

DAQ & Online - CM44

Y. Karadzhov

UNIGE - DPNC

April 1, 2016

Step IV is in progress and major modifications in the DAQ hardware and software are highly undesirable.

Only minor interventions in the DAQ since the last collaboration meeting. Mostly addressing the recommendations made by the MPB:

- The DAQ repository now maintains frozen releases, making possible to easily **roll back** to an earlier version if needed.
- Instructions explaining how to roll back the DAQ equipment software are available here
<https://micewww.pp.rl.ac.uk/projects/operations/wiki/DAQ>
It is as simple as running a single script on any of the DAQ computers.
- The software version used during the data-taking is now recorded in the CDB.

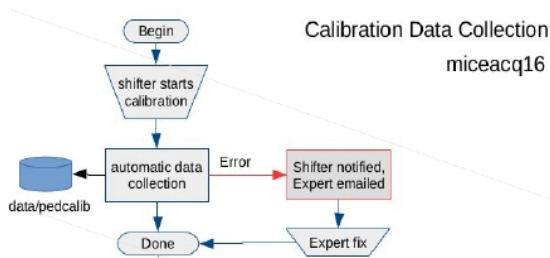
Tracker DAQ

No major changes to Tracker DAQ and Controls since last CM.

Tracker Calibration

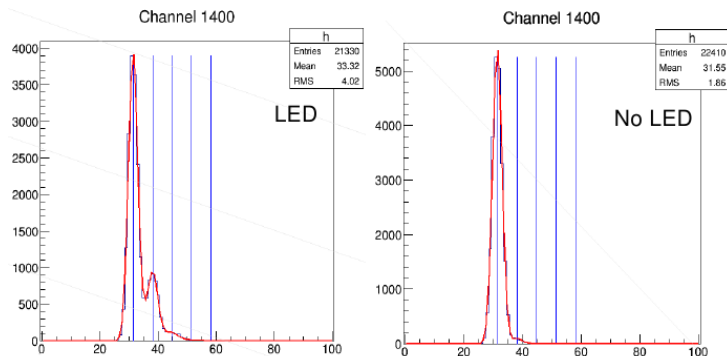
Improvements made to calibration process to facilitate regular calibration updates:

- Master calibration performed at beginning of ISIS cycle
- Daily updates to calibration are verified against “master” performance.

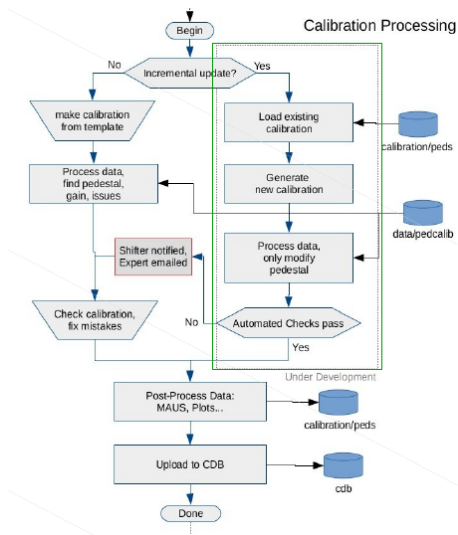


Tracker Calibration

- A calibration is required in order to convert from the ADC values to a meaningful **npe** value.
- The calibration procedure contains two runs with the internal light injection system enabled/disabled.



Automating the “per shift” calibration updates



Online running still under development.

Online Monitoring

- The communication interface between the Online Monitoring system and EPICS is complete.
- Alarms having appropriate severity levels are raised in Run Control in case of problems.
- We still have bugs in the communication between the background process and the GUI.

Rhys Gardner is about to defend his PhD thesis in the next few months. Thanks to Rhys for the good job.

We need a new person to take the responsibility for the Online Monitoring

Why do we need a new Online Reconstruction framework?

The old Online reconstruction:

- Based on the open source python package Celery
- Used in MICE since 2010.

Even during Step I the old framework was set to process 1 of every 20 spills.

The installation of the two trackers and the EMR revealed big problems:

- Dramatic increase in the amount of raw data to be processed.
- Addition of new complex reconstruction algorithms.

The result was:

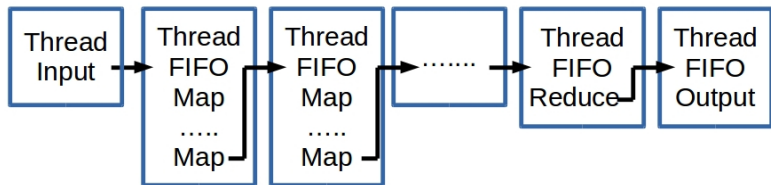
- A huge increase in the processing time needed to reconstruct a single spill.
- The Online reconstruction became unstable.

Divide and conquer

Parallel computing: the large problem can be divided into smaller ones, which are then solved at the same time.

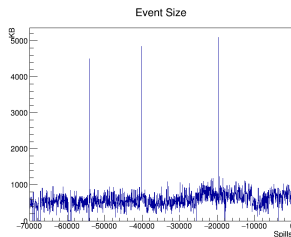
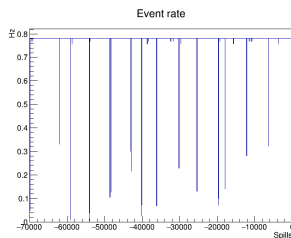
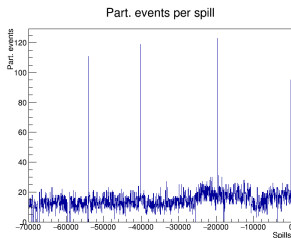
How does this work

- MAUS is used as a third party. All original MAUS maps and reducers are used without changing anything.
- The data processing is driven by a bunch of classes written c++11.
- Conveyor-like implementation of the job distribution.
- The code allows for a single-thread or a multi-thread processing.

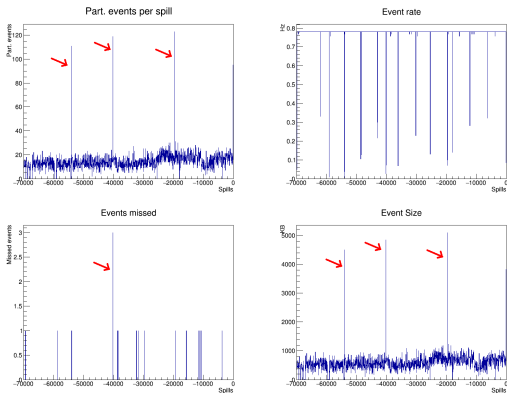


The new framework for the Online reconstruction was tested the last two data-taking periods.h

Reducer, monitoring the Online reconstruction. On the histograms: spill 0 is the last processed spill and the older spills have negative Id numbers.



More sophisticated tests in February and March



Pion Ref. runs taken at the beginning and at the end of the day. The Online reconstruction starts having problems when the DAQ event size is bigger than ~ 6 GB. Investigations have shown that most probably this is a network issue. Will investigate further, during the shut-down period.

The Online Reconstruction daemon

- A background process checks periodically (every 30 s.) if the main online reconstruction process still runs. If because of some reason, the reconstruction process is dead, the daemon will restart it.
- The daemon also checks the memory consumed by the main process. If the memory usage exceeds 16 GB the main process will be restarted.
- If needed the Online Reconstruction can be stopped manually by running the script **stop_processing.sh**. This will stop the main process and will kill the daemon.

On-line event viewer

Mihailo Savic - University of Belgrade

- The idea is to implement a reducer which would export events from the last reconstructed spill to HepRep format.
- Files would be stored in a selected directory keeping only events (files) from the last reconstructed spill.
- HepRApp would run independently in automatic update mode (iterates through all files in selected directory)

Conclusion

- Online is stable since quite some time.
- Working to constantly improve the subsystems.
- Couple of minor issues have to be addressed, but in general the Online is in a good shape and is ready for Step IV.