ELSEVIER

Journal logo

A DAQ processing system for the KM3NeT shore station

E.G. Anassontzis [a], A. Belias [b], A.Fotiou [b], S. Koutsoukos [a], A. Papaikonomou [a,*], L.K. Resvanis [a,b]

[a] *Physics Department, University of Athens, Greece*

[b] *NESTOR Institute for Astroparticle Physics, National Observatory of Athens, 24001 Pylos, Greece*

On behalf of KM3NeT consortium

Elsevier use only: Received date here; revised date here; accepted date here

Abstract

The framework for the KM3NeT shore DAQ system is based on the Internet Communications Engine, ICE. The task of the system includes control, data acquisition and processing, pre-selection of events for storage and further processing and on-line monitoring of the KM3NeT neutrino telescope. We describe the overall shore DAQ system and discuss in particular the processing, storing and monitoring tasks. We present our experience with implementations for the DAQ systems which have been built to support the foreseen demonstrator of the KM3NeT shore station.

Neutrino Telescope; Data Acquisition; Software Architecture;

1. Introduction

The deep-sea neutrino telescope in the Mediterranean Sea being prepared by the KM3NeT consortium [1], will contain thousands of optical sensors to readout. Each optical sensor module, the Digital Optical Module (DOM) [2] will be equipped with 31 photomultiplier tubes (PMTs) to detect Cherenkov light from neutrino-induced muons. The PMTs will be housed in pressure resistant glass spheres. Moreover, each DOM, will be equipped with a hydrophone, a LED beacon, an inclinometer and a compass module, and will contain a power converter board and readout electronics [3] with the interfaces to the power network and the photonic data transmission network [4], and it will be connected to the shore station optically and electrically.

To realize a km^3-scale neutrino telescope, about 10000 DOMs will be operated under a central control from the shore station and the collected data from each DOM must be on-line processed in the shore station.

* Corresponding author.; e-mail: antonios.papaikonomou@cern.ch

Concepts of the on-shore DAQ have been presented in earlier studies for KM3NeT [5][6] . Following the readout requirements, we present here the software for the Data Acquisition (DAQ) processing system under development for KM3NeT and early experience with the prototype.

2. KM3NeT shore DAQ requirements

The main concept of the KM3NeT's DAQ system, is to retrieve all data collected from the neutrino telescope DOMs, minimizing the off-shore trigger activity. All data are sent to shore, where the filtering and main processing occurs.

Time over threshold data from PMTs which are digitized in the DOM and data from the other installed sensors mentioned in section 1, will be routed in time slices. The data in each time slice is sent to an array of processing units. This includes also the information collected by Earth and Sea sciences' instruments.

The DAQ system is designed such that it allows for the online processing of the collected raw data, so that extraction of physics event candidates and calibration constants will be feasible during the acquisition procedure. At the same time, the results of the processes, accompanied by the associated raw data, will be stored in the shore station.

The raw data will include information about event candidates, the sensors (for calibration purposes) and the calibration parameters. An on-shore single command control structure will control and operate the telescope.

The software design permits a multiple online monitoring at various levels, such as physics event and hardware monitoring. The possibility to monitor the system remotely through the web network is included as well.

Furthermore, all local databases are expected to be updateable. In addition, access to the data will be provided for remote facilities on request for further off-line processing and long term storage.

2.1. Processing requirements

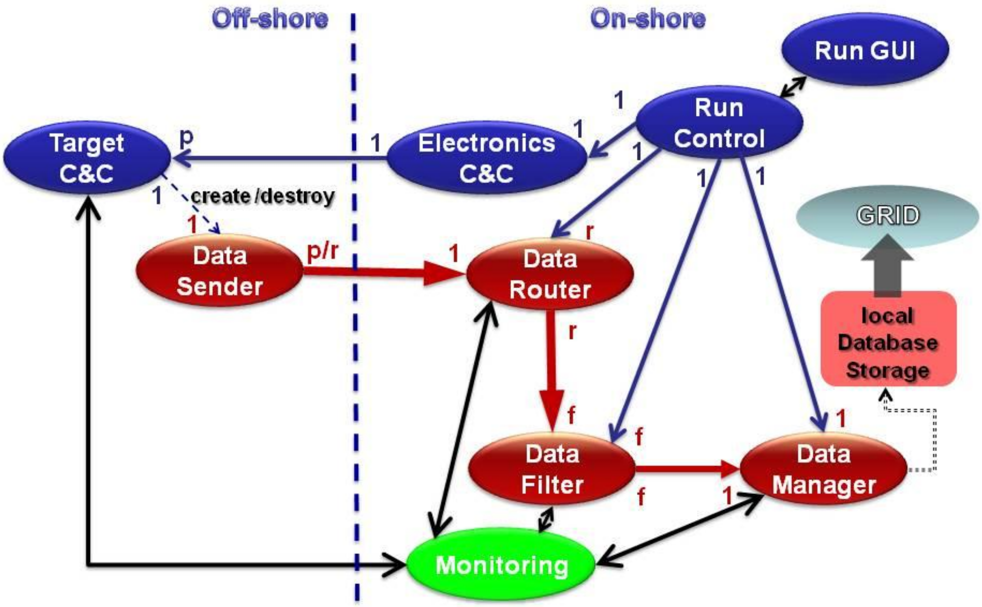

Fig. 1: Task/Object diagram of KM3NeT's DAQ system. The objects in ellipses stand for software tasks. The arrows show the direction of the communication/data flow. The symbols at the edge of the arrows indicate the type of the communication, for example: 1 Run Control agent communicates with "r" Data Routers, whereas "r" Data Routers communicate with "f" Data Filters. Main design by S. Anvar [7], inspired by ANTARES and augmented for KM3NeT.

The on-line processing has to satisfy the following main requirements:

i. be able to accommodate the bandwidths for all data to shore of all DOMs, acoustic sensors and various slow control sensors

ii. be flexible in accommodating various event selection algorithms

iii. have scalability through modular functionality
iv. have long term maintainability

Solutions to these matters have already been found. The use of a 1 Gbit Ethernet takes care of i., while the use of the Internet Communications Engine (ICE) middleware [8], described section 3.2, satisfies ii. and iii. Point iv. can be arranged by using off-the-shelf, mass market hardware components.

2.2. DAQ system

The DAQ processing system consists of chain of interacting software tasks, shown in Fig. 1. Each task is designed to serve a certain purpose. For example, a task will distinguish between operations taking place on-shore and those happening off-shore. Those tasks are interconnected through TCP/IP and have the following main purpose:

The Data Router's function is to propagate the data of the same time interval via a buffering system to the processing farm and then to the database and archiving systems. The routing takes place through a dedicated switch fabric.

The Data Filter is the first place where all data for a particular time frame can be processed in one memory. It will be able to apply different filter algorithms on the same time slice. Furthermore, it will process successive snapshots of the data from the whole telescope, each covering a time span of 10-100 ms, and provide the first results; events of interest based on algorithms of track finding, high data rates and other parameters. In addition, it will have an external trigger input and an built-in option to generate a trigger for an external alert, in case hints of interesting events have to be examined closer as soon as possible e.g. by another experiment.

The tasks of the Monitoring in the DAQ are: (a) to display information useful for the operation and control of the telescope and (b) to check on the status of the DAQ itself, the telescope's hardware and (c) to display the outcome of the filters.

The Data Manager's main function is to administer the storage for the results of each filter. In addition to the results of a data filter, the DAQ will save the raw data which were used in the data filter including the data from an extended time window, in case interesting information appears on the edge of the data sets. The data will then be stored locally and subsequently distributed to remote computing centers for offline analysis.

The basic part of the on-shore tasks is the Run Control unit. Managed via a Graphical User Interface (GUI), it communicates with several components of the on-shore DAQ system. There is a direct connection to all Data Routers, all Data Filters and the Data Manager, as well as to the Control and Configuration (C&C) Electronics. The latter communicate with groups of C&C of all placed instruments off-shore. These controllers, in turn, are able to create a "one-to-one" communication node to the tasks responsible for sending all the data to shore, the Data Senders in each DOM that could be monitored simultaneously from shore. A Target-C&C task and a Data Sender Task are present in each DOM.

Each group of Data Senders dispatches the data to a Data Router, which then propagates the information to a set of Data Filters. Last but not least, all data end up in the Data Manager, a task that reaches the final decision whether the data is worth storing in the local database storage system. Both the Data Routers and the Data Filters, as well as the Data Manager, can be monitored online.

It is important to emphasize that the local database storage will be accessible over the network and data collected will be provided to other facilities using the GRID infrastructure.

3. Shore DAQ system prototype

The minimal configuration of the shore DAQ system consists of a Run Control, a Data Router, a Data Filter, a Monitoring and a Data Manager. The last four components will be the subject of discussion here; one may read about the Run Control in [7].

3.1. Hardware implementation

In order to accomplish the tasks of the shore DAQ software, it is required to arrange a minimum set of equipment, that is a single multi-core PC, an 1 Gbit Ethernet card and a storage device; in this case was a Network Attached Storage (NAS), supporting an 1 Gbit Ethernet connection over 2 ports.

However, to fulfill the needs for a large volume neutrino telescope, multiple hardware systems will have to be used at various stages according to the function of each software task. Here we present the hardware system used for a demonstrator model with 4 DOMs.

The function of the Data Router, the Data Filter and the Monitoring is bound to the use of a PC server farm, consisting of multi-core PCs with multiple Network Interface Cards (NICs), a manageable Gbit switch infrastructure. In addition, a dedicated database for expert monitoring and APIs for familiar tools, like ROOT [9] will be useful. For the Data Manager a network attached storage volume, like NAS, will be implemented in the system, supporting multiple RAID levels and having a dedicating uplink connection to remote computing centers.

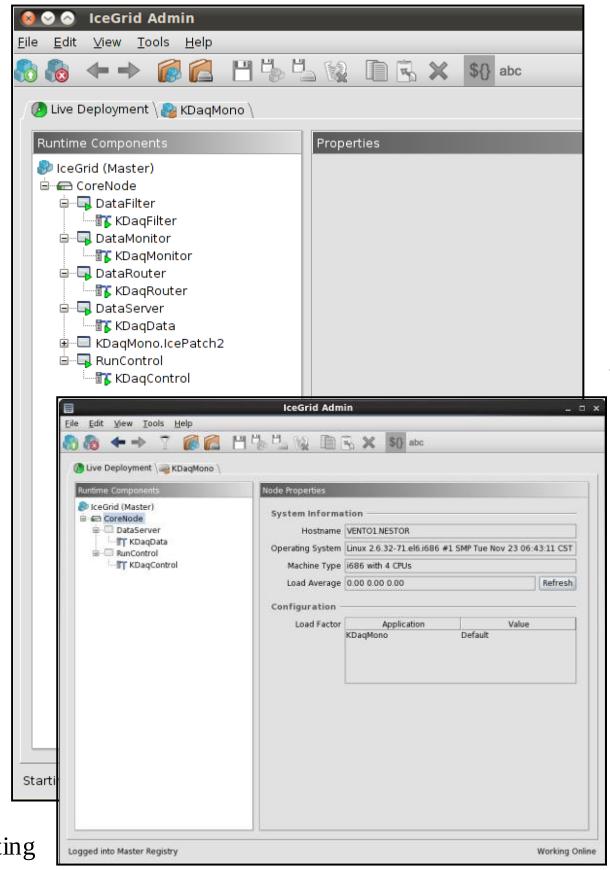

Fig. 2: Using the ICEGRID administrator to control the software tasks

3.2. Middleware implementation

As already mentioned, the design of the DAQ chain is based on the ICE middleware. It is a modern, widely deployed object-oriented middleware with multi-language support. It applies both to the needs of the different software tasks as well as to the hardware of the prototype and to the hardware intended to be used in the future telescope.

ICE provides the possibility of easily creating objects that stand for the software tasks of the DAQ chain. These objects function as separate entities performing their dedicated task on a separate node and communicate with each other over a TCP/IP connection. A GRID interface, called IceGrid and included in the middleware, provides the possibility to administrate all different nodes. This allows the selection of dedicated CPUs for a specific software task and allowing for load-balancing to avoid a system overload. Fig. 2 shows an example of administrating such an IceGrid node via a GUI.

4. Experience in the lab

In the lab, steps forward to testing the prototype have been made, using the ICE middleware on 1 CPU unit. The current implementation of the framework design, realized by [10][11] for KM3NeT, has been used for laboratory tests at the NESTOR institute. The continuation of this effort is used to implement the Data Router, the Data Filter, the Monitoring and the Data Manager. Fig. 2 demonstrates the instantiation of these objects, which can hold different algorithms.

Using a simulator, it was achieved to route pseudo-data, create data sets and store them locally.

By means of a java application, one is able to look at the acquisition files.

This work is in progress and in the near future we will proceed to tests with actual DOMs.

5. Outlook

The design of the KM3NeT's DAQ chain is flexible, scalable and maintainable over a long time. Compliance with on-shore DAQ requirements is reached. The software infrastructure will be based on the ICE middleware, currently in use by KM3NeT.

So far our experience showed that implementing ICE-based software for the shore DAQ is feasible and capable to accomplish the KM3NeT tasks. The basic steps in accomplishing the chain have been made.

The next steps are to complete a Data Router, a Data Filter, a Data Manager and a Monitoring, suitable for the KM3NeT telescope and have it stress tested with a data generator.

In the near future, first tests with actual DOM hardware will take place.

Acknowledgments

This study was supported by the European Commission through KM3NeT-PP, EU FP7, Grant agreement no.: 212525.

References

[1] KM3NeT Consortium, http://www.km3net.org.
[2] KM3NeT TDR: ISBN 978-90-6488-033-9.
[3] Talk by F. Louis (these proceedings).
[4] Talk by J. Hogenbirk (these proceedings).
[5] A. Belias, NIM A 602 (2009) 143.
[6] A. Belias et.al, NIM A 626-627 (2011) S176.
[7] Talk by S. Anvar (these proceedings).
[8] "The Internet Communications Engine". ZeroC, Inc., http://www.zeroc.com/index.html.
[9] Rene Brun and Fons Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.
[10] S. Anvar, Nuclear Science Symposium Conference Record, 2008. NSS '08. IEEE: 10.1109/NSSMIC.2008.4775103.
[11] F. Chateau, S. Anvar, Real-Time Conference, 2007 15th IEEE-NPSS: 10.1109/RTC.2007.4382768.