

## ARCHITECTURE OF THE ATLAS ONLINE PHYSICS-SELECTION SOFTWARE AT LHC

S. Armstrong, J.T. Baines, C.P. Bee, M. Biglietti, A. Bogaerts, V. Boisvert,  
M. Bosman, S. Brandt, B. Caron, P. Casado, G. Cataldi, D. Cavalli,  
M. Cervetto, G. Comune, A. Corso-Radu, A. Di Mattia, M. Diaz Gómez,  
A. dos Anjos, J. Drohan, N. Ellis, M. Elsing, B. Epp, F. Etienne, S. Falciano,  
A. Farilla, S. George, V. Ghete, S. González, M. Grothe, A. Kaczmarska,  
K. Karr, A. Khomich, N. Konstantinidis, W. Krasny, W. Li, A. Lowe,  
L. Luminari, H. Ma, C. Meessen, A.G. Mello, G. Merino, P. Morettini,  
E. Moyse, A. Nairz, A. Negri, N. Nikitin, A. Nisati, C. Padilla, F. Parodi,  
V. Perez-Reale, J.L. Pinfold, P. Pinto, G. Polesello, Z. Qian, S. Rajagopalan,  
S. Resconi, S. Rosati, D.A. Scannicchio, C. Schiavi, T. Schoerner-Sadenius,  
E. Segura, J.M. de Seixas, T. Shears, S. Sivoklokov, M. Smizanska, R. Soluk,  
C. Stanescu, S. Tapprogge, F. Touchard, V. Vercesi, A. Watson, T. Wengler,  
P. Werner, S. Wheeler, F.J. Wickens, W. Wiedenmann, M. Wielers, H. Zobernig

THE ATLAS HIGH LEVEL TRIGGER GROUP \*

presented by P. CASADO<sup>†</sup>

*European Organization for Nuclear Research, CERN, 1211 Geneva 23,  
Switzerland*

Given the extremely high bunch crossing rate foreseen at the Large Hadron Collider (LHC) and the general-purpose nature of the ATLAS particle physics experiment, after the hardware-based first level trigger, an efficient and flexible trigger software is needed for the online selection of events. This filtering of events is organized in two levels: the second level trigger and the event filter. Both levels are referred together as High Level Trigger (HLT). A coherent approach to event selection across the HLT has been taken. Thus a common core software framework has been designed to maximise the usage of offline interfaces and software components, whilst allowing sufficient flexibility to meet the different interfaces and requirements of the two different levels, notably those of performance and robustness. This paper describes the architecture and high level design of the selection software and shows how the implementation meets the challenges of the ATLAS environment.

\*<http://atlas.web.cern.ch/atlas/groups/daqtrig/hlt/authorlists/como2003.pdf>

<sup>†</sup>Work supported by the Marie Curie Fellowships Program

## 1. Introduction

The Large Hadron Collider(LHC) is expected to start data taking in 2007 at CERN (European Organisation for Nuclear Research). Proton-proton collisions will be produced at a center of mass energy of 14 TeV and design luminosity of  $10^{34}\text{cm}^{-2}\text{s}^{-1}$ . The bunch crossing frequency will be 40 MHz which will have to be reduced to about 100 Hz by the Trigger and Data Acquisition (DAQ) systems in order not to exceed the foreseen storage capability and meet the physics requirements of the ATLAS experiment<sup>1</sup>. Given the high granularity and complexity of the ATLAS detector, the event size will be of the order of 1.5 MBytes, which will result in a storage rate of up to a few hundred MBytes/s.

The ATLAS detector is a complex High Energy Physics (HEP) apparatus, which is composed of different tracking subdetectors, a solenoid, electromagnetic and hadronic calorimeters, muon subdetectors and a toroid, as well as the Trigger and DAQ systems<sup>2</sup>. The ATLAS Trigger system is organised in three levels. The first level (LVL1) is hardware-based and has to reduce the input rate from 40 MHz to 75 kHz in less than  $2.5\ \mu\text{s}$ . The result of the LVL1 selection contains information about the type of trigger and position of the possible particle candidates that cause the event to be accepted. After a positive LVL1 decision the data are transferred from electronic pipeline memories to distributed buffers in so called Read-Out Systems (ROs). The second (LVL2) and third (Event Filter -EF-) levels are software based systems running on linux PC farms. They are referred collectively as High Level Trigger (HLT). They use full granularity data of all detectors and can also combine information from different detectors. The LVL2 accesses a few percent of the total detector data in a Region of Interest (RoI) provided by the LVL1 result by direct network request to the ROs. This reduces the event rate to about 2 kHz. After a positive LVL2 decision the complete event is assembled and made available to the EF, which has access to the full detector data and the latest calibration and alignment information. Events accepted by the HLT are stored on tape for further analysis. The goal is to achieve an average decision time of 10 ms and 1 s for LVL2 and EF, respectively, although the system could easily scale to accommodate larger execution times if needed.

## 2. HLT Selection Software

The HLT selection software has been designed with Object Oriented methodology and is implemented in C++. It can run in the LVL2, EF and

offline environments. This is possible because the differences between the environments are hidden by well defined interfaces and the same framework (Athena<sup>3</sup>) is used. The boundaries between LVL2 and EF and between EF and offline are flexible, thus allowing the migration of algorithms and software components from one level to another, for example, to suit different running conditions or increased maturity of the software.

In order to perform a fast rejection, the event processing is divided into a set of steps and decisions. The criteria for continuing at the end of a step is in the form of abstract signatures which describe the physics processes ATLAS is searching for. If the event does not match any of these signatures it is rejected. If several LVL1 RoIs are present in a given event, the processing is organised horizontally (e.g. for two seeds divide the processing in steps and analyse for a given step the first seed and then the second one) and not vertically (process completely the first seed and then the second one).

The HLT selection software is subdivided into four main sub-packages: (1) The Steering organizes the processing of the HLT algorithms, so that the necessary data are produced for the trigger decision. (2) The Event Data Model (EDM) covers all data entities in the event and is used by the other components of the HLT selection software to communicate information about the event. The EDM is imported from the offline software. (3) The Data Manager provides the means for accessing the event data during the trigger processing. The current implementation is based on the transient event store of Athena. It provides the necessary infrastructure for the EDM. (4) The HLT Algorithms can either reconstruct the event or check hypothesis about previously calculated quantities. They are called by the steering, use the EDM and send data requests to the Data Manager.

### 3. Data access

A fundamental part in the design of the HLT architecture is the way in which an HLT algorithm accesses the data. The HLT algorithm requests data in a RoI using the offline transient data store (Athena Storegate<sup>4</sup>) as interface and the Region of Interest package, a software component which returns Storegate pointers to the data inside the specified detector geometrical region. These data are in form of C++ objects, convenient for the algorithm (e.g. calorimeter cells which hold energy and position), contrary to the data coming from the detector electronics which is in raw format (i.e. a binary file with the read-out channel information). The software which transforms one representation into another is the raw data converter. The

raw data conversion is done on demand (the conversion into data objects is only done for the data inside the requested RoI), and it only occurs if the necessary objects are not in memory already (“caching”). In this way the same software is used in the offline and in the HLT online running environment. The only difference is that when executed offline the data source is a file while direct network access to ROSs is used in the LVL2, or to memory resident events in the EF case.

#### 4. System Performance

In order to measure the system performance, algorithms to trigger on electrons with  $p_T > 30$  GeV were executed on simulated di-jet events with  $p_T > 25$  GeV that would be accepted by LVL1. This type of events account for a large contribution to the input rate of LVL2. The software ran in dual-2.4 GHz Xeon machines with 1 Gbyte of memory. Data were requested in a Region of Interest of size  $\Delta\eta \times \Delta\phi = 0.4 \times 0.4$  around the provided LVL1 result, which is the estimated required size to efficiently identify electromagnetic particles.

The time taken by the calorimeter raw data converter was measured. The result is 3-4 ms for a region of  $\Delta\eta \times \Delta\phi = 0.4 \times 0.4$ . The time taken by the Region of Interest package was  $< 0.10$  ms for the calorimeter code. The algorithmic part of the calorimeter HLT algorithm took  $500 \mu\text{s}$ . These results, in which offline software components are used, are well within the targeted goal for the LVL2 time budget.

These results were confirmed with measurements taken in a testbed, a realistic environment in which the data input is close to that of the data taking experiment. In this setup the time to retrieve the calorimeter raw data from the ROSs over the network can be measured, the result being around  $900 \mu\text{s}$ . In order to provide a baseline measurement of the minimum time the data conversion process would take, a special raw data converter for the calorimeter was developed, without the constraints of the offline environment. Fig. 1 (left) shows the total latency distribution for di-jet events for a region of  $\Delta\eta \times \Delta\phi = 0.3 \times 0.3$ . As can be seen in the figure, over 95% of the events are processed within 5 ms. Fig. 1 (right) shows the main contributions to the total latency. Pure feature extraction in the calorimeter is completed within  $500 \mu\text{s}$ , while data preparation and conversion are typically completed within 1.8 ms for 95% of the events, to be compared with the 3-4 ms measured using the offline software. These results serve as guidance for further system performance improvements in the raw data converters. An issue yet to be studied is the robustness of

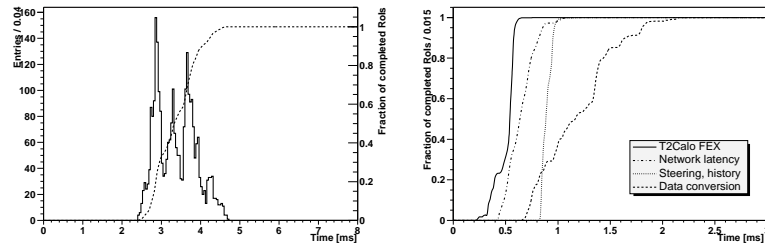


Figure 1. Total latency for RoI processing (shown at left) in the LVL2 calorimeter trigger for di-jet events at low luminosity. The dotted curve is the integral of the distribution, showing that 95% of the events are processed within 5 ms. The four main contributions to the total latency are shown (right) as curves of integrals. The contributions are (in order of decreasing importance): data preparation and conversion, framework overheads, network access time, and algorithmic processing.

the software which at LVL2 has to process three orders of magnitude larger rate than at the reconstruction farm.

## 5. Conclusions

The ATLAS HLT group has achieved a working solution for the architecture of the online selection software.

The design principles of the ATLAS online selection software have been validated, namely: It has been demonstrated that the RoI mechanism works with small overheads; and it has been shown that acceptable processing times are achievable using software which is coherent with offline.

## Acknowledgments

The authors would like to acknowledge the help and support of the Atlas Data Acquisition Group and the Atlas Offline and Detector software groups.

## References

1. ATLAS Detector and Physics Performance Technical Design Report, Volumen I, ATLAS TDR 14, CERN/LHCC/99-14; and Volumen II, ATLAS TDR 15, CERN/LHCC/99-15.
2. ATLAS High-Level Trigger, Data Acquisition and Controls, Technical Design Report, ATLAS TDR-016. ATLAS HLT/DAQ/DCS Group.
3. <http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html>
4. *The StoreGate: a Data Model for the Atlas Software Architecture*, Proceedings CHEP03, La Jolla, California, March 24-28, 2003.