DAQ Architecture Design of Daya Bay Reactor Neutrino Experiment

Fei Li, Xiaolu Ji, Xiaonan Li, and Kejun Zhu

***Abstract*—The Daya Bay Reactor Neutrino Experiment consists of seventeen detectors distributed in three underground experimental halls. Each detector has a separate VME readout crate that contains the trigger and electronics modules. The data acquisition (DAQ) system reads data fragments from electronics and trigger modules and concatenates them into an event in each crate. The DAQ monitors the data quality, merges the event streams from each hall, and records these data to disk. The DAQ architecture is designed as a multi-level system based on the BESIII DAQ and ATLAS TDAQ systems using embedded Linux, VME bus system, advanced commercial computers and distributed network technologies. This paper presents the main DAQ design requirements, and the hardware and software architectures.**

***Index Terms*—Data acquisition systems, distributed computing, high energy physics instrumentation computing, real time systems.**

I. Introduction

The Daya Bay Reactor Neutrino Experiment is a neutrino-oscillation experiment designed to measure the mixing angle θ_{13} using anti-neutrinos produced by the reactors of the Daya Bay Nuclear Power Plant (NPP) and the Ling Ao NPP in Shenzhen, China. The basic experimental layout consists of three underground experimental halls, one far and two near, linked by horizontal tunnels as shown in Fig. 1. The distances between Daya Bay(DB) near, Ling Ao(LA) near and Far halls with surface tunnel entrance are about 300, 1600 and 2000 meters [1].

Eight identical cylindrical anti-neutrino detectors (AD) will be deployed as shown, with 192 photomultiplier tubes (PMT) in each. The ADs are deployed within water pools designed to reduce background radiation and are instrumented to detect cosmic rays. Segmented as two separate water Cherenkov detectors, the inner and outer water shield detectors contain between 123 and 212 PMTs that surround the ADs. A resistive plate chamber (RPC) detector covers each water pool. Overall there are seventeen independent detectors in the Daya Bay experiment (eight ADs, six water shields, and three RPC detectors). Each detector will be readout using an independent VME crate that contains both trigger and detector readout electronics. A custom distributed clock system provides a synchronized 10 MHz clock and an absolute time stamp for triggered events.

Manuscript received June 14, 2010; revised December 01, 2010, February 18, 2011, and April 08, 2011; accepted April 16, 2011. Date of publication July 12, 2011; date of current version August 17, 2011. This work was supported by the Ministry of Science and Technology of China (No. 2006CB808103).

The authors are with the Institute of High Energy Physics, Chinese Academy of Sciences, 100049 Beijing, China (e-mail: lifei@ihep.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2011.2158657

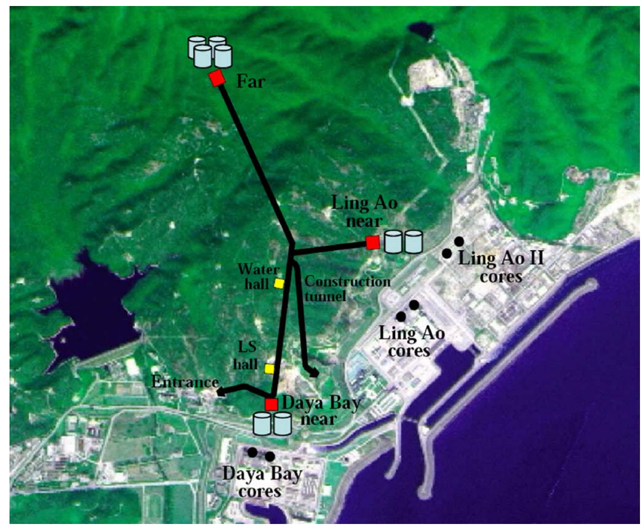

Fig. 1. Default configuration of the Daya Bay experiment.

The Daya Bay data acquisition (DAQ) system receives interrupts from the trigger module in each crate, reads-out data fragments from all modules resident in the crate with chain block transfer (CBLT) mode, and concatenates them into an event, crate by crate in parallel. Then DAQ transmits the data to back end to do event monitoring and event stream merging which orders the events using the trigger time stamp. The DAQ system also provides an interactive interface to the electronics, trigger and calibration systems. The DAQ system design impacts the capabilities and dependability of the overall integrated experiment.

II. System Requirements

A. Run Control Requirement

Since the detectors are independent, the experiment requires multiple independent DAQ systems in each hall. The three experimental halls are far away from the surface control room which is located outside of tunnel. The run control should be configurable and flexible, allowing it to work seamlessly for data taking controlled from the surface or in a detector hall. The run control should allow both global operation of all detector systems in all three detector-halls, and operation of sub-sets of detectors whenever debugging or commissioning is required. The system should also allow for multiple run-control operations in parallel in any detector hall so that debugging and commissioning can be conducted simultaneously in more than one location.

TABLE I
SUMMARY OF RUN MODE REQUIREMENTS

Run mode	AD	Water Pool	RPC
Physics	Y	Y	Y
Electronics Diagnosis	Y	Y	N/A
Pedestal	Y	Y	N/A
AD Calibration	Y	Physics	Physics
Water Shield Calibration	Physics	Y	Physics
Mineral Oil Monitoring	Y	Physics	Physics

Y means this detector will work according to the corresponding run mode.

B. Run Mode Requirements

The DAQ should be capable of taking different data types from detectors running together in different run modes. Table I shows the summary of run mode requirements. The DAQ also needs to automatically control data taking, assort and exchange parameters with external interactive systems during calibration and monitoring running modes.

C. Data Readout and Processing Requirements

Event building is executed within each VME crate. There are two types of VME detector readout electronics, one for PMTs (ADs and water shield detectors) and another for the RPC detectors. All readout electronic modules are custom modules designed and manufactured by engineers of the Daya Bay collaboration, requiring special integration and debugging with the DAQ.

All data created within a hall should be merged. Merging should be configurable for any combination of detectors and different run modes. Furthermore, the events should be sorted in the merged stream by the time stamp read from the trigger module. A policy to name the merged stream is needed. Finally, event data must be recorded to a disk array.

Raw event data must be processed and analyzed online for real time monitoring of the detectors and readout systems.

D. Throughput Requirements

The expected maximum physics data throughput rate is less than 500 KB/s for one crate and less than 1.5 MB/s for one hall assuming a baseline trigger rate. The total normal physics data throughput rate for all 3 halls is expected to be about 3 MB/s [1]. These estimates would increase with implementation of full waveform digitization, noisy PMTs, or implementation of additional triggers (e.g., LED calibration triggers). The system design should have sufficient flexibility to allow for background studies or noise research. Consequently, the design DAQ event rate could reach 1 kHz with a 2 kilobyte event size resulting in a throughput less than 2 MB/s per PMT detector crate. As such, the overall DAQ must be capable of managing a total experimental hall throughput of up to 35 Mbytes/s.

The DAQ system is also required to have a negligible readout dead-time. This requires fast online memory buffers that can hold multiple detector readout snapshots while the highest level DAQ CPUs perform online processing and transfer to permanent storage.

E. Other Common Requirements

Hardware parameters, electronic layout and software setup should be configurable. The DAQ also needs to provide a graphical user interface to display the running status and for monitoring using histograms. Bookkeeping should include run configuration, run parameters, run summary, running status, errors and logs. Finally, the experimental site is far away from most collaboration members such that remote operations should be possible with good stability and robustness.

III. ARCHITECTURE AND HARDWARE SYSTEM DESIGN

Based on construction experience for BESIII DAQ [2], [3], the Daya Bay DAQ architecture is designed as a multi-level system using embedded Linux, advanced commercial computers and distributed network technology. The hardware and network system had a united design and common devices with detector control and offline systems. The onsite computing system is shown in Fig. 2. The system has been designed entirely with a gigabit ethernet network. Due to the distances involved, the three experimental halls connect to the surface through single mode fibers. Double peer to peer single mode fiber cables connect the front-end and back-end networks. Double switches will be placed in each hall for redundancy and reliability.

The DAQ system can be separated into two parts: VME front-end system and back-end system. The front-end readout system is a real-time embedded system based on a VME bus. Each VME crate holds a VME system controller MVME 5500, an embedded single-board computer manufactured by Motorola. It is based on a PowerPC CPU and Universe II chip for VME bus interface. The operating system running on the controller is TimeSys, a commercial embedded real-time Linux with version 2.6.9 kernel. The system throughput can exceed 2.5 kHz with a 2 kilobyte event size, which is sufficient to meet the experimental requirements [4].

All computers except a local control PC are placed in the surface computer room for more convenient management and maintenance. The Daya Bay experiment also adopted the blade server based computing farm to construct the back end system. Computing farms are widely used in the field of high-energy physics. Blade server technology greatly increases server density, lowers power requirements and cooling costs, eases server expansion and simplifies data center management [5]. The DAQ has two x3650 servers acting as files servers for farm and data storage. Nine blade servers serve as computing nodes for data gathering and data quality monitoring.

IV. SOFTWARE ARCHITECTURE DESIGN

The DAQ system needs a distributed software solution to meet above requirements and architecture design. ATLAS TDAQ software implements a CORBA based inter-process communication framework [6]–[8]. And BESIII DAQ software is ported to the framework [9]. The Daya Bay DAQ software is designed and developed based on the ATLAS TDAQ software and BESIII DAQ software. Functionally, the DAQ software can be divided into two layers as shown in Fig. 3: the data flow software and the online software. The data flow software is responsible for all the processing of physics data, including

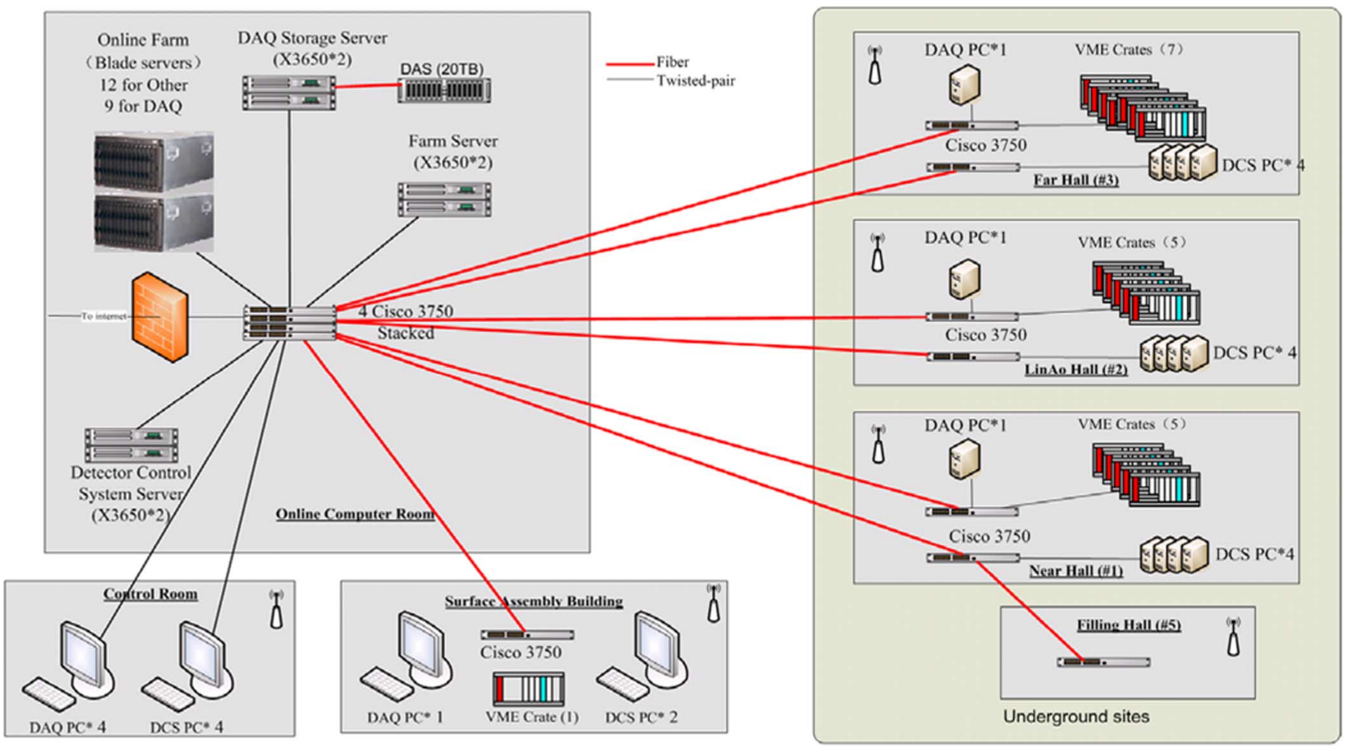

Fig. 2. Architecture of DAQ.

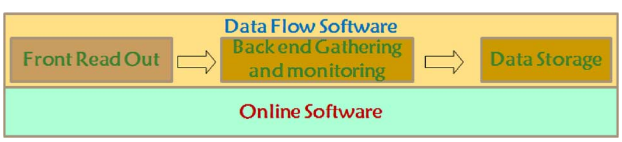

Fig. 3. Software architecture of DAQ.

receiving and transporting the data to storage. The online software is responsible for all aspects of experimental and DAQ operations and controls during data-taking. The online software provides services to data flow.

A. Online Software Design

The Online Software is customized from the ATLAS online framework and migrated to a TimeSys Linux/PowerPC embedded system environment. The ATLAS Online Software is the global system software of the DAQ system to configure, control, and share information. It is able to start, stop and synchronize the order of all programs on all processors. It is a distributed software framework based on CORBA. It provides essentially the 'glue' that holds the various sub-systems together. It does not contain any elements that are detector specific as it is used by all the various configurations of the DAQ and detector instrumentation. It also provides the interface between the human user and the DAQ system.

The Online Software architecture is based on a component model and consists of three high level components, called packages.

◦ **Control** contains sub-packages for the control of the DAQ system detectors. Control sub-packages exist to support DAQ system's initialization and shutdown, to provide control command distribution, synchronization, error handling, and system verification. The Process Manager (PMG) provides the basic process management functionality in a distributed environment. It starts, stops and monitors processes on the different DAQ hosts. To execute the requests, the PMG runs agents on all the machines and uses the information provided by the user and/or the configuration databases. A Graphical User Interface (GUI) allows the user to send commands to control as well as to monitor the system.

◦ **Databases** contains sub-packages for configuration of the DAQ system and detectors. Configuration sub-packages exist to support the system configuration description and access to it, record operational information during a run and access to this information.

◦ **Information Sharing** contains classes to support information sharing in the DAQ system. Information Sharing classes exist to report error messages, to publish states and statistics, to distribute histograms built by the sub-systems of the DAQ system and detectors, and to distribute events sampled from different parts of the experiment's data flow chain.

The interaction between the Online Software packages is shown in Fig. 4. The Control makes use of the Information Sharing and of the Databases packages. The Databases package is used to describe the system to be controlled. The Information Sharing package provides the infrastructure to obtain and publish information on the status of the controlled system, to report and receive error messages, and to publish results for interaction with the operator [10].

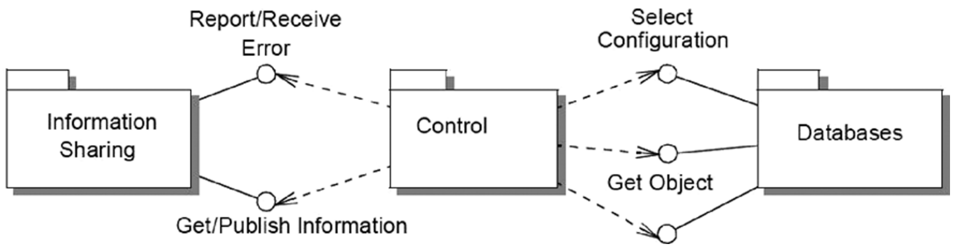

Fig. 4. Interaction diagram of online component.

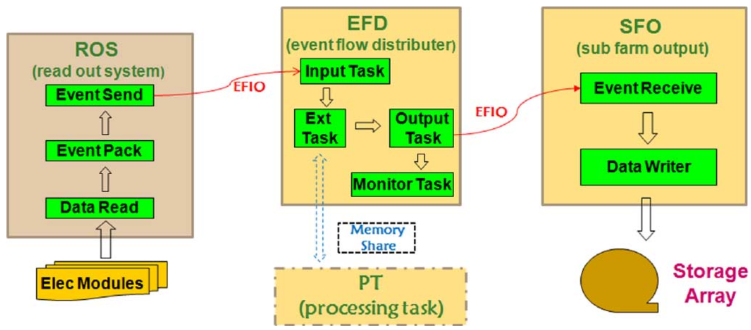

Fig. 5. Data flow component diagram.

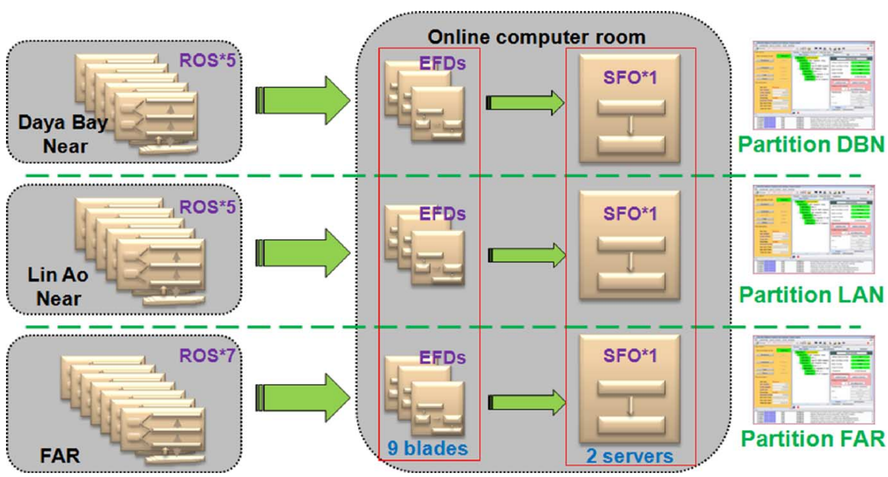

Fig. 6. Deployment design diagram.

B. Data Flow Software Design

Fig. 5 shows the data flow components diagram of the Daya Bay DAQ. It was migrated from the ATLAS TDAQ backend data flow software and BESIII front readout software [11]–[15]. The read out system (ROS) runs on a Timesys Linux/PowerPC. One ROS runs on one MVME 5500 controller. It reads data, packs events and sends the data to the back-end.

Other components run on a SLC4/X86. The Event flow distributor (EFD) can receive data from multiple ROSs through an input task. The linked ROS data will be merged together then sent to sub farm output (SFO) by EFD. The event flow input output (EFIO) service package is used to deal with data transportation among the data flow component.

The monitoring task of the EFD will parse sent data and fill data quality monitoring histograms, then publish the data to an information sharing server. The external task of the EFD can share the event data with an external process task (PT). PT is used for an option to do further data processing or analysis. Data analysis will require significant CPU resource such that EFD and PT should be run on blade servers.

SFO can also receive multi EFDs' data, merge and sort events by trigger time, then record to data files. The SFO must run on file servers for better storage performance with locally mounted disks.

C. System Deployment Design

The DAQ system can be separated into multiple partitions to minimize correlations among hardware and software of different detectors. Multiple detectors can run and be controlled together as a partition. The participants of a partition can be configurable. Each detector can be an individual partition by itself. 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 and configuration file. Run control and communications are only carried out inside a partition. Parallel running depends on different partition configuration files.

A typical partition deployment design is shown in Fig. 6. Each ROS corresponds to a VME crate and the detector. The whole system is separated into three partitions for three experimental halls. Each partition includes all ROS of one hall, several EFDs and one SFO. Several EFDs of one partition can be linked to specific ROSs according to these ROSs' throughput or detector type. All event data of one hall will be recorded to one single data file stream by a SFO.

V. Summary

Most servers and the surface network have been installed and are working well, and the first two ADs have been assembled. A full integration test was conducted onsite in surface assembly building. One VME crate housing thirteen electronic PMT readout modules and one trigger module was used to readout each of the assembled ADs separately. The DAQ operated one MVME 5500 running one ROS, one blade server running one EFD, one file server running one SFO. A maximum trigger rate of 1.05 kHz with a 10 kilobyte event size reading out noise signals was achieved. The maximum trigger rate for an event size of 2 kilobytes was 2.8 kHz. A 72-hour stability test with an approximate 1.5 kHz event rate and 2.3 kilobyte mean event size was successfully completed without any crashes.

A DAQ software test for multiple VME crates is set up in laboratory. It composes of seven software emulated ROS running on MVME 5500, seven corresponding EFD running on two blades and one SFO running on one storage server. The servers and VME crates connect by more than 2 kilometers fiber cable. The maximum event rate for an event size of 2 kilobytes can reach 8 kHz with data stream merging and time sorting now. Each ROS-EFD-SFO data flow link can handle above 1.1 kHz event rate on average.

Based on the ATLAS TDAQ and BESIII DAQ, the architecture design of the Daya Bay DAQ was achieved. The single subsystem (one PMT detector) DAQ software was developed rapidly with minimal manpower and reusing ATLAS and BESIII software. Preliminary test results show that the DAQ performed well and the architecture design meets current experimental requirements. More detail design and development are ongoing. Further validation testing is planned in the coming installation and commissioning phases.

References

[1] Daya Bay Collaboration, "Daya Bay Project Technical Design Report (v2)," Jan. 22, 2008.

[2] BESIII Collaboration, "The construction of the BESIII experiment," *Nucl. Instr. Meth. A*, vol. 598, no. 1, pp. 7–11, Jan. 2009.

[3] M. Ablikim *et al.*, "Design and construction of the BESIII detector," *Nucl. Instr. Meth. A*, vol. 614, no. 3, pp. 345–399, Mar. 11, 2010.

[4] X. Ji *et al.*, "Research and design of DAQ system for Daya Bay reactor neutrino experiment," in *Proc. IEEE Nuclear Science Symp. Conf. Record*, Dresden, Germany, 2008, pp. 2119–2121, N30-76.

[5] F. Li *et al.*, "Design and implementation of BESIII online farm," in *Proc. IEEE Nuclear Science Symp. Conf. Record*, Dresden, Germany, 2008, pp. 2112–2113, N30-72.

[6] R. Jones *et al.*, "Applications of CORBA in the ATLAS prototype DAQ," *IEEE Trans. Nucl. Sci*, vol. 47, pp. 331–335, Apr. 2000.

[7] ATLAS DAQ Backend Software Group, "ATLAS DAQ Back-End Software High-Level Design," 1998.

[8] ATLAS TDAQ group, "ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report," 2003.

[9] F. Li *et al.*, "Distributed inter process communication framework of BESIII DAQ online software," (in Chinese) *Nucl. Electron. Detection Technol.*, vol. 26, no. 5, pp. 605–608, Sep. 2006.

[10] I. Alexandrov *et al.*, "Online software for the ATLAS test beam data acquisition system," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 3, Jun. 2004.

[11] H. P. Beck *et al.*, "The base line data flow system of the ATLAS trigger and DAQ," *IEEE Trans. Nucl. Sci.*, vol. 51, pp. 470–475, Jun. 2004.

[12] C. Haeberli *et al.*, "ATLAS TDAQ data collection software," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 3, pp. 578–584, Jun. 2004.

[13] H. Zhu *et al.*, "The implementation and application of statecharts in real-time system in high energy physics," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 3, pp. 1032–1038, Jun. 2006.

[14] F. Li *et al.*, "Online data processing and analyzing in BESIII DAQ," in *Proc. 16th IEEE-NPSS Real Time Conf. Record*, Beijing, China, 2009, pp. 458–460.

[15] Q.-X. Zhang *et al.*, "The control of data flow in event building of BESIII DAQ software," (in Chinese) *Nucl. Electron. Detection Technol.*, vol. 26, no. 3, pp. 302–306, May 2006.