ATLAS HLT Rate Prediction and Resource Estimation Tool

Bruno Balthazar, 30/08/2013

Supervisor: Yu Nakahama

Abstract

My project comprised developing a tool for estimating the ATLAS High Level Trigger event rate for the LHC 2015 upgrade. This uses the data taken during the 2012 LHC run and scaling factors based on Monte Carlo samples.

Having finished that, my project was further extended to develop the tool to estimate the resource used for the primary chains in the trigger menu.

Introduction

One of the problems in data taking during the LHC run is the limited storage capacity. For ATLAS, this rate is limited to 1000 Hz at High Level Trigger (HLT, see below). This means that we should store only the events that are actually interesting from the physics point of view. This is where the trigger comes in.

The trigger menu consists of a series of triggers that select which events to keep and which to throw away. This can consist of selections based on energy (threshold energy), eta or phi (the objects direction), among others. In the ATLAS, the trigger consists of mainly two parts: the Level 1 (L1) trigger, and the High Level Trigger (HLT), which is itself subdivided in the Level 2 trigger (L2) and the Event Filter (EF).

The L1 trigger selection is done by the hardware, and so is done online and is very fast. The HLT selection is done offline, by software.

The LHC upgrade is going to increase the colliding protons energies from 8 TeV to 14 TeV and the luminosity (among others). This means we expect a significant increase in the event rates, so a new trigger menu and rate estimation are needed.

This comprised my first task, and to do it I adapted a tool already developed for the L1 rate estimation and included HLT signatures, using for that Monte Carlo samples to scale the 8 TeV data to 14 TeV, the energy of the LHC upgrade. I will describe the method further in the Method section below.

My second task then consisted of developing a tool to estimate the total time taken by the algorithms in the trigger menu. This is important since we only have a limited amount of CPU resource available, and the algorithms that are used should be such that they reject as much of the uninteresting events as possible, while also keeping the run time low, since data is being taken at a very fast rate. It is also particularly important for this second run because the ATLAS group is planning to use complex algorithms to move from offline ones to trigger level ones, which have a better resolution and higher rejection power. This requires a good compromise between.

We would like to use this tool to predict the time of the algorithms for the LHC upgrade, and also to include HLT chains and to be able to quickly modify the trigger chains to test which ones are most useful. To do this, I used the information from the times taken for each algorithm as well as the rejection factors for each, and applied this in the 8 TeV sample. This is further discussed in the Method section below.

Method and results

1. HLT rate estimation tool

Since at Level 1 there are only a few signatures, extending the rate estimation tool to HLT required introducing more signatures and cuts, for example isolation cuts, quality cuts (loose, medium, tight etc.), among others.

We also needed to include some L1 signatures, because some HLT triggers required the corresponding L1 trigger to simulate the L1 cut. This improved our results significantly specially at low pT, where some inefficiency was expected (particularly for electrons and muons).

At 8 TeV, we have measured the rates and we can compare these to Monte Carlo (MC) samples. These can have jet, W, Z, ttbar components, so we need to include them all, with the appropriate weight, to obtain the full rate. To understand this I compared the rates at 8 TeV from MC samples with those measured during the 2012 run, obtaining good agreement. Plots of the results at 8 TeV are shown below, figure 1.

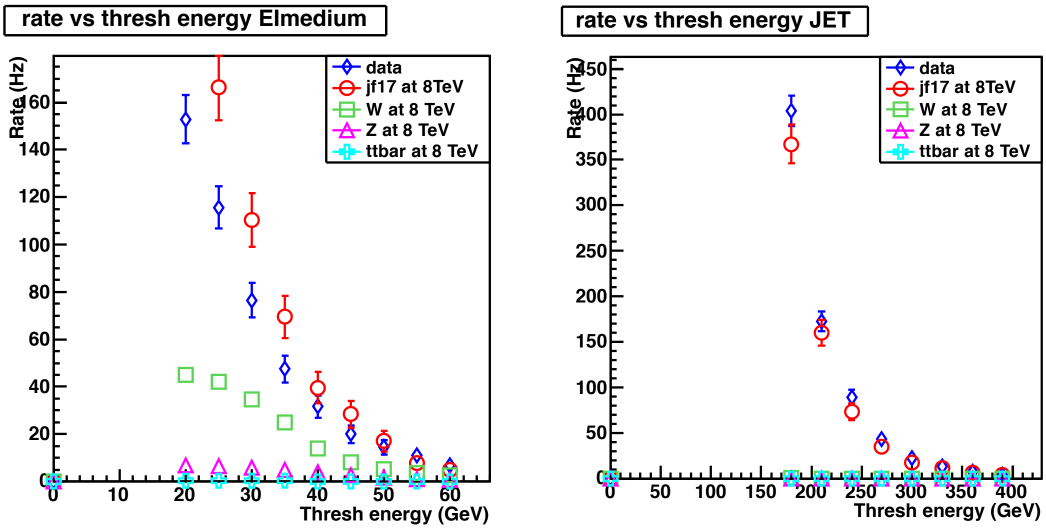

Figure 1: Plot of rate versus threshold energy for the data sample and different MC samples: W, Z, ttbar and jf17 (which includes jets and in this case also the contributions from the other MC samples). The left plot is for the eimedium trigger, and the right one for the jet trigger

Since there are some differences between the MC and data samples, we quantify this difference in a parameter called the mcfudgefactor, which is the approximately constant ratio between the rates for the MC sample to the data sample. Plots of the mcfudgefactor versus threshold energy for different signatures can be seen below, figure 2.

Similar plots were made for the scaling factor from 8TeV to 14TeV for the MC samples, figure 3. These also show a dependence on the threshold energy.

Finally, we extrapolated the data sample from 8 TeV to 14 TeV, using the scaling factors obtained from the MC samples. These rates are shown in the plots below in figure 4 These results are very valuable in the sense that we can now quickly estimate the rate needed by a trigger menu, and will be used by the official trigger menu group.

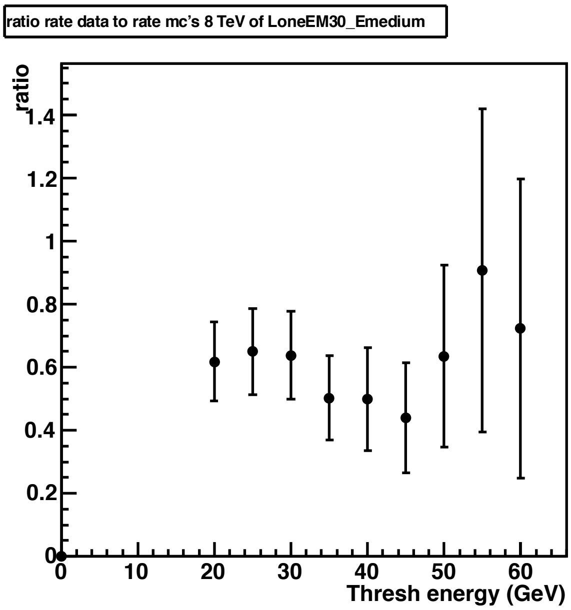

Figure 2: Plot of mcfudgefactor versus threshold energy, i.e., the ratio of data rate to MC rate at 8 TeV, for the emedium trigger (with L1 trigger simulation included)

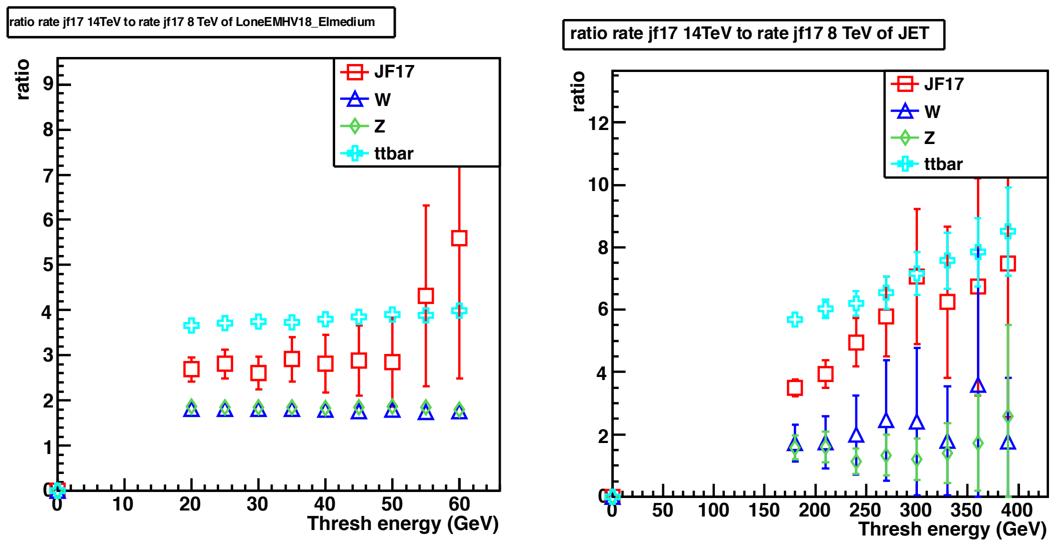

Figure 3: Plot of scaling factor for MC samples from 8 TeV to 14 TeV for different MC samples: W, Z, ttbar and jets (jf17, which doesn't include the other MC samples in this case). The plot on the left is for eimedium trigger (with L1 trigger applied), and the right plot is for the jet trigger

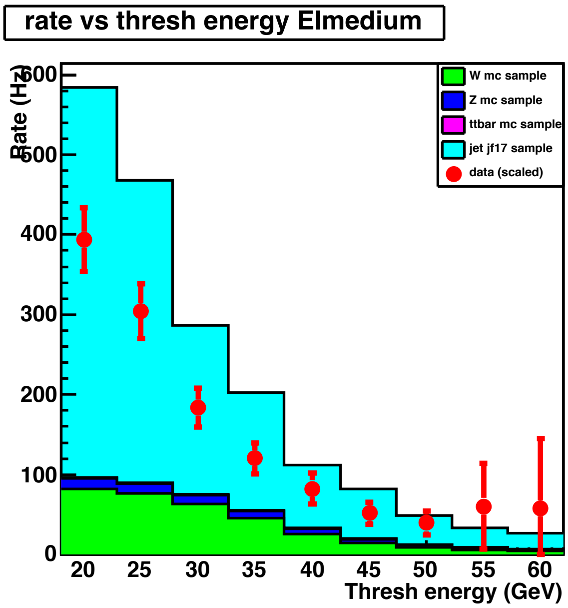

Figure 4: Plot of rate versus energy threshold for the eimedium trigger. In red dots the rates obtained from scaling the data from 8 TeV are plotted. The histograms show the data obtained from the corresponding MC sample at 14 TeV.

2. Resource estimation tool

This consisted of estimating the resource usage for Level 2 algorithms. A trigger chain consists of many steps, in which there are different sequences. Each step usually comprises a part of the whole trigger selection. For electrons and jets for example, the L1 trigger is followed by calorimeter clustering, then calorimeter selection, followed by tracking and then cluster-track matching. These sequences are divided usually into 2-3 steps. In each sequence, algorithms are called to perform the desired action.

When an event passes the L1 trigger(s), it seeds L2 trigger chains that are performed one at a time at each step. So at each step, the trigger makes the decision to reject or keep the event. The time to make the decision is then added to that chain's run time. After making the decision, it goes to the same step for the next chain, and when the last chain is run, it moves to the next step of the first chain. If a trigger chain rejected the event in the previous step, then it no longer runs this step, but if it decides to keep the event, it will run this step too and so the time for this step to run is added to the trigger chain's run time. It is also important to note that the same chain may be triggered multiple times in an event by different objects (e.g. e24vhi_tight1 can be triggered by two different electron objects that have threshold energies above 24GeV), and hence comprise independent chains.

To obtain the resource usage I needed the rejection factors for each trigger chain at each step, and also the time at each step. This however was not available, neither was the time per sequence call, so I had to sum them from the algorithms call time. This means that to obtain the time at each step, I had to sum the times of each algorithm to obtain the sequence time, and then sum the times at each sequence.

The rejection factors per sequence were obtained from online steering histograms, the algorithms in a sequence were found from TMXML files, and the sequences comprising a chain as well as the algorithm call time were obtained from the cost-monitoring webpage.

I proceeded to write all this information to external .csv files, which I kept as external input files for the tool. I developed a tool to read from these files chains specified by the user and with the information from these files I could develop an automatic tool that calculates the total time for each chain as well as the total time for an event. I focused on non prescaled triggers since they are simpler and include all the primary triggers. Some plots are shown below in figures 5 and 6, and an estimation of a trigger menu with primary triggers is shown in table 1.

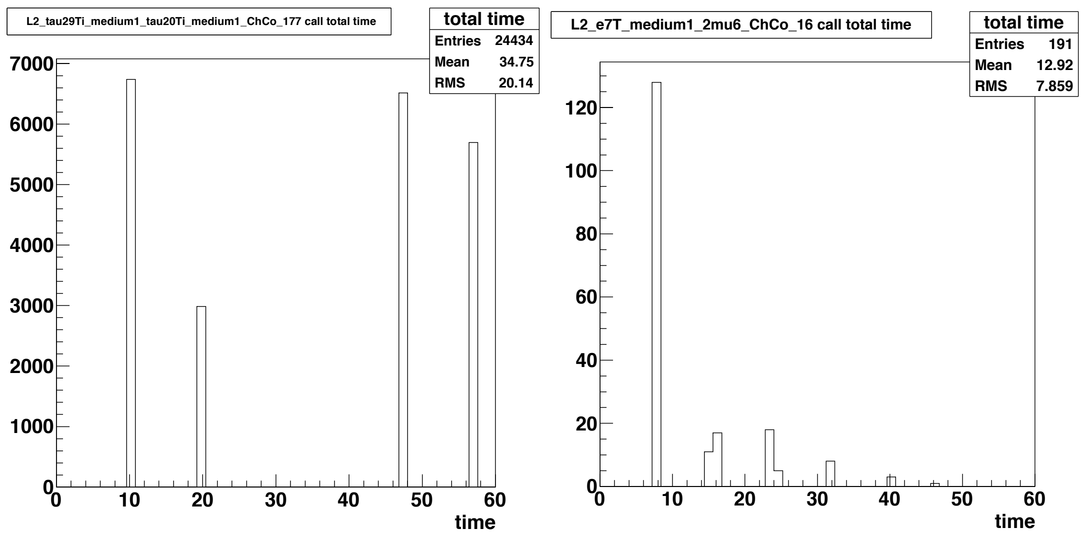

Figure 5: Histograms of number of events and the time spent in each trigger chain (in ms). The left histogram is for the L2 tau29Ti_medium1_tau20Ti_medium1, whereas the right one is for the L2 e7T_medium1_2mu6

L2 trigger	Mean time (ms)	Active Events
L2_vtxbeamspot_FSTracks_L2Star_B_ChCo_2053	1010.9	1726
L2_2mu6_Bmumux_ChCo_165	132.0	1329
L2_tau29Ti_medium1_tau20Ti_medium1_ChCo_177	40.9	24434
L2_tau20_medium1_mu15_ChCo_837	24.7	3493
L2_tau38T_medium1_xe40_tight_ChCo_189	21.8	2309
L2_mu18_tight_2mu4T_Jpsimumu_ChCo_873	20.0	10970
L2_tau38T_medium1_ChCo_474	14.9	52153
L2_e7T_medium1_2mu6_ChCo_16	12.9	191
L2_mu24_tight_ChCo_944	12.8	10970
L2_b40_tight_4j40_c4cchad_ChCo_752	12.8	1726
L2_b15_medium_4j15_a4TTem_ChCo_775	11.8	1726
L2_4j20_a4TTem_ChCo_339	11.3	1726
L2_mu18_tight_e7_medium1_ChCo_542	10.2	10970
L2_g_nocut_hiptrtL2_ChCo_992	8.9	36521
L2_j165_c4cchad_ChCo_758	8.7	3764
L2_b140_loose_j140_a10TTem_ChCo_726	8.3	3764
L2_2tau38T_medium1_ChCo_931	8.0	11464
L2_j75_c4cchad_xe55_ChCo_59	6.3	656
L2_tau125_medium1_ChCo_257	6.3	10029
L2_e24vh_medium1_e7_medium1_ChCo_789	6.1	36521
L2_xe55_ChCo_658	5.1	682
L2_e24vh_medium1_EFxe30_ChCo_373	4.2	36521
L2_e24vhi_medium1_ChCo_178	4.1	36521
L2_2e12Tvh_loose1_ChCo_719	3.3	11808
L2_g120_loose_ChCo_536	3.0	13157
L2_2g40_loose_ChCo_160	2.9	13157
L2_e24vh_loose0_3j15_a4TTem_ChCo_704	0.5	36521

Table 1: Mean time and active events for a trigger menu including the primary triggers, as well as a calibration trigger, L2_vtxbeamspot_FSTracks_L2Star_B_ChCo_2053 . Note the very high mean time of this trigger chain.

I have ignored caching (storing information on the algorithms already called) because this makes the whole process a lot more difficult. To avoid this problem one could use only one trigger chain for each significant group, i.e., to avoid trigger chains that wouldn't be processed.

I didn't have time to extend this to 14 TeV using minimum bias samples, or to include event filter (EF) to obtain the full HLT, however the framework was already developed, so modifying it to include these shouldn't be much more complicated.

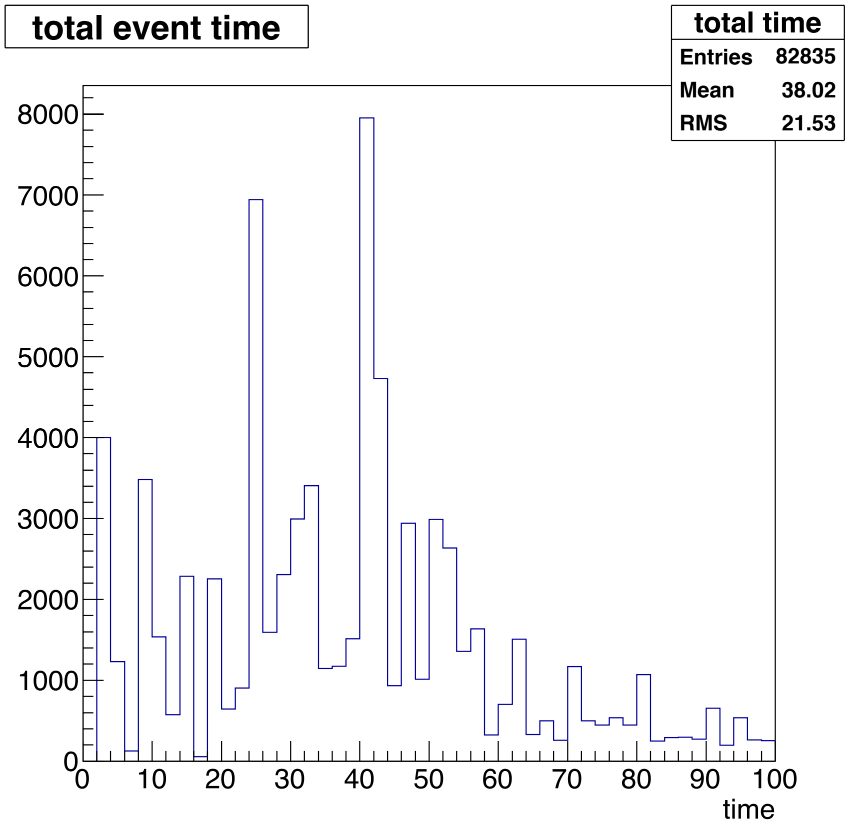

Figure 6: Histogram of number of events and the time spent in any trigger chain (in ms). This shows the amount of time each event spent in the L2 trigger.

Further discussion

The HLT rate estimation tool was developed to a very high level (while still keeping it simple, fast and very useful). However during the time of testing/implementation the MC samples available were not the most significant ones for this study, since they were not available at the time. In the future we plan to filter these samples again to further improve the reliability of our results. It should also be noted that HLT rates are particularly difficult to estimate, and we still obtain very good agreement (at least at 8TeV where we can compare with the data taken). Further signatures could be included if needed, even though we already included the primary and most used ones.

The resource estimation tool already has most of the framework implemented and is already a very useful tool, but could have potential improvements. These include b-jets and b-physics triggers, and also non prescaled triggers, in particular calibrating ones. This is because even though they aren't primary triggers, they are the ones that require most CPU time, and hence in order to compare our results with the cost monitoring webpage we need to include them. Also, as mentioned, one should take into account caching, either by including it in the tool (somehow), or by only including one trigger chain for each cached group, as mentioned above. Finally, we also need to extend this to include 14 TeV minimum bias samples and HLT trigger chains.

I also feel that it should be emphasized how complicated the trigger for the ATLAS is, and how difficult any kind of estimation of the sort I performed is, due to the amount of data being taken.