

Disentangling $\bar{P}ANDA$'s time-based data stream

M. Tiemens on behalf of the PANDA Collaboration

KVI - Center For Advanced Radiation Technology, University of Groningen, Zernikelaan 25,
9747 AA Groningen, The Netherlands

E-mail: m.tiemens@rug.nl

Abstract. To allow the online selection of events, the readout of the $\bar{P}ANDA$ detector will reconstruct particles online. Several algorithms to perform this task in one of the subsystems, the electromagnetic calorimeter, are being developed. The algorithms are discussed, and a simple test case shows that they appear to behave similarly in terms of the ability to reconstruct events, and the time it takes to do this. However, some peculiarities, like the fact that the algorithms show the best performance in what is expected to be the worst-case scenario, still require additional investigation.

1. Introduction

This work discusses some software developments for the future $\bar{P}ANDA$ experiment. For an introduction, the reader is invited to read other publications on the experiment in this proceedings series. The design of the detector is shown in Figure 1. This paper will mainly focus on one subsystem, the electromagnetic calorimeter (EMC), shown in purple in Figure 1. The device consists of four parts: a Barrel, containing 11,360 scintillation crystals and a Backward and Forward Endcap, containing 524 and 3856 crystals, respectively. The faces of the crystals are approximately 2×2 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.

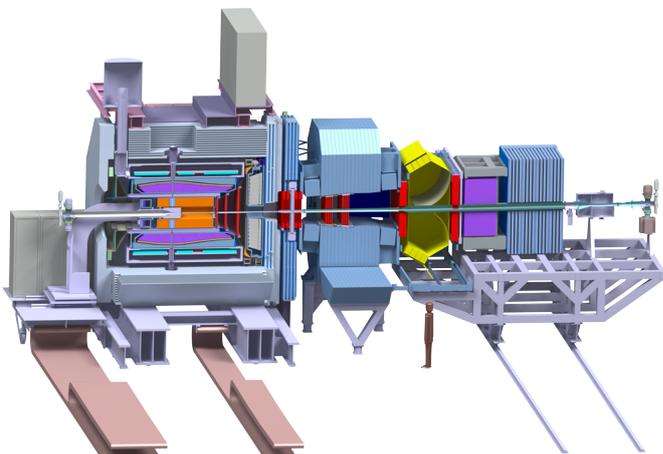


Figure 1: CATIA drawing of the $\bar{P}ANDA$ detector. *Courtesy:* A. Cebulla, FZ Jülich, 2014.

An important feature of the detector is the ability to reconstruct detected decays online, and perform the event selection based on those reconstructions. To be able to do this, 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 discussed below. Two of these are eligible for implementation on the online platform, providing vital information for the event selection.

2. Cluster Forming

An interaction between a proton and an antiproton will be referred to as an *event*. In the $\bar{P}ANDA$ experiment, events are generated at a rate of $2 \times 10^7/s$ at the design luminosity. The interaction may lead to the production of secondary particles. The majority of final-state particles at the end of the decay chain of these reactions will create a particle shower inside the EMC crystals. Due to conservation of momentum, the shower spreads out laterally (i.e., transverse w.r.t. the momentum vector of the incoming particle), which may cause the shower to create hits in neighbouring crystals as well. In this context, a *hit* is a crystal with an energy deposition above the detection threshold. The formation of particle showers creates groups of hits in the calorimeter, which are called *clusters*.

3. Cluster Finding

To find the four-momentum of the impinging particle, which is needed to be able 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. The next sections describe the established (“standard”) method, which cannot be implemented online, and the proposed online versions. The final part will compare the methods against each other.

3.1. Method 1: Standard Cluster Finding

The currently implemented version in the offline software package PandaRoot[1] takes the stream of hits, and treats each new hit as a separate cluster, unless it neighbours¹ to an existing one. In that case, the hit is absorbed in that cluster.

Advantages:

- + Widely used, well tested.

¹ As the time domain is important, in all references, “neighbour” means close in space *and* time.

Disadvantages:

- Slow (either each cluster size has to be recalculated when a new hit is processed, or a loop over each existing hit is needed to check for neighbours).
- When two hits from the same cluster are processed as first and second hit in the stream, they will be put into separate clusters when they are not direct neighbours. However, in general, the central hit in a cluster will have the lowest timestamp, and will therefore be processed first, so the probability for this failure to occur is minimal.

3.2. Method 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.

Advantages:

- + Fast (it is only needed to loop over hits once).

Disadvantages:

- There are still many iterations involved, making the solution possibly less suitable for implementation online.

3.3. Method 3: Distributed Cluster Finding

Before proceeding to the method, a brief introduction to the readout chain is required.

3.3.1. The Readout Chain of the \bar{P} ANDA Detector The light created in each crystal is collected at the backface by photosensors, which are read out by digitisation modules. Each module can read out 16 crystals, and sends the information it extracted on to a data concentration stage, which performs some more signal processing. Each Data Concentrator (DC) reads out 8 digitisation modules, and hence has access to 128 crystals – a sizeable portion of the calorimeter. The Data Concentrators send their information to the Event Building stage, which performs the online event reconstruction.

3.3.2. Distributed Cluster Finding As the Data Concentrators have access to a sizeable section of the EMC, it may be possible to look for clusters at this level. This will reduce the load at the later stages. The idea of distributed cluster finding is to first look for clusters in such a section using the Online Cluster Finding Method. However, as can be seen in Figure 2, it is possible that a cluster is located at the edge of two or more sections. In this case, such clusters will be split, so they will need to be merged later on (see, for example, the white-and-gray cluster in Figure 2). Therefore, for the clusters found at this stage, called *preclusters*, only their (x, y, t) location² and radius will be calculated (i.e., no energies or advanced cluster properties). The radius is the circumcircle of the precluster (coloured circles in Figure 2). In the next stage, preclusters are merged if the distance between them is smaller than the sum of their radii.

Advantages:

- + Each DC processes a small sample of the dataset → Fast.
- + The next stage can use preclusters instead of hits → Fast.

² The idea is to use a 2D map of the EMC, where each crystal is a single point, so the z information is not needed here.

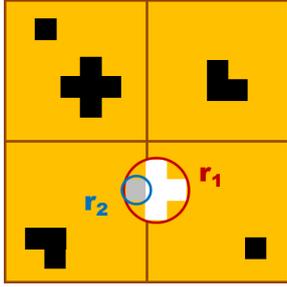


Figure 2: Illustration of precluster forming in Data Concentrators (yellow squares). Hits are indicated by black, gray, and white squares. More details can be found in the text.

Disadvantages:

- Some additional operations are needed, such as the calculation of the precluster location and radius.
- If preclusters have very elliptical shapes, the radius will be greatly overestimated. However, as the incoming particles hit the crystals orthogonal³, the shower shape is expected to be (approximately) circular.

4. Comparing Methods

The next task is to compare these methods. Two tests on an artificial simulated data set, $5000 p\bar{p} \rightarrow \gamma\gamma$ events at a centre-of-mass energy of $2.081 \text{ GeV}/c^2$, are carried out. It must be noted that the results of these tests are very preliminary, as both the algorithms and the simulation framework are still under development. For each method, three interaction-rate scenarios are considered: High Rate (HR, $2 \times 10^7/s$), Low Rate (LR, $2 \times 10^5/s$), and Event-Based (EB, completely disjoint events).

4.1. Test 1: Ability to reconstruct events

The first test checks how many of the original 5000 events can be recovered. Because of the forward boosted particle production, the raw cluster energies do not give much information. The invariant mass of the parent $p\bar{p}$ system is taken for comparison. The spectrum (see Figure 3) shows a peak around that value, which is fitted and integrated to obtain the number of reconstructed events. The results are shown in Figure 4a. The high rate scenario consistently shows higher yields, implying that it is more efficient than the ideal (event-based) case. As the high rate case suffers from pile-up and event mixing effects, this result is unexpected. A check with the information from the event generator will need to be performed to see if the reconstructed events correspond to the simulated events. However, due to technical difficulties, that information is not yet available.

Quantitative, all methods obtain similar yields. However, the Online method appears to be slightly less effective, as can also be seen in Figure 3.

4.2. Test 2: Time needed to reconstruct events

As the data is produced at a high rate (see section 3), the runtime of the algorithms is another key parameter to be checked. The results are shown in Figure 4b for the HR case, relative to method 1 (Standard). As desired, the Online method is the fastest, and the two steps of Distributed Cluster Finding (precluster finding and merging) are separately the fastest. However, no solid conclusions can be drawn at this point, because this test was performed on a CPU, and

³ The crystals in the EMC are oriented to face inwards, pointing just behind the interaction point (to avoid particles to get undetected in the gaps between the crystals).

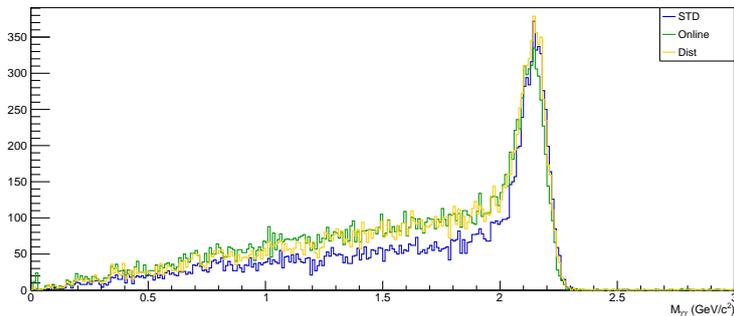


Figure 3: Comparison between the three cluster finding methods for the high rate scenario. Legend: STD = Standard Cluster Finding, Online = Online Cluster Finding, Dist = Distributed Cluster Finding.

the final algorithm will run on an FPGA⁴. The performance likely differs on such a device. Hardware prototypes containing FPGAs are available, but the cluster finding algorithm has not yet been implemented on them. That will be done when the algorithm has been debugged and streamlined. Hence, at this moment, the comparison on CPUs is the best that can be done.

5. Summary

This paper discussed three methods to search for clusters in the EMC of the future $\bar{P}ANDA$ experiment. All methods performed similarly in both event reconstruction and runtime, with the Online algorithm being the fastest overall. The two steps in Distributed Cluster Finding, precluster finding and merging, are separately the fastest, as desired. The yield was higher in the high rate scenario, which is not what would be expected. This will be investigated.

After fixing the forementioned bugs, the algorithms will be tested using more complicated decays, such as $h_c \rightarrow \gamma\eta_c \rightarrow 7\gamma$, and by including background from e.g. a DPM background generator. The effect of so-called bump splitting, where two overlapping clusters, that are identified as one, are separated, also has to be investigated. Ideally, this should be included, but it will consume additional resources which may not be available. It should therefore be investigated how much will be gained by including this procedure, and then design a lighter version.

References

- [1] Spataro S *et al* (PANDA Collaboration) 2010 *IOP Publishing - Journal of Physics: Conference Series* **331** 3

⁴ A Field-Programmable Gate Array (FPGA) is an integrated circuit containing an array of programmable logic blocks, connected by a collection of reconfigurable interconnects.

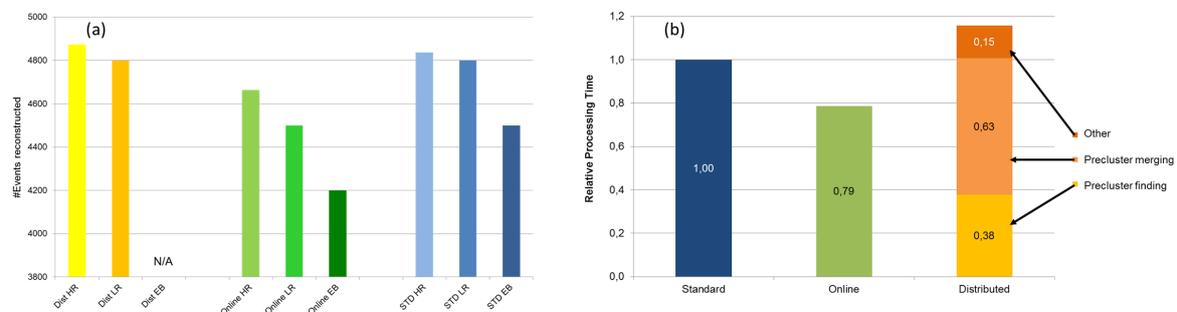


Figure 4: Comparison between the three cluster finding methods: (a) Between the number of reconstructed events in three scenarios, (b) Between the runtimes for event reconstruction, relative to the Standard method, for the high rate scenario.