EPJ Web of Conferences **116**, 05003 (2016)
DOI: 10.1051/epjconf/201611605003

Live monitoring and quasi-online event reconstruction for KM3NeT

Tamas Gal[a] for the KM3NeT Collaboration

ECAP, University of Erlangen-Nuremberg, Erwin-Rommel-Str. 1, 91058 Erlangen, Germany

Abstract. KM3NeT is a new generation neutrino telescope in the abyss of the Mediterranean Sea. It will instrument a volume of several cubic kilometres of sea water in its final configuration. Currently, the project is in its first phase with the aim of constructing and installing 31 detection units up to 700 m in height, each equipped with 18 digital optical modules. The optical modules are equipped with 31 3-inch photomultipliers to detect the Cherenkov light of charged secondary particles produced in high-energy neutrino interactions. This contribution describes a live detector monitoring system, which enables real-time parameter control and a reconstruction of events soon after the data acquisition. It also allows a rapid response to or provision of external alarms of multi-messenger campaigns. The data acquisition system of KM3NeT provides pre-filtered data in event form, as well as general detector status messages. The events will be processed almost in real-time – with a delay in the range of minutes – using fast reconstruction mechanisms. This allows for high-level monitoring of the detector status using derived distributions, such as time and charge distributions and event rates. The resulting data is displayed on a web page using a dedicated, flexible web service. The same service also displays low-level monitoring data such as trigger rates, PMT hit rates and the general status of the optical modules.

1. Introduction

KM3NeT is a new generation neutrino telescope in the abyss of the Mediterranean Sea. It will consist of several hundreds of digital optical modules (DOMs), each holding 31 photomultiplier tubes (PMTs). The DOMs send the acquired data along with calibration data through an optical network to the on-shore computer centre. In the following, a short description of software package Restless Oyster (ROy) is presented, which allows real-time monitoring of the detector by providing an easy to use web interface.

2. The concept of Restless Oyster – a software package for real-time monitoring

The main concept of the Restless Oyster[1] online monitoring package is to provide a lightweight, easy to setup monitoring system which can be smoothly integrated in any existing environment. It uses standard software for both the back-end and the front-end, which most modern operating systems already include.

[a] e-mail: tamas.gal@km3net.de

[1] http://royweb.readthedocs.org

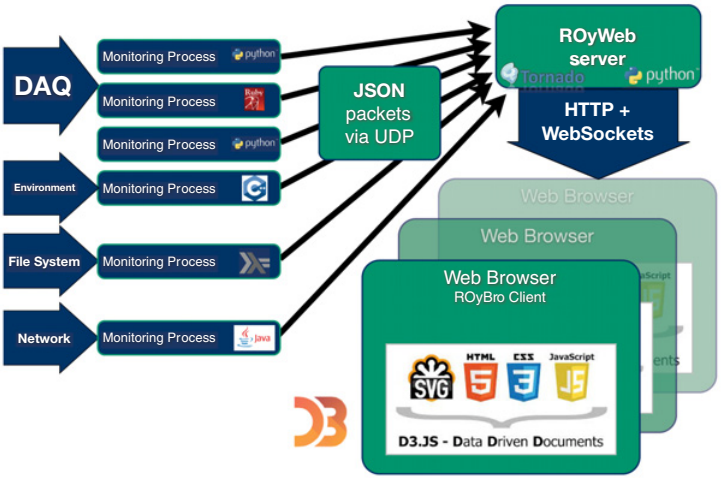

Figure 1. The data flow of the ROyWeb system. The blue boxes on the left represent the ROyCruncher, which are tiny daemons collecting and analysing online data and reading out sensors. The results are sent as JSON packets via UDP to the ROyWeb server. A web browser pointed to the ROyWeb server address is connected through HTTP to receive the web interface and through WebSockets to establish a permanent data stream which is then used to distribute the collected results in real-time.

The fact that ROy uses a central web server for data distribution allows to shift the workload of generating live graphs (like scatter plots or histograms) to the users workstation– namely the web browser –so the actual visualisation overhead is completely separated from the central monitoring server. In addition to that, each user is able to manage an own dashboard completely independent from others.

2.1 Components

ROy is a software package which consists of three main parts. ROyCruncher: small, standalone monitoring process or even a part of a bigger software package to read out sensors or calculate different parameters which need to be live monitored. ROyBro: the web interface which takes care about the user interaction, session management and data visualisations. ROyWeb: the main component of ROy, acting as a data dispatcher between ROyCruncher and ROyBro.

2.2 Data flow

Figure 1 shows the data flow of ROy. The ROyCrunchers collect all kinds of information like temperature, compass alignment, summary data of events, network usage etc. and send them as JSON packets through UDP to the ROyWeb web server. ROyWeb immediately stores the collected data in a local SQLite database and also redistributes them to all connected web clients via individual web socket connections between itself and the ROyBros.

2.3 Implementation

The core of ROyWeb is written in Python and is based on the Tornado[2] web server. The web client ROyBro is written in JavaScript and uses the D3.js[3] framework for data visualisation and HTML5/CSS

[2] `http://www.tornadoweb.org`
[3] `http://d3js.org`

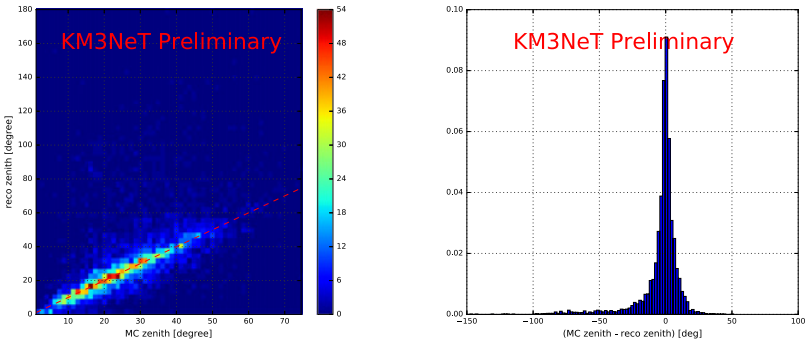

Figure 2. ROyFit – Fast muon reconstruction algorithm on 21857 MUPAGE events (atmospheric muon bundles) simulated for one detection unit of the KM3NeT neutrino telescope.

for the graphical user interface. Since operating systems like Linux and Mac OS X already include Python and a modern web browser like Safari, Firefox or Chrome, there is mostly no need to install any other dependencies in order to use ROy. The source code is MIT licensed and available on `https://github.com/tamasgal/royweb`.

3. ROy in the KM3NeT project

ROy is used both as a low-level monitoring tool for e.g. live display of PMT/DOM rates, trigger rates, time-over-threshold histograms, event size etc. as well as a high-level monitoring tool for quasi-online event reconstructions including fit parameters, reconstructed muon directions etc., which yield a nice overview of the detector performance.

Preliminary fit performance results of the fast muon reconstruction algorithm mainly based on BBFit[1] are shown in Fig. 2.

Reference

[1] ANTARES Collaboration (2011), *doi:10.1016/j.astropartphys.2011.01.003*