Prototype Shared Memory, ROOT based online monitoring frontend for DAQs

Raman Sehgal*, P. K. Netrakanti, L. M. Pant, and R. G. Thomas
Nuclear Physics Division, Bhabha Atomic Research Centre, Mumbai - 400085, INDIA
** email: sc.ramansehgal@gmail.com*

Introduction

The monitoring of any Data Acquisition (DAQ) system requires filling and presenting a large number of histograms and graphs. A prototype of shared memory based online monitoring system based on ROOT framework has been developed for this task. The system is designed in such a way that it can act as a super module which will provide GUI to access the data fetched from any Data Acquisition system, provided that an appropriate interface with the Data Acquisition stream is available that will store the data in a defined format.

Online vs. Offline analyis

Data handling in High Energy Physics activities are categorized as Online and Offline. The Online attribute is given to all components active during the transfer between the front-end electronics (detector(s), triggers, sensors) and the Permanent Data Storage (hard disk, tape, CD). The Offline activities cover the libraries, algorithms and simulation sessions used before and after the data collecting process. This work present an Object-Oriented front-end developed to provide an online data monitoring for DAQs. This is easy to understand tool which can be plugged in with any DAQ which can pack the acquired data in desired ROOT tree format and store it in shared memory.

The ROOT system

The ROOT [1] system provides a C++ based complete Object Oriented framework with all the functionalities needed to handle and analyze large amounts of data.

The backbone of the ROOT architecture is a layered class hierarchy where most of the classes inherit from a common base class TObject, which implements the common behavior for ROOT classes. All ROOT objects can be uniformly stored using a proprietary file format. To achieve higher performances and to have complete syntax checking of the code, compiled C++ code can also be used in the ROOT framework.

Architecture

The system is designed to provide a common tool to start the data acquisition and visualization. To achieve this a multithreaded application is developed that provides very simple GUI to start the DAQ and Visualization in two different threads. The architecture is based on shared memory based producer-consumer model where the producer thread fetched the data from DAQ and put it in shared memory buffer, in addition to this it also writes the data to ROOT file, which can be used later for offline analysis.

The second thread named as consumer thread fetches the data from shared memory buffer and displays the statistics in real time in the form of ROOT histograms. The application provides full control over these two threads, i.e. one thread can be started and stopped independently of other. These two threads are created in Qt framework[2]. Figure 1 below shows the Main-window that will create these two threads.

Fig 1: Main window of the application

The schematic presented in figure 2 below shows the architecture of the application

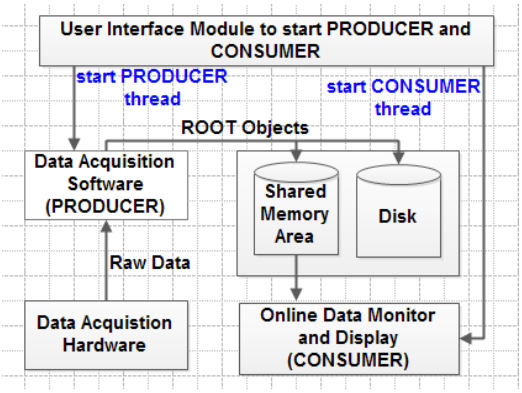

Fig 2 : Schematic of the architecture using producer consumer model

Implementation Details

The tool is written using ROOT Object-Oriented framework. It provide some classes to create the GUI to select the crate number and channel number within the crate, other classes are there that will do the actual work of displaying real time histograms. It needs one configuration file that describes the DAQ system configuration, based on which it creates the GUI. It is having only two threads, out of which one thread will start the Data acquisition application and the other thread start the real time visualization of data as dynamic ROOT histograms. This tool can show the data corresponds to multiple channel as different histograms simultaneously, but with the current limitation of only one dynamic histogram. Later on we will provided the functionality that enables it to show multiple dynamic histogram simultaneously.

Dummy Data used for testing

The dummy data used for testing is random data which is packed into the data structure which corresponds to data structure generated by VME-DAQ setup at RPC-lab, NPD , BARC. The data generated by the setup consist of ROOT trees. The ROOT tree is having one branch for each channel. The data for each event consist of variable size vector i.e. for each event the size of vector corresponds to the number of hits in particular channel. Once the data vector is ready, it is filled in the ROOT tree. And after every "n" number of events which can be decided by user, the data is written by this producer program to shared memory and also to the persistent storage in ROOT file format. The consumer program fetches the data from this shared memory location and display it as dynamic ROOT histograms. So once the DAQ system is switched on, there is no need to wait till the end of run to see the statistics, it can be seen online, once the producer starts filling the shared memory area. Figure 3 below shows the histogram window as displayed by this tool.

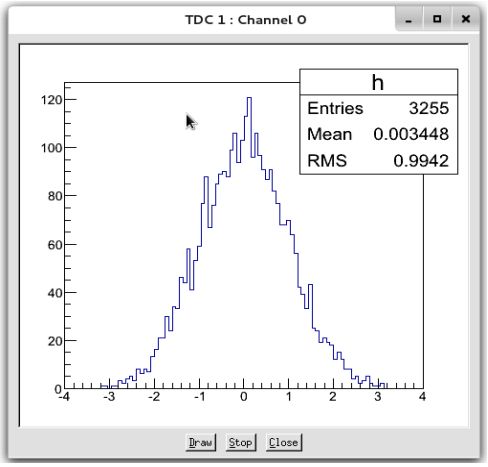

Fig 3 : Histogram window for one of the TDC channel

Conclusion

In this work we have presented a ROOT based online monitoring front-end tool for visualization of data acquired by DAQ system in real time. This tool is easy to integrate with any existing DAQ system just by casting the data into the data structure it accepts and put them in shared memory buffer.

References

[1] R.Brun and F.Rademakers, "ROOT – An Object Oriented Data Analysis Framework", Nucl. Inst. Method Phys. Res ., vol.A389, pp. 81-86, 1997

[2] Qt 3.1 Tutorial and Reference, Troll Tech AS, http://www.trolltech.com/