The NEMO Trigger and Data Acquisition System

Tommaso Chiarusi

INFN - Sezione di Bologna, Viale Berti-Pichat 6/2, 40127 Bologna, Italy

Abstract

The second phase of the NEMO project represents a valuable occasion to test a new Trigger and Data Acquisition System (TriDAS), designed to scale up to the km3. Because of the deep sea optical background, the NEMO "all data to shore" approach requires to handle a large continuous data-stream from off-shore to on-shore. The TriDAS consists of 4 computing layers: hit managing into time-coherent aggregates, data selection according to possible concurrent trigger algorithms, composition of the selected events into post-trigger files and finally persistent data storage. The finalized design of TriDAS adapted for NEMO Phase 2 is reviewed together with the dedicated on-line monitoring tool and networking architecture.

Keywords: NEMO, neutrino telescope, trigger and data acquisition system

1. NEMO Phase 2: the expected throughput

The NEMO project [1] of INFN, as part of the KM3NeT Consortium [2] activities, is dedicated to study the technical issues of a neutrino telescope on a cubic kilometer scale in the Mediterranean Sea [3]. In year 2012 it will complete its final goal by deploying in the abyssal site of Capo Passero a prototype Detection Unit (DU) [4] and by activating the corresponding data acquisition for long duration runs. Besides being a continuous and hence precious source of deep sea environmental monitoring, it will represent a very important test of all the underwater mechanical and electronic components as well as the online data handling and filtering facilities located on-shore.

A general overview of the NEMO project is given in [5]. It is here convenient to recall the most relevant elements of the NEMO Phase 2 DU.

The NEMO Phase 2 DU is to be located in the Capo Passero site at a depth of 3500 m and 100 km off from the southern coast of Sicily. While the *standard* NEMO DU is a tower made of 16 instrumented floors, interleaved by 40 m, the prototype Phase 2 DU is made of 8 floors only, and it is 500 m high. Each floor hosts 4 optical modules, made of glass spheres each containing a 10" PMT [6] and its Front End Module (FEM) read-out electronics, plus various environmental and positioning subdetectors [7]. On each floor, the data stream from the four optical modules is gathered by the Floor Control Module (FCM) electronic board; the stream from all the floors is multiplexed at the base of the DU according to the DWDM protocol and sent to shore through the 100 km optical fibers inside the EOC cable. Once de-multiplexed, each data stream originated from one FCM is addressed a specific board, called Ethernet Floor Control Module (EFCM), identical to the FCM offshore but with a further feature which allows the ethernet connection to the rest of the data acquisition system. The details of NEMO Phase 2 electronics are reported in [8]. To minimize the number of possible points of failure in the abyssal site, no hardware triggers are added to the underwater detector, and all the measured signals are sent on-shore. The total available bandwidth for the optical data from the DU is 2 Gbps (and can be extended up to 5 Gbps) which is well suited for the specific optical conditions in the NEMO Phase 2 case. In fact, the expected optical background at the Capo Passero abyssal site is dominated by single photon hits on the PMTs after ^{40}K decays which have been measured and reported in [9]. The paper shows that the contribution of bioluminescence to the optical signal, which strongly enhances the optical noise in shallower locations, is significantly suppressed at the NEMO depth. The NEMO Phase 2 DU is thus expected to produce an average throughput of 250 Mbps. It is shown also that a 10" PMT with a minumum charge threshold set at 0.5 photoelectrons (p.e.) is subjected to a continuous hit rate of about 40 kHz. Each detected single photon pulse is sampled by the FEM and arranged by the FCM in a hit record with a mean size of 28 bytes.

It is important to mention that the NEMO electronics were designed to deal with a 150 kHz continuous single rate on each PMT without dead time; such a single rate per PMT would cause a global 2 Gbps throughput from the entire standard NEMO DU, and it is assumed as the reference condition for a conservative design of the NEMO on-shore Trigger and Data Acquisition System (TriDAS).

2. The NEMO TriDAS

The TriDAS is the main computing structure for on-line optical data aggregation, selection and distribution. It was designed to scale with the dimension of the underwater detector, and it is modular. It is composed of the following elements: the **Hit Managers** (HMs) receive the data stream from a defined number of EFCMs, which represent a *sector* of the DU. One HM subdivides the collected data stream in a sequence of time ordered bunches of data, each called the *Sector Time Slice* (STS); the sequences of STS refer to the succession of time windows of a fixed duration, typically 200 ms. Sharing a common time reference, all the HMs arrange their STSs according to the same intervals of time, which are referred to as *Time Slices* (TSs). The STSs assembled by all the HMs referring to the same TS are sent to one Trigger CPU. The **Trigger CPUs** (TCPUs) aggregate the incoming coherent STSs in a *Tower Time Slice* (TTS); each TTS is analysed by the various existing filtering algorithms. Different concurrent trigger algorithms may run on the same TCPU, all acting on the same TTS. Each trigger algorithm lives on an independent thread, which is started after loading the relative plugin-library on the TCPU start. Anytime a trigger condition is satisfied, a subset of data called *Triggered Event* (TE) occurred within a fixed time window (typically 6 μs) around the trigger seed is sent to the Event Manager. The **Event Manager** (EM) collects the TE selected from each trigger algorithm; if hits are duplicated, the EM merges the TE and then writes the selected data in the so called *Post Trigger* (PT) file ; finally, the EM drives the storaging of each PT file on persistent media. The **TriDAS System Controller** (TSC) is the front end of the TriDAS to the user of the detector. It allows to configure, operate and monitor the TriDAS activities.

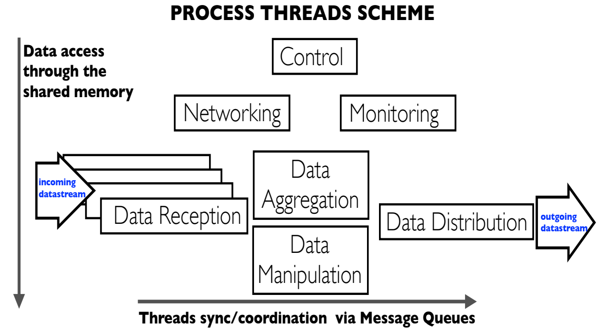

Figure 1: The prototype design for all the TriDAS elements, i.e. the HM, TCPU, EM and TSC processes. See text for a detailed description.

2.1. TriDAS design

The TriDAS elements HM, TCPU, EM and TSC are implemented in C++ and run on servers with 2 GHz 8-core Intel®Xeon®CPUs, 6 GB RAM and the Scientific Linux 5.5 (Boron) operating system. The design of each one of them follows the same prototypical design (see Figure 1) based on POSIX threads [10]. Control and Networking threads provide services, drive the usage of the shared memory and grant the connection among the TriDAS elements. Although HM, TCPU, EM and TSC handle different data structures, they follow the same action scheme: buffering the incoming coherent data through various input channels; data aggregation into more complex structures; data manipulation and data distribution. Within the same process, the threads are synchronized by the System V IPC Message Queues [11]; TCP/IP sockets are opened between EFCMs and HMs and between the HMs and the TCPUs. The sending processes are clients for the receiving ones, which consequently play the role of servers. The Monitoring threads are interfaced to an external monitoring system (which will be discussed in section 4) through multicasting services. Such services are granted by a data dispatcher facility called ERMES, based on [12] [13] and specialized for the NEMO requirements. Also the TE, whose stream determines a low throughput from the TCPU to the EM (see next section), are sent via the ERMES services.

2.2. TriDAS dimensioning and scalability

Each HM handles the incoming data stream from a sector of the DU. The number of EFCMs belonging to one sector depends on their effective throughput. Running on the servers mentioned above, it was extensively tested that an HM can easily deal with a global incoming data stream of about 1 Gbps; it corresponds to the output of 8 EFCMs (2 sectors for the standard NEMO DU) and 150 kHz continuous single rate per PMT. These performances are well beyond the requirements for the NEMO Phase 2 setup, given the expected single rate of 40 kHz on each PMT and the Phase 2 DU being composed of 8 floors only; since the floors could eventually be grouped in 4 sectors of 2 EFCMs each or 2 sectors of 4 EFCMs each the number of necessary HMs ranges from 2 to 4. The use of a single HM for all the DU was excluded in order to avoid a single point of failure which could cut the whole data acquisition.

The number of the TCPU servers is directly related to the speed of the trigger algorithms. It has been already mentioned that various trigger threads can be simultaneously applied to the same TTS. While a TTS is under analysis, the TCPUs keep buffering the other incoming TTSs, and wait for all the trigger threads to complete their job before starting with a new TTS. This implies that the slowest algorithm biases the TCPU speed-performances. Given the CPU clock frequency $R_{CPU} = 2$ GHz, the size of each hit $S_{hit} = 224$ bit (in the hypothesis of single photon hits) and the maximum bandwidth between one HM and a TCPU $B_{HM\rightarrow TCPU} = 1$ Gbps, the maximum number of CPU clock cyles available per hit is given by $N_{ACC} = R_{CPU}S_{hit}/B_{HM\rightarrow TCPU} = 448$. If a trigger algorithm uses per hit a number of CPU clock cycles $N_C > N_{ACC}$ the number of TCPU must be increased, accordingly.

For a given N_C, a continuous single rate per PMT ν_K, the number of PMTs in the DU N_{PMT} and the TCPU

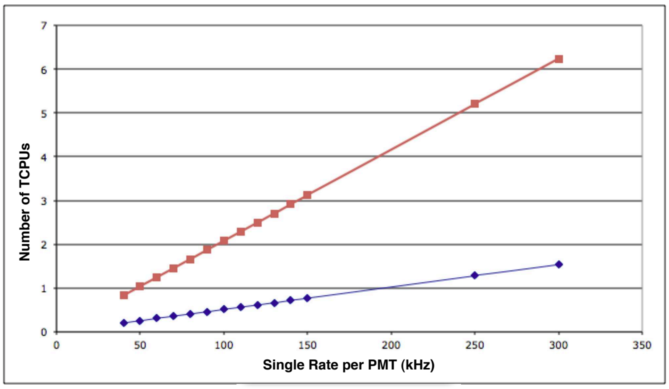

Figure 2: The computed number of needed TCPU versus the continuous single rate on each PMT for the NEMO Phase 2 DU (blue line) compared to that for the NEMO standard DU (red line), assuming N_C equal to 350 and 650 for NEMO Phase 2 and NEMO standard case, respectively. See text for details.

clock frequency R_{CPU}, the number of needed TCPU is $N_{TCPU} = \nu_K N_C N_{PMT}/R_{CPU}$.

The most elementary topological trigger is searching for time coincidences between hits detected by close PMTs of the same floors, within a given time window (typically 20 ns). It is called *Simple Coincidences* (SC) trigger. It needs at any step a time-ordered list of hits, one per PMT; getting the ordered list on a NEMO standard DU with 64 PMTs takes about 330 CPU clock cycles per hit. The maximum estimated CPU load per hit for the full SC trigger search is 650 CPU clock cycles, which indicates that more than one TCPU is needed. In the case of the NEMO Phase 2 DU, it is conservative assuming a N_C= 350 which indicates that 1 TCPU could be enough (see Figure 2). Similarly for the HM, at least 2 TCPU will be used to avoid single point of failure also at the trigger layer of the TriDAS chain. Moreover, in order to remove any possible bottleneck effect in data transmission from the HMs and the TCPUs, which could drive to increasing delays in the data acquisition, the minimum number of TCPU must be at least equal or greater than the number of HMs.

The TriDAS for the NEMO Phase 2 will host also a GPU server which is deputed to test the parallel designed trigger reported in [14] on a subset of the TTS assembled by one TCPU.

The data stream exiting from the TCPU is significantly suppressed with respect to the incoming one. Considering the NEMO standard DU, the SC trigger (one of the most conservative algorithms) is expected with a rate of 14 kHz if assuming the usual 150 kHz continuous single rate per PMT. If each TE time window extends for 6 μs around the trigger seed occurrence, the output data stream from each TCPU is 160 Mbps/N_{TCPU}, i.e. the global post-trigger data stream drops to $\sim$ 8% of the incoming throuput from the DU. It is not a very strong data filtering, but indicates that just one EM is capable to deal with such a limited post-trigger data stream. In the case of the NEMO Phase 2 DU, with the expected 40 kHz continuos single rate per PMT, the throughput reduction sets to 0.3 % of the unfiltered data stream. The expected storage load is thus less than 50 GB per day, i.e. about 20 TB per year, which can be managed with standard storage systems technologies.

3. Networking architecture

The TriDAS is only one part of the full NEMO control system, which is located on-shore. Together with the EFCMs, it involves also the GPS servers, giving the precise absolute time, the database which stores all the detector settings and the acquired slow control data, and finally the *Data Manager* (DM) which is the abstract definition for the collection of services available for the user to control the detector run. In particular, the DM arranges the proper setup for all the TriDAS elements, which is configured, started and controlled via TCP/IP socket communications through the TSC. Moreover, during the run, the TriDAS is continuously reporting to the DM its status by means of UDP packets sent with a fixed latency, typically of 1 s. All the network connections between the servers of the TriDAS, the DM and the other components of the NEMO control system are based on 1 Gbit Ethernet technology and, with the exception of the TriDAS, the communication is at low throughput. In order to optimize the networking perfomances, separating the high throughput traffic from the low band occupancy communications, and profiting of the double ethernet connections of the modern server architectures, two subnets were designed according to the sketch in Figure 3.

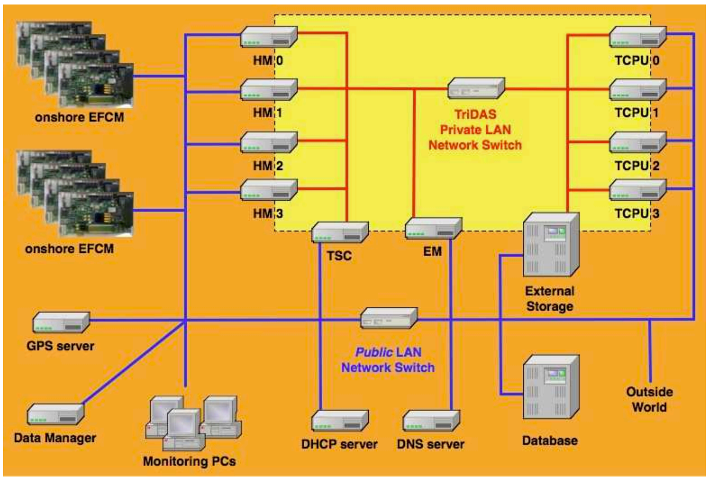

Figure 3: The on-shore networking architecture design for NEMO Phase 2

A private LAN is set among all the TriDAS servers, but it is mainly dedicated to the HMs communication towards the TCPUs. Within this network, all the TriDAS servers have their own IP addresses statically assigned to one of their two network devices. The second ethernet device is connected to a second LAN, which is called ”public” since it can be accessed from outside the NEMO control system, e.g. from the INFN departments and labs or even via web. Besides the TriDAS, to the public LAN are connected also the EFCMs, the DM and all the rest of the NEMO control system PCs. The devices connected through the public LAN receive a dynamic assigned IP address thanks to local DNS and DHCP servers.

4. Monitoring tools

Dedicated C++ software for monitoring the data elaborated by the TriDAS, together with the TriDAS own functionalities, have been developed by integrating the already mentioned ERMES dispatching facilities with the ROOT [15] and Qt [16] frameworks.

ERMES functions allow processes to send to a central dispatching unit various kind of data structures (from a simple value to extended bunches of data) with a customizable latency. Each data structure is identified by a tag and can be retrieved from the dispatcher memory by any monitoring tool connected to the LAN.

Graphical user interfaces (GUIs) were designed with multiple scopes: software called *Vis* [17] allows to plot any single data value versus time and to quickly perform some statistical analysis on it. Finally, an *Event Display* [18] was developed to check a sample of the Triggered Events, showing for example the hit charge distribution or the pulse waveform from the hit samples.

Figure 4 sketches the TriDAS monitoring connections, allowing the visualization of sensible parameters which are representative for each phase of data manipulation. Two or more ERMES dispatchers can be used to collect and distribute physics information (solid line arrows) as well as that concerning the system functionality (dashed line arrows) of the chain HM-TCPU-EM-Storage.

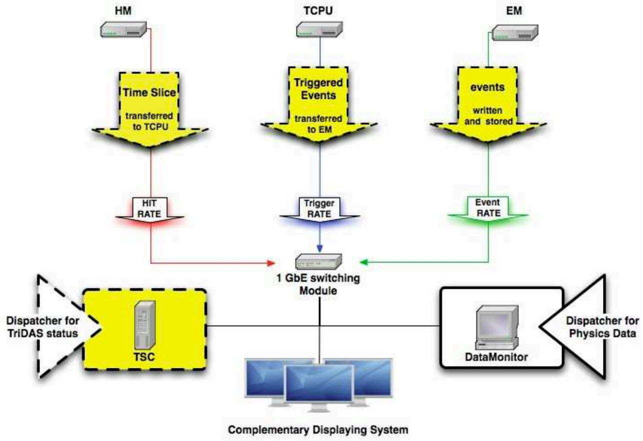

Figure 4: Sketch of the TriDAS monitoring connections. The dashed arrows indicate the information stream relative to the functionality of TriDAS, which are collected by the TSC. The other arrows are relative to physics information, collected by a dedicated server, the DataMonitor. The Complementary Displaying System is the front-end for a final user.

Together with the API expressely dedicated to the NEMO project, public software like Nagios [19] and Ganglia [20] are used to monitor the TriDAS servers and to publish the sensible data on the web.

5. Conclusions

In the year 2012 the prototype Detection Unit of NEMO Phase 2 will be deployed in the abyssal site of Capo Passero. The on-line Trigger and Data Acquisition System was finalized and integrated into the 1 Gbit Ethernet network present at the Capo Passero shore control base. The TriDAS design is scalable with the incoming throughput from the underwater Detection Unit and with the complexity of the data filtering algorithms. Performances of the TriDAS and the quality of the selected data can be monitored online by means of dedicated APIs and graphical user interfaces.

Acknowledgments

I want to thank Giuseppe Terreni who is involved in the TriDAS design and development since the beginning. I'm also grateful to Andrea Paolucci for his contribution to the TriDAS monitoring and network design. Special thanks also to G. De Rosa, for the TSC development, and to F. Ameli, F. Simeone, R. Masullo, T. Capone, M. Morganti and all the members of NEMO DAQ Working Group.

References

[1] NEMO web site, http://nemoweb.lns.infn.it.
[2] KM3NeT Consortium web site, http://www.km3net.org.
[3] T. Chiarusi, M. Spurio, High-energy astrophysics with neutrino telescopes, Eur. Phys. J. C 65 (2010) 649.
[4] G. Cacopardo, these proceedings.
[5] M. Taiuti, et al., Nucl. Intrum. Meth. A 626-627 (2011) 25.
[6] E. Lenora, et al., these proceedings.
[7] S. Viola, these proceedings.
[8] F. Simeone, et al., these proceedings.
[9] The NEMO Collaboration, Report to ApPEC Peer Review, http://nemoweb.lns.infn.it/sites/sitereport.
[10] B. Nichols, D. Buttlar, J. Farrel, Pthreads programming, O'Reilly, 1998.
[11] J. S. Gray, Interprocess communications in Unix®, Prentice Hall PTR, 1998.
[12] V. Maslenikov, et al., CASPUR Consortium, http://afs.caspur.it/temp/ControlHost.pdf.
[13] Controlhost distributed data handling package, http://www.nikhef.nl/∼ruud/HTML/choo_manual.html.
[14] B. Bouhadef, et al, these proceedings.
[15] ROOT analysis framework, http://root.cern.ch.
[16] QT cross-platform application and UI framework, http://qt.nokia.com/.
[17] D. Bonfigli, Bachelor Thesis in Informatics, University of Bologna, 2008.
[18] A. Riccardo, Bachelor Thesis in Informatics, University of Bologna, 2009.
[19] Nagios, http://www.nagios.org/.
[20] Ganglia monitoring system, http://ganglia.sourceforge.net/.