EPJ Web of Conferences **280**, 08001 (2023) https://doi.org/10.1051/epjconf/202328008001
RICAP-22

Scalable solutions for the Control Unit of the KM3NeT DAQ system

Cristiano Bozza[1*] for the KM3NeT Collaboration

[1]University of Salerno and INFN Gruppo Collegato di Salerno, Dipartimento di Fisica, Via Giovanni Paolo II 132, 84084 Fisciano, Italy

Abstract. The neutrino telescopes of KM3NeT are being incrementally expanded, and will reach their final size in the coming years. New versions of optical modules running new versions of firmware and new instrumentation for calibration are being introduced in the originally repetitive lattice. The inner architecture and data flow of the Control Unit of the KM3NeT telescopes is described, with information about computational and architectural complexity. The current goal is to control two full blocks of the KM3NeT/ARCA detector, i.e. 4370 CLBs and 128340 photomultipliers for 230 detection units, with a single mid-range commercial server machine. The system is designed with software protections and fault tolerance for hardware failure.

1 Introduction

The KM3NeT Collaboration is incrementally building and operating two neutrino telescopes in the Mediterranean Sea: ARCA[†], located off the shore of Sicily is devoted to neutrino astronomy ([1]); ORCA[‡], off the French shore is optimised for the study of neutrino oscillations ([2], [3], [4]). Both are water Cherenkov detectors with 3D lattices of *DOMs* (Digital Optical Modules) [5], each hosting 31 *PMTs* (photomultipliers) plus a compass, tiltmeter and humidity and pressure sensors, all controlled by a *CLB* (Central Logic Board); DOMs are grouped in DUs (*Detection Units*), which are vertical strings, few hundred meters tall, anchored to the seabed, of 18 regularly spaced DOMs. The spacing of DOMs is wider for ARCA than for ORCA, reflecting the different energy ranges of the neutrinos they are targeted at (up to PeV scale for ARCA, GeV scale for ORCA). In full configuration, ARCA will consist of two *blocks* of 115 DUs each, with in total 4140 DOMs, 128340 PMTs and 230 DU-base modules, instrumenting more than 1 km^3 of water; ORCA will have 115 DUs, 2070 DOMs, 64170 PMTs and 115 DU-base modules, covering 0.007 km^3 of water. Sea currents change the shape of the detectors; the position of detector elements is measured by an acoustic system including piezo elements in the DOMs, LED beacons and external hydrophones. A Calibration Unit and an Instrumentation Unit monitor water properties. Optical data from the PMTs and acoustic data from the piezos and

∗ Corresponding author: cbozza@unisa.it

†Astroparticle Research with Cosmics in the Abyss

‡Oscillation Research with Cosmics in the Abyss

hydrophones are transmitted to the control station on shore and processed by a computing farm, whereas data from other instruments are used for monitoring and recorded in a remote relational Database. At the time of the RICAP 2022 conference, 19 DUs were already taking data in ARCA and 10 in ORCA. At a given time KM3NeT has indeed more than two detectors in operation: each DU in the test benches at the end of the integration and before deployment is handled as an independent detector, managed by the same system.

2 The Control Unit

With the telescopes in their final configurations, about 400,000 parameters (HV, temperature, power, humidity, compass, tiltmeter, etc.) need to be controlled every 60 seconds in ARCA and 200,000 in ORCA. Some are input parameters (e.g. HV), others would raise alarms (e.g. humidity increase) if out of bounds. In addition, the *CU* (Control Unit) [6] checks the status of the *TriDAS* (Trigger and Data Acquisition System) and adjusts it according to the current task (calibration or physical data taking). DOM CLBs run two flavours of firmware and operate according to a state machine; DU-base module CLBs also have two variants of firmware for power control; a similar state machine also defines the behaviour of the TriDAS processes.

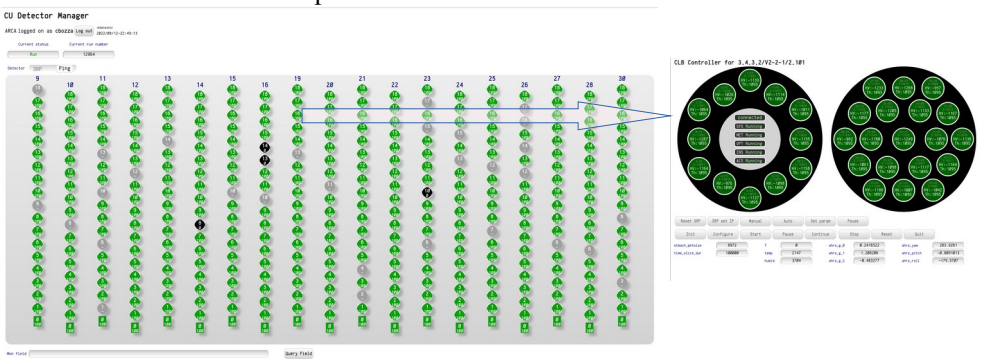

Fig. 1. Screenshot of the Detector Manager graphical user interface, showing the overview of the status of the full detector and a detailed view of one DOM.

The CU is a suite of fault-tolerant, partially independent software services. It has an overall state controller, the *MCP* (Master Control Program). The *DM* (Detector Manager) (Fig. 1) translates this into the set of actions needed to consistently govern each CLB, sets the operating parameters of PMTs (such as voltage and threshold) according to the *runsetup* defined on the MCP and reads slow monitoring and book-keeping data, staging them into a temporary area. The *TM* (TriDAS Manager) ensures that the *DataQueue* processes prepare data for the *Optical Data Filters* to perform online triggering, sending the results to a *DataWriter* for rolling storage in the local site before transmission to final storage; *Acoustic Data Filters* write their acoustic signal triggers directly to temporary storage. Finally, the *LAP* (Local Authentication Provider) provides user identification, dynamic role management of the machines, self-diagnosis and automatic load balancing in case one or more servers experience hardware problems. The *DPM* (Dynamic Provisioning Management) function of the LAP monitors local CU and TriDAS processes, preventing unwanted excess CPU or memory usage and restarting stopped processes in case of need. In case of degraded power, such as can happen if a server fails, LAPs of the other servers recognise the failure and reallocate the processes to the remaining machines. A *DBI* (Data

Base Interface) uploads datafiles from the local staging area to the remote Database and acts as a local cache to ensure continuity of data taking in case the Internet link to the remote Database is down. The most CPU-intensive task are handled by the DM. Currently in ARCA with 19 DUs there are 76 incoming UDP datagrams/s from the CLBs, each 175 B/s/CLB while the outbound UDP packets towards CLBs are as small as 19 B/s/CLB. The CPU load due only to monitoring tasks is 1.39 core on an Intel Xeon E5-2640@2.00 GHz. This results in 18 MB of monitoring data every 10 minutes, i.e. about 1 TB/year.

3 Performances

In its full configuration, the ARCA detector will have more than 4000 CLBs needing supervision and communication with the DM, which is the most critical component of the CU from the point of view of performances. A lightweight UDP-based messaging protocol has been developed for this application. Data must frequently be sorted by time or source, via optimised sorting methods. Writing continuously to disk or DB is inefficient, and all CU processes have an internal buffer to optimise writing tasks in bunches of 32 MB or 10 minutes. The monitoring information also needs to be presented in a GUI (*Graphical User Interface*), with hundreds of parameters represented in JSON and transported over HTTP to each subscribing client, the graphical rendering being left to the client's browser. Once the list of data to be extracted is defined, a direct-access path is created and no further sorting and searching occurs. A robust design must prevent thread starvation (i.e. thread pools exhausted or growing beyond the number of available cores) and deadlocks (both internal, due to data consistency requirements and external, due to cross-process logical error loops). In order to choose a safe attitude, multiple sockets are allocated with properly sized buffers and they are shared among CLBs in fixed groups (automatically defined); there is one readout thread per socket, which distributes data to CLB controller entities in the DM software, each with its own queue. This architecture cancels any chance of thread race. All data logging uses multiple memory slots to avoid thread races; logs are sorted at flush time, when data are copied to disk. Monitoring data from CLBs are processed by threads that are allocated as specified by configuration parameters: worker threads visit the CLB controller entities in a *round-robin* fashion, to log data in their queues and perform the needed reactions and adjustments. As a result of all this care taken in DM design, measurements done in regular data acquisition days in ARCA with 18 DUs show (Fig. 2) that the CPU load scales almost linearly and not faster than $N \log N$ (N being the number of DUs). Extrapolating these figures to a full *block* of 115 DUs, 4 CPU cores on the currently used server should suffice; a scalable solution with multiple DMs can be set up if needed.

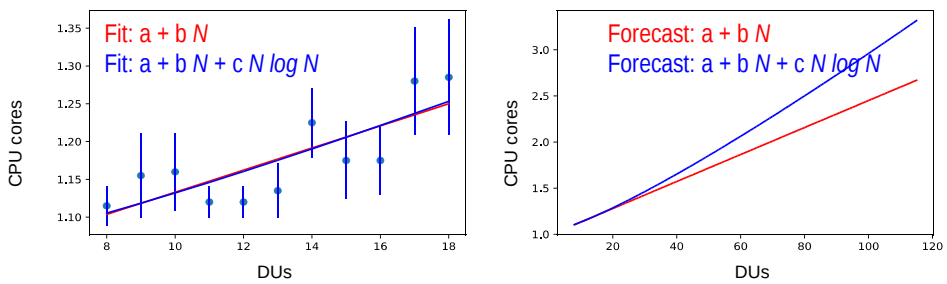

Fig. 2. CPU load (ARCA 18 DUs) on Intel Xeon E5-2640 v2 @ 2.00GHz (left) and extrapolation to full ARCA *block* of 115 DUs.

4 Overload protections

CU services are unlikely to cause CPU or memory overloads in normal conditions. On the other hand, TriDAS processes require normally a significant amount of resources and are already now spread over several servers. Protections are set up against runaway CPU or memory consumption. DataFilters and DataWriters are single-thread applications, so each one cannot saturate more than one core. This allows for intrinsic protections, by avoiding instantiating too many processes. DataFilter memory buffers are statically allocated. On the other hand, DataQueue memory buffers tend to grow if there is insufficient computing power (which may happen in case of hardware failure of one or more DataFilter servers). In this case, the resident LAP of each server watches for overall memory consumption and stops/restarts all local processes if the available memory falls below a tunable limit. In case DataWriter memory buffers grow too much, the resident LAP takes similar action.

5 Flexibility, code development and maintenance

The CU drives very different entities, because of the timespan of construction of the KM3NeT detectors. CLBs have two different variants and run two different versions of firmware. CLBs in base modules host a separate component for power functions that can have different version of firmware. Additional devices that need to be controlled by the CU are the Calibration Unit and the Instrumentation Unit, and more can appear in the future. The codebase must be flexible enough to allow evolution. The codebase is entirely in C# running on the Mono CLR and has only the Oracle Data Provider as external dependency to connect to the remote Database. Binary images are tagged and self-checked on startup, the fingerprint being added to all data. Loops, filters and sort operations are implemented in functional-programming fashion by means of LINQ, which makes the code expressive and readable. Behaviours are described with action tables. The source code for each device controller is kept in a single file. Event reactions are implemented as well-identified actions. Interprocess calls run on an HTTP library and also the graphical interface uses pure HTTP and Javascript. Static type checking and unit tests are used to actively prevent mistakes.

6 Conclusions

The Control Unit software for the KM3NeT Data Acquisition has been presented. It is a mature software suite, but evolving to support new versions of firmware for existing devices, new devices and adapting to changing scenarios. The code is produced to be clear and easy to maintain, with a mission to prevent, mitigate or eliminate risks of data loss. Performances are well under control and one commercial server should manage two *blocks* of 115 DUs. The architecture is fault tolerant and allows multiple resources standing by.

References

1. S. Aiello et al. (KM3NeT Collab.), Eur. Phys. J. C, 10.1140/epjc/s10052-021-09187-5
2. S. Aiello et al. (KM3NeT Collab.), JHEP, 10.1007/JHEP03(2022)055
3. S. Aiello et al. (KM3NeT Collab.), Eur. Phys. J. C**,** 10.1140/epjc/s10052-021-09893-0
4. S. Aiello et al. (KM3NeT Collab.), JHEP, 10.1007/JHEP10(2021)180
5. S. Aiello et al. (KM3NeT Collab.), JINST, 10.1088/1748-0221/17/07/P07038
6. S. Aiello et al. (KM3NeT Collab.), Comp. Phys. Comm., 10.1016/j.cpc.2020.107433