src/lib/RootController.cc. FUNCTION_NAME: virtual void
daq::rc::RootController::receive(const char*, const char*, const char*). LINE_NUMBER: 316. DATE_TIME:
1386776531.", params="trans: INITIALIZE  from IGUI. ", quals="RunController ", chained="0", gh=1755720156
```

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

```
[ers]
TIME_PREFIX = ^t=
MAX_TIMESTAMP_LOOKAHEAD = 20
TRANSFORMS-host = transform-host-ers
TRANSFORMS-source = transform-source-ers
SHOULD_LINEMERGE = true
BREAK_ONLY_BEFORE = ^t=\d{10},\srn=\d+,\s
SEGMENTATION = ers
EXTRACT-fields = ,\stext="(?<text>.*)",\scontext="(?<context>.*)",\sparams="(?<params>.*)",\squals="(?
<quals>.*)",\schained=(?<chained>.+),\sgh=\d+$
```

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 are 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

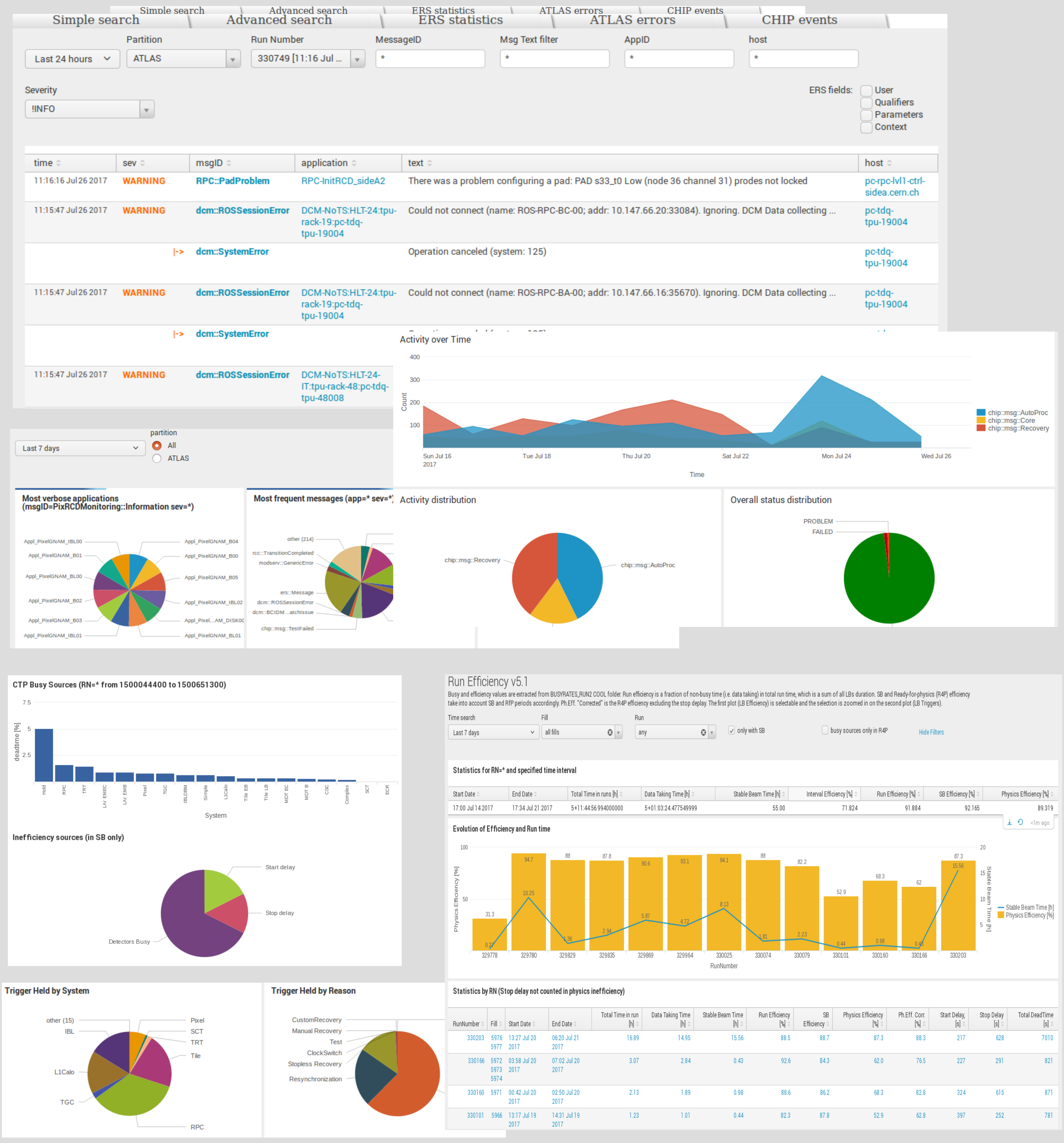
Detailed information about ATLAS data-taking efficiency (fractions of dead-time induced by ATLAS detectors per lumi-block) is stored by Central Trigger in an Oracle database. It is being indexed by Splunk and a number of dashboards are available for experts, allowing to check efficiency evolution by run, to analyze sources of inefficiency by subsystem and by type.

CHIP actions dashboard

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.

Access Manager logs browser

This is a classical Splunk application, where log files from distributed applications are collected and indexed on a server, and dashboards are provided with filtered information relevant for different use cases. This application is being developed.

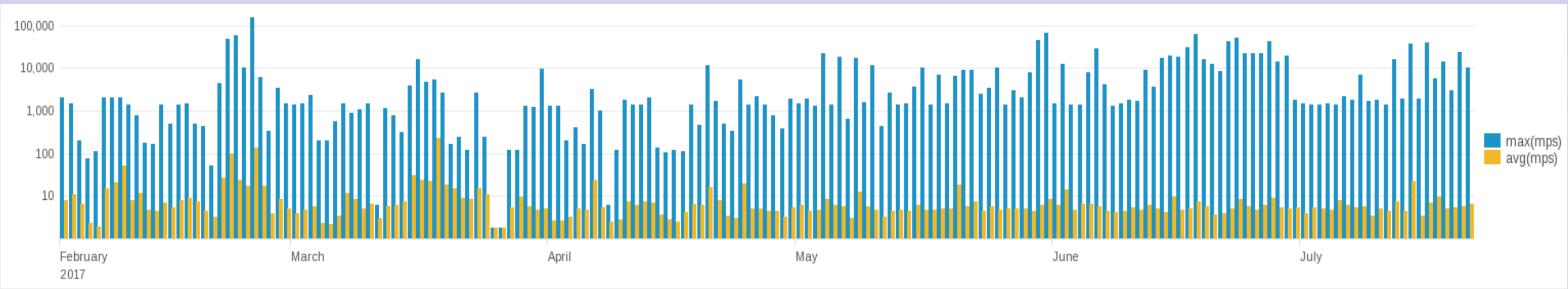


Partition: ATLAS | Run Number: 330749 [11:16 Jul ... | MessageID: * | Msg Text filter: * | AppID: * | host: *
Last 24 hours
Severity: !INFO
ERS fields: □ User □ Qualifiers □ Parameters □ Context

time	sev	msgID	application	text	host
11:16:16 Jul 26 2017	WARNING	RPC::PadProblem	RPC-InitRCD_sideA2	There was a problem configuring a pad: PAD s33_t0 Low (node 36 channel 31) prodes not locked	pc-rpc-lvl1-ctrl-sidea.cern.ch
11:15:47 Jul 26 2017	WARNING	dcm::ROSSessionError	DCM-NoTS:HLT-24:tpu-rack-19:pc-tdq-tpu-19004	Could not connect (name: ROS-RPC-BC-00; addr: 10.147.66.20:33084). Ignoring. DCM Data collecting ...	pc-tdq-tpu-19004
	|->	dcm::SystemError		Operation canceled (system: 125)	pc-tdq-tpu-19004
11:15:47 Jul 26 2017	WARNING	dcm::ROSSessionError	DCM-NoTS:HLT-24:tpu-rack-19:pc-tdq-tpu-19004	Could not connect (name: ROS-RPC-BA-00; addr: 10.147.66.16:35670). Ignoring. DCM Data collecting ...	pc-tdq-tpu-19004
	|->	dcm::SystemError			
11:15:47 Jul 26 2017	WARNING	dcm::ROSSessionError	DCM-NoTS:HLT-24-IT:tpu-rack-48:pc-tdq-tpu-48008		

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 26M 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.