CERN
Summer Student Programme

WORK PROJECT REPORT

under the supervision of Antoine Marzin (PH-ADT-TR) and Kristof Schmieden (PH-ADT-TR)

Monitoring of ATLAS Central Trigger Processor

Florentin Delaine

PH-ADT-TR

August 22, 2014

Abstract

The ATLAS Level-1 (L1) trigger is the first stage of the trigger chain. It must reduce the input data rate from the initial bunch-crossing rate of 40 MHz to a maximum of 100 kHz as required by the HLT and the subdetectors readout systems. The final decision of the L1 is taken by the Central Trigger Processor. During the Long Shutdown 1, upgrades are carried out on the hardware of this system in order to stretch its limits and improve its flexibility for Run 2. The software also needs to be updated to handle the new functionalities of the hardware. The development of status monitoring applications is the main activity of this Summer Student project.

Contents

1 Introduction

1.1 Central Trigger Processor

The Central Trigger Processor (CTP) combines information from the calorimeter and the muon triggers in order to make the decision to keep or not an event. The requirement of the Level-1 (L1) is to reduce the data rate from 40 MHz to 100 kHz after the Long Shutdown 1 (LS1). Given the bunch-crossing interval which is of 25 ns, and the requirements made for the decision latency (2.5 µs), every delay must be as short as possible. Therefore its efficiency is crucial [1] [2].

During Run 1, some characteristics of the CTP already reached their design limits and so an upgrade of the device will be performed to provide more flexibility during the Run 2 which will work through higher energy and higher luminosity than before at the Large Hadron Collider (LHC).

1.2 Upgrade during LS1

The purpose of this upgrade is:

- to add a new topological processor to improve multi-object selection;
- to double the number of trigger inputs on the PIT bus backplane (from 160 to 320) to include the inputs from the topological processor;
- to double the trigger items (from 256 to 512);
- to use up to three partitions in parallel, in order to perform detector calibrations;
- to double the notch groups for bunch-by-bunch masking of the trigger items (from 8 to 16).

To do so, the CTPCORE, the CTPOUT and the custom backplane will be replaced. At the same time, the software will also go through changes, new ones have to be designed as some others have to be adapted [3].

2 CTP monitoring

Among all the software updates, the CtpMonitoring package needed to be redesigned in accordance with the changes made on other software like the CtpConfigurator or the CtpMonitoringServer. This is shown in figure 1 which represents the new architecture of the CTP software.

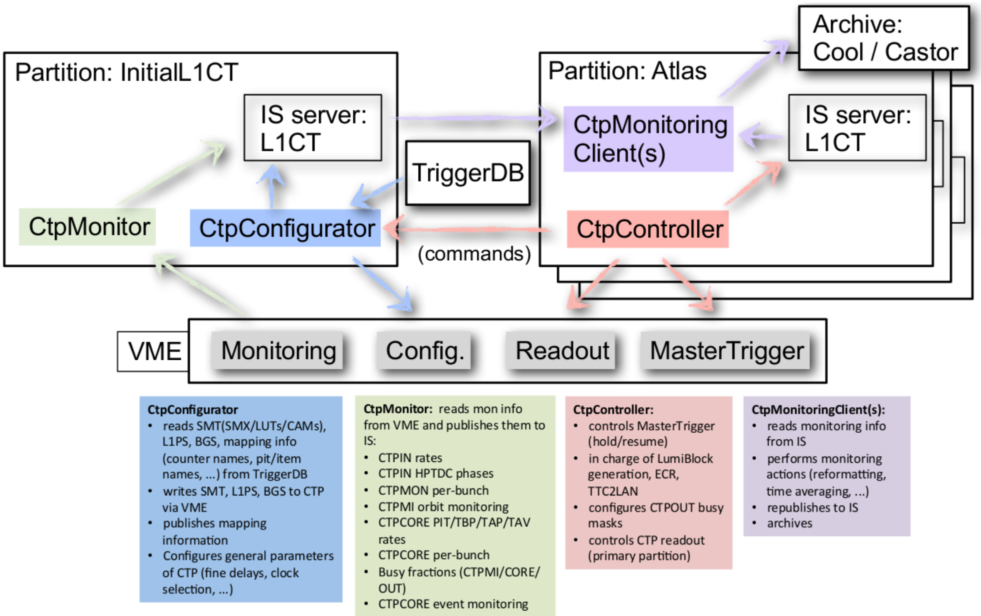

Figure 1: New architecture of the CTP software

2.1 Tasks of the CtpMonitoring application

Several monitoring tasks have to be handled by the CtpMonitoring package applications:

- Rate monitoring: matching of counters and names of trigger inputs, collecting of instantaneous rate and time-integration over Lumiblocks, performed by the CtpcoreTriggerRateMonitoring, the CtpinRateMonitoring and the CtpmonRateMonitoring;
- Status monitoring: active monitoring of the CTP boards configuration with comparison to the parameters read in the trigger database. This is performed by the subpackage CtpStatusMonitoring which is described in the section 3;
- Phase monitoring: matching of counters and names in order to produce time-trend plot, performed by the CtpinPhaseMonitoring;
- Busy monitoring: matching of counters and names, collecting instantaneous busy fractions, and time-averaging of fractions over Lumiblocks, performed by the CtpBusyMonitoring;

- Data quality monitoring: quality control of the input data, performed by the CtpDataQualityMonitoring.

However, this list is not complete and some other tasks may have to be carried out, like the temperature and voltage monitoring of the CTP, performed by the CtpTempAndVoltageMonitoring. This is why the development of the package has to facilitate the development of any client.

2.2 Modular development

The CtpMonitoring package provides a monitoring base client class, which inherits from the Controllable class of RunControl, with all the methods to control the application implemented. The package also defines specific common exceptions which can be thrown by the monitoring softwares and the handler to manage them. Finally

2.3 Information Service

In order to share data between applications, the Information Service (IS) has been developed as part of the ATLAS TDAQ project. It is using a client-server architecture and object model to store information. IS allows to publish and to read information but also to subscribe to it and the server will send notifications when changes are made [4].

This tool is widely used in the L1 software as it allows applications to communicate indirectly and easily, especially for monitoring softwares which needs to gather informations from various sources (figure 1).

3 CTP Status Monitoring

The status of the modules of the CTP must be monitored in order to confirm that the configuration from the database matches the configuration in the hardware. In this way, the current status of each board has to be compared to settings previously entered. For each module, a standalone application must perform this and give information to the Control Room, integrated into the main software of the TDAQ.

3.1 Modular development

As the informations monitored on each board are different, independent applications were designed (see 3.2.1). Each one is built around the monitoring client from the CtpMonitoring package and an analyser created for this package.

3.2 Status monitoring clients

3.2.1 List of the clients

Several monitoring client applications were designed:

- CtpcoreStatusMonitoringApp;
- CtpinStatusMonitoringApp;
- CtpmonStatusMonitoringApp;
- CtpcalStatusMonitoringApp;
- CtpmiStatusMonitoringApp.

According to their names, each one is designed for the board it is named after.

3.2.2 General process

The working of each application is very simple. At calling of RunControl commands, subscriptions to the IS publications are managed first, including a step to retrieve information already existing in the server to initialize the status analyser. Then, the application waits for notifications from the IS repository and once it happens, it updates its data about the module. The checking of the status is performed when the updated informations come from the monitoring server and a timer starts right after that to ensure a regular monitoring of the board, otherwise there might be an issue.

On exit, the application unsubscribe to the IS publications and then destroys the analyser and the monitoring client.

3.3 Examples

This section aims to present some parameters monitored and prove their relevance.

3.3.1 CtpcoreStatusMonitoring

This application monitors for example many delays like the Event Counter Reset delay, the ORBIT delay and the delay of the two pre-scaled clocks. These delays are important to keep the consistency of the timer informations exchanged with the High Level Trigger or the Timing, Trigger and Control (TTC). If one value would come to change without any reason and without noticing it, this could make the data acquired useless.

3.3.2 CtpinStatusMonitoring

An interesting parameter monitored is the pipeline length of each cable connected to a board of the CTPIN. These length are useful in order to take in account the delay introduced by the cables. These are hardware delays which can not be avoided and contribute to the non synchronization of the subdetectors. Like for the delays presented for the CTPCORE, a change in these values could make the recorded data useless.

4 Conclusion

The upgrade of the Central Trigger Processor involved the development of new software tools for communication between hardware devices and databases. The work carried out during this project enabled to design the CtpStatusMonitoring package and to add new features to the CtpMonitoring package which will be used in the monitoring software during Run 2, and is modular enough to suite with further improvements.

References

[1] S Ask, D Berge, P Borrego-Amaral, D Caracinha, N Ellis, P Farthouat, P Gãllnö, S Haas, J Haller, P Klofver, A Krasznahorkay, A Messina, C Ohm, T Pauly, M Perantoni, H Pessoa Lima Junior, G Schuler, D Sherman, R Spiwoks, T Wengler, J M de Seixas, and R Torga Teixeira. The ATLAS central level-1 trigger logic and TTC system. *Journal of Instrumentation*, 3(08):P08002, 2008.

[2] ATLAS Level-1 Trigger Group. ATLAS Level-1 Trigger Technical Design Report. Technical report, ATLAS, June 1998.

[3] G Anders, D Berge, H Bertelsen, M Dam, E Dobson, N Ellis, P Farthouat, C Gabaldon, M Ghibaudi, B Gorini, S Haas, M Kaneda, S Maettig, A Messina, C Ohm, T Pauly, R Pottgen, R Spiwoks, M Stockton, T Wengler, and S Xella. The upgrade of the ATLAS Level-1 Central Trigger Processor. *Journal of Instrumentation*, 8(01):C01049, 2013.

[4] S Kolos, G Boutsioukis, and R Hauser. High-performance scalable Information Service for the ATLAS experiment. Technical report, CERN, Jun 2012.