Web Run Control for ATLAS Trigger and Data Acquisition

A web-based solution for monitoring and controlling the data taking sessions

Andrei Kazarov - University of Johannesburg, SA
Aimilianos Koulouris - CERN

Introduction

The ATLAS experiment at the Large Hadron Collider (LHC) operated very successfully in the years 2008 to 2024, in periods known as Run 1, 2 and 3. ATLAS Control and Configuration (CC) software [1] is the core part of the ATLAS Trigger and DAQ (TDAQ) system, it comprises all the software required to configure and control the ATLAS data taking. It provides essentially the components that hold the various ATLAS sub-systems together. During recent years, more and more applications in the CC software were made available as web applications, thus making them easily available to experts for remote operations. The importance of web-based solutions raised substantially in recent years when teleworking became a standard working environment, even in a physics experiment. The important missing part, however, was the main data taking control and monitoring application known as “IGUI” (Integrated Graphical User Interface, [2]), the front-end GUI tool used by operators in the ATLAS Control Room to steer the data taking sessions.
Here we present a web application called “WebRC” (Web Run Control), which provides functionality of the data taking monitoring and control from a web browser: presenting expandable tree of all DAQ applications, reflecting the changes in their states, browsing their log files, subscribing to applications messages, monitoring different DAQ parameters, Trigger rates, subdetectors busy information. In short, WebRC allows experts to promptly assess the state of the data taking and to investigate possible issues.

Requirements

We identified the following factors affecting the choice of the technology for the WebRC:
- Its back-end part needs to be easily integrated with main TDAQ services (Run Control, Information Service and Error Reporting Service - ERS) which are all accessible via CORBA middle-ware
- The front-end part should offer rich set of widgets, allowing to implement GUI features similar to widgets available in IGUI
- It shall be well scalable and conservative in resource usage, supporting many user sessions and connecting to multiple TDAQ sessions in parallel
- Support of dynamic and interactive web features like Ajax or Web Sockets, allowing to visually reflect fast changes in data-taking conditions
- The development stack should include minimal number of technologies and should not depend on other frameworks and back-end libraries in order to facilitate maintenance and allow fast learning curve for newcomers

The technology

After a survey and a prototype development, a solid candidate was identified, the Apache Wicket: an open source, component oriented, pure-Java web application framework [3]. Developed since 2004, it remains one of the mainstream Java server-side frameworks, allowing to develop a powerful Java server application which integrates naturally into TDAQ software. A great advantage of Wicket is that it allows to develop Java-only back-end application, not involving any Java-script front-end, which simplifies a lot development and long-term maintenance.

Implementation

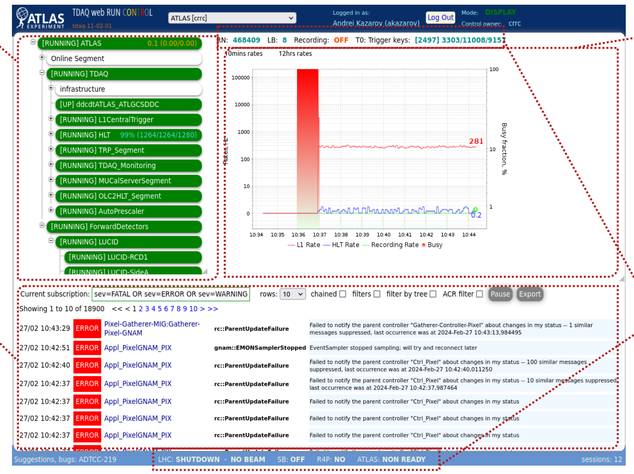
The server side of WebRC is a single multi-threaded Java application which provides HTTP access to render Wicket components, modeling different elements of running DAQ data-taking session, like a hierarchical tree of DAQ applications and streams of application message. The front-end part is a simple HTML page (including CSS for styling the representation), containing stubs for the visualization of the Wicket Java components, following a typical Model-View design pattern. Visualization is managed internally by Wicket, leaving to the programmer only the back-end part of the business logic. The single back-end application handles multiple user sessions and can connect to different independent data-taking sessions (known as “partitions”) The server application uses standard TDAQ software libraries to access all needed information, similarly to it’s standalone counterpart application, the IGUI. The single web page integrates the following main elements: view on all running applications organized in a tree (known as Run Control tree), a panel providing few on all application messages passed via ERS system, and a contextual monitoring graphics dashboard. The application can run in two modes: DISPLAY and CONTROL. The former provides monitoring-only capabilities, and this is the default mode for ATLAS operations. The details of the functionality provided by the WebRC single-web page application in DISPLAY are presented in the figure below.

Run Control tree

Represents all running applications in the selected partition (O(10⁴)), dynamically reflecting their Run Control state, error state and busy status, also displaying some contextual monitoring information for certain elements like occupancy of the trigger farm, crucial parameters of the Level-1 trigger etc.

ERS panel

Allows to subscribe to application messages and to apply advanced filtering criteria for displaying only relevant messages in the table. This is very crucial aspect of the functionality, allowing experts to focus only on relevant messages.



Dashboard panel

A panel which displays an external monitoring dashboard, associated to the selected element in the tree. Each user may customize the URL and has its own monitoring profile, stored in the browser’s cookie. The default dashboard shown here plots the Trigger rates and Busy fraction – the most crucial DAQ parameters to monitor.

Misc

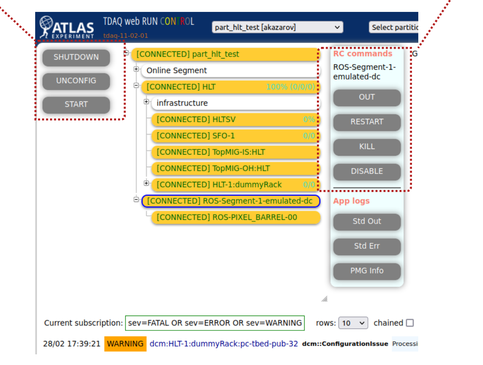
Miscellaneous information from ATLAS and LHC monitoring – the idea is to provide to the experts the most relevant information from a single web page.

The Control mode

A more advanced deployment of WebRC is done in the TDAQ test-bed cluster at CERN: there it runs in the Control mode, allowing full control over DAQ sessions (partitions) started on the test-bed. This includes sending global data taking session control commands (e.g. “START”/”UNCONFIG” as in the screenshot on the right), making configuration changes like disabling or enabling certain parts of the configuration, and also sending individual signals to any application in the tree, like “Restart” or “Kill”. WebRC switches to the Control mode when the user name matches the owner of the partition. For that, any user WebRC session needs to be authenticated with help of CERN SSO portal. This is achieved by hiding the front-end WebRC page (URL) behind a CERN single-sign-on (SSO) proxy application deployed in CERN IT Kubernetes cluster [4].

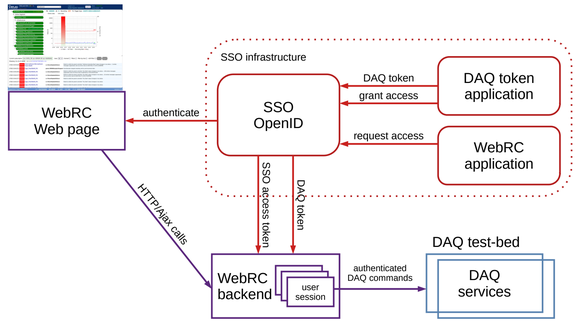
Partition control

Application control



Authentication and token exchange

WebRC relies on the CERN SSO infrastructure for authentication. This provides not only the basic user authentication, but allows the user applications registered in SSO to get an SSO Access Token and then to exchange it for application-specific tokens, leveraging the technologies known as JWT and Open ID Connect (OIDC). Every authenticated user session in WebRC makes a requests to the SSO portal and gets its own DAQ user token [5], provided by another application deployed and registered in the OIDC infrastructure. This token allows each session to get authenticated when accessing the DAQ test-bed services on behalf of the user and to request any action like if the user was issuing DAQ commands from a command line. In particular this makes possible to provide full control through the lifetime of a user sessions in test-bed, from bootstrapping the partition infrastructure till it’s full shutdown.



Conclusion

WebRC is a single-page web application used for monitoring and controlling TDAQ data taking sessions in ATLAS experiment and in TDAQ test-bed. A single powerful Java-only application allows to serve many dozens of user sessions and to facilitate the remote monitoring of the ATLAS data taking sessions and handling the problems by experts. A safe and powerful authentication OIDC-based mechanism allows full access to TDAQ resources in the test-bed for users directly from the web browser.

References

[1] G. Lehmann Miotto et al., Nucl. Instrum. Meth. A623, 549 (2010)
[2] G Avolio et al 2011 J. Phys.: Conf. Ser. 331 022032
[3] Apache wicket, https://wicket.apache.org
[4] https://paas.docs.cern.ch/4._CERN_Authentication/1-use-cern-sso/
[5] https://gitlab.cern.ch/atlas-tdaq-software/daq_tokens

22nd International Workshop on Advanced Computing and Analysis Techniques in Physics Research ACAT 2024, 11-17 March 2024, Stony Brook, USA