Specification and Simulation of the ALICE Trigger and DAQ System

T. Anticic, G. Di Marzo Serugendo, P. Vande Vyvre, O. Villalobos Baillie, P. Jovanovic

For the ALICE collaboration

Abstract

The ALICE Trigger and Data Acquisition (TRG/DAQ) System is required to support an aggregate event building bandwidth of up to 4 GByte/s and a storage capability of up to 1.25 GByte/s to mass storage. The system has been decomposed in a set of hardware and software components and prototypes of these components are being developed.

It is necessary to verify the system design, its capability to reach the expected behavior and the target performances, discover possible bottlenecks and ways to correct for them, and explore alternative algorithms and new architectures.

To achieve this the complete TRG/DAQ system has been formally specified, and the verification of the expected behavior has been performed through the execution of the specification. Two tools were used for this: Foresight, and Ptolemy.

1 Introduction

ALICE [1] is an experiment dedicated to explore heavy ion interactions at the LHC. Its exceptionally large aggregate event building bandwidth of up to 4 GByte/s and a storage capability of up to 1.25 GByte/s to mass storage presents unique challenges. It is a complex experiment, having detectors of differing data gathering times, readout times, and data volumes. Further, the trigger system involves 4 separate event types : central, minimum bias, dielectron, and dimuon, each of which has its own subset of detectors read out. These subsets can overlap, e.g. a detector involved in dimuon events can also be involved in central events.

It is necessary to be able to predict the behavior of this system in order to find and correct any bottlenecks. And, as the ALICE final architecture is not defined, it is necessary to explore alternative algorithms and new architectures. A detailed and realistic simulation has been created to achieve this. The formal specification was initially realized using a commercial tool called Foresight [2]. The Ptolemy [3] simulation environment was used for the final implementation.

2 Simulation tools

2.1 Foresight

A Foresight specification is made of hierarchical data flow diagrams, finite state diagrams, and pieces of procedural modeling language. The specification provides an unambiguous description of the system. The semantics of the specification provides a model of the system whose behavior is very close to the behavior of the system.

The Foresight simulation consists of the real-time execution of the specification. It offers debugging functions like animation of diagrams, breakpoints, and monitor windows.

2.2 Ptolemy

Ptolemy is a heterogeneous simulation environment developed as a free, open source project at Berkeley. It supports a wide variety of models of computation, so called domains. For the ALICE DAQ simulation the discrete event domain is used. It is an extremely efficient tool, this being the main reason for transferring the Foresight specification to Ptolemy. Ptolemy is C++ based, and user defined modules are also written in C++. The modules are independent of each other, and communicate with each other via messages. Complex systems are set up using ptcl, a Ptolemy tcl variant.

As the simulated time needed to reach plateauing can take days, the Root [4] package (an open source analysis software developed at CERN) is used for online plots to monitor in detail the progress of the simulation. This combination has proved very efficient. Further, the fact that in ALICE the standard programming language is C++ and Root the standard analysis package, makes it easy for other ALICE members to use and extend the simulation [6].

3 Overall system

The main components of the ALICE DAQ system specification consists of an Interaction Source, Trigger System, Detectors, DAQ, and Permanent Data Storage. The Interaction Source feeds the Trigger System with events, which in turn emits L0, L1, and L2 signals to the detectors. The detectors send data to the DAQ, and communicate with the Trigger System. The DAQ performs sub-event building, Level 3 decisions for further selection of dielectron events, event building, and transfer of data to the Permanent Data Storage (PDS).

All the known and assumed ALICE architectures, rates, event type probabilities, buffer sizes, parameters, data sizes and distributions are used [5], unless indicated otherwise. The full specification involves thousands of independent units working in parallel, with data flow rates and volumes between them monitored in detail using Root.

3.1 Interaction Source

The Interaction Source generates events according to a Poisson distribution, and decides what type the event is. It can also make background events, used for Past/Future protection only. Mixed event types are also supported.

3.2 Trigger system

The trigger System takes care of L0, L1 and L2 triggers (occurring at t_{L0}, t_{L1} and t_{L2} respectively) and Past/Future (P/F) protection, which is used to prevent readout of events contaminated by the occurrence of another event during the data gathering phase. Each event type has its own P/F protection time. For mixed events, at any time P/F is checked and it is not satisfied, the type with a smaller P/F protection time is tried. If P/F protection is satisfied for this type, then the event type is changed.

- As soon as an event is received from the Interaction Source the P/F protection is tested. If satisfied, the Trigger System next checks if the detectors involved in this event type are free. If some detectors in a detector subset are busy for a mixed event type, then the event type with fewer detectors is tested to see if all the detectors can be read out.
- If all detectors are available, the Trigger System waits until t_{L0}, and sends a L0 signal to the detectors if P/F is satisfied.
- At t_{L1}, P/F is checked again. If satisfied, a L1 signal is sent to the detectors, and the Trigger System waits until t_{L2}.
- If at t_{L2} P/F is satisfied a L2 ACCEPT is sent to the detectors. Otherwise, a L2 REJECT signal is sent.

3.3 Detector

A detector takes inputs from the trigger system, and sends data to Readout Receiver Cards (RORC) via Detector Data Links (DDL). Dielectron events involve only a subset of DDLs.

- On receiving a L0 signal, a detector becomes "busy", with the Trigger System not sending any more L0 signals to it.

- If no L1 signal arrives at t_{L1}, the event is rejected, and the detector becomes "free". Otherwise, after a readout time the event is placed in a multi event FIFO buffer queue (one for each DDL).
- After the readout time the detector is free again (unless any multi event buffer is full), and waits for the L2 signals. In the meantime it can receive further L0 and L1 signals. If a L2 REJECT arrives the event is removed from the buffer queue. If it is a L2 ACCEPT, then the status of the RORCs are checked. If any RORC is busy or full then the detector waits until they become available before sending the data.

3.4 DAQ

- The Event Destination Manager (EDM) is responsible for assigning a Global Data Collector (GDC) to an event (corresponding to the modulo of the event) and sending this information to the Local Data Concentrators (LDC).
- The RORC receives data from the detector via DDLs. A bus is used to transfer the data from multiple RORCs to a LDC.
- The LDC waits until the data from all the RORCs is received and then the subevent is built and sent to the GDC via a switch. Only one stream at a time is permitted.
- The procedure for LDCs is modified for dielectrons, as a Level 3 implementation is used to increase the percentage of true dielectrons by having further data processing. The subevent is reduced in size to simulate cluster building, and forwarded to Level 3, which rejects a percentage of dielectrons. This decision is returned to all the LDCs. If accepted, the subevents are, like before, forwarded to the GDC, and are discarded otherwise.
- A GDC waits for all the LDCs to send the sub-events via the switch. The event is sent to the PDS when it is complete.

4 Results

The time needed to simulate one second within the Ptolemy environment is about 1 minute on an Intel PII 400 MHz, permitting realistic periods to be investigated.

A total rate of 6000 Hz is used in the results presented, with a rate of 1200 Hz each for central, dimuon, dielectron, minimum bias, and background events. No mixed events at this point are used.

The simulation runs show that better rates are achieved if the currently envisaged GDC event assignment is modified by having the GDC with the largest available buffer dedicated to an event. Similarly, results are improved by enabling an LDC to have a multi-stream output. With these changes the PDS is fed in all tested architectures the design rates of 1.25 GBytes/s.

Limitation of rates/data volume is achieved by data back-pressure, starting with the GDCs getting filled, then the LDCs, RORCs, DDLs, with finally the detectors getting busy, and not being able to receive any more L0 signals from the trigger system. All GDCs get filled completely, but this is not true for the other components, as rendering one detector busy for an event type limits the rates for all the detectors (and hence DDL, RORC, LDC buffer occupancy) in the detector set for the event type.

Table 1 shows the L2 rates for different scenarios investigated (A, B, C, D).

- Scenario A. All events are taken, subject only to DAQ limitations. Dimuon and dielectron rates get reduced by a large factor from their maximum rates. This is due to central and minimum bias events overwhelming the system with their order of magnitude larger data volumes.
- Scenario B. As it is important from a physics perspective to take as many dimuon and dielectron events as possible, a limitation of 18 Hz (to leave enough remaining bandwidth)

is imposed on central and minimum bias rates. The resulting rates are much better for dimuons and especially for dielectrons.

- Scenario C. By dedicating GDCs for dielectron and dimuon events the limitation on their rates due to GDC back-pressure of central and minimum bias data is removed. Then the only coupling of the different event types will be due to the busy status of shared detectors. The dimuon rates have significantly improved. However, the same is not true for dielectron rates, as for them (and unlike for dimuons) the shared detectors with central and minimum bias detector sets limit the rate.
- Scenario D. Increasing DDLs, RORCs, and LDCs of detectors which have their LDCs filled does not improve the rates. The reason for this is that it is the rates from the GDC to the PDS, which are at maximum, that limit the final rates. Thus, it is not important which detector gets busy first.

Table 1: L2 rates for different Trigger/DAQ scenarios

type	A	B	C	D
dimuon	420	750	1050	1050
dielectron	20	220	200	200
central	20	18	17	17
minimum bias	20	18	17	17

5 Conclusion

A realistic simulation of the ALICE DAQ has been developed, which permits detailed investigation of event rates and data volumes involved. The tool is fast, and easy to set up and modify.

It has been verified that the system can successfully sustain the bandwidth it is designed for. However, limitation have been discovered regarding rates of individual event types. Dedicating GDCs for event types solves this for dimuons, but dielectron rates cannot be increased this way. Also, increasing the number of DDLs, RORCs, and LDCs past a reasonable number will not increase rates, at least in the configurations explored.

Further architectural designs are being investigated to try to increase the dielectron rates. Also, more detailed specifications are being worked on, like making a realistic simulation of the Gigabit Ethernet switch, or the DAQ software framework.

References

[1] http://alice.web.cern.ch/Alice/.

[2] Foresight-Systems, inc., Austin, TX 78759. http://www.nuthena.com.

[3] Ptolemy project, Heterogeneous modelling and design, UC Berkeley, EECS http://ptolemy.eecs.berkeley.edu/ptolemyclassic/pt0.7.1/index.htm.

[4] Rene Brun and Fons Rademakers, ROOT - An Object Oriented Data Analysis Framework, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.

[5] O. Villalobos Baillie, D. Svoboda, P. Vande Vyvre. Data Acquistion, Control and Trigger. Internal Note DAQ ALICE/98-23, CERN, 1999.

[6] http://trinity.irb.hr/alicedaq/.