DQM Implementation

Boqun Wang MPI for Physics Belle II Trigger and DAQ Workshop August 27, 2019

System Overview

- DQM is used for checking detector status and finding possible problems during data taking.
- Histograms are filled by DQM modules running on HLT/ExpressReco working nodes.
- The transferring of the DQM histograms involves multi-stages of shared memory buffers on the HLT/ExpressReco working nodes and control nodes. Finally all histograms are transferred and merged on DQM server (dqmsrv1).
- More advanced analysis jobs like fitting and reference superimposing can be done on the DQM server by using the DQM analysis modules.
- The plots are served on the canvas server and can be accessed by web browser in the DAQNET.
- A DQM mirror site has been prepared for people outside of KEK.
- After each run, all DQM histograms are saved to a root file on quick analysis server for offline check and validation. They are transferred to KEKCC and the mirror site for offline checking.

System Overview (Phase 3)



DQM Module

- basf2 modules to fill histograms for certain sub-detectors or other purpose of online monitoring, e.g. physics quantity like R2.
- Running on the work nodes of HLT/ExpressReco.
- The DQM module is running in parallel and the flag *c_ParallelProcessingCertified* must be set.
- Only histograms with "LIVE" option will be transferred to backend (not implemented yet).
- The DQM module has the access to the raw data from detector and the reconstruction information (track, vertex, etc) from HLT/ExpressReco.
- The implementation of the DQM module is the duty of the sub-detector experts and the source code is managed under each sub-package.
- To include the DQM module in the online processing, one should edit the config file under dqm/scripts/daqdqm/.
- The number of the histograms should be limited and only the necessary histograms should be implemented in the DQM module.

DQM Transfer

- The following transfer scheme is used for phase 2 and phase 3. It might be replaced by the new scheme with ZeroMQ (see Nils' talk).
- Two standalone programs for the histogram transfer:
 - hserver: receives histograms from various sources, merge the histograms with the same name and save them to the shared memory.
 - hrelay: transfers the histograms saved in the shared memory to the hserver process on another machine.
- Another module, DqmHistoManager, which runs in the same path with the DQM module, collects and transfers the histograms filled by the DQM module to the hserver process.
- Finally, all histograms are transferred, merged and saved to the shared memory on DQM server for following processing.
- The transfer interval is about 2 minutes under current configuration.
- During the histogram transfer, if for some reason the histograms from previous runs haven't been cleared, it might contaminate the histograms for the following runs.

DQM Analysis Module

- The histograms saved in the shared memory of the DQM server (dqmsrv1) can be accessed by multi processes to perform high level analyses.
- An input module (DQMHistAnalysisInput) load the histograms for following processing.
- The analysis module can do:
 - Fit the histogram and draw the curve and parameters.
 - Change the background color to indicate the current working condition and whether experts should be notified.
 - Draw additional lines and marks on plots to indicate good and bad regions.
 - Export features extracted from the plots to EPICS PVs for other programs (CS-Studio) to use.
- The analysis module also has the access to Belle II database.
- Currently most detectors have or plan to have their own analysis modules.
- TCanvas objects are generated by analysis modules to display the plots and other analysis results.

Online Display

- After the processing of the analysis modules, the TCanvas objects generated are transferred by an output module (DQMHistAnalysisOutputRelay) to a canvas server.
- A canvas server is a root macro which uses THttpServer to serve the TCanvas objects. It can be accessed by web browser and other programs in the DAQNET.
 - To access from inside the KEK network, a ssh tunnel on bdaq is necessary.
- On the client side, the JSROOT library is used to render the plots and provide some interactive functionalities like zooming and set log scale display.
- The online display is shown on the monitoring screens in the Belle II control room in Tsukuba Hall B3.
- For control room shifters, there is a page for them which contains selected plots from all detectors to make it easy for non-experts to check their running conditions. More detailed plots can be checked by detector experts in other pages.

Reference Plots

- The reference plots are used to check whether the plots from real data is OK or not.
- They are provided by sub-detector experts as root files and are scaled and superimposed with real plots in real time.
 - The reference root files are managed under git repository <u>https://stash.desy.de/projects/B2G/repos/daq_dqm_gui/bro</u> <u>wse/references/phase3</u>
 - They are deployed to the DQM server manually.
- Currently the normalization of the reference is based on the integral of the plot. More methods of normalization are under consideration.

Display Layout

- The plots for a sub-detector can be grouped into different pages based on the relations between plots.
 - For example, one page for all the occupancy plots and another one for all the cluster size plots.
- Sub-detector experts can specify the layouts of their histograms in a certain page.
- JSON files are used for the layout specification. It is easy to write and easy to interpret in the Javascript.
 - One only need to define the name of the plot and the its position in a certain layout.
- Some simple layouts are defined by using CSS (Cascading Style Sheets), e.g. 6X4 grid layout.
 - More complicated one can also be defined with CSS grid.
- The JSON and CSS files are managed in a git repository <u>https://stash.desy.de/projects/B2G/repos/daq_dqm_gui/browse</u> under web_files/detectors/css/ and web_files/detectors/phase3/.

Mirror Site

- One mirror site on DESY has been prepared for people outside of KEK:
 - https://dqm.belle2.org
 - Log in with Belle II DESY credential.
- A program running on the DQM server (dqmsrv1) takes the snapshot of the canvas server every 5 minutes and transfers it to the mirror site.
- The canvas server running on the mirror site reads the snapshot file (root file) and serves the canvases for public access.
- Access speed is a bit slow from KEK, but much faster from Europe.
- The mirror site also hosts the histograms from past runs and offline plots for sub-detectors (see next page).

DQM Root Files

- After each run, a root file is generated which includes all the histograms for this run.
- In the beginning of phase 3, the root files are generated on each work node and they are merged together into one file. This is very time consuming when the number of histograms is very large (> 7000).
- During phase 3, the scheme for generating the root files were changed to use the histogram transferred to the DQM server. The time for file generation is reduced significantly. More details see Itoh-san's and Nils' talk.
- The root files are transferred to quick analysis server at /data1/dqm/dqmsrv1/e0008/dqmhisto/ and synchronized to KEKCC at /group/belle2/phase3/dqm/dqmsrv1/e0008/dqmhisto/.

DQM for Past Runs

- The DQM root files are transferred to the mirror site for offline checking:
 - <u>https://dqm.belle2.org/past_runs/index_hlt_exp8.htm</u>
 - Only physics, cosmic and beam runs are transferred.
 - The root files are reprocessed offline to generate plots from analysis modules.
 - These files on mirror sites can be downloaded, checked with JSROOT browser or displayed with the same layout as in online display.
- Synchronization interval: ~30 minutes.
- Reference display is also enabled for the past runs.
- Sub-detectors can also use this mirror site to store their own plots for offline checking.

Offline Development

- For the DQM module development, the HistoManager can be used to save histograms to output root file.
- For the offline development of the DQM analysis modules, it's difficult to set up the DQM histogram generation and transfer for the offline testing on machines with basf2 installation.
- Two modules can be used for the input and output for the analysis modules development:
 - The DQMHistAnalysisInputRootFile module can read a root file containing histograms and use them as the input for the following modules in the same path.
 - The DQMHistAnalysisOutputFile module can save the histograms and canvases generated by the analysis modules in the same path to an output root file.
 - Example: dqm/analysis/examples/file_dqm_test.py
- Some scripts in the daq_dqm_gui repository can read root files with histograms or canvases and serve them in a local web server. This is useful for the layout development in the local machine.
 - web_files/serve_canvas.C and web_files/testserver.C

Summary

- DQM system was running during phase 2 and phase 3 data taking.
- Reference plots are superimposed on the display.
- Offline histograms for past runs can be checked at the mirror site.
- Preparation for next data taking in October:
 - Basically the same setup as the spring run (possibly with ZeroMQ for histogram transfer).
 - Automate some of the management jobs (reference files and layout files deployment).
 - Knowledge transfer to new DQM liaison person (Longke Li).