Research and design of DAQ system for LHAASO experiment

MINHAO GU[1,2], KEJUN ZHU[1,2], JIAN ZHUANG[1,2] FOR THE LHAASO COLLABORATION
[1] *Institute of High Energy Physics, Beijing, 100049, China*
[2] *Key Laboratory for Technologies of Nuclear Detection and Electronics, Beijing, 100049, China*
gumh@ihep.ac.cn
DOI: 10.7529/ICRC2011/V03/1134

Abstract: The tasks of LHAASO data acquisition system (DAQ) are carrying out the hardware configuration, reading out the data from electronics and monitoring the status of detector running. The system must be flexible and universal. It could be used for all subsystems even with different types of front-end electronics. So in the design, the system is divided into two parts, the font-end and the back-end software. The former runs on a single board computer with real-time Linux operating system and reads out data directly from electronic boards through VMEbus. This part depends on the subsystem with different front-end electronics. The back-end software, which controls and monitors the whole DAQ system, is designed to meet the common requirements of all subsystems. This paper presents the major requirements, design of back-end DAQ software and implementation of front-end for LHAASO experiment.

Keywords: LHAASO DAQ.

1 Introduction

The Large High Altitude Air Shower Observatory (LHAASO) project is proposed to study the gamma ray astronomy from 40GeV to 1PeV, and cosmic ray physics from 10TeV to 1EeV [1]. The proposed system consists of 5 subsystems. The layout of LHAASO detectors is showed as Figure 1.

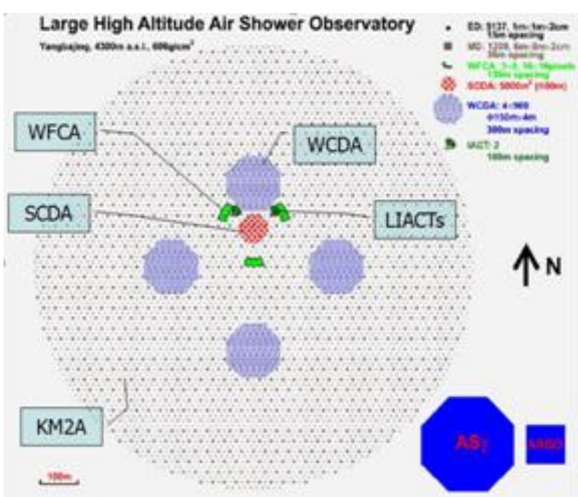

Figure 1. LHAASO layout..

DAQ software is required to be developed for at least two of these subsystems:

- A $1km^2$ EAS array (KM2A),
- 4 water Cerenkov detector arrays (WCDAs).

The LHAASO-KM2A consists of 5000 charged particle detectors and 1000 muon detectors covering an area of 1 km^2. The LHAASO-WCDAs is composed of 4 huge water tanks, and each has independent electronics and DAQ system.

2 Requirements

According to the design of electronics system, data is collected by front-end electronics (FEE) modules located in several VME crates. DAQ software needs to readout the data fragment through VME back planes, transport to back-end servers and build them into full events for storage. Events and fragments need to be processed by DAQ software for the purpose of monitoring.

System	Number of channels	Event rate (kHz)	Data throughput(Mbytes/s)
LHAASO-KM2A	6000	≈ 1	--
LHAASO-WCDA	900(×4)	≈16	≈ 13(×4)

Table 1. Performance requirements

Electronics configuration can be changed easily for data acquisition runs.
The data throughput and event rate of relevant subsystems are listed in table 1, which presents the performance requirements for the DAQ software system.

3 Software Design

The design of LHAASO data acquisition software is based on Atlas TDAQ framework, including the online system and the data flow. We must adjust the data flow design and add more components and applications to satisfy the needs of LHAASO experiment.

3.1 Data Flow Design

The data flow components in DAQ software are responsible for receiving and processing physics data, transporting the data to storage. Figure 2 shows the data flow design for LHAASO and the following paragraphs show more details of each layer in its design.

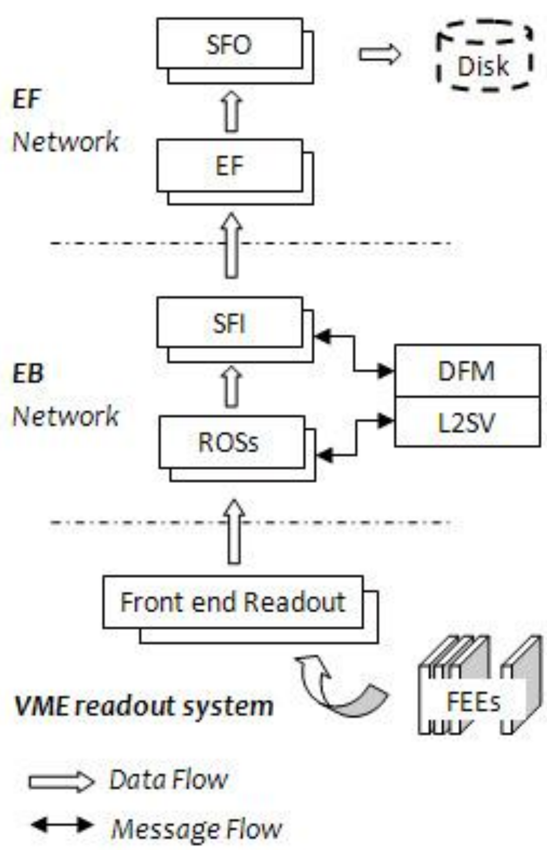

Figure. 2. Data Flow design.

1. Front end readout layer

This layer runs on the VME controller with embedded Linux operating system, reads data from FEE and sends fragments to back-end ROS (Readout subsystem) PCs. The response latency of software should make sure that no dead time is introduced.
Test results [4] show that on MVME5500 with TimeSys Linux OS, the average throughput of VME CBLT can reach 18Mbytes/s and the VME interrupt latency of CPU is 16.5us. LHAASO is proposed to use similar or better hardware which can meet the performance requirements.

2. Event building layer

When all fragments of an event are buffered in ROSs, the L1ID of the event is send to DFM (Data Flow Manager) by L2SV (Level 2 Supervisor). Then, according to some load balancing method, the DFM chooses a SFI (Sub Form Input), to collect these fragments and build a full event [2].
The ROS, DMF and SFI components transport messages and data through Ethernet, and perform the event building task.

3. Event filter and storage layer

The EF (Event Filter) components were originally designed to do the event level of software trigger [5], analyze the event data and decide whether to send this event to the next level of data flow components. The LHAASO experiment requires that all events triggered by electronics are stored future offline analysis, instead of online filtering tasks. But the interface of EF is still kept for the implementation of online monitoring system on this layer.
Finally, event data is stored to disk or transferred to IHEP by SFO (Sub Farm Output) component [5].

3.2 Front-end Application Design

The front-end application configures the FEEs through the VME back plane, reads data out and reports status to the back-end software.
Due to its design, LHASSO DAQ can be applied to different front-end electronics. The components directly related to a specific type of FEE can be developed and built into libraries independently, which are loaded dynamically by front-end application framework (see Fig 3).

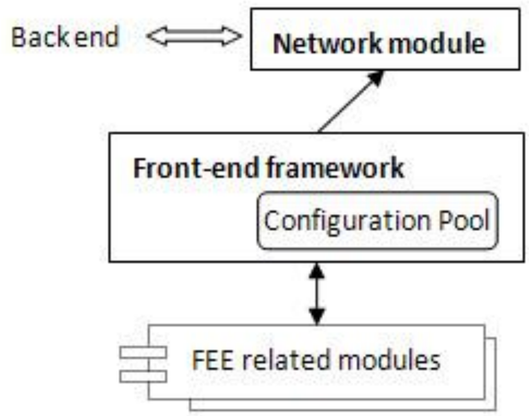

Figure. 3. Front-end software design

3.3 FEE Configuration Download

Configuration parameters need to be modified sometimes. It cannot be hardly coded.
Configurations for electronics are stored in back-end PCs. When users pass the CONFIGURE command, back-end application will download these parameters to the configuration pool in front-end software framework (see Figure 3). The configurations are listed as items with a name and a type property. The properties are described in XML format at back end and mapped values in front-end configuration pool. So the FEE readout modules are able to refer to these values when configuring FEEs.
When user adds a new item to the configuration file, it will be downloaded automatically to the front-end. So, developers don't need to care about the configuration download while developing FEE related modules.

3.4 Front-end Status Report

The front-end framework provides an interface to FEE related modules for status reporting. Status messages are classified into different levels. The front end sends leveled messages through this predefined interface.
On the back end, the FESV (Front End Supervisor) module receives the message and the TDAQ Run Control system decides whether to stop data acquisition or not (see Figure 4).

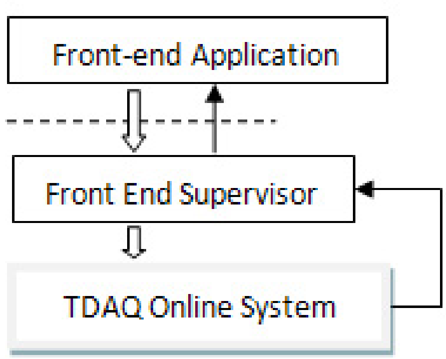

Figure 4. Front-end status report design

4 Deployment Design

DAQ system can be separated into multiple partitions to minimize correlations. Partitions which use no conflicting resources can run separately. Each partition is independent of each other logically. Each partition has its own hardware and software resource. Run control and communications are only carried out inside a partition. Parallel running depends on different partition configuration files [3].
As to LHAASO, each subsystem is defined as one partition in TDAQ software and has data flow, control and monitoring components running independently. Shutting down one of these systems does not affect the running of others. The scale of DAQ software for each subsystem is configurable according to the number of readout VME crates and the data throughput requirements.
The general deployment design is shown in Figure 5.

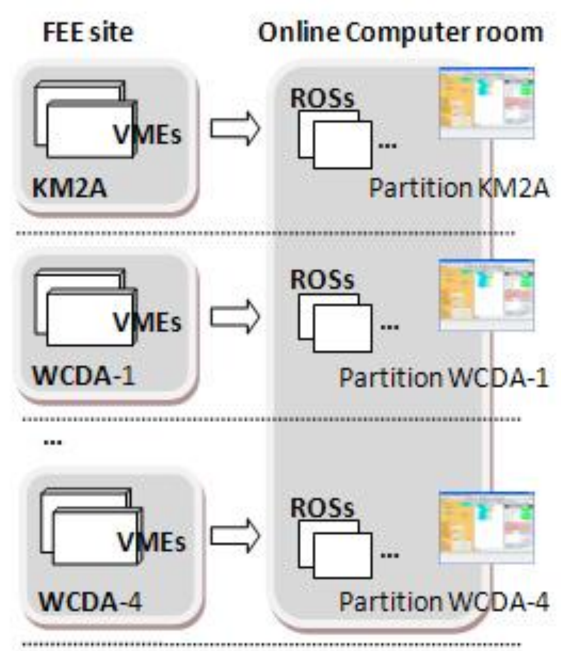

Figure 5. Software deployment

5 CONCLUSION

Architectural design of LHAASO DAQ is done based on TDAQ, after analyzing the requirements of LHAASO DAQ system. In the design, the components related to a specific FEE module are separated from the framework. So, it can be applied to different subsystems. A simulated front-end module has been developed for software test.
More effort is still needed to refine the design, and improve the performances to meet the throughput requirements.

6 Reference

[1] Huihai He, LHAASO Project: detector design and prototype, 2009, 31th ICRC
[2] Fei Li, Research and Implementation of BESIII DAQ Online Distributed Data Monitoring [D], Beijing: Institute of High Energy Physics of Chinese Academy of Science
[3] Fei Li, Xiaolu Ji, Xiaonan Li, Kejun Zhu, DAQ architecture design of Daya Bay Reactor Neutrino Experiment, 2010, Real Time Conference (RT)
[4] Xiaolu Ji, Fei Li, Mei Ye, Yuan An, Kejun Zhu, Yifang Wang, Research and Design of DAQ System for Daya Bay Reactor Neutrino Experiment, 2008, IEEE Nuclear Science Symposium Conference Record, N30-76
[5] ATLAS HLT/DAQ/DCS Group, ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report, 2003, ATLAS TDR-16