

DAQ Architecture Design of Daya Bay Reactor Neutrino Experiment

Fei Li, Xiaolu Ji, Xiaonan Li, Kejun Zhu

Abstract—The main task of the data acquisition (DAQ) system in Daya Bay Reactor Neutrino Experiment is to record antineutrino candidate events and other background events. There are seventeen detectors in three sites of Daya Bay experiment. Each detector will have a separate VME readout crate that contains the trigger and DAQ electronics modules. DAQ system reads event data from front end electronics modules, concatenates data fragments of the modules and packs them to a subsystem event, then transmits to the backend system to do data stream merging, monitoring and recording.

The DAQ architecture design is a multi-level system using advanced commercial computer and network technology. The architecture requires multiple independent or part joint DAQ systems at each site. The run control will be configurable and flexible; it should allow both global operation of all detector systems in all three sites, and operation of parts of detectors whenever debugging or commissioning is required. The system should allow each subsystem in any detector site to start its own run control in order that debugging and commissioning can be conducted in parallel.

This paper presents the DAQ main design requirements, the system architecture design and the software architecture design.

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 of Daya Bay consists of three underground experimental sites, one far and two near, linked by horizontal tunnels as shown in Fig 1.



Fig 1. Default configuration of the Daya Bay experiment.

Four anti neutrino detector (AD) are deployed in water pool at the far site and two ADs at each of the near sites. The water pool is separated as two water Cherenkov detectors, inner and outer water shield detectors, around the ADs. And one RPC detector is covered the whole water pool at each site. Overall there are seventeen detectors of Daya Bay experiment. Each detector will have a separate VME readout crate that contains the trigger and electronics modules.

DAQ system is the interactive interface with front-end electronics, trigger, detectors and calibration system. It is used to read data from the front-end electronics modules resided in the VME crates, concatenate the data fragments into a subsystem event, crate by crate in parallel, then transmits to the backend system to do data stream merging, monitoring and recording.

DAQ system has direct influence to the capability and dependability of the integrated experiment system. This paper presents the DAQ main design requirements, the system architecture design, the hardware system design and software architecture design.

II. SYSTEM REQUIREMENTS

A. Run Control Requirements

There are independent front-end read out subsystems for all detectors in three experiment sites. The experiment requires multiple independent DAQ systems at each site. 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. And run control should allow both global operation of all detector systems in all three detector sites, and operation of parts of detector sites whenever debugging or commissioning is required. The system should allow each subsystem in any detector site to start its own run control in order that debugging and commissioning can be conducted in parallel.

B. Run Mode Requirements

DAQ should take different data from the detectors run together when different run mode. TABLE I shows summary of run mode requirements. DAQ need automatically control data taking, assort and exchange parameters with external interaction systems for calibration modes.

TABLE I. SUMMARY OF RUN MODE REQUIREMENTS

| Run mode | AD | Water Pool | RPC |
|-----------------------|----|------------|---------|
| Physics | Y | Y | Y |
| Electronics Diagnosis | Y | Y | - |
| Pedestal | Y | Y | - |
| AD Calibration | Y | Physics | Physics |

Manuscript received May 28, 2010.

F. Li et al. are with the Institute of High Energy Physics, Chinese Academy of Sciences, P.O.Box 918-1 100049 Beijing, China (telephone: +86-10-88236046, e-mail: lifei@ihep.ac.cn)

| | | | |
|--------------------------|---------|---------|---------|
| Water Shield Calibration | Physics | Y | Physics |
| Mineral Oil Monitoring | Y | Physics | Physics |

C. Data Processing Requirements

Event buildings are only executed in each VME crate. DAQ should read out data fragments of all modules in each crate and building them to a subsystem event. There are two type VME electronics system to read out, one is PMT for ADs and water shield detectors with three type electronics modules, another is RPC with two types electronics modules.

Then the event data should be gathering, merging and recording to disk. All data of one site detects should be merged together. The participants should be configurable to switch merging or not. And the event should be sorted by trigger time stamp in merged stream. A regulation to name the merged stream is needed.

The raw event data must also be processed and analyzed at online for real time monitoring the detector and data taking status.

D. Throughput Requirements

TABLE II shows data throughput estimation for each subsystem in each detector site for the baseline trigger scheme. It is estimate that the expected data throughput rate is <1.5 MB/second/site. The total data throughput rate for all 3 sites is estimated to be about 3MB/s. These estimates would increase with full waveform digitization or additional triggers. So the DAQ design event rate should be reached 1 kHz when all channels of one AD are fired. Then the throughput will less than 2MB/s (PMT crate) when event size is less than 2KB.

TABLE II. Summary of data rate estimations

| Detector | Description | Trigger Rates (Hz) | | | Occ | Ch No. | data rate (kB/s) |
|--------------------|---------------|--------------------|--------|---------|------|-----------|------------------|
| | | DB | LA | Far | | | |
| \bar{p} module | cosmic- μ | 36 × 2 | 22 × 2 | 1.2 × 4 | 100% | 192 | 1408 |
| | Rad. | 50 × 2 | 50 × 2 | 50 × 4 | | | |
| Inner water shield | Rad & noise | 50 | 50 | 50 | 10% | 123/169 | 31 |
| | cosmic- μ | 250 | 160 | 15 | 70% | | 526 |
| Outer water shield | Rad & noise | 50 | 50 | 50 | 10% | 168/212 | 41 |
| | cosmic- μ | 250 | 160 | 15 | 30% | | 309 |
| RPC | Rad. | 2000 | 2000 | 3000 | 10% | 32/module | 217 |
| | cosmic- μ | 186 | 117 | 11 | | | 11 |
| site totals | (kB/s) | 1042 | 788 | 714 | | | 2544 |

kB/s = 1000 bytes per second

Meanwhile, DAQ system is required to have a negligible readout dead-time (<0.5%). 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. The system design should have sufficient flexibility.

E. Other Common Requirements

Hardware configuration and description, software running parameter and so on should be configurable. DAQ need also provide a graphic user interface for running status and histograms monitoring.

Bookkeeping should include run configuration, run parameters, run summary, running status, error and logs.

III. ARCHITECTURE AND HARDWARE SYSTEM DESIGN

The DAQ architecture design is a multi-level system using advanced commercial computer and network technology as

shown in Fig 2. According to site location, DAQ system can be separated to two parts: VME front-end system, backend system.

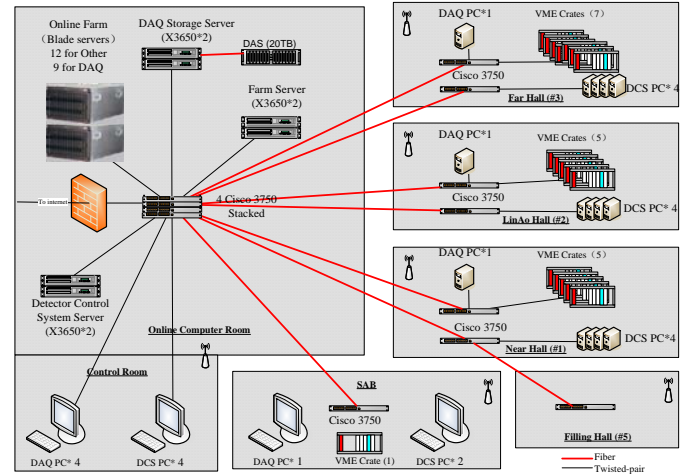


Fig 2. Architecture of DAQ

The front-end readout system is a real-time system based VME bus. Each VME crate holds a VME system controller, some front-end readout modules and one trigger module. The VME controller, an embedded single-board computer, is used to collect, preprocess, and transfer data. It is based on a PowerPC CPU and Universe II chip for VME bus interface. TimeSys, the operation system run on controller, is a commercial embedded real-time Linux with version 2.6.9 kernel. The readout system is sufficient to meet the bandwidth requirements^[1].

Daya Bay DAQ also adopted the blade server based computing farm to construct back end system. Computing Farms are used widely in the field of high energy physics. Blade server technology greatly increases server density, lowers power and cooling costs, eases server expansion and simplifies datacenter management^[2].

Double peer to peer fiber cables connect the front end and back end network. And double switches were placed in each site for redundancy and reliability.

IV. SOFTWARE ARCHITECTURE DESIGN

DAQ software of Daya Bay is designed based on the BESIII DAQ software and ATLAS TDAQ software^[3]. Functionally, the DAQ software can be divided into two layers as Fig 3: data flow software and online software. The data flow software is responsible for all the processing of physics data, receiving and transporting the data to storage. The online software is responsible for all aspects of experimental and DAQ operations and control during data-taking. And the online software provides services to data flow.



Fig 3. Software architecture of DAQ

A. Online Software

The Online Software is customized from ATLAS online framework and migrates to Timesys linux/PowerPC

embedded system environment. 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 and detectors. Control sub-packages exist to support DAQ system initialization and shutdown, to provide control command distribution, synchronization, error handling, and system verification.
- **Databases** contains sub-packages for configuration of the DAQ system and detectors. Configuration sub-packages exist to support 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.

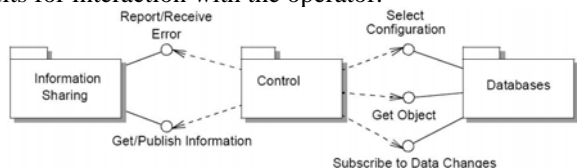


Fig 4. Interaction diagram of online component

B. Data Flow Software

The Fig 5 shows data flow component diagram of Daya Bay DAQ, it is migrated from ATLAS TDAQ backend data flow software and BESIII front readout software^[4,5]. Read out system run on Timesys real time Linux/ PowerPC, read data, pack to event and send to back-end. Other components run on SLC4/X86. Event flow distributor (EFD) can receive data from multi ROSs through input task. The linked ROS data will be merged together then send to sub farm output (SFO). SFO can also receive multi EFD data, merge and sort event by trigger time, then record to data file.

The monitor task of EFD will parse sent data and fill some data monitoring histograms, than publish to IS server. The external task of EFD can share the event data with external process task (PT). PT is used for an option to do further data process or analysis.

The event flow input output (EFIO) service package is used to deal with data transportation among the data flow component.

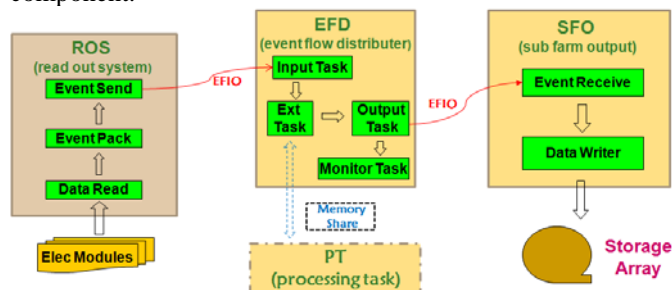


Fig 5. Data flow component diagram

C. System Deployment Design

DAQ system can be separated into multi partitions to minimize correlations. Multi subsystems can run and control together as a partition. The participants can be configurable. Each subsystem can be an individual partition too. 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. Running in parallel depend on different partition configuration files.

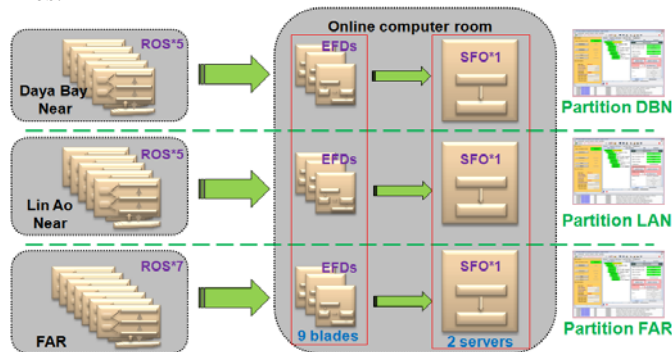


Fig 6. Deployment design diagram

A typical partition deployment design show as Fig 6. Each ROS correspond to a VME crate and the detector. Whole system separated to tree partitions for three experiment sites. Each partition includes all ROS of one site, several EFD and one SFO. Several EFD of one partition can link to specific ROS according to these ROSs' throughput or detector type. All event data of one site will be recorded to only one data file stream by SFO.

V. SUMMARY

Based on ATLAS TDAQ software and BESIII DAQ software, the architecture design of DYB DAQ had been archived. And the single subsystem (one detector) DAQ software version had been developed and validated during the AD mini dry test in this year beginning. The test result shows a well performance and the design can meet current experiment requirements. But many detail design and development work need to do yet.

REFERENCES

- [1] Ji, Xiaolu et al, "Research and design of DAQ system for daya bay reactor neutrino experiment", 2008 IEEE Nuclear Science Symposium Conference Record, N30-76, P 2119-2121
- [2] Li, Fei et al, "Design and Implementation of BESIII Online Farm", 2008 IEEE Nuclear Science Symposium Conference Record, N30-72, P 2112-2113
- [3] <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerDAQ>
- [4] Beck, Hans Peter et al, "The base-line DataFlow system of the ATLAS Trigger and DAQ", IEEE Transactions on Nuclear Science, Volume 51, Issue 3I, June 2004, P 470-475
- [5] Li, Fei et al, "Online data processing and analyzing in BESIII DAQ", 2009 16th IEEE-NPSS Real Time Conference - Conference Record, p 458-460