Remote Event Client Implementation in FairRoot Framework

D. Kresan, F. Uhlig, and the FairRoot Group
GSI, Darmstadt, Germany

Introduction

FairRoot is a framework, which is used by future FAIR experiments for an implementation of the simulation, reconstruction and data analysis algorithms [1]. Requirement of both, the basic framework features and the user-defined tasks developments, is the usage of the same code for online and offline event reconstruction. In this report we will present an extensions to several base classes, which was done to provide the possibility for communication with a Data Acquisition (DAQ) server.

Online Data Processing

The possibility of online event reconstruction is an essential point for most of the experiments at FAIR, which are aiming to perform high-intensity measurements without a hardware trigger. Therefore, we implemented an additional functionality to the framework, which is sketched in figure 1. In order to avoid overloading of the main DAQ server with connections from multiple clients, an intermediate stage called ”Remote-Event Server” [2] was introduced. There is also a possibility to connect a client directly to a file on disk, in order to support also the offline analysis. The class design allows supporting multiple data formats from a server or from a saved file. In the next subsection we shortly describe the Multi Branch System (MBS) [3] format used in many DAQ server implementations at GSI.

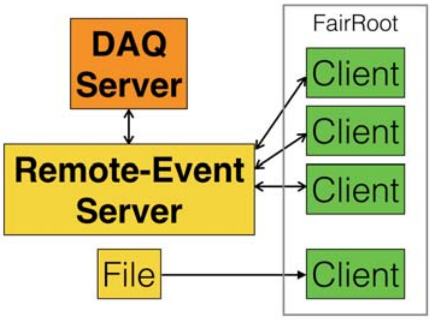

Figure 1: Illustration of the communication flow between a DAQ server and the FairRoot framework.

MBS Format

The data coming from the DAQ contains response from multiple detectors and is formed in sequences of events, each containing several sub-events. Each sub-event corresponds to one detector. A detector can be then identified via TYPE and SUB_TYPE values in the sub-event header. This header in a sub-event is then followed by the data.
A support for other data formats is provided, as was already requested by the CBM collaboration.

Server Side

The Remote-Event Server is to be started on a separate computer node. It connects directly to the specified DAQ server, and transmits the data using ROOT sockets to a specified network port. The name of the node, where this server is started, has to be used in a client configuration, described in the next section. For the testing purpose, one can send events with random numbers without connecting to a DAQ server.

Client Side

The Remote-Event Client is part of FairRoot. The steering class *FairRunOnline* implements an event-loop, and user-specified tasks execution. It gets the data from an object of the concrete implementation of the abstract base class *FairSource*. Detector-specific ”unpacker” classes then process the received raw data events. The unpackers fill the data as arrays of ROOT objects, which are taken as input for the hardware calibration and further analysis.
This dedicated class design allows a user to easily switch in his analysis between online-streaming and stored data by changing just one line in the steering macro.
Further developments will focus on a class *FairRootSource*, which will be inherited from *FairSource* and will read simulated data from a ROOT file. This will allow to combine two steering classes *FairRunAna* (currently used for the analysis of Monte Carlo simulations) and *FairRunOnline* into a single implementation.

Summary

We presented in this report an extension to the FairRoot framework for communication with a Remote-Event Server. The implementation will be used in the detector tests for the future R^{3}B experiment. These experimental tests provide unique possibility for validation and verification of the dedicated detector-specific reconstruction algorithms.

References

[1] http://fairroot.gsi.de

[2] H. Göringer,
http://www-aix.gsi.de/∼goeri/mbsnew/online.html

[3] H.G. Essel, N. Kurz,
http://web-docs.gsi.de/∼mbs/v51/manual/gm_mbs_c.pdf

doi:10.15120/GR-2014-1-IT-09