Control and Configuration Software for the ATLAS DAQ system in LHC Run 2

I. Alexandrov1 G. Avolio2 M. Caprini3 A. Chitan4 A. Corso-Radu5 A. Kazanov6 A. Kazymov6 G. Lehmann-Miotto7 M. Mineev7 A. Santos1 I. Soloviev1 G. Unel1 V. Visalie8

1University of California, Irvine, USA; 2INFN “Kurchatov Institute”; 3FMP, St. Petersburg, Russian Federation; 4CERN, Geneva, Switzerland; 5JINR, Dubna, Russian Federation; 6National Institute for Physics and Nuclear Engineering, Bucharest, Romania

Requirements extension

In a large, heterogeneous system such as the ATLAS TDAQ, it is essential to be able to verify the correctness of the firmware and software components. Therefore, a TDAQ functional testing framework existed and was used already throughout Run 1. Additional requirements were identified with the experience gained during data taking:

• "Experts shall be able to define the order in which tests should be executed for a component; the sequence may dynamically change based on the result of completed tests."
• "Experts shall be able to define the order in which inter-related components should be tested; the test sequence may change based on the result of completed tests." • "Experts shall be able to define what should be done upon failure of a test or a component to further diagnose the issue or recover."

Functional testing is used by:

• The Run Control (RC) system that periodically verifies the functioning of the components it is in charge of;
• The Central HLT and Information Processor (CHIP) that executes tests to diagnose problems;
• The Operator or a DAQ expert who manually executes tests via a dedicated graphical user interface, e.g. in case when a detailed assessment of the system healthiness is needed after a power cut.

Software refactoring

The Run Control (RC) system steers the data acquisition by starting and stopping processes and by carrying all data-taking elements through well-defined states in a coherent way. Given the size and complexity of the TDAQ system (2000+ PCs, 10000+ applications, 9000+ network ports, ...), errors and failures are bound to happen and must be dealt with in ATLAS this task is carried out by an expert system. The RC and expert system components were tightly coupled in the first implementation of the software, with the consequence that their separation of duties became increasingly harder with the additions of features during operations in Run 1. Therefore these two components were completely redesigned, using a distributed run control tree and a separate central expert system application, the Central HLT and Information Processor (CHIP).

Applications in the ATLAS TDAQ system are organized in a tree-like hierarchical structure (the run control tree), where each application is managed by a parent Controller. The root node of the tree is the Root Controller. Controller applications are responsible to keep the system in a coherent state by starting and stopping their child applications and by sending the proper commands needed to make the system suitable for data-taking. Controller applications interact with CHIP by informing it about any changes. CHIP acts as an intelligent system having a global view on the TDAQ. It is capable of recovering from error conditions and guiding the TDAQ system through automated procedures in order to take data efficiently.

The RC has been re-implemented with modern and efficient C++ technologies such as C++11, Boost and Intel Threading Building Blocks. CHIP has been designed and implemented based on a third party open source Java-based Complex Event Processing (CEP) engine, Esper.

From Run 1 to Run 2

The ATLAS experiment at the Large Hadron Collider (LHC) operated very successfully in the years 2008 to 2013, identified as Run 1. It achieved an overall data taking efficiency of 94%, largely constrained by the irreducible dead time and a system, and how they were performing in course of Run 2. Despite these being specific to the system, many solutions can be considered and adapted to different distributed DAQ systems.

Code modernization

Message Transport System (MTS)

MTS is a part of the requirements that led to a complete redesign and new implementation to reach its actual role (fast and reliable transport layer for TDAQ Error Reporting System messages). The redesigned system is reliable, scalable and its performance has been improved. The plot below shows the ratio of messages reported in MTS during ATLAS operations in Sept-Oct 2018. In these conditions MTS handled a maximum rate of up to 100 kbit of delivered messages.

Information System Archiver, P-BEAST (a Persistent Backend for the ATLAS TDAQ)

P-BEAST is a new component designed and implemented to archive operational monitoring information for analysis by experts. It provides CORBA and REST interfaces for data access. Its implementation is based on Google protobuf (data persistence), CORBA (internal protocol and user programming interface) and libmicrohttpd (Web server). The push on the right demonstrates how much data was archived per day by P-BEAST during operations in Sept-Oct 2018.

Resource Manager

After an initial review and simplification of the requirements, the component underwent partial changes with the introduction of Boost multi-index containers. As a result the code base has been reduced by 40% against the previous implementation thus leading to a more maintainable system. The plot above shows that the resource manager introduces a negligible overhead to the initialization of the RC tree.

Software applications

The ATLAS DAQ system in LHC Run 2

Web applications

P-BEAST Dashboard

This web application offers an interface to visualize any operating monitoring data published by the TDAQ system through configurable and customizable dashboards. The data is provided by P-BEAST and the application is based on the Grafana project, adapted to support a custom data source within the AngularJS framework.

ELSA

The ATLAS electronic logbook (ELSA) is a web-application used to record and share messages about ATLAS data taking activities by system operators, experts and automated services. The information is stored in an INDELE database. The application of an MVC driven architecture has allowed one to focus code development on specific features of the project, while profiting from the reliability of established third-party technologies such as the Spring framework. The tool also provides an HTTP based REST API, such that other programs can access its features.

Conclusions and Outlook

The Control and Configuration software has contributed to the physics results obtained by the ATLAS experiment during Run 1 by ensuring smooth and efficient data taking. It was completely revised during 2013-2014 in order to accommodate additional requirements, improve maintainability and profit from advances in IT technologies; all this was done applying minimal changes to APIs, such that the large amount of client code would not need significant adaptations. The Control and Configuration software has proved to be stable and well performing LHC Run 2 (2015 - 2018) and is prepared to face the new challenges that will arise in Run 3 operations, after further modernization in different components foreseen during Long Shutdown 2. This experience has also demonstrated that the overall modular architecture of the control and configuration system is flexible and supports partial upgrades, as well as step-wise modernization of its components; this is fundamental for a system that is foreseen to run for the next 20 to 30 years and that will undergo several more upgrade iterations.