Planning the updates on DUNE Offline Data Quality Monitoring

1. Framework is built for protoDUNE, so we can test it under data taking context;
2. Current User Interface: https://dunedash.edi.scotgrid.ac.uk/;
3. Plan to update de tool to be used in August 2024.

The future user interface

Move from Dash to Django application!

protoDUNE VD

Cold Box

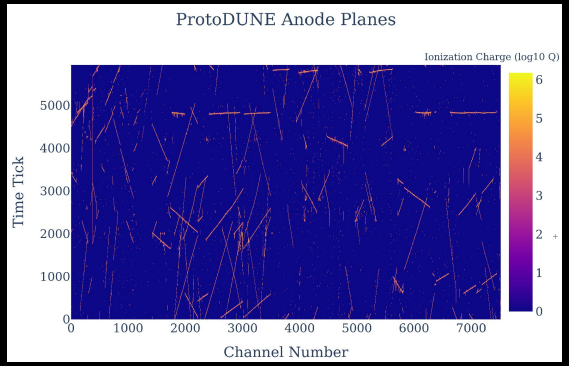
ProtoDUNE Offline Data Quality Monitoring

Run

26952

Event

51



Updates:

- Separate U, V and W planes for each APA;
- Format them into interactive deep zoom images;
- Add bad channels and low frequency noise;
- Mask zeros to give more contrast between no and low signals.
  (Brett Viren's suggestions, I am currently working on them!)

The future user interface

Move from Dash to Django application!

protoDUNE VD

Cold Box

ProtoDUNE Offline Data Quality Monitoring

Run

26952

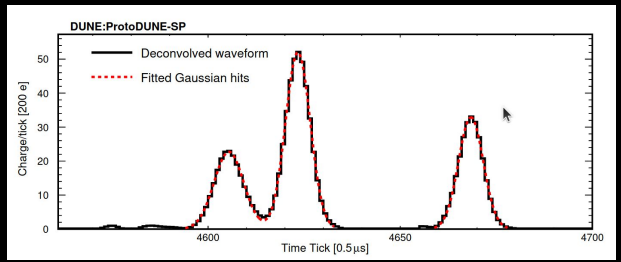
Event

51

Insert new plots

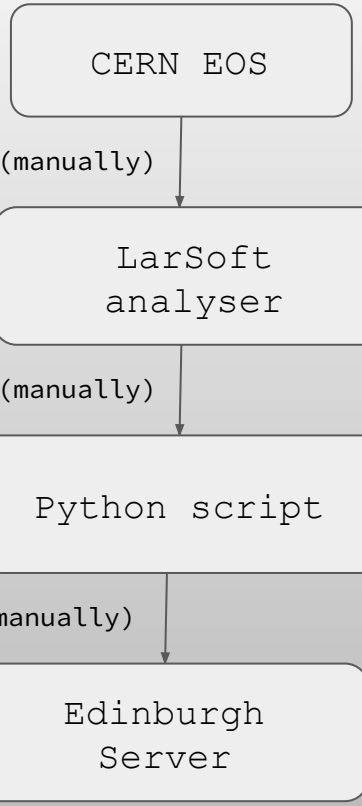
Updates:

- **Waveforms' RMS versus wire channel;**
- **Noise spectrum versus frequency;**
- **Hit finding:**



B. Abi et al 2020 JINST 15 P12004

Path of the Data



Approximately 1 hour after data is recorded by DAQ (Wenlong)

Takes around 30 minutes to process each .hdf5 datafile

Outputs the usual ROOT file
- \+ Numpy arrays (file.npz)
- \+ For the future plots, will you want to export processed files (file.txt)?

Takes around 40 seconds to generate a single high resolution image. For 100 events, 1 hour!

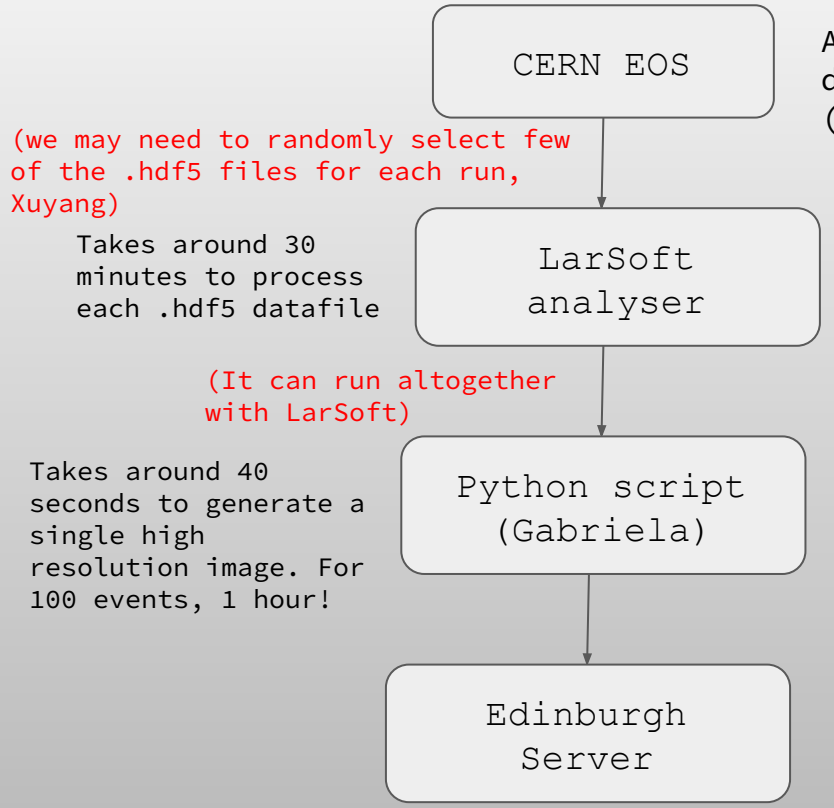
- Extracts the run and event numbers;
- Plots the heatmap of ionization charge and exports it as run_{runnumber}_event_{eventnumber}.png

- Gets the .png images plus a .npz which metadata is included;
- Displays the figure according to user request!

Path of the Data - future

Team:

Gabriela Vitti Stenico
Xuyang Ning
Robert Currie
Wenlong Yuan



Approximately 1 hour after data is recorded by DAQ (Wenlong)

Outputs the usual ROOT file
+ Numpy arrays (file.npz)
+ For the future plots, will you want to export processed files (file.txt)? -> Xuyang

- Metadata -> Elasticsearch
- Images -> S3

- I still need to learn how to interface Django app with s3 and elasticsearch services.

Questions

1. Is there a way to optimize LarSoft processing time? We have the **standard2_reco_protoduneHD.fcl** for event reconstruction in protoDUNE. Only two producers are active in my fhicl file:

```
reco: [ #ophit,
        #opflash,
        #opslicer,
        tpcrawdecoder,
        wclsdatahd
      # gaushit,
      # nhitsfilter,
      # reco3d,
      # hitpdune,
      # pandora,
      # pandoraWriter,
      # pandoraTrack,
      # pandoraShower,
      # pandoracalo,
      # pandoracalonosce,
      # pandorapid,
      # pandoraShowercalo,
      # pandoraShowercalonosce,
      # pandoracali,
      # pandoracalipid,
      # emtrkmichelid
      # crttag,
      # crtreco,
      # anodepiercerst0,
      # pandora2Track,
      # pandora2calo,
```

Necessary for signal processing. Are there other services we can disconsider in this fhicl to reduce time of processing?

2. Channel mapping. Following this one. Is it correct? Can I use this code to dump the wire channels?

```
art::ServiceHandle<geo::Geometry> geo;
{
  double xyz[3];
  double abc[3];
  int chan;
  int cryo = geo->Ncryostats();
  for (int c=0; c<cryo;++c){
    const geo::CryostatID cid(c);
    int tpc =geo->NTPC(cid);
    for (int t=0; t<tpc; ++t){
      const geo::TPCID tpcid(cid,t);
      int Nplanes=geo->Nplanes(tpcid);
      for (int p=0;p<Nplanes;++p) {
        const geo::PlaneID planeid(tpcid,p);
        int Nwires = geo->Nwires(planeid);
        for (int w=0;w<Nwires;++w){
          const geo::WireID wireid(planeid,w);
          geo->WireEndPoints(wireid,xyz,abc);
          chan=geo->PlaneWireToChannel(wireid);
          std::cout << "FLAG " << chan << " " << c << " " << t << " " << p << " " << w << " " << xyz[0] << " " << xyz[1] << " " << xyz[2] <<  " " << a
        }
      }
    }
  }
}
```


THE UNIVERSITY OF EDINBURGH


DUNE
DEEP UNDERGROUND
NEUTRINO EXPERIMENT

Thank you!