SUMMER STUDENT REPORT

ATLAS TRIGGER SOFTWARE VALIDATION TOOL WEBSITE

LAZZARINI Etienne
Student in Computer Science at EISTI
EP-ADT-TR

Supervisors: Rafal Bielski, Daniele Zanzi

Summer 2019

Contents

1 Introduction

The LHC has gone through and will go through a lot of upgrades to keep up with the new technologies available and provide enhanced results.

We are currently in the Long Shutdown 2 (LS2) to prepare for the Run 3. But the major ATLAS improvements are planned for the Run 4 in 2026, when the LHC will be upgraded into the High-Luminosity LHC (HL-LHC).

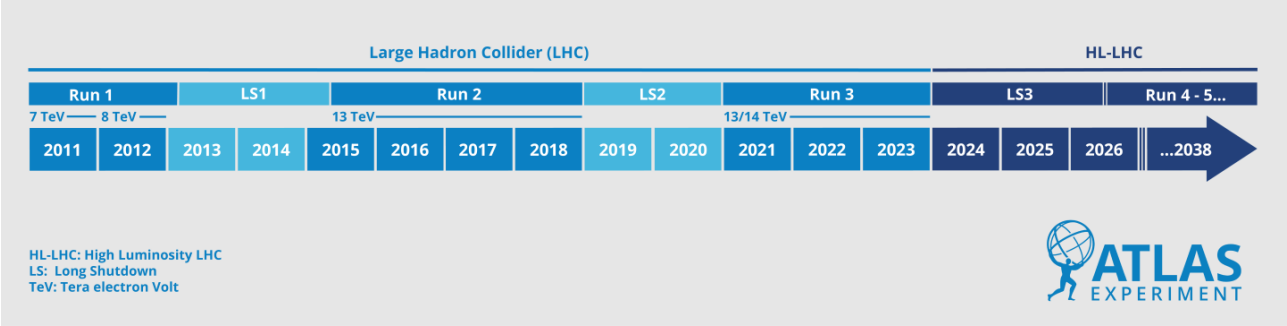

Figure 1: LHC and HL-LHC timeline for ATLAS

In order to cope with the increased amount of data produced with the HL-LHC, hardware and software upgrades are presently being performed on the detector and on the Trigger System. The Trigger Software is being written from scratch to improve its performance and implement new technologies. As it is the software that will select the valuable collisions, its role is central in the ATLAS experiment. It needs to be strict to not store useless collisions, but not too strict to avoid discarding valuable ones. Such a software is very complex and requires cooperation of a lot of developers. ATLAS software, including Trigger, make use of a continuous integration (CI) strategy, meaning that each modification in the code will be verified and ran by automated tests in order to detect problems early. This continuous integration is coupled with ART (ATLAS Release Tester) tests which assess the latest code version every day. ART runs on the code including all merged changes from the day.

I was part of the team that supervises and coordinates the ART continuous integration for Trigger.

1.1 Continuous integration

The ART continuous integration is strongly linked to the use of multiple software:

- Git and GitLab, a distributed version control system for managing code
- JIRA, an issue tracking software for bug management and project development
- Jenkins, an automation server for building, deploying and automating project
- AFS and EOS, the CERN's file systems that we can access to by logging into lxplus, the interactive logon service to Linux for all CERN users.

Thanks to Git, we can create different branches for the same project. Each branch contains a different project version that is developed at its own pace, and is independent of the others. Each night these branches, with their own modifications of the day and all the tests in them are built, compiled and executed with Jenkins on data previously collected by ATLAS. Each ART test outputs some log files, finally stored on EOS, that describe how well it did.

These builds created every night are called Nightly and are structured in Packages which are composed of tests. There are two different types of test: Build and Grid. The first ones are executed directly in Jenkins whereas the latter are sent to the Grid, the CERN computing network, as they need more resources to run.

The role of the Validation Team is to monitor all these test results in order to detect any issues, called Bug, at their earliest stage. Once a Bug is spotted, a JIRA ticket is created to track and solve it. JIRA tickets hold a lot of useful information: we know the person to who they are assigned, we can discuss about it, we know its status: open, closed, reopened, and we know the severity of the Bug.

This important task is handled by the Shifters, people who are shifting each week. They submit a JIRA ticket for every Bug detected. Reading all the test log files is a task impossible for a human, and this is why we do it by computer. The ART-Monitoring-Script, written in Python, is one of the validation tools. It analyzes all

the test results, detects Bugs, tracks JIRA tickets and generates a website that displays this information. To achieve the last part, the script files need to be updated each time a JIRA ticket is submitted. The Shifter needs to enter the JIRA ticket ID as well as other information, as the branch for which is was submitted. Then the Shifter commits his modification with Git, logs in into AFS and runs manually the script one more time to update the Bug information in the website. Tests finish to run at different times of the day, so the script is scheduled in AFS to run every hour.

Another validation tool is used by the Shifters: a website. This website displays all the information computed by the ART-Monitoring-Script in a more readable and structured way.

2 Motivations and goals

The Trigger software is undergoing a major update and its continuous validation is crucial. Hundreds of ART tests are running every night and need to be monitored to detect the potential issues.

However, because the Trigger section used to have two separate shifts during data taking, online release coordinator and trigger validation shifter, there were two completely separate websites. One for data-taking specific tests and another for everything else.

The main idea behind this project was to centralize the information into one website. This new website will implement all the functionalities of the old ones, enhance the user experience and bring new features.

As it will be a validation tool, it will also need other improvements to allow the validation team more efficient checking, better JIRA tickets management and understanding of the overall status. Moreover, it should also enhance the user experience and add new features such as memory monitoring.

This new website should also be very modular to scale with the different new features that the Trigger software major upgrade may need in the future.

3 Mission

3.1 Workflow

Every week I had meetings with my supervisors and other members of the team when they were available. It was a moment to discuss about the progression of the project, the implementation ideas and which functionalities the website should have.

My first mission was to get accustomed to the current state of the validation tools. The analysis and the website generation were all done by the ART-Monitoring-Script, written by Yasuyuki Okumura, a member of my team. I spent my first weeks on gradually improving my knowledge of the script operation, as the data generation, analysis of the test results, and data presentation were strongly linked together.

Our first decision was to clearly separate the data generation from the data presentation. By doing so, the code will be far more modular and easy to maintain or to upgrade in the future. Indeed, by keeping this strong interaction between them, the small modification of one could produce unwanted alterations in the other one. A very dangerous feature that may lead to a lot of issues.

The next step was to find a replacement to the old data presentation and a way to make it communicate with the data generation. Finally, we have decided that the ART-Monitoring-Script would output JSON files, which would be read by the data presentation script. For this last part, I have suggested and used the framework Angular as my supervisors left me the initiative to do so.

Once this main step was achieved, I worked on GitLab to improve the Shifters routine and switch the hourly script execution from AFS cron jobs to GitLab. I used

GitLab Pipelines, a collection of scripts, to simplify Shifters' task and have a better control of the script execution. The GitLab interface is more user-friendly than the AFS one.

The last stage was to deploy the website and it was split in three steps:

1. The ART-Monitoring-Script generates the old website and also outputs the JSON files used for the new one. Both websites are running, the old one is still the default one. Shifters use the new GitLab Pipelines.
2. After my presentation on the 28/08 about how to use the new website, it becomes the default one. The old website is still kept as a back up.
3. The old website is removed, the ART-Monitoring-Script only outputs JSON files.

3.2 Challenges

Before starting the development of the website I had to study and decide how to implement it, what technologies to use and how to store the data generated by the ART-Monitoring-Script.

I had learned through my projects at EISTI how to build dynamic websites. A dynamic website is perfect when the data it is composed of changed regularly. And this was exactly my case: each day we have new Nightly data to monitor and to display. I wanted to use the same technologies as the one that I have learned and developed with: a dynamic server side interacting with a database.

However, the website is hosted on EOS, one of the CERN file system, and does not allow the use of a dynamic server in the same conditions as I was used to. One cannot use high-level framework easily and it comes with a great performance cost. One of the solution would have been to contact the CERN IT service to see with them if I could use such technology. But I knew that it would have taken a lot of time and they wouldn't have done an exception for me. With a low-level server-side

the use of a database is pretty limited, as one would need to spend a lot of time to implement everything from scratch. Moreover at this time me and my supervisors thought it was not possible to have access to a database.
I could not utilize what I was used to and I needed to find another solution, learn it and implement it.

A dynamic server-side was not realisable but a dynamic client-side was. It does not generate the HTML in the server before sending it back to the client browser, but instead generate them directly in the client browser. A website using dynamic client-side generation is called a Single Web Page Application. Because the HTML page changes directly in the browser, the client only needs to get a single page from the server. After that everything happens in the browser. There is still some communication with the server in order to get the data to display. A database would have been great but as it was not practical, a substitute was used: JSON files.

JSON is a standard data interchange format, mostly used to make a web application communicate with a server. Although the modularity of a database was lost, JSON files are perfect for this case and are easy for the ART-Monitoring-Script to output.

The development plan was now completed, however, the technology to generate the Single Web Page Application was yet to be decided. As I was already familiarized with Angular, a JavaScript framework developed by Google, I suggested my supervisors to use it. They trusted my judgement and entitled me to do so, giving me a great responsibility in the choices of the website implementation.

Another challenge was the implementation of the GitLab Pipelines. Some basic script functionalities were not working properly inside the Pipeline. We figured out that it was due to how Pipelines were working internally. But as we could not change how Pipelines are operated, we had to find some other ways to bypass these problems by looking deeply into the Linux documentation.

4 Results

The new website I developed improves all the old features and bring upgrades, as a better user experience and a chart interface highly customisable, to help the validation process. Furthermore, thanks to the great structure of Angular applications, it is highly modular and can easily receive more enhancements. If you want to see the result you can access the website at:
https://test-atrvshft.web.cern.ch/test-atrvshft/ART_monitor/

5 Conclusion

This first internship in an international organization was an unforgettable first step into the international professional world. I hope my job here will help the ATLAS Trigger validation team and will allow a better development management. The upgrades for the ATLAS experiment are only at their beginning, and they will be difficult.

This experience was definitely rewarding, both professionally and personally. I earned a lot of background in web development, continuous integration and project management. This knowledge will surely help me in my next years.

I would like to come back at CERN to work there. And I recommend the Summer Student Programme for anyone who has the scientific curiosity.

6 Acknowledgements

First, I would like to thank CERN for providing me this unique experience and opportunity to get engaged in one of the biggest international experiments. Many thanks to the Summer Student Team for their support and work. I really enjoyed my project and it was a pleasure to work with the ATLAS Trigger Validation team. Many thanks to my Summer Student friends for this beautiful summer.

Finally, I would like to express my deepest appreciation to my supervisors Rafal Bielski and Daniele Zanzi for their guidance, patience and the trust that they placed on me by letting me responsibilities.