STEERING THE ATLAS HIGH LEVEL TRIGGER

G. Comune, Michigan State University, East Lansing, Michigan, USA
J. Haller, CERN, Geneva, Switzerland
P. Morettini, I.N.F.N Genova, Genova, Italy
R. Stamen, S. Tapprogge, Institut für Physik, University of Mainz, Mainz, Germany
S. George, Royal Holloway, University of London, London, UK
C. Schiavi, Universita' of Genova & I.N.F.N. Genova, Genova, Italy

Abstract

This paper describes the *Steering* mechanism of the ATLAS High Level Trigger (HLT). The Steering software is responsible for the implementation of the *seeded* and *stepwise* execution of algorithms in a portion of the full event called Region of Interest (RoI). The *Steering* is responsible for the global event accept/reject decision based on a static configuration matched against the dynamic event outcome in terms of *Trigger Conditions* validated by the Trigger algorithms. In the case of an event being accepted the Steering is in charge of the creation of the *Detailed Event Result* and in order to enable this it provides tools for reconstructed objects serialization and a fast data navigation mechanism that allows to organize the objects in memory with logical relations and all objects in an RoI back to the initial RoI seed.

INTRODUCTION

The ATLAS experiment is in its final construction phase at the Large Hadron Collider at CERN (Geneva). ATLAS is a general purpose experiment designed to carry out searches for the Higgs and of possible physics beyond the Standard Model at the TeV energy scale. It has a three level Trigger system (Figure 1) in charge of reducing the initial 40 MHz bunch crossing rate down to the final 200 Hz that is sent to permanent storage. The ATLAS Level-1 Trigger is hardware based while the Level-2 together with the third level, called Event Filter (EF), forms the High Level Trigger and run on farms of commercially available PCs executing an offline-like software suite called the HLT Selection Software (HLTSSW). HLTSSW reuses at a large extent the ATLAS offline Athena/GAUDI [1] framework. Algorithms at Level-2 have access only to "*Regions of Interest*" (RoIs) where high p_T activity has been previously detected by the Level-1 Trigger through a full scan of the ATLAS calorimeters and the muon spectrometer. The event lays dispersed in fragments at the Read Out Buffers [2] and data corresponding to the small geometrical region pinpointed by the RoI is pulled over the network when requested by the algorithms. In the case of a Level-2 *accept* the event is built and shipped to an EF processing node and algorithms have access to the full event in memory. The option is left open there for them whether to access the event data on a per RoI based fashion or fully. The reconstruction in a RoI is carried out in a *stepwise* fashion and the outcome of one algorithm is used as a "*seed*" for the algorithm that follows in the *reconstruction chain*.
Level-2 algorithms are offline like algorithms purposely developed in order to cope with the stringent latency requirements. EF algorithms are directly imported form the offline reconstruction software and adapted to work in a RoI seeded fashion.

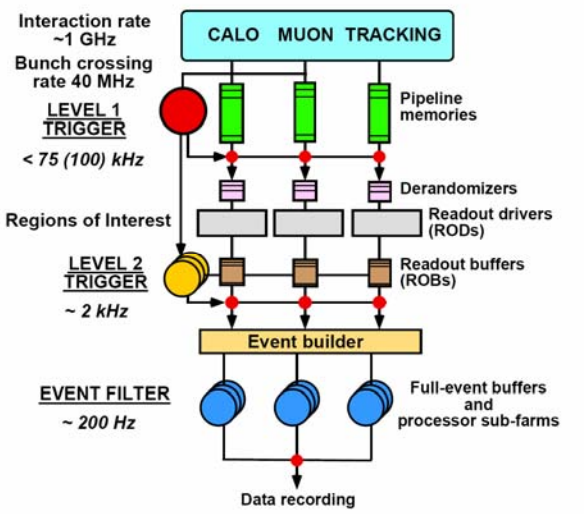

Figure 1: Schematic view of the ATLAS Trigger. The Level-1 is hardware based and scans the calorimeters and the muon spectrometer with a coarse granularity in search for Regions of Interest. Level-2 and Event Filter run on PC farms and perform partial reconstruction within the RoIs found by the Level-1

THE *STEERING* SOFTWARE

The Steering software implements the basic concepts of the algorithmic event reconstruction and selection. It runs unmodified on Level-2 and Event Filter machines and is responsible for:

- Unpacking the Level-1(Level-2) Result into the initial seeds used by the Level-2(EF) algorithms,

- Sequencing the Trigger algorithms execution and rejecting an event at every step of the selection chain if it does not fulfil at least one in a set of trigger conditions
- Globally accepting an event after all reconstruction steps have been carried out
- The creation of a detailed Event Result that contains all the trigger related information created during the event reconstruction
- Providing a fast data navigation mechanism that allows trigger objects to be *connected* with some logical *relation* establishing a data structure that connects objects back to their corresponding initial RoI seed
- Providing tools for object serialization that are used to obtain a flat machine independent and persistent representation of all objects created during the event reconstruction. Those tools are used to create the Detailed Event Result.

SEEDED AND STEPWISE RECONSTRUCTION

The goal is to achieve the Level-2 trigger decision within the foreseen 10 ms latency. In order to do this the event selection takes full advantage of the partial event reconstruction carried out only within the RoIs.

The reconstruction in an RoI is broken down in sequential steps (Figure 2). At every step one or more algorithms are executed and they *validate* the triggering embodied by a *Trigger Element* (TE) which is basically a Boolean condition. The execution of one algorithm sequence is driven by a static configuration that informs the Steering what particular algorithm (statically defined) must be executed in case a (dynamic) trigger condition (TE) is active. At the end of every step the Steering takes the decision whether to reject an event or not. This is done by matching the (static) expected combinations of satisfied trigger conditions (*signatures*) against the actual (dynamic) outcome of the event processing.

An event can be (early) rejected at any step of the reconstruction chain if the event does not fulfil any of the configuration signatures. *Signatures* are organized in chains and if at any step a signature is not satisfied the signature chain is deactivated and not checked any more at any future steps. In a similar fashion reconstruction chains that lead to trigger conditions that cannot generate any future satisfied signature are deactivated in order to minimise the processing time. What is just being described is extremely relevant for the high p_T physics programme. There are other uses cases where more complex sequencing and decision scenarios are foreseen. *Topological* triggers are one such case. They are triggers where information of two or more RoIs is combined together to form some derived quantity (i.e. invariant mass formed from two electrons) and consequently some selection cuts are applied on that quantity.

Another example that deviates from the simple sequencing schema described at the beginning of this section is the treatment of secondary RoIs. Secondary RoIs are RoIs formed and accepted by Level-1 but that do not satisfy cause the event accept. These RoIs cannot be let free to initiate reconstruction chain straight away like triggering primary RoIs do because they come at rates too high for the HLT to cope. Their reconstruction can only be started after some other additional condition as been satisfied and the rate at which the secondary RoIs are reconstructed is reduced to a manageable level.

The Steering software must be able to cope with both the simpler and the more complicated sequencing scenarios.

CONFIGURATION AND OPERATIONS

It has already been mentioned that the Steering relies entirely on a static configuration for its sequencing and decision mechanism. The HLT Configuration [3] provides the Steering with pairs of *sequence-signature* objects for every foreseen step in the Trigger selection and it makes sure that the Level-1 and HLT configuration are kept consistent. *Signatures* are organized in chains and it is vital that signature chains do not get *"entangled"*. A configuration is entangled if one ore more signatures are part of several different signature chain.

The ATLAS Trigger is a very complex system. A largely entangled Trigger configuration should be highly disfavoured because any changes to a signature definition (i.e. pre-scale factor, p_T thresholds …) would inevitably affect the whole system making its understanding a daunting task. Amongst other issues a fundamental task like estimating the overall efficiency of the Trigger with respect to a particular physics signature becomes an highly involved procedure. As a final remark an highly disentangled system allow to greatly simplify operations like dynamic pre-scale changes during data taking.

The configuration tools make sure that only Trigger configuration like the one in Figure 3 are allowed.

PERSISTENCY

The Steering provides tools for objects persistency and serialization. A dictionary based generic serialiser and a custom hand written serialiser technique is available. Objects are created by algorithms during the event reconstruction and need to be transformed into a machine independent format that can easily be added to the payload of the *Detailed Event Results* produced by both HLT trigger levels.

At Level-2 serialized objects become seeds for the EF reconstruction. All objects created both at Level-2 and at EF are finally permanently recorded on storage for later offline usage. At offline recorded objects are used for a wide variety of activities ranging from trigger optimization, debugging, efficiency/rejection curves

creation, debugging, monitoring and finally development and improvement of selection strategies and assessment of their impact on the overall physics performance of the ATLAS detector. The initial choice taken of reusing offline software in both HLT trigger levels together with a clear separation between algorithms that carry out reconstruction and create feature objects *(feature extraction* algorithms) and algorithms that verify conditions and apply cuts on a feature and/or combinations of several features (*hypothesis* algorithms) and do the actual selection verifying trigger conditions allows a complete reuse of the same algorithms and configuration to perform the aforementioned offline activities. Strategies and configurations developed in an offline environment are guaranteed to deliver the exact same performance when used/ported in the real online environment. The implementation of the Steering allows performing Trigger optimisations in an offline environment reusing the same code and even more importantly the same configuration that is used in the online environment. In an offline environment one can switch off hypothesis algorithms and run the computing intensive algorithms alone. This will produce the feature objects in a persistent format in the Event Result. Once the features have been extracted one can proceed to optimizing the Trigger by rerunning in an iterative fashion the Steering over the feature object this time switching off the feature extraction algorithms and allowing only the lightweight hypothesis algorithms to run (Figure [4]). Optimizing the trigger entails scanning the hypothesis algorithms cuts parameters' space in search for the best combination of cuts. This is a very computing intensive task that can be done in a generic and scalable fashion thanks to the Steering design and implementation.

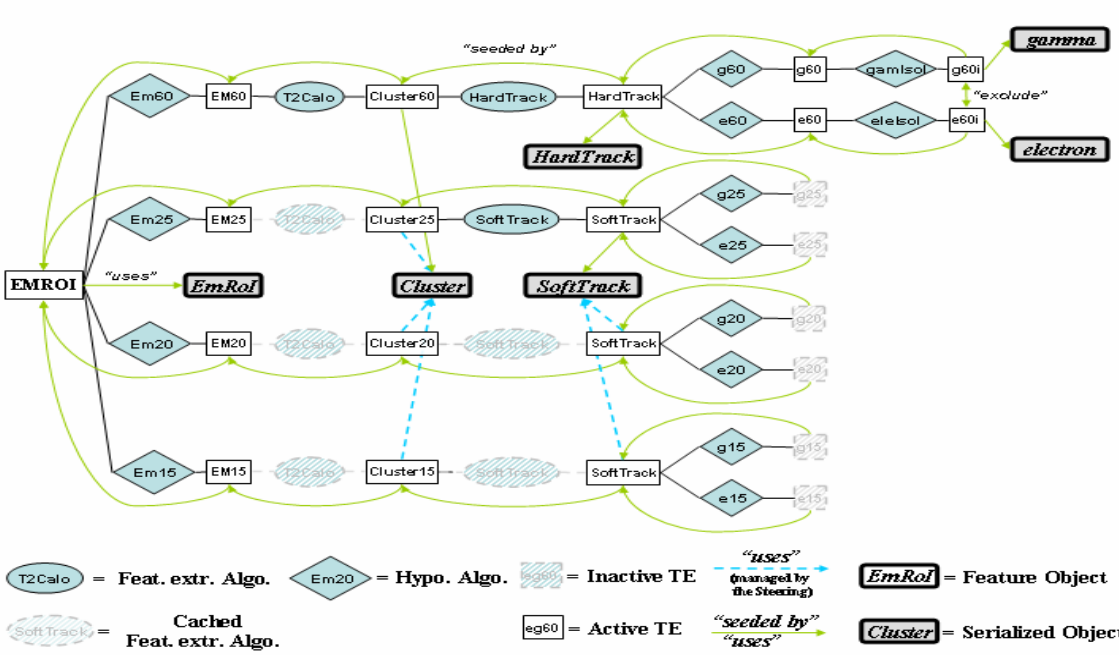

Figure 2: An example of how the Steering performs the seeded and stepwise reconstruction starting from the initial RoI. The Initial electromagnetic RoI is unpacked and a chain for each of the satisfied Level-1 pT thresholds is created. In each of this chains the Steering executes a sequence of algorithms which alternatively extract physics features (electromagnetic Cluster, Track, ..) and apply cuts on one feature object or combinations of several feature objects. The former type of algorithms is called Feature Extraction (FEX) algorithms while the latter are called Hypothesis (Hypo) algorithms. In the picture it is shown how FEX algorithms, which are computing intensive are only executed once on a particular RoI and the outcome is cached and replicated on all the chains in the same RoI. In this particular example the calorimeter algorithms (*T2Calo*) executes once on the "EM60" input trigger condition and creates the *Cluster* feature.

TIMING OF A REALISTIC CONFIGURATION

Timing the overhead added by the Steering execution implies having a realistic configuration available which currently is not the case. It is know from experiments similar to ATLAS that the number of trigger items (signatures) is in the order of few hundreds. An important missing information is the number of algorithms and finally how many objects will be serialized and how big they will be. In Table 1 it is shown the overhead added by the Steering mechanism including the Detailed Event Result formation and handling. The times in that table have to compare with the requirements of the Level-2 latency of 10 ms.

Table 1: time taken by the Steering in a configuration with 100 signatures as a function of average number of initial RoIs, number of steps the reconstruction is broken into and average size of the Detailed Event Result formation, obtained with a Pentium4 2.8 GHz processor.

<# RoIS>	<steps>	Result Size (kB)	Time (ms)
5	5	2.8	1.15
8	5	3.2	1.21
5	8	2.3	1.38

CONCLUSIONS

In this paper the basic ideas of the Steering mechanism of the ATLAS HLT system and its current implementation have been presented. It has been show that the current implementation is able to cope with the execution of the seeded and stepwise reconstruction in the case of a realistically sized configuration.

ACKNOWLEDGMENTS

The authors want to thank the ATLAS TDAQ community support and collaboration in this work.

REFERENCES

[1] Athena framework: https://uimon.cern.ch/twiki/bin/view/Atlas/AthenaFramework

[2] The ATLAS Collaboration, "ATLAS High Level Trigger, Data Acquisition and Controls – Technical Design Report", CERN, Geneva, Switzerland, CERN-LHCC-2003-022, 2003.

[3] H. von der Schmitt et al. "A configuration system for the ATLAS Trigger", these proceedings

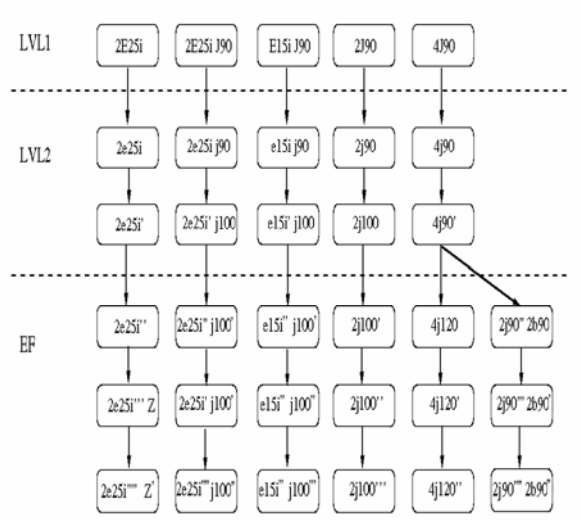

Figure 3: example of a HLT configuration where signature chains are only allowed to split and never to merge. Changes to pre-scales or cuts in a signature chain only affect the trigger response for that particular physics signature

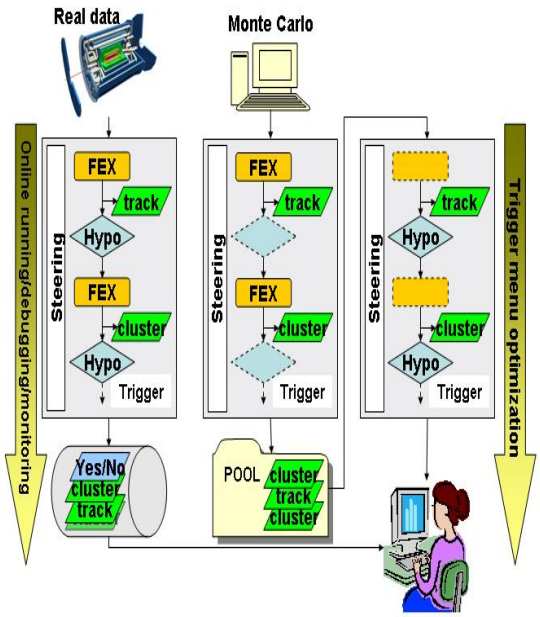

Figure 4: An example of how the exact same Trigger code and configuration can be used in an online and in an offline environment without any changes