Iterative jet finding in the ATLAS trigger for the HL-LHC

Lorena Bucurú, Carlos Buitrago

6th COMHEP Talk Transcript

When moving out, choosing which things to take with you and which things to leave behind can be a very difficult decision, since there is no algorithm in our brain that chooses for us which things are worth to pack and which ones are not. These kind of algorithms exist however, in the trigger of a particle detector. These are useful to decide which events are of interest and which events we want to keep. Finding which events are of interest is a great and demanding challenge at hardware and software level, even more when taking into account the HL upgrade which will be implemented in the ATLAS experiment in the future.

In jets we can find clues of new physics, therefore jet identification is extremely important. Although the jet finding trigger is based on a fixed cone algorithm at hardware level, the algorithm to be implemented in the global trigger of the HL upgrade is more sophisticated, based in a simple and robust process, backed up on the iterability of the standard anti-kT algorithm. This global trigger was developed with the objective in mind of producing jet findings at online level with the highest possible precision and efficiency, so that the overlap between trigger data and analysis selected data is maximized, all of this while taking into account the resource and time limitations given by the detector.

An important factor to consider for the new algorithm are the physical signals that it has as objective signals. The algorithms used in the current trigger work good enough for events with one isolated jet, however they leave a lot to be desired in terms of multi-jet signals, and specially when the jets are spatially close together. The anti-kT algorithm on the other hand works very well with these kind of signals, which are our objective signals. It has to be noted however, that the new algorithm should work to reconstruct single jets, but it is to be expected that it's performance is not a lot better to that of the ones that are present in the current trigger.

The next question then arises: "Why can't the standard anti-kT be implemented on the trigger at the hardware level?" That is mainly due to it's iterative nature, which doesn't allow the algorithm to be run on the FPGAs present in the ATLAS calorimeter under latency restrictions. The basic functioning of the anti-kT algorithm goes as follows: the algorithm takes as inputs a list of objects, with these objects the distances d_{ij} (distance between the object i and the object j) and d_{iB} (distance between the object i and the beam). If d_{ij} is smaller than d_{iB} the algorithm recombines the objects i and j into a new object of which a new η, ϕ and p_t are calculated. On the other hand if d_{iB} is smaller than d_{ij} the algorithm marks i as a jet. The recombination process of i and j is known as the "E scheme" and it is because of the iterative calculation of η and ϕ that the standard anti-kT is not suited to run on a FPGA. This can be solved by changing the recombination method to the "Winner Takes All scheme", in which the η and ϕ of the new recombined object is just the value of η and ϕ of the object with higher p_t between i and j.

The algorithm uses as input the topoclusters in the calorimeter, which are adjacent cells in the calorimeter that surpass certain energy threshold. Another latency issue that must be solved is the fact that the number of inputs for the algorithm is too high in a event of interest. The solution to this problem is to define certain regions of interest (RoIs) on which jets are prone to appear and on which the algorithm must be run. These RoIs are defined using the sliding windows algorithm, which finds local maxima on a grid of data. The input that this sliding windows algorithm uses to define the RoIs are the gBlocks, which are groups of 3x3 gTowers (regions of 0.2 x 0.2 in the calorimeter barrel, and bigger in the endcaps). These gBlocks are chosen as the input of the sliding windows since they can be obtained before the topoclusters and they give us information about the energy stored in the calorimeter cells.

The general process goes as follows: RoIs are defined from the identification of local energy maxima, which are ordered from highest to lowest energy. The topoclusters around these local maxima are collected and then the anti-kT algorithm is run using these topoclusters as input. Finally the overlap is removed, so that if a jet is reconstructed in more than one RoIs it is only assigned to the region with the most part of its energy.

Our work (in progress) is to verify the efficiency of the trigger, to study the benefit of the RoIs and to check methods to reduce the jet finding time such as limiting the number of iterations done by the algorithm exploiting the fact that the topoclusters are organized by their energy. In the following slides the trigger efficiency is plotted in terms of the p_t of the 4th jet for three different p_t thersholds. In the first plot the full scan version of the algorithm is used, this means that the algorithm is run over the whole calorimeter (RoIs are not defined). In the second plot four RoIs are defined.