Online Cluster-Finding Algorithms for the $\bar{P}ANDA$ Electromagnetic Calorimeter

M. Tiemens[a], M. Kavatsyuk[a], P. Schakel[a]
[a]*KVI-CART, University of Groningen, The Netherlands*
on behalf of the $\bar{P}ANDA$ Collaboration

***Abstract*—In this work, the performance of a proposed algorithm to search for clusters in the $\bar{P}ANDA$ electromagnetic calorimeter (EMC) in real time, which provides vital information for the online event selection procedure, is discussed. Two implementations will be discussed and compared to each other and to a third, less suited algorithm (at least, as far as online usage is concerned). After this, the implementation in the readout hardware and concepts for the following data collection network will be presented.**

I. $\bar{P}ANDA$ Experiment

A new experiment, called $\bar{P}ANDA$ ($\bar{p}$ ANnihilations at DArmstadt), is being developed to perform precision measurements in the energy range between 1.7 and 5.5 GeV/c^2, where Quantum Chromodynamics becomes non-perturbative for charmonium systems[1]. Over the last years, several new charmonium-like hadrons were discovered, called XYZ states [1], possibly consisting of more than 3 quarks. The $\bar{P}ANDA$ spectrometer will be able to perform precision measurements on the properties of these and other predicted, as-of-yet unobserved, exotic states of matter, such as glueballs, which can be directly populated through the proton-antiproton interactions. However, the production cross section for those states is five orders of magnitude lower than that of conformal states [2]. It is furthermore assumed that they exhibit a similar event topology, rendering a conventional triggered readout unusable. To solve this, i.e. to make an event selection, the experiment features an advanced, intelligent readout system, that tries to reconstruct detected decays online.

The main focus will be on one subsystem of the detector, the electromagnetic calorimeter (EMC). The device consists of four parts: a Barrel, containing 11,360 $PbWO_4$ scintillation crystals and a Backward and Forward Endcap, containing 524 and 3,856 crystals, respectively. The faces of the crystals are approximately 2 cm $\times$ 2 cm. This is complemented by a sampling (shashlik) calorimeter, placed in the forward spectrometer a few meter downstream of the interaction point, to cover very small polar angles.

To be able to perform the online event reconstruction, complete information on the final-state particles is required. Some common particles, like photons and electrons (and positrons), are reconstructed using input from the EMC. Three algorithms to perform these reconstructions will be evaluated. Two of these are eligible for implementation on the online platform, providing vital information for the event selection.

II. Cluster Finding

Particles hitting the EMC crystals deposit their energy by creating an electromagnetic particle shower. Momentum conservation causes the shower to spread out over multiple crystals, leading to the formation of clusters. To find the four-momentum of the impinging particle, which is needed to perform the online event reconstruction later on, it suffices to add the individual hits in a cluster. However, the high interaction rate, combined with the large variety of intermediate states that can be directly formed, having different decay times, may lead to pile-up and event mixing. These features complicate the task of assigning hits to the correct clusters considerably. As these features occur in the time domain, the timestamp of each hit will play a key role in disentangling the hits. For this reason, the simulation that has been designed to reproduce this structure is called *time-based* simulation.

The distribution of events through time follows a Poisson distribution, with the mean time between two events determined by the interaction rate. This creates a bunched structure in the final data stream. The size of the bunches, called *timebunches*, can be controlled by a time threshold. Tuning this parameter can help put hits from a single (or few) event(s) into a single timebunch, aiding the assignment of hits to clusters. If multiple events are present in a timebunch, their corresponding hits are likely located in different parts of the detector, because apart from being forward boosted, the decays are isotropic. There are multiple methods under development to search for clusters in these timebunches, but the main aim is to develop a method that can be easily implemented to process the data online. As the data in the EMC is produced at a rate of 80 GB/s, this algorithm needs to consume as little resources as possible. These are the methods under consideration:

1) *Default Cluster-Finding*: The currently implemented version in the offline software package PandaRoot[3] takes the stream of hits (pre-selected from the same primary interaction), and treats each new hit as a seperate cluster, unless it neighbours[2] to an existing one. In that

Manuscript received December 9, 2016.

[1]Charmonium is the bound state of a charm and an anticharm quark

[2]As the time domain is important, in all references, "neighbour" means close in space *and* time.

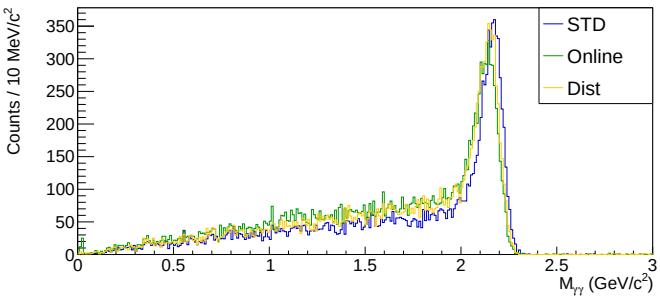

Fig. 1: Comparison of the number of reconstructed events for the three cluster-finding methods, obtained by taking all combinations of two photon candidates.

case, the hit is absorbed in that cluster.

2) *Online Cluster-Finding*: The first proposed version to be eligible for online usage loops over all pairs of hits in the input stream to establish neighbour relations between them. It then uses this information to merge them into clusters. The input stream consists of complete collected data from a given short time period, e.g. 40 μs.
3) *Distributed Cluster-Finding*: The data concentration stage already has full access to the calorimeter data for a subsection of the EMC (about 128 crystals per device). Making use of the advanced processing capabilities of the readout hardware, it is possible to search for clusters at this stage, reducing the load at later stages. The online cluster-finding method will be used to identify clusters. As clusters may be split at the edge of such a section, the algorithm will make so-called preclusters containing only minimal information on the location and size. In a later stage, it will be checked if the preclusters need to be merged using that information, and then the final clusters will be formed.

These methods are tested using several datasets, generated using the time-based framework. The first test case is 5000 instances of $p\bar{p} \to \gamma\gamma$ at an antiproton beam-momentum of 1.5 GeV/c at an interaction rate of 20 MHz (which is the worst-case scenario in terms of pile-up and event mixing). As can be seen in Figure 1, they exhibit a comparable performance in this case. The two methods for online cluster finding show a slightly depleted peak, because the probability to split a cluster into more smaller, low-energy clusters is higher for those algorithms. This is partially corrected in Distributed Cluster-Finding, because the use of radii for the preclusters allows to recombine some of them.

As speed is key, also the processing time is compared. No solid conclusions can be drawn, however, because this test was performed on a CPU, and the final algorithm will run on an FPGA[3]. The performance likely differs on such a device. The Distributed Cluster-Finding algorithm is currently being implemented on an FPGA in a protoype of the EMC readout hardware (see III), but has not yet been tested. Hence, at this moment, the comparison on CPUs is the best that can be done.

[3]A Field-Programmable Gate Array (FPGA) is an integrated circuit containing an array of programmable logic blocks, connected by a collection of reconfigurable interconnects.

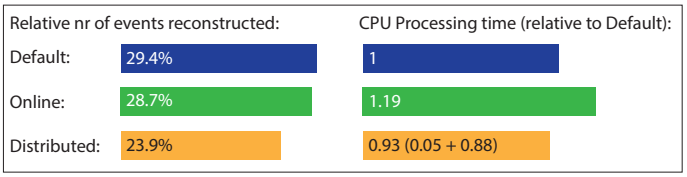

Fig. 2: Comparison between the three cluster-finding methods for the h_c decay channel. Events were generated at a rate of 200 kHz. (left) Relative number of succesfully reconstructed h_c mesons. (right) CPU processing time needed, relative to the *Default* method.

To get a better idea of the yield and speed of the algorithms, a more challenging dataset with 5000 events of $h_c \to \gamma\eta_c \to \gamma\pi^0\pi^0\eta \to 7\gamma$ is generated. This channel was chosen because it can be fully reconstructed using information from the EMC only, and because it features a high photon multiplicity. That increases the probability for pile-up, so the recovery performance for those events can also be checked. Figure 2 shows how the three algorithms stack out against each other, in terms of yield (left) and processing time (right). There is no huge difference in performance between the methods; the *Distributed* method perform slightly worse in terms of yield, but shows excellent time performance. Because of these features, and because its concept fits the architecture of the readout system, that method is chosen for implementation.

III. Readout

Recently, the distributed cluster-finding algorithm has been implemented in a prototype of the readout hardware, specifically on a device that is called a Data Concentrator (DC). The FPGA-based Data Concentrators are part of the readout chain, taking data from the digitisers as input. They combine data streams, sort the data, and provide synchronisation. The hardware implementation enables testing the performance in the online environment. Figure 3 shows the comparison between the results of the algorithm for a simulation in PandaRooT (revision 28955, with ROOT v5.34 and FairRoot v-15.11, shown in blue) and a simulation of the implementation of the

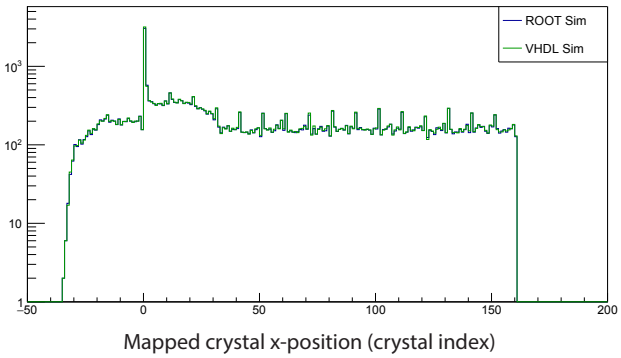

Fig. 3: Results for the mapped X-position, an arbitrary quantity that is constructed by the Distributed Cluster-Finding algorithm: For a simulation using PandaRoot (blue), and for a VHDL simulation (green).

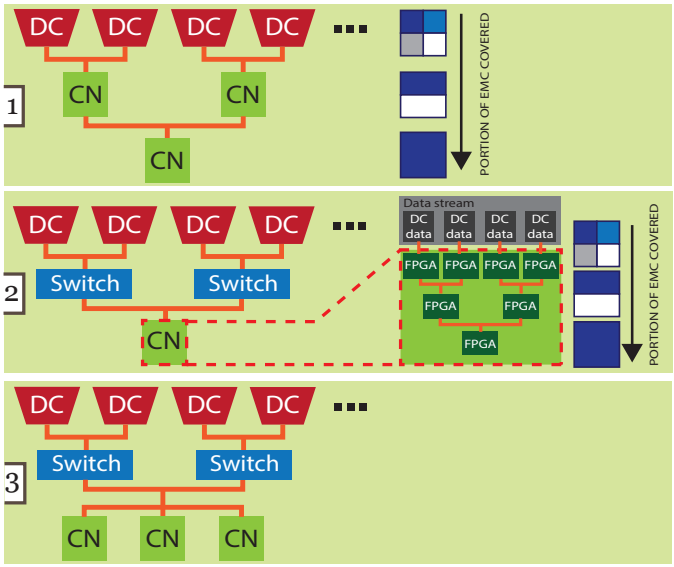

Fig. 4: Graphical representation of the three conceptual options for the Data Collection Network (see text).

same algorithm in VHDL (using Vivado v2015.3, shown in green). As can be seen, the results agree. This implies that the real implementation will perform the same as the simulation.

IV. Data Collection Network

As stated before, the use of the distributed cluster-finding algorithm fits in the design of the readout system of the calorimeter. The question remains what is to be done with the output produced by the Data Concentrators, which will have done partial clustering on the subset of the data that is available to them. This section describes three conceptual options for the Data Collection Network. The options are represented graphically in Figure 4.

1) **The *physical* intelligent network** | A network of Data Concentrators combines data from two (or more) subsets, covering an ever growing portion of the EMC, until it is completely covered. In combining the sets, the network re-sorts the input stream of preclusters, and can perform advanced processing tasks like merging neighbouring preclusters and removing low-energetic clusters[4].
   - *Advantages*: The data is combined in small steps, shrinking the total amount of data on the way. At the endpoint of the network, less computational resources will be needed.
   - *Disadvantages*: The large number of small steps can introduces a larger latency in the system, and some work might be done twice. In addition, the nodes of the network are required to have processing power.
2) **The *emulated* intelligent network** | A simple network of switches collects the data. The data is fed to Compute Nodes[5], which internally combines datasets from the participating Data Concentrators, like the physical intelligent network does.
   - *Advantages*: The simplicity of the network makes it much more cost-effective.
   - *Disadvantages*: It is much more difficult and meticulous to implement the merging algorithm.
3) **The non-intelligent network** | Like in the previous option, a simple network of switches collects the data. The data is fed to Compute Nodes, which take a timeslice of data from the *complete calorimeter* and perform the advanced processing tasks on that dataset, such as precluster merging, removing low-energetic clusters, and setting cluster properties.
   - *Advantages*: The simple network is very cost-effective.
   - *Disadvantages*: The load at the endpoint of the network becomes very heavy, because each node will need to process data from the entire calorimeter.

[4]That is, as long as they are not near the edge of the section of the EMC under investigation, since in that case, they might be part of a bigger cluster.

[5]FPGA-based devices that are designed to perform high-level reconstructions in real time

V. Conclusion and Outlook

The use of a distributed cluster-finding algorithm reduces the load and allows parallel preprocessing, which fits the free data-streaming concept of the PANDA experiment. The algorithm has already been implemented in an FPGA, and it produces the same results as in the PandaRooT simulation.

Several concepts for the Data Collection Network are being explored, but future investigations on the expected data rates and resources of the candidate network nodes are needed to provide a recommendation which option to use.

References

[1] An overview of XYZ new particles, X. Liu et al., arXiv:1312.7408 [hep-ph] (2014).
[2] New spectroscopy with PANDA at FAIR: X, Y, Z and the F-wave charmonium states, E. Prencipe, J.S. Lange, A. Blinov, arXiv:1512.05496 [hep-ex] (2015)
[3] S. Spataro ($\bar{P}ANDA$ Collaboration), IOP Journal of Physics: Conference Series, 331-3 (2010).