The Control Unit of KM3NeT detectors

The KM3NeT Collaboration[‡][*]
[‡]https://www.km3net.org/km3net-author-list-for-icrc-2019
E-mail: cbozza@unisa.it

The KM3NeT Collaboration has developed a flexible software suite named Control Unit to control and manage its neutrino detectors in the Mediterranean Sea, as well as all the benches where the DAQ hardware is integrated and tested. The Control Unit is highly modular and can work with or without direct coupling to the central database system. In order to ensure highest availability, maximum detector livetime and trouble isolation (no "single-point-of-failure"), the Control Unit is built of disconnected services which can individually continue operation even if others are temporarily missing. In case of hardware failure of one or more of the hosting computers, the Control Unit is able to automatically reconfigure itself and triggering/postprocessing resources to minimize the downtime to few seconds without any human intervention. This mitigates the data loss in case of overlap of failures with transient neutrino emission phenomena. In-house developed, high-performance web-based interfaces for humans as well as for machines allow easy integration of the Control Unit with monitoring tools. A user-friendly job system allows easy planning and management of data-taking campaigns.

Corresponding authors: Cristiano Bozza[1], Tommaso Chiarusi[2], Ronald Bruijn[†][3]
[1] *University of Salerno and INFN Gruppo Collegato di Salerno, Fisciano, Italy*
[2] *INFN Sezione di Bologna, Bologna, Italy*
[3] *University of Amsterdam, Amsterdam, Netherlands*

36th International Cosmic Ray Conference -ICRC2019-
July 24th - August 1st, 2019
Madison, WI, U.S.A.

*for collaboration list see PoS(ICRC2019)1177
†Speaker.

1. Introduction

KM3NeT [1] detectors [2,3] consist of blocks of Detection Units (DUs), which are vertical strings of 18 Digital Optical Modules (DOMs), each containing one Central Logic Board (CLB) that reads out and digitizes the signals from 31 photomultipliers (PMTs) and some additional instruments. Each DU has its foot anchored to the seabed and is kept stretched by a buoy. In its full configuration, a *building block* of KM3NeT contains 115 DUs, and hence 2070 CLBs and 64170 PMTs. Two *building blocks* would span more than 1 km^3. The Control Unit (CU) is a suite of software services for fine-grained control on each part of the detector and triggering/processing programs, with the goal of maximizing the uptime and hence the scientific output. As shown in Fig. 1, the CU interacts with various components and fulfills several tasks:

- reading the detector definition and operating parameters from the Central Data Base [4];
- controlling the CLBs and applying operating parameters (e.g. PMT voltage and threshold);
- reading out instruments (temperature, humidity, compass, tiltmeter, current, link signal strength etc.) and log parameter values to the Database as a data stream;
- configuring and running data processing for the Triggering and Data Acquisition System (TriDAS) [5];
- managing resources and ensuring stable operation despite hardware failures.

The CU is not built as a single process but as a set of programs each living in its own process for several reasons, among which it is worth recalling:

- increased code modularity;
- ability of each service to run even if all others are down;
- ability to upgrade parts of the CU with the detector running;
- scalability to large detectors and dynamic resource provisioning.

In the next sections, the following CU processes will be described:

- Data Base Interface
- Master Control Program
- Detector Manager
- TriDAS Manager
- Local Authentication Provider

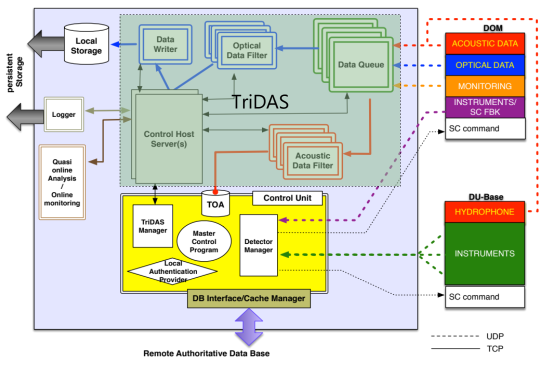


Fig.1 The Control Unit and its relationships with KM3NeT detectors and online software.

All CU processes incorporate a custom-made HTTP(S) server, which allows exposing a Web GUI. On top of that, the HTTP protocol is used also to establish communications between any two processes, using a lightweight remote procedure call library (named "Server Application with Web Interface"). Common web browsers can be connected and used as debuggers to inspect and emulate inter-process calls provided the necessary authentication is performed.

2. The Data Base Interface

KM3NeT detectors are managed in a centralized way already at the integration stage. As a consequence of the fact that all the information about a detector is stored in the Central Data Base (DB), along with consistent running parameter sets ("runsetups") and calibration information, an Internet link is needed with the Control Unit of each station (off-shore detector control stations as well as test benches); in addition, run book-keeping and instrument monitoring data must be made available on the Central DB as soon as possible for analysis. A direct link from each service of the CU would require authentication credentials to the KM3NeT DB to be stored and/or made accessible to the program, which should in addition have embedded SQL code with explicit reference to the data model representation in relational format. In order to optimize the access pattern, reduce the need for access credentials and the dependency on the external Internet link, a Data Base Interface (DBI) service has been included in the CU, as shown in Fig. 2. The DBI maintains a local file cache that mirrors the Data Base content, and in most cases serves the needed datasets without accessing the LAN. Conversely, writing to the DB for a CU service is as simple as writing a file in the proper staging directory: the DBI will take care of parsing it and addressing the proper tables and procedures. This also effectively decouples the relational data model from the internal representation of CU data, concentrating it in a single service and with a unified management of authentication credentials. All references to the DB schema object are bounded within the source code of the DBI. In case new tables are added or old ones need to be refactored, the changes would involve only one program, and the format of input/output files would stay unchanged. In case of a failure on the Internet link, data are not lost but data writing is delayed.

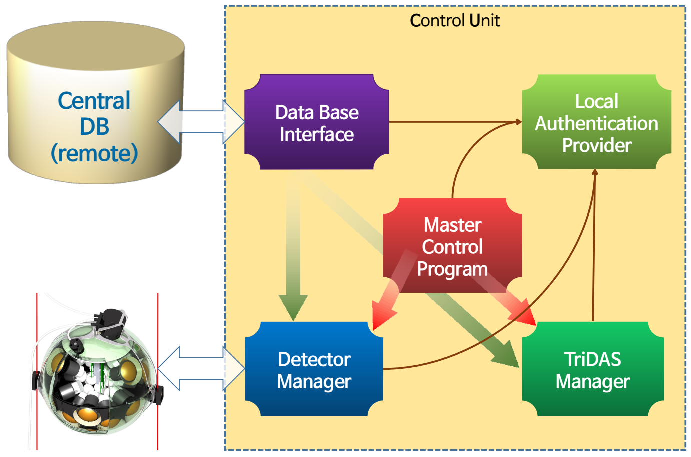


Fig.2 The components of the Control Unit and their interactions with the Data Base and the detector. Red arrows sketch the control flow, green arrows represent data flow.

3. The Master Control Program

The concept of detector in KM3NeT includes neutrino detectors deployed in the sea as well as single DOMs or DUs in test stations. Every time the configuration of a detector changes (i.e. addition of new DUs underwater, switch to another DOM or DU to be tested), the same CU is attached to a new detector. The Master Control Program (MCP), as sketched in Fig. 2, is the single authority to identify the current detector, the current runsetup, the "run number" (see below) and the set of data-taking jobs to be executed. For more efficient offline data handling, data taking is split into "runs", which are timespans during which the operating parameters are kept constant. In addition, the MCP defines the current "target", i.e. a flag that specifies whether the detector must be kept *"Off"* (all instruments and photomultipliers powered off), *"On"* (ready to start with instruments working and PMTs turned on but their output discarded) or *"Run"* (all instruments working and PMTs turned on and taking data). Some calibration parameters are also fine-tuned when needed (approximately on a monthly basis) and the MCP ensures overall consistency on this aspect too. Every time one of the above mentioned condition changes, the MCP notifies the Detector Manager and the TriDAS Manager. The recipients of the notification download the needed information from the DBI (i.e. the MCP message contains no data, just identifiers of data objects to retrieve). The communication works in push-pull mode. Both the Detector Manager and the TriDAS Manager also regularly poll the MCP to ask whether the conditions have changed: for high fault tolerance, even the case that the MCP message was lost because one of the recipients was temporarily unavailable is accounted for.

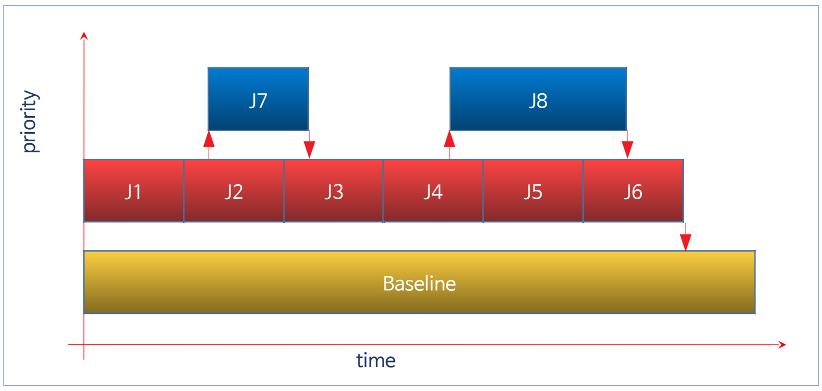


Fig.3 Jobs and priorities in data taking scheduling.

For highest effectiveness and comfortable usage by detector supervision operators, the changes of running conditions are almost never applied manually (although this is kept as a possible working mode): KM3NeT detectors are preferably operated by specifying a job schedule, as shown in Fig. 3, with each job being defined by a runsetup, a target, an application timespan and a priority. In case of multiple jobs active at the same time, the one with the highest priority will preempt all others. As soon as its timespan elapses, the MCP falls back to the second highest-priority job in the queue. As most data-taking campaigns last several months with nearly identical parameters, the related jobs can be generated automatically. A common setup is to have in priority 0 a very long baseline job that would simply keep the detector "*On*", automatically generated jobs in priority 1 and "*Run*" target to take data, and use priority 2 for special data taking (e.g. multimessenger alerts, calibration runs, development, maintenance, etc.). A job does not correspond to a single run, but a new run number will be generated every time the MCP switches to a different job. All this information is book-kept in the Central DB.

4. The Detector Manager

The Detector Manager (DM) is the CU process that is in charge of interacting with the hardware components of the detector. All instruments and PMTs are reachable via their CLBs. The DM communicates with the CLBs on a dedicated network segment using the Simple Retransmission Protocol (SRP), a lightweight layer on top of the UDP protocol to trace missing datagrams that may need to be sent again. From the point of view of the DM, the detector is a set of CLBs, each with its own finite state machine, which can be in one of the following states:

- *Idle*: the CLB is waiting for commands.
- *Stand-by*: the CLB expects configuration data.
- *Ready*: the CLB has been configured with a consistent set of operational parameters and is ready to start data taking. PMTs are on if allowed by input parameters.
- *Running*: PMTs are on and their output is sent to the TriDAS.
- *Paused*: PMTs are on but the data flow is kept on hold.

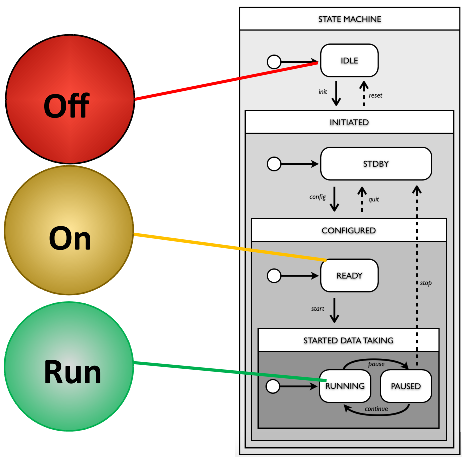
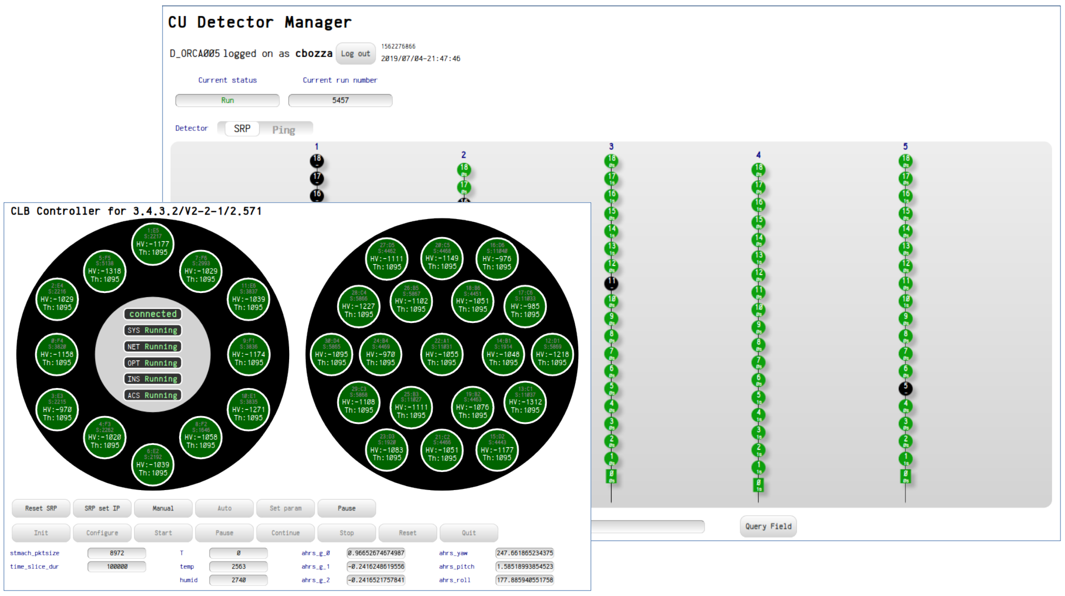


Fig.4 Targets and states of the state machine.

Some states are transient states, whereas others correspond to the MCP targets, as shown in Fig. 4. In particular, *Off* maps to *Idle*, *On* to *Ready* and *Run* to *Running*. The DM controls each CLB individually and, if a problem is detected on any of them (e.g. wrong state, wrong parameter, unresponsiveness, etc.), the run is not stopped, but the DM performs recovery asynchronously on-the-fly to put the CLB back on track with the others. Usually data retrieval from CLB instruments and monitoring is done by sampling every second, but slower rates are possible if needed. All parameters are sent to *datalog* files in the staging area of the DBI ready to be written to the DB; continuously updated values of all parameters are made available for online readout in a *Virtual Directory Tree* that is constantly refreshed to display the latest status of the detector. Such technology is used both to support a Graphical User Interface (GUI), as shown in Fig. 5, and to provide a fast and direct insight to external programs and scripts that may need to access and control the DM. It is even possible to add alternate graphical interfaces for specific applications just by placing the corresponding HTML files into the dedicated directory of the DM installation.


CU Detector Manager

D_ORCA005 logged on as **cbozza** 1562276866 2019/07/04-21:47:46

Current status: Run

Current run number: 5457

CLB Controller for 3.4.3.2/V2-2-1/2.571

stmach_pktsize	8972	T	0	ahrs_g_0	0.96652674674987	ahrs_yaw	247.661865234375
time_slice_dur	100000	temp	2563	ahrs_g_1	-0.2416248619556	ahrs_pitch	1.58518993854523
		humid	2740	ahrs_g_2	-0.2416521757841	ahrs_roll	177.885940551758

Fig.5 Screenshot of the Detector Manager user interface running in a browser. The view in the background shows the status of the whole detector, DOM by DOM, sketching the DU topology. The foreground view shows the status of a single DOM, with the PMTs of the top hemisphere on the left and those of the bottom one on the right.

Although the data rate handled by the DM is relatively small, the number of simultaneous connections to be managed (one per CLB) is not trivial and the amount of logging and transformations needed to make data available in various formats (and specifically high level ones, such as JSON) is CPU-intensive. Estimates based on experience gained in testing stations and with the first DUs deployed on the seabed in detector sites suggest that a 32-core machine should be able to control a full *building block* of 115 DUs.

5. The TriDAS Manager

KM3NeT detectors implement the “all-data-to-shore” policy, so very limited data reduction is performed at the level of the CLB. PMT signals are encoded as the timestamp of the pulse and the total time the PMT output exceeds a set threshold (“Time-over-Threshold”). The continuous stream of PMT pulses is segmented in timeslices at the level of the single CLB, but overall event reconstruction requires that timeslices from all CLBs be assembled together to cover the whole volume of the detector. This is done by Data Queue processes that route detector-scale timeslices to Optical Data Filters. They run the triggering and postprocessing algorithms to produce triggered events, sent to one or more Data Writers. The shape of the detector changes under the action of seawater currents and needs to be reconstructed from the arrival times of acoustic beacon signals measured by hydrophones hosted in the DOMs. The Acoustic Data Filter processes the acoustic data stream from the CLBs and provides data that are later used (offline) to fine-tune the detector shape. Dispatchers provide a convenient way to send control information and data to multiple destinations. The aforementioned processes are altogether called the Trigger and Data Acquisition System (TriDAS), running on a local computing farm. The TriDAS Manager (TM) is the CU process that supervises and manages all instances of the TriDAS programs. They implement the same state machine as the CLBs, and the TM takes care of putting them in the state that is consistent with the current target. The TM sets the triggering policies and position/timing calibrations as specified by the MCP. Like the DM, the TM features a Virtual Directory Tree to expose the status of TriDAS processes and saves logging information to datalog files to be uploaded to the DB by the DBI.

6. The Local Authentication Provider

The CU processes normally run in a local subnet that is accessible from the outer world through VPN or bastion servers. This is enough to make them reasonably safe from malicious accesses. Nevertheless, the actions available to detector operators controlling the acquisition for a weekly shift are a subset of those that are allowed to detector managers and data-taking experts. The Local Authentication Provider (LAP) synchronizes its local access database with the Central DB, retrieving the current credentials and permissions for each user to log in and perform actions. On top of this authentication service, the LAP also provides identification for the distributed CU system. All processes ask the LAP to know on which IP and port a certain service is allocated to send notifications (e.g. see the case of MCP notifying DM and TM of a new run number).

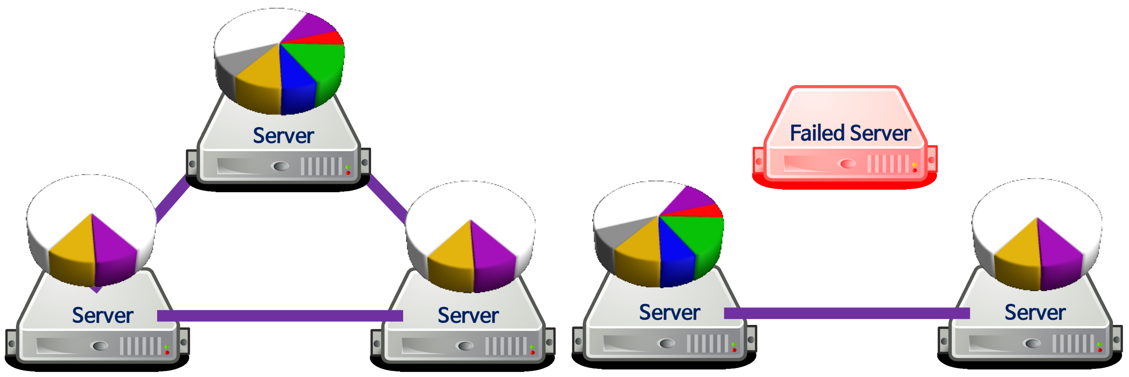


Fig.6 Example of Dynamic Resource Provisioning and Failover. Left: full configuration. Right: configuration with a single failure. The pie charts sketch the computing resources (CPU, memory) used by different CU and TriDAS processes. In the downgraded configuration, all services (all colors) are present, although with different overall proportions. Service relocation follows priority rules.

The goal of maximizing the uptime of the KM3NeT detectors, and specifically to be always ready in case of transient phenomena, recently suggested to upgrade the role of the LAP to offer "Dynamic Resource Provisioning and Failover" (DRP-F) [6]. This allows ensuring fault tolerance even in case of hardware failure of one or more of the CU and TriDAS servers. In this operating fashion, there is one LAP running on every CU and TriDAS computer in the shore station/testing station. Each LAP runs diagnostic checks locally to determine if the machine is running properly (e.g. testing also connectivity on all network interfaces). LAPs know each other through the IP address. Each LAP continuously polls the status of other machines. If one declares that it is in a fault state or is unresponsive, all others know that it must be temporarily disregarded. All CU and TriDAS processes are reallocated on the remaining servers, using compatibility rules (for example, memory requirements are relevant for Data Queues and Data Filters, not for the Detector Manager) and a priority system that is based on the IP address, as shown in Fig. 6. This ensures that there is no single point of failure or central authority: in case of failure of one server, each LAP will independently work out the new allocation of processes and consistency of the result is ensured by the algorithm. No server is so critical that it cannot be replaced automatically. Likewise, when a new machine joins the team (for replacement or upgrade), registering it on one of the existing LAPs ensures that the information is quickly propagated to all others, so all servers know it and it knows all others. Tests performed so far indicate that this DRP-F system can switch from one configuration to another in a time between 5 and 10 seconds, which includes the time needed to shut down some of the existing processes, start new ones, switching to a new run, etc. Depending on the available resources, it is possible to set up configurations for the CU and TriDAS with single, double or multiple fault tolerance. Of course, running with multiple failures downgrades the available computing power (which should anyway be sized to provide some redundancy), but it mitigates the impact of hardware failures and the need for specialized experts to reconfigure the infrastructure. When the failed machines are repaired or replaced, it is sufficient to connect them to the network to have the system recover full power again.

7. Conclusions

The CU is the result of a design process that has been guided by the principles of flexibility, scalability, reliability and long-term planning. It is built on well-tested and widely adopted technologies, whereas few custom-designed components are completely controlled by the KM3NeT Collaboration. The technology of Dynamic Resource Provisioning at the same time simplifies administration and allows software to circumvent also hardware problems.

References

[1] S. Adrián-Martínez et al. (KM3NeT Coll.), *J. Phys. G.* **43** 1 (2016)

[2] S. Aiello et al. (KM3NeT Coll.), *Astroparticle Phys.* ***111*** (2019), 100

[3] S. Adrián-Martínez et al. (KM3NeT Coll.), *J. High. Energy Phys.* 2017 **5** (2017), 8

[4] A. Arnauld and C. Bozza, *EPJ Web of Conferences* **116** (2016) 07004

[5] T. Chiarusi and C. Pellegrino, *EPJ Web of Conferences* **116** (2016) 05005

[6] C. Bozza, *EPJ Web of Conferences* **207** (2019), 06008