ATLAS ONLINE DETERMINATION AND FEEDBACK OF THE LHC BEAM PARAMETERS

J. Cogan*, R. Bartoldus, E. Strauss SLAC, Menlo Park USA
D. Miller, University of Chicago, Chicago USA
On Behalf of the ATLAS Collaboration

Abstract

The High Level Trigger of the ATLAS experiment relies on the precise knowledge of the position, size and orientation of the luminous region produced by the LHC. Moreover, these parameters change significantly even during a single data taking run. We present the challenges, solutions and results for the online luminous region (beam spot) determination, and its monitoring and feedback system in ATLAS. The massively parallel calculation is performed on the trigger farm, where individual processors execute a dedicated algorithm that reconstructs event vertices from the proton-proton collision tracks seen in the silicon trackers. Monitoring histograms from all the cores are sampled and aggregated across the farm every 60 seconds. We describe the process by which a standalone application fetches and fits these distributions, extracting the parameters in real time. When the difference between the nominal and measured beam spot values satisfies threshold conditions, the parameters are published to close the feedback loop. To achieve sharp time boundaries across the event stream that is triggered at rates of several kHz, a special datagram is injected into the event path via the Central Trigger Processor that signals the pending update to the trigger nodes. Finally, we describe the efficient near-simultaneous database access through a proxy fan-out tree, which allows thousands of nodes to fetch the same set of values in a fraction of a second.

INTRODUCTION

ATLAS is one of several large experiments situated on the Large Hadron Collider (LHC) at CERN [1]. To cope with the LHC's extremely high bunch crossing rate, currently ≈ 20 MHz, ATLAS makes real time decisions to accept or reject event data using a trigger system [2]. Several algorithms in the software trigger, such as vertex finding and b-jet identification, rely on specific knowledge of the luminous region to maintain a high identification efficiency for interesting physics events.

However, the parameters of the luminous region–the beam spot–change significantly during the course of an LHC fill. The width typically grows $\approx 3\ \mu m$ while the position can drift by $5 - 10\ \mu m$. To operate at maximum trigger efficiency, we measure and communicate the beam spot parameters to the trigger farm in near real time. We require over 100,000 vertices to measure the 1,300 colliding bunches in the current LHC structure. Running a vertexing algorithm on the trigger farm itself provides access to the large statistics needed at a high rate. Since each trigger process operates independently of the others, our challenge becomes coordinating them to measure the beam spot and then feed it back to each of them for use in tracking algorithms. This measurement and feedback is accomplished without disrupting data taking.

* jcogan@cern.ch

EXPERIMENTAL SETUP

While ATLAS is comprised of several sub detectors, we will focus on the silicon trackers and the trigger system. We use these systems to observe and reconstruct charged particle track and vertices.

ATLAS Silicon Trackers

The Pixel detector, comprised of three concentric cylindrical layers and three end-cap discs on each side, has an excellent hit resolution of $\sigma_{r\phi} \approx 10\ \mu m$ in the plane transverse to the beam axis, and $\sigma_z \approx 115\,\mu m$ in the direction of the beams. The Semi-Conductor Tracker (SCT), surrounds the Pixel detector and has four concentric cylindrical layers in the central region, as well as nine disks on each side. The SCT's silicon is shaped in strips and provides $\sigma_{r\phi} \approx 17\mu m$ and $\sigma_z \approx 580\mu m$ resolutions. The trigger system uses data from both detectors for tracking and vertexing, however the final vertex resolution is dominated by the Pixel hits. They cover an acceptance range of $|\eta| < 2.5$ in pseudo-rapidity, or about 10 degrees from the beam axis.

ATLAS Trigger

ATLAS employs a three level trigger system, one in hardware (L1) and two in software called the High Level Trigger (HLT). A detailed description can be found elsewhere. Here we restrict ourselves to the points related to the beam parameter determination.

Central Trigger Processor For our purposes, the L1 Central Trigger Processor (CTP) performs two important roles. First, the CTP tells the ATLAS Data Acquisition System what is the current LumiBlock (LB). A LB is a period of data taking with nearly equivalent detector conditions, often lasting 60 seconds. It is important that all the events in the same LB use the same beam spot for consistency checking. Second, the CTP has a small data fragment it adds to the data stream as the events are read into the HLT. This fragment contains information about the L1

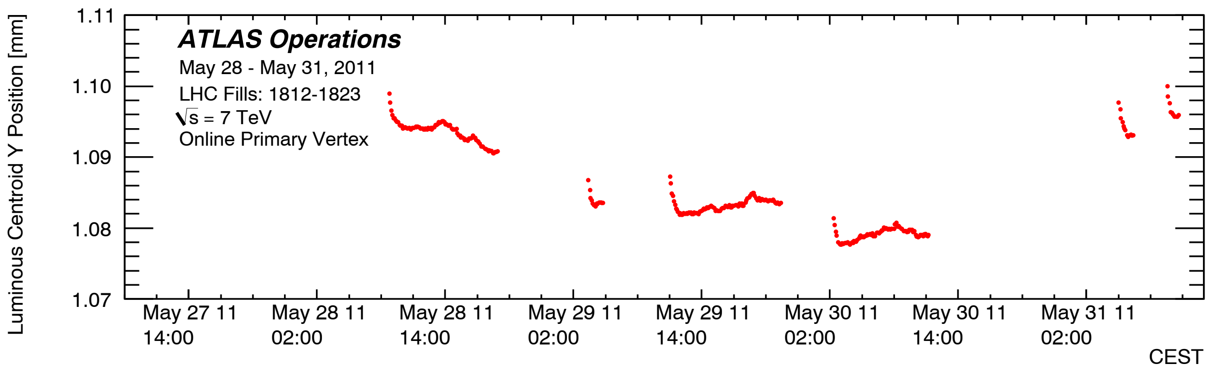

Figure 1: Time-variation of the luminous centroid position in y measured in the High Level Trigger every five minutes, for six separate LHC fills recorded over the span of four days.

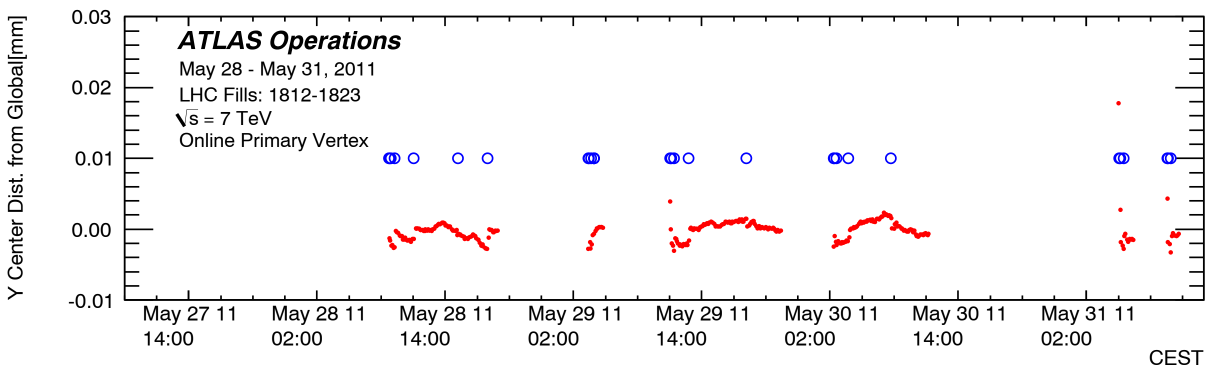

Figure 2: Time-evolution of the deviation of the luminous region position in y. The deviation represents the difference between the most up-to-date measurement and the value stored in the conditions database. The blue circles mark the time of the updates while the red dots show the deviation in millimeters as a function of time. Each red data point uses five minutes of data calculated on the High Level Trigger Farm, for six separate LHC fills recorded over the span of four days.

trigger decisions, the LB, and which set of beam spot parameters to use.

ATLAS High Level Trigger The ATLAS HLT comprises two software triggers: Level-2 (L2) and the Event Filter (EF). The L2 and EF sequentially decide to accept or reject events based on physics objects they have reconstructed. The HLT is run on commercially available server racks with more than 6,000 L2 and EF processors each.

The L2 trigger is the first level to access tracking information and is able to process approximately 75,000 events per second, only examining each event's respective Region of Interest. Due to the presence of tracking information and the high attainable rate of events, the beam spot algorithm runs on the L2 farm for all events that passed certain L1 multi-jet triggers.

The Beam Spot Algorithm

Events which pass the L1 trigger are considered by the beam spot algorithm if they contain sufficiently high quality vertices. When the algorithm processes an event, it calculates the position in x, y, z of the three vertices in the event with the highest number of tracks. Each of these positions is added to local histograms and is used to compute the center of the beam spot and its width (still convolved with the detector-resolution). To calculate the width of the beam spot we employ a split vertex method (described elsewhere [3]), whose primitives are also histogrammed.

METHODS

While building and maintaining this system presents many challenges in tracking and commissioning, we will focus on the hurdles related to the coordination and cooperation of the High Level Trigger Farm.

As we discuss the various components of the feedback system we will mention the latency each one introduces. We define latency to be the time between an event passing the L1 trigger and our first ability to execute an update based on that information. The total latency is approximately 5 minutes.

Gathering

As the beam spot algorithm runs, it accumulates statistics in the local histograms. First we must aggregate these histograms along with those from other applications running concurrently, across all 6,000 L2 processors. We accomplish this via a *Gatherer* tree, in which the leaf nodes are the trigger processors. Every 40 seconds, the ≈ 240 trigger processors in the same server rack send their histograms to the rack level gatherers which sums and stores them locally. At the next 40 second interval the ≈ 30 rack level gatherer send their histograms to the top level gatherer where they are summed to produce farm level histograms. From here the farm level histograms are pushed to a historical archive, where asynchronous monitoring applications existing off the L2 or EF farms can query them. To mitigate possible out-of-phase problems between the gatherer and the beam spot monitoring tool, we wait another 60 seconds, bringing the total latency at this step to 140 seconds.

Calculation

Once the histograms are summed across the L2 farm, they are analyzed to extract the beam spot parameters. This requires fitting Gaussian functions to the vertex position histograms to find the distribution mean and width, which correspond to the beam spot center and raw width respectively. The split vertex method is used to calculate the ATLAS vertexing resolution which is then deconvolved to produce a corrected width. Here we calculate both the beam-average parameters as well as per bunch measurements.

We log the calculated parameters, regardless of whether an HLT beam spot update was executed. The parameters are published to the ATLAS online conditions database (COOL) and the LHC logging database (TIMBER). The complete calculation and logging adds 15 seconds to the update latency.

Update Criteria

We do not update the HLT with each new set of beam spot parameters. While the system could handle the continuous feedback, it complicates the following data analysis and adds little extra precision. Thus, there are four criteria, any of which can trigger an update. Each criteria is based on the *current* and *nominal* beams spot. The *current* beam spot refers to the set of parameters just calculated from the HLT histograms. The *nominal* beam spot, on the other hand, refers to the beam spot parameters in use by the HLT. These criteria are completely configurable and were selected to optimize the performance of the displaced vertex triggers. They are as follows:

- The position difference (current to nominal) is greater than 10% of the width in any direction.
- The width difference is greater than 10% the nominal width in any direction.
- The statistical error on a parameter drops by 50%.
- The current beam spot has a valid status flag, while the nominal has an invalid flag. This flag is used to invalidate the beam spot after a beam dump.

The Central Trigger Processor

To update the parameters used by the HLT, the monitoring tool must first indicate to the 13,000 trigger processes that their beam spot is out-of-date. Furthermore, it is imperative that at any given LB the entire trigger farm is using the same set of beam spot parameters.

To accomplish the update, the monitoring application publishes the new parameters to a temporary network location and informs the CTP of the new beam spot.

The CTP writes the beam spot parameters into the conditions database with an interval of validity starting at the next LB, N. At the same time it updates its data fragment with the LB number of the next beam spot update. When the first event in LB N arrives the HLT nodes discover there is a new beam spot available from the information in the CTP fragment. Each trigger process then reloads the parameters from the conditions database. In this way, over 13,000 nodes (L2 and EF) are told there is a new beam spot, with a clean LB transition.

Since the CTP waits for the next LB this step adds 60 seconds to the update latency.

Proxy Tree

There is a final problem: how can 13,000 processes query the conditions database simultaneously? We have two advantages in this situation. Firstly, each process is asking the same question, "what is the beam spot for LB N?". Secondly, each process receives the CTP fragment (and thus queries the database) with some random time offset, $\approx 50\ ms$ relative to the other processes. This offset comes from the average event processing time.

To exploit these advantages we employ a database proxy tree built on the CORAL abstraction layer. Instead of directly querying the conditions database, the trigger process asks a database proxy which may or may not already have the response cached locally. If a proxy does not have the response it passes the request further up the tree. If the top level proxy does not know, it asks the CORAL server which queries the Oracle conditions database. As the response goes back down through the tree, each proxy stores the query-response pair locally and thus prevents unnecessary database access in the future.

Thus, only a few trigger processes need to wait the full round trip time to query the database; most nodes wait only for their nearest proxy to retrieve the answer from local memory. On average, each process only waits $\approx 10\ \mu s$. As the waiting time is short compared to the average processing time at L2, this induces no dead time on the HLT. Finally, the top level servers, CORAL and Oracle, are saved from attempting to field 13,000 requests.

RESULTS

The automatic beam spot determination and feedback has been in place since early 2011. In the following section we summarize its feedback performance and some of the physics results produced from its logging output.

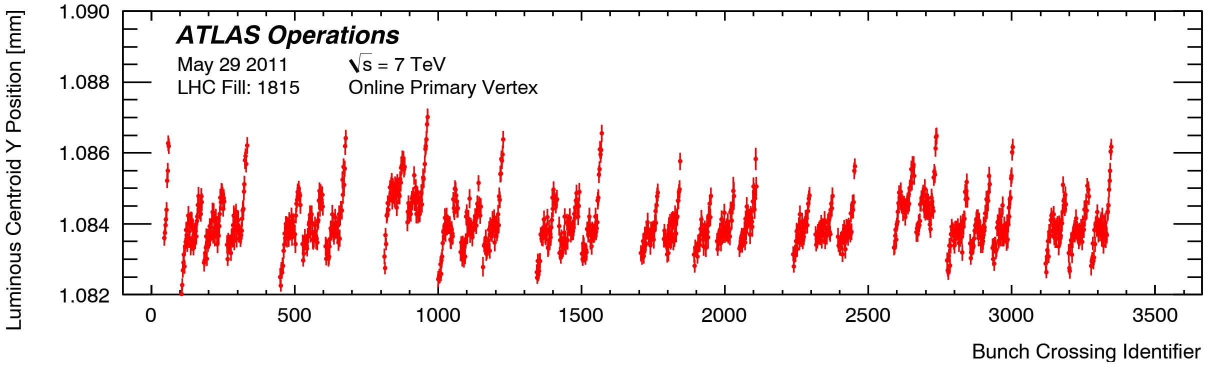

Figure 3: The luminous centroid position in y measured in the High Level Trigger for each of the 1024 colliding bunch pairs separately. Distinct structures are visible, with variations of up to 5 μm and repeating patterns across the injected bunch trains.

Feedback

Here we discuss four measures of the feedback mechanism: latency, frequency, system load, precision.

- Latency: As discussed in **Methods** section, the update mechanism has a minimum amount of time between an event being observed and an update it might trigger. The total latency is $\approx$ 4 minutes which is shorter than typical beam behavior and satisfies the trigger needs.
- Frequency: At the beginning of a fill the update mechanism first fires five minutes after collisions begin (because of latency) and then 4-5 times in the first 30 minutes due to the rapid decrease of statistical errors. After this early phase, beam orbit drift and emittance blow up are the most common causes of beam spot updates. These updates happen much less frequently, typically occurring only once every few hours.
- System Load: Once the HLT nodes receive the new CTP fragment, they must all fetch the new beam spot. However due to the proxy tree this causes only a 10 ms pause for each process on average. This load is so small that it induces no system dead time.
- Precision: As discussed in **Update Criteria** section the feedback mechanism maintains the difference between current and nominal beam spot parameters small by occasionally updating the HLT. In Figures 1 and 2, we see the current and nominal beam spots are kept within a few microns despite 30 micron variations in the current value over the course of a week.

Physics

The occasional updates of the beam spot used by the HLT keeps tracking-triggers operating at a high efficiency. In addition to the updates, the monitoring tools also record the parameters across time and for each colliding bunch separately. We provide time lines of the position and width in all three directions, and the tilt with respect to the beam axis of the luminous ellipsoid. Furthermore each value is available for offline analysis and stored with the statistical error from the fitting process.

The online system is uniquely positioned to perform the 1,300 per bunch calculations because of the high rate of good vertices available at L2. Capitalizing on the high rate of events, we calculate the center of the luminous region for each bunch to within $\approx 1\mu m$, which is better spatial resolution than the LHC dedicated hardware can obtain. Examining the bunch position variation within a train, shown in Fig. 3, we see unambiguous evidence for beam-beam kicks from long range collisions at the LHC. Long range collisions occur when a bunch in beam 1 passes near, but not through, a bunch in beam 2. Each bunch exerts an electromagnetic force on the other distorting its orbit and shape.

CONCLUSION

We have developed a system to calculate and monitor the LHC beam parameters using the ATLAS High Level Trigger Farm. Furthermore, if the nominal beam spot parameters used by the HLT tracking algorithms differ significantly from the current values an update is executed while the system continues to collect data.

We coordinate the communication to and from the 13,000 independent HLT processes by exploiting the gatherer and proxy systems already in place. Finally, the HLT's beam spot is maintained within microns of the current value while inducing no dead time on the data taking system.

REFERENCES

[1] ATLAS Collaboration "The ATLAS Experiment at the CERN Large Hadron Collider," *JINST*, vol. 3, p. S08003, 2008.

[2] ATLAS Collaboration "Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics," no. CERN-OPEN-2008-020, 2009. `http://cdsweb.cern.ch/record/1125884`

[3] ATLAS Collaboration "Characterization of Interaction-Point Beam Parameters Using the pp Event-Vertex Distribution Reconstructed in the ATLAS Detector at the LHC," ATLAS-CONF-2010-027 `http://cdsweb.cern.ch/record/1277659/`