Modeling Resource Utilization of a Large Data Acquisition System

Alejandro Santos[1][2], Pedro Javier García[3], Wainer Vandelli[1], and Holger Fröning[2]

[1] Physics Department, CERN, Geneva, Switzerland
[2] Institute of Computer Engineering, Ruprecht-Karls University of Heidelberg, Germany
[3] Computing Systems Department, University of Castilla-La Mancha, Albacete, Spain

Abstract. The ATLAS 'Phase-II' upgrade, scheduled to start in 2024, will significantly change the requirements under which the data-acquisition system operates. The input data rate, currently fixed around 150 GB/s, is anticipated to reach 5 TB/s. In order to deal with the challenging conditions, and exploit the capabilities of newer technologies, a number of architectural changes are under consideration. Of particular interest is a new component, known as the Storage Handler, which will provide a large buffer area decoupling real-time data taking from event filtering. Dynamic operational models of the upgraded system can be used to identify the required resources and to select optimal techniques. In order to achieve a robust and dependable model, the current data-acquisition architecture has been used as a test case. This makes it possible to verify and calibrate the model against real operation data. Such a model can then be evolved toward the future ATLAS Phase-II architecture. In this paper we introduce the current and upgraded ATLAS data-acquisition system architectures. We discuss the modeling techniques in use and their implementation. We will show that our model reproduces the current data-acquisition system's operational behavior and present the plans and initial results for Phase-II system model evolution.

Keywords: Simulation model, OMNeT++, Data Acquisition

1 Introduction

Data-acquisition systems for high-energy physics experiments have demanding computing resource requirements. They are complex systems, needing to process data in real time. The ATLAS experiment [1] at CERN will be facing new requirements in terms of data throughput for the upgrade starting in 2024.

There is still a significant uncertainty with respect to the technologies to be used for the new system implementation and their availability at the time of upgrade. The existing data-acquisition system has proven to be adequate for the current experiment conditions, but it will undergo major changes to fulfill the new requirements. The work presented in this paper explores the use of discrete simulations to model and study data-acquisition systems. Ultimately, the aim is modeling the future system to explore architecture, provisioning and advanced techniques such as compression and storage under different technology scenarios. In order to develop a trustworthy model the current data acquisition system is analyzed first. The results of simulating the current TDAQ system with that model will be presented.

2 Current ATLAS Data Acquisition System

The existing ATLAS Trigger and Data Acquisition system (TDAQ)[2] selects relevant data for the experiment's goals in real-time. The architecture of the current TDAQ system is shown in Figure 1. Data are transfered from the detector in the form of an *event* through serial links. An event is composed of many *fragments*, one for each serial link.

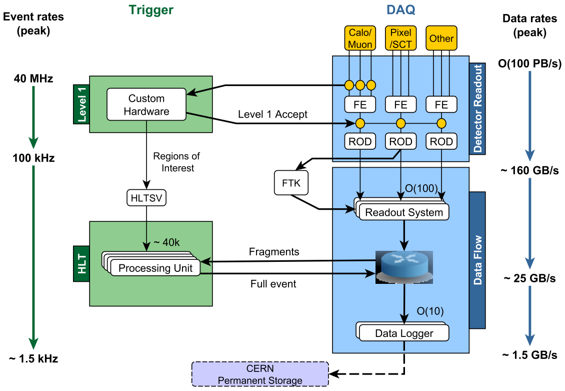

Fig. 1: Current ATLAS Trigger and Data Acquisition architecture.

The TDAQ system has a two level trigger system. The first level is implemented with custom electronics reducing the 40 MHz event rate to 100 kHz. The second level, called the High-Level Trigger (HLT), is implemented by a farm of commodity servers connected via Ethernet network. It reduces the event rate from 100 kHz to 1 kHz by selecting interesting events which are then sent to permanent storage.

The HLT is coordinated by the High-Level Trigger Supervisor (HLTSV), a dedicated software process which assigns each arriving event to an available Processing Unit (PU). Each PU is executed on a dedicated HLT CPU core and analyzes events by reading their fragments from the Readout System (ROS). The ROS system provides an interface between the ATLAS detector custom electronics and the commodity Ethernet network. The ROS also provides buffering of unprocessed fragments, implemented as a distributed computer system of many machines. Specific physics analysis algorithms executed by the PU process request individual fragments from individual ROS machines, but not all fragments are required to process an event. Dataflow to and from PU processes on each HLT server is coordinated by a Data Collection Manager (DCM) process. Each DCM provides the arbitration of the network access for PU processes.

3 The Future ATLAS Data Acquisition System

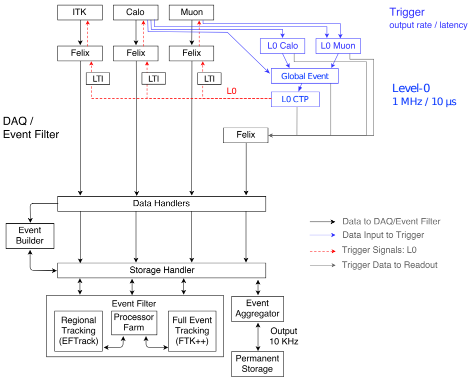

Fig. 2: Future ATLAS Trigger and Data Acquisition architecture.

The baseline architecture for the Phase-II ATLAS data-acquisition system is shown in Figure 2. The Level-0 trigger reduces the event rate from 40 MHz to 1 MHz and the Event Filter (EF) reduces the event rate from 1 MHz to 10 kHz. Data acquisition challenges in the upgraded system include the higher data throughput of 5 TB/s and the greater processing power required for the EF. One key component of the future ATLAS system is the Storage Handler, which will provide temporary buffering for event data. The equivalent component in the current system is the ROS, and the study of a ROS model will bring us better understanding of the Storage Handler.

4 The Model for the Current Data Acquisition System

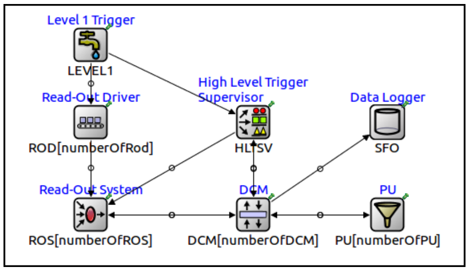

Fig. 3: OMNeT++ simulation model for the simplified ATLAS TDAQ system.

Figure 3 shows the implementation of the simulation model for the current ATLAS data-acquisition system in OMNeT++[3]. The model represents a simplified version of the current TDAQ system. The network is assumed to be ideal with infinite capacity and no packet loss. The simulation assumes that, for small intervals of time, the conditions of the experiment remain constant. The only modeled delay is the event processing delay for the PU processes.

Input values for the simulation are extracted from ATLAS real operational monitoring data, and include the *incoming data rates*, *fragments sizes*, and *processing delays* for each PU. Another input value is the *ROS request sizes*, which is the distribution of the number of requested fragments per ROS. These operational data are processed in several steps. First, outliers in the data are removed by discarding the *outer fences* values [4]. Next, values are averaged in five minute intervals. Processing times are already averaged over smaller time intervals and they have to be normalized to be re-averaged again.

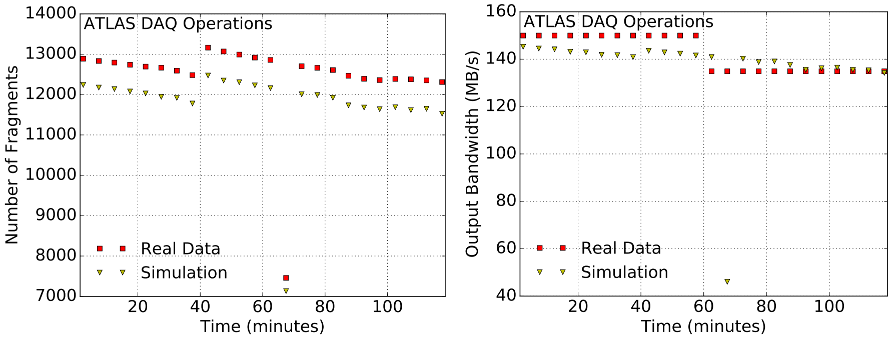

Fig. 4: Comparison of the simulation results and real data for the average number of fragments in the ROS and the average output bandwidth of the ROS.

5 Simulation Results

Simulations are executed on a Xeon E5645 2.4 GHz machine with 24 GB of RAM. Each simulation is executed independently for 60 simulated seconds and takes ~6 hours to complete. In total, 24 simulations are executed over 2 hours of consecutive data. Figure 4 shows some of the simulation results, with an outlier at minute ~70. The real system stopped due to external conditions and the simulation does not reproduce this behavior. The number of fragments results differ due to missing delays in the model of ~10 ms, and output bandwidth results differ due to the low resolution of the real data and network retransmissions.

6 Conclusion

A simulation model has been developed to study the behavior of the current ATLAS TDAQ system. Results show a good and stable agreement between simulation and real data, with a relative error below 5%. Simulation results can be further improved by adding more accurate simulation of components of the TDAQ system and network latencies to the model. It can then be used as the basis to studying the behavior of the candidate architectures for the new system.

References

1. ATLAS Collaboration. Performance of the ATLAS detector using first collision data. *JHEP*, 09:056, 2010.
2. Astigarraga, ME Pozo (on behalf of the ATLAS Collaboration). Evolution of the ATLAS trigger and data acquisition system. In *Journal of Physics: Conference Series*, volume 608, page 012006. IOP Publishing, 2015.
3. Andras Varga. Omnet++. In *Modeling and Tools for Network Simulation*, pages 35–59. Springer, 2010.
4. Songwon Seo. A review and comparison of methods for detecting outliers in univariate data sets. Master's thesis, University of Pittsburgh, 2006.