INTEGRATION OF ATLAS SOFTWARE IN THE COMBINED BEAM TEST

M. Dobson, CERN, Geneva, Switzerland, on behalf of the ATLAS Collaboration

Abstract

The ATLAS collaboration at CERN operated a Combined Beam Test from May until October 2004. Collection and analysis of data required integration of several software systems that are developed as prototypes for the ATLAS experiment, due to start in 2007. Eleven different detector technologies were integrated with the Data Acquisition system and were taking data synchronously. The DAQ was integrated with the High Level Trigger software, which will perform online selection of ATLAS events. The quality of the data was monitored at various stages of the Trigger and DAQ chain. The events were stored in a format foreseen for ATLAS and were analyzed using a prototype of the experiments' offline software, using the Athena framework. Parameters recorded by the Detector Control System were recorded in a prototype of the ATLAS Conditions Data Base and were made available for the offline analysis of the collected event data. The combined beam test provided a unique opportunity to integrate and to test the prototype of ATLAS online and offline software in its complete functionality.

INTRODUCTION

Combined Test Beam

The ATLAS Combined Test Beam system combines many pieces of the ATLAS detectors and runs the system synchronously to take data in the same environment. The aim is to run as many detectors together in an environment closest to the one foreseen for the final system, in order to test and evaluate performance and functionality of the detectors, and test the combined performance of the system.

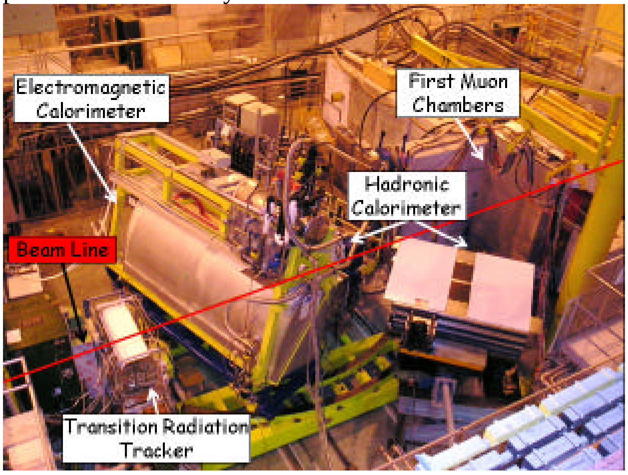

Figure 1: Test Beam detector setup

The Test Beam has been active for many years now and the number of detectors in the beam line has been increasing steadily. Previous Test Beam studies in ATLAS were using dedicated detector software. This year, most of the detectors are present and all are running with final or near final (prototype) hardware and software over a period of eight months.

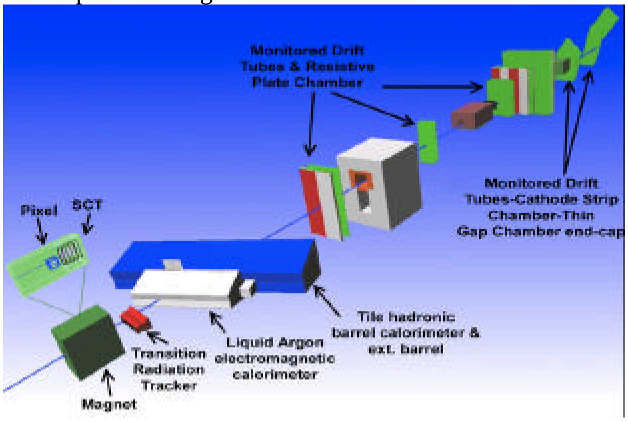

Figure 2: Geant 4 visualization of Test Beam

A photo of some of the detectors in Test Beam can be seen in Figure 1. Figure 2 shows all the detectors present in the Combined Test Beam, as modelled in the Geant 4 framework used for the simulation of events [1].

ATLAS Software Systems

Many software systems are present in the Test Beam setup and have been integrated to allow combined data taking and analysis. The central part of the software in Test Beam is the Data Acquisition (DAQ) software which enables many detectors to take data in a synchronised way and combine the data into files which use the ATLAS online Event Format. The DAQ software links to the other software systems for the data access and processing functions.

For the data input it must integrate with the detector readout hardware and software. For the Trigger part, it integrates with the Level-1 trigger via the same interface as the readout, and with the High Level Trigger (HLT) software for the upper two levels of the trigger system, namely Level-2 Trigger (LVL2) and Event Filter (EF). In turn the HLT software uses many parts of the Offline Software [2], itself responsible for the analysis of the data in the offline framework, with access to full geometry and calibration data. The Offline software also contains all the functionality for event simulation.

TDAQ SYSTEM

Overview

The Trigger and DAQ (TDAQ) system of ATLAS [3] is illustrated in Figure 3. This shows the flow of data from the detector readout at the top of the diagram, to the Mass Storage system at the bottom right.

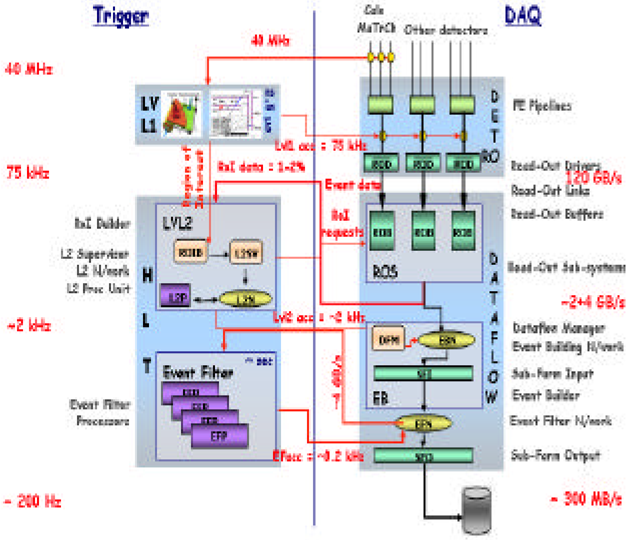

Figure 3: The TDAQ system

The data come from the detectors and are stored in Front End Pipeline memories while the Level-1 trigger (hardware) takes a decision. This is based on data from the calorimeter and the Muon trigger detectors only. Data for only those events accepted by the Level-1 trigger are then passed to the detector Readout Drivers (RODs) and then readout into the Readout Buffers (ROBs) of the Readout System (ROS), the first element in the TDAQ chain. In parallel the regions of the detector which contain interesting physics, Regions of Interest (RoI) as found by the Level-1 trigger, are passed to the Level-2 trigger system supervisors (L2SV) through the Region of Interest Builder (RoIB). The L2SV assigns each event to a processor (L2P), which will request data for the RoIs only (approximately 1-2% of the total event data) and will decide to accept or reject the event, decision which is returned to the supervisor, who passes this onto the DataFlow Manager (DFM).

The DFM is responsible for assigning events to SubFarm Inputs (SFI) who will request the data from all the ROSs and build the event. Once built, it is available for processing by the Event Filter (EF). An Event Filter Processor (EFP) will request an event from the SFI, and analyze the data. The decision to accept/reject the event is sent back to the DFM and the event data sent to the SubFarm Output (SFO). Once the data is in the SFO, it is then exported to mass storage for later processing in the offline environment.

All components of this system are used in the Combined Test Beam system.

TDAQ in Test Beam

The various components of the TDAQ system which are present in Test Beam include elements of the LVL1, the RoIB, the L2SVs and L2Ps, the DFM, SFI, SFO, and the EF farms. One of the EF Farms is local in the Test Beam, another one is semi-remote and located in the Computer Centre, and one or more are remote farms, connected via gateways in the Computer Centre.

All these components represent the biggest possible system in the Test Beam, and although this is the goal to achieve during this period, not every component will be integrated with the others at all times.

INTEGRATIONS

Detector / DAQ Integration

The DAQ software application providing a uniform interface to detectors is called ROD Crate DAQ (RCD), and this usually runs on the Readout Driver (ROD) Crate processor.

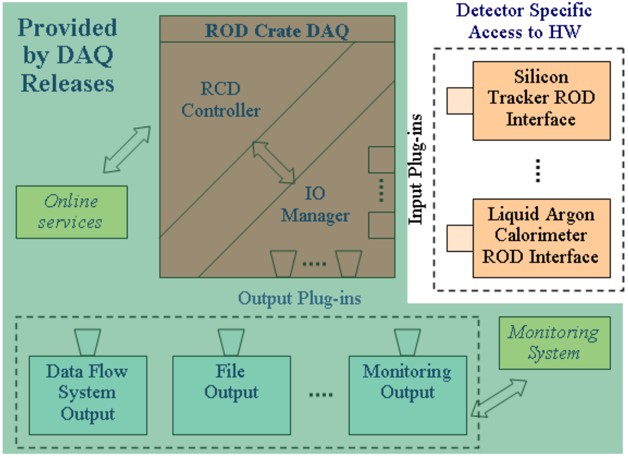

Figure 4: ROD Crate DAQ and detector interface

It is composed of two distinct processes, the RCD controller and the RCD IO manager, as illustrated in figure 4. The former deals with talking to the Online Services (such as Control, Information exchange, Error reporting) and the latter deals with the Input and output functions provided by a plug-in mechanism. This allows the dynamic loading of plug-ins at run time according to the configuration parameters of this application in the Online Configuration Database. Two of the available plug-in types are Input and Output. The first one deals with getting the data into the system from the different detectors. The second one deals with the output of data from RCD, which includes sending the data to the ReadOut System, and to the monitoring.

The detectors have only to implement the very thin interface of the input plug-in in order to offer their data to the system. There are only three methods to implement, one to configure the detector HW, one to find out available data and one to request the data. A set of guidelines and rules for detectors to follow, have made the detector integrations straightforward.

DAQ / HLT Integration

The framework of HLT applications are DAQ based, and use all the underlying DAQ services for data transfer, control and information exchange.

The HLT releases are synchronised and built against both DAQ releases and Offline releases. The latter requirement is because the HLT uses many parts of the Offline software as explained later.

DAQ software and Offline software have some core software in common, which is used throughout, for example the Event Format library.

HLT Gatherer for Monitoring

Monitoring tasks can monitor event data at various levels in the Readout, Trigger and Offline elements of the system. These Monitoring Tasks produce information (for example histograms) about certain features of the events.

When multiple elements produce monitoring information which are the same and one wants to view this information merged over all the elements producing it, there is a need to gather this information and make it available. This kind of task is particularly useful for gathering histograms from the different processing elements of a farm and showing the joint status of the information. To this end the Monitoring Gatherer was developed by the HLT to do this function. The Gatherer is able to serve the combined information to a variety of monitoring applications such as a Monitoring Display for the shift crew or some archive service for the monitoring information.

An illustration of this Gatherer and the Monitoring principle behind it is given in Figure 5.

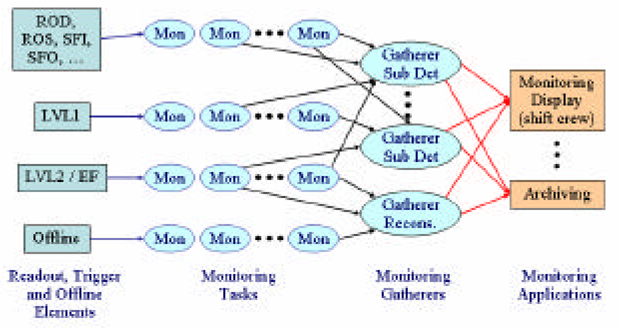

Figure 5: The Gatherer for combining monitoring information from multiple levels

HLT / Offline Integration

The HLT integrates Offline reconstruction and analysis algorithms and tools for data processing in its farms.

There are special requirements for running inside the LVL2, which are due to the inherent multi-threaded nature of their applications. Therefore any Offline code used in the LVL2 must be thread safe, and Offline developers need an environment to test this. To this end, a multi-threaded Athena (AthenaMT) environment was set up (see Fig. 6).

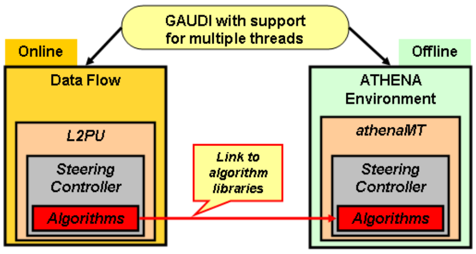

Figure 6: AthenaMT for Online / Offline development

Thanks to this, all the algorithms for the LVL2 were developed in the multi-threaded Offline environment then used as is in the LVL2. Due to the performance requirements of the LVL2 system the algorithms are developed specifically for it. In contrast, the EF algorithms are adapted Offline algorithms running in the Online environment.

BEYOND HLT AND BEFORE OFFLINE

After the HLT processing, the event data are sent to Mass Storage (CASTOR system) for later retrieval and analysis by offline processing.

Non event data, some needed for offline data analysis, are called Conditions data. They include such things as detector status and settings (from the Detector Control System), DAQ information, run information, and detector calibration and alignment. The Conditions data is currently written to an implementation of the Conditions Database based on MySQL.

Run bookkeeping information (for example beam energy and beam type) is copied from the Conditions DB to the ATLAS Metadata Interface (AMI) for cataloguing the available event data. The data stored in AMI is used for the pre-selection of suitable data prior to running reconstruction and analysis algorithms on it. AMI holds metadata for both Real data and Monte Carlo data for the Test Beam setup.

TEST BEAM MONITORING

In Test Beam, much monitoring (and reconstruction) was performed semi-online, that is in the Offline environment immediately after a run, as the monitoring and reconstruction algorithms were initially developed there. They have since been easily moved to the online monitoring in the EF, and used there when this was present in the data taking.

The work presented here is preliminary and much of the data was taken without prior detector calibration.

Standalone detector Analysis

This section will show only one reconstructed quantity (out of many) calculated for a single detector.

Figure 7 shows the reconstructed momentum as given by the Muon detector. The beam was composed of 120 GeV muons, and the momentum was reconstructed with a mean of 108 GeV, and a sigma of 4.4 GeV. The energy loss of 12 GeV seen by the Muon detector corresponds to the energy loss in the material (~ 6.5 m of Iron) in front of the Muon detectors. Indeed many other detectors are in front of the Muon system, notably the calorimeters, and also a beam stop just before the precision Muon chambers.

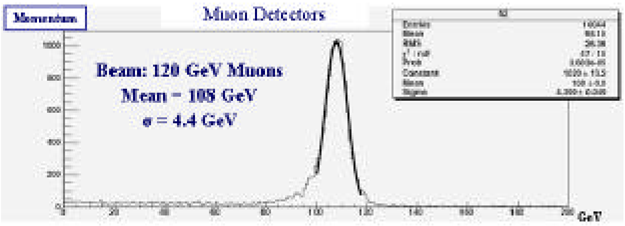

Figure 7: Muon Momentum Reconstruction (preliminary)

Combined Analysis

In addition to the many quantities monitored or reconstructed for individual detectors, many correlations between detectors are done, as well as combined reconstruction.

Figure 8, shows the energy reconstruction across both the Electromagnetic and Hadronic Calorimeters. The beam is made up of pions and electrons of 180 Ge V. The energies from the different detectors are added in the proportions shown in the figure. The factor of 0.9 is obtained by simple optimization. The detector response factor between electrons and pions has not yet been applied to the pion energy. This factor is about 1.5.

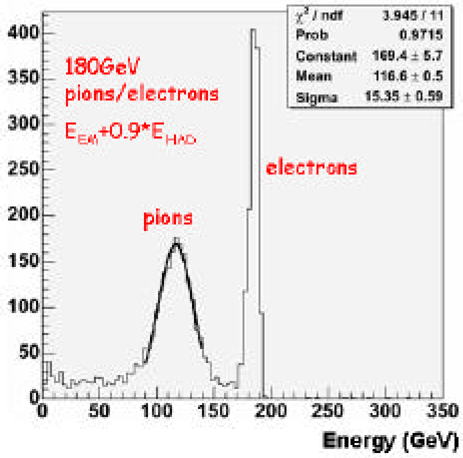

Figure 8: Combined Pion Energy reconstructions in the Calorimeters

Figure 9 shows a correlation of one of the track parameters as reconstructed in the inner tracking detectors (Pixel and Transition Radiation Tracker detectors) and in the Muon detectors (Trigger and Precision chambers). One can see that there is a good correlation of the two types of detector, which mean that the same tracks are being seen in the detectors, even if there is an offset between them, indicating that the detectors are not aligned in the same plane. These results are before detector calibration.

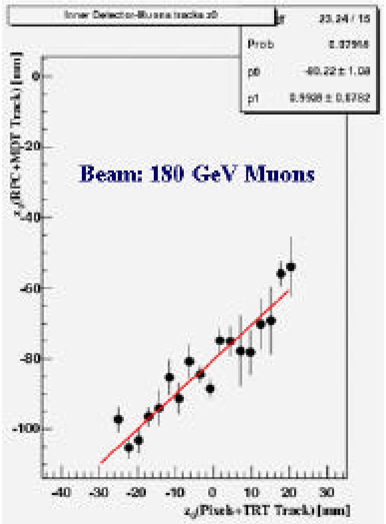

Figure 9: Correlation between z_0 of the track reconstructed in the Tracking detector and Muon detector (z_0 is the offset in z of the track at the x = 0 plane)

CONCLUSIONS

The Combined Test Beam has been the chance to integrate systems which would otherwise not always have been done. The integration of many software systems has been a success, especially as they are all semi-final software for ATLAS which is used in Test Beam, not special Test Beam software

Much data has been taken since the beginning of Test Beam and will be analyzed for some time to come (~ 1 TB data).

Throughout this exercise, lessons have been learned and will be fed into the participating systems for the forthcoming and very important stage, commissioning.

ACKNOWLEDGEMENTS

This paper has reported on very diverse work which has involved many groups within ATLAS. We would therefore like to thank the Detector groups present in Test Beam, the TDAQ group, the Offline Software group, in particular the event reconstruction and analysis groups working for Test Beam, and all the detector shift crews who have enabled the Combined Test Beam to take so much data and make it a success.

REFERENCES

[1] A. Rimoldi, “The simulation of the ATLAS experiment: present status and outlook”, CHEP04 proceedings, 2004.

[2] A. Farilla, M. Gallas, “Offline Software for the ATLAS Combined Test Beam”, CHEP04 proceedings, 2004.

[3] ATLAS Collaboration, “ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report”, CERN/LHCC/2003-022.