The Antares Computing Model

Claudio Kopper (for the Antares Collaboration)

NIKHEF, Science Park 105, 1098 XG Amsterdam, The Netherlands

Abstract

Completed in 2008, ANTARES is now the largest water Cerenkov neutrino telescope in the Northern Hemisphere. Its main goal is to detect neutrinos from galactic and extra-galactic sources. Due to the high background rate of atmospheric muons and the high level of bioluminescence, several on-line and off-line filtering algorithms have to be applied to the raw data taken by the instrument. To be able to handle this data stream, a dedicated computing infrastructure has been set up.

The article covers the main aspects of the current official Antares computing model. This includes an overview of on-line and off-line data handling and storage. In addition, the current usage of the "IceTray" software framework for Antares data processing is highlighted. Finally, an overview of the data storage formats used for high-level analysis is given.

Keywords:

1. Overview

In order to achieve its main science goal of detecting neutrinos from galactic and extra-galactic sources, the Antares neutrino telescope relies on a dedicated infrastructure of software and hardware that allows for real-time data recording in the presence of a continuous background of atmospheric muons and bioluminescence. This article presents the infrastructure that has been set up to

Email address: `claudio.kopper@nikhef.nl` (Claudio Kopper (for the Antares Collaboration))

sustain a high data rate and minimize down-time for the neutrino telescope. It also details the analysis software tools and data storage formats that are being used to study the recorded data off-line.

2. Data Acquisition System

The ANTARES data acquisition system [1] uses an all-data-to-shore scheme, that sends all incoming data to the detector shore station, where it is triggered and filtered. This keeps the actual detector more simple and allows for addition of new trigger algorithms after the initial detector design (if necessary long after the detector has been built).

The DAQ itself consists of multiple software packages, implemented in either C++ or Java which are communicating using the "controlhost" [2] infrastructure. Controlhost provides a high-performance subscription-based messaging layer built on top of TCP/IP. The on-shore part is running on standard Linux systems, whereas the off-shore hardware runs the vxWorks real-time operating system.

All signal digitization is done off-shore using a custom designed front-end-electronics board [3] before being sent to shore using a controlhost stream. There, the incoming continuous data stream is split into so-called "time slices" of 104.85 ms, buffered and then sent to individual nodes of a dedicated computer cluster in a round-robin fashion for triggering. This allows for a high level of parallelization without introducing too much dead-time.[1]

ANTARES uses several different trigger algorithms, from simple multiplicity triggers to more advanced algorithms using full-sky scans and even simple track-fitting algorithms. These algorithms are dynamically turned on and off according to the bioluminescence background rate to keep the post-trigger bandwidth and CPU load within a sensible range. A special trigger is implemented

[1] Each trigger node only sees the specific time slices sent to it, so events that occur during the change-over between two time slices are lost or distorted. By choosing a rather long time slice duration, this effect is kept to a minimum.

where during short time-spans the full stream of incoming data can be written to disk. This mode is currently being triggered by incoming gamma ray burst alerts. It is also very useful for systematic studies of the detector behavior.

All triggered and filtered data is finally saved in ROOT-files and stored for off-line processing on a hybrid disk/robotic tape storage system (HPSS [4]) at the "Centre de Calcul de l'IN2P3" located in Lyon, France. Data transmission from the shore station to Lyon is done on a run-by-run basis using automated scripts, with each run having a size of about 2 GB.

3. Off-line Processing

The Lyon Computing Center also provides a large fraction of the CPU resources used in Antares. This is advantageous since the main data storage facility is located at the same place where the central processing is done.

As most of the event filtering has already been performed during data taking, there is only a single post-processing level applied off-line after that. It adds an improved calibration and uses more detailed geometry data. The latter is necessary since reconstructing the constantly changing detector geometry from acoustic pinger and compass data is not done in real time and thus isn't available during data-taking. All re-calibrated data is then subjected to a list of reconstruction algorithms, including different muon track reconstructions, cascade-finding algorithms and several energy estimators. All these algorithms are combined using the "IceTray" software framework, originally developed by the IceCube collaboration. [5] The framework allows setting up an analysis chain composed of several modules responsible for the various steps analysis steps. Each module may use the output of its preceding modules. Adding new algorithms for track reconstruction is very easy and does not require re-compiling code or changing existing modules. This allows to improve the analysis chain very easily while keeping algorithms that are tested and known to work unchanged.

An intermediate output containing all results is then saved in IceTray's internal “i3” format for further conversion into “n-tuple” files for final analysis.

4. Analysis level

In addition to the framework-specific file “i3” file format, the output data after off-line processing are saved in formats more suited for analysis. There are two different formats used for this at the moment, one of them being a ROOT-file based DST format inspired by code from Auger, called “antDST”. This format has the advantage that a large set of tools to work with it is already existing and that it is well documented.

The other format (“tableio”) in use is available directly from within IceTray and allows for easy booking of data into either ROOT-files or files in the standardized hdf5 [6] format. It can be easily extended to include new data structures without changing the overall file format. This is something that is slightly harder to do with the monolithic antDST code. Having hdf5 available as an output format allows for processing of data without any dependencies on ROOT. This might for example be advantageous for long-term data storage, as ROOT is constantly under development and has a large set of dependencies. Older ROOT versions might not always compile on all current platforms.

Both file formats are indexed and thus provide quick access to data. They also allow applying various selection criteria in a highly optimized manner, something that is essential during data analysis. An example of this would be retrieving all events that pass a certain quality cut. Both file formats allow doing this within a few seconds for files with sizes on the order of several gigabytes.

5. Optional on-line analysis

The data stream from the detector can optionally be used for on-line analysis in close-to real time. On-line analysis can be useful for target-of-opportunity style analyses, where the neutrino telescope triggers other observations. It can also be an important tool for detector monitoring, allowing for on-line event displays.

Using the subscription-based controlhost interface, analysis clients can have the triggered and filtered data stream sent to them at any time. This works for local clients at the shore stations as well as for remote clients that receive the data using a TCP/IP tunnel. In addition to a file-based reader, the “IceTray” framework provides an on-line interface for the real-time data stream. Given sufficient CPU power, any off-line analysis could thus be run on-line without the need for modifications.[2]

Events are typically sent with a delay of several seconds due to buffering and the time needed for triggering.

6. Simulation

Monte-Carlo event simulation is done using a variety of tools responsible for generating events, propagating particles through the detector medium, generating Cerenkov light and propagating this light to the detector’s optical modules. [8] All of these tools write a standard text-file format containing their simulation results, which is then used for further processing. At this point, all simulation tools used in Antares are stand-alone tools and do not depend on the IceTray framework.

The DAQ trigger processor code, in turn, has a special simulation mode that allows it to read the output files generated by those program. It is also responsible for adding simulated noise hits from bioluminescense and for applying the PMT and electronics simulation. It finally applies the trigger and saves all events in a format compatible with the on-line output format. This, in turn, can then be read by IceTray. From this point on, the reconstruction chain is identical to the off-line processing chain used for real data. Monte-Carlo truth information is retained throughout this processing and is available per event in the final output files.

[2] Antares uses a fast reconstruction algorithm especially suited for real-time reconstruction [7], but the interface allows for any existing reconstruction to be run.

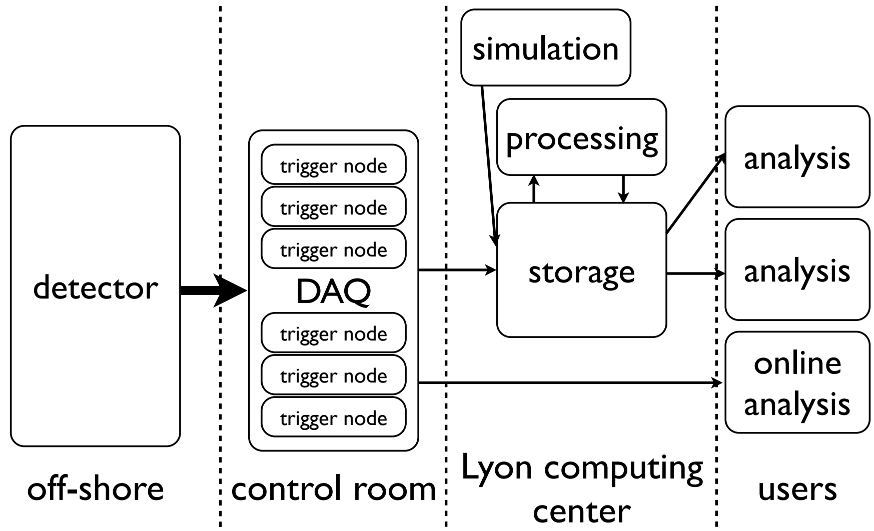

Figure 1: Summary of the ANTARES computing model from off-shore data taking to triggering and filtering in the shore station to off-line processing of data and analysis by users. Monte-Carlo simulation and a real time on-line analysis bypassing the off-line processing step are also shown.

7. Summary

The Antares computing model presented in this article is summarized in figure 1. All stages from data-taking through off-line processing and conversion to summary files for analysis have been set up to make optimal use of the available computing power at both the shore station and the main computing center for Antares in Lyon. The optional availability of an on-line data stream for real-time analysis and the integration of Monte-Carlo simulation tools allow for prompt analysis and verification of the recorded data.

References

[1] J. A. Aguilar et al., Nucl. Instr. Meth. A, 2007 vol. 570 p. 107

[2] R. Gurin, A. Maslennikov, *Controlhost: Distributed data handling package*, Technical report, CASPUR Inter-University Computing Consortium, Rome, 1995

[3] J. A. Aguilar et al., Nucl. Instr. Meth. A, 2010 vol. 622 pp. 59-73

[4] http://www.hpss-collaboration.org/

[5] IceCube Coll., T. DeYoung, *IceTray: A Software Framework for IceCube*, International Conference on Computing in High-Energy Physics and Nuclear Physics (CHEP2004), 2004, available from http://www.chep2004.org/

[6] The HDF Group, *Hierarchical data format version 5*, 2000-2011, http://www.hdfgroup.org/HDF5

[7] J. A. Aguilar et al., Astropart. Phys., 2011 vol. 34 pp. 652-662

[8] C. Kopper, Nucl. Instrum. Meth., 2011 vol. A626-627 pp. S243-S245