

Data Quality Monitoring : Automatic MOnitoRing Environment (AMORE) Web Administration Tool in ALICE Experiment

Imre Nagi

Electrical Engineering Departement, School of Electrical Engineering and Informatics

Institut Teknologi Bandung, Indonesia

imre.here@yahoo.co.id ; imre.nagi@cern.ch

Supervisor : Barthélemy von Haller

ALICE-DAQ CERN

August 16, 2013



Abstract

ALICE (A Large Ion Collider Experiment) is the heavy-ion detector designed to study the physics of strongly interacting matter and the quark-gluon plasma at the CERN Large Hadron Collider (LHC). The quality of the acquired data evolves over time depending on the status of the detectors, its components and the operating environment. To get an excellent performance of detector, all detector configurations have to be set perfectly so that the data-taking can be done in an optimal way. This report describes a new implementation of the administration tools of the ALICE's DQM framework called AMORE (Automatic MonitoRing Environment) with web technologies.

Keywords: **ALICE, DAQ, AMORE, CERN, DATA QUALITY MONITORING**

1 Introduction

1.1 The ALICE Experiment

ALICE (A Large Ion Collider Experiment) is a heavy-ion detector that has been built for studying the physics of strongly interacting matter and the quark gluon plasma at extreme energy densities. For this reason, this experiment is carrying out a comprehensive study of hadrons, electrons, muons, and photons produced in the collision on heavy nuclei. This detector is also doing a comprehensive research in proton-proton collisions as a comparison with lead-lead collisions.

1.2 Data Quality Monitoring

Data Quality Monitoring (DQM) is an important aspect of every High-Energy physics experiment, especially in the era of LHC where the detectors are extremely sophisticated devices. DQM involves the online gathering of data, their analysis by user-defined algorithm and the storage and visualization of the produced monitoring information.

1.3 ALICE Data Acquisition

The ALICE Data-Acquisition system (DAQ) handles the data flow from the sub-detector electronics to the archiving on tape. The first layer of computer, the Local Data Concentrators (LDCs), read out the event fragments from the optical Detector Data Links (DDLs). Several LDCs will be needed in collecting the data from a single sub-detector. The event fragments aggregated in sub-events are then transferred to a second layer of computers called the Global Data Collectors (GDCs) for performing the event building. DQM software runs on dedicated servers connected to the event building network. The data samples that feed the DQM nodes are intercepted on either the LDC's or the GDC's depending on the needs.

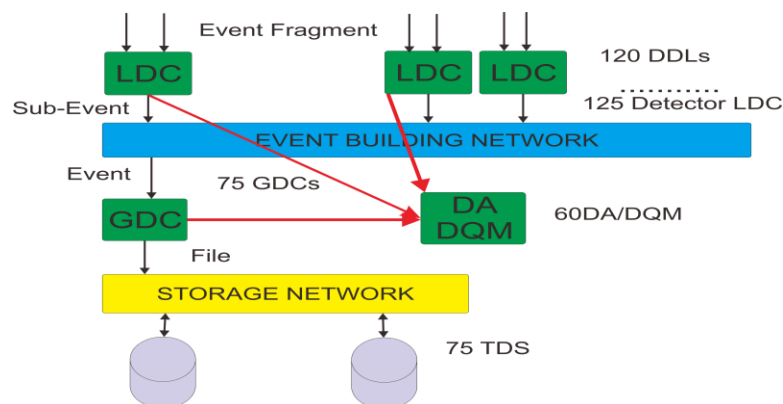


Figure 1. Simplified dataflow of ALICE Data Acquisition

2 AMORE: A DQM Framework For ALICE

2.1 Architecture

AMORE is based on a publisher-subscriber paradigm where a large number of processes, called **agents**, execute detector-specific decoding and analysis on raw data samples and publish their results in a pool. **Clients** can then connect to the pool and visualize the monitoring results through a dedicated user interface.

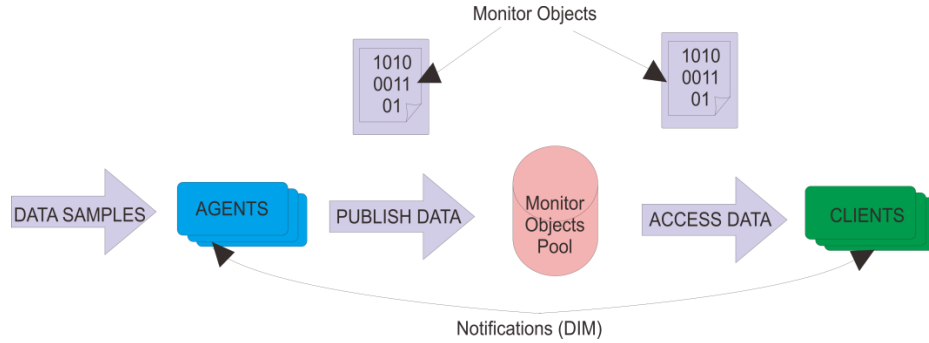


Figure 2. Publisher-subscriber paradigm in AMORE

2.2 AMORE Data Pool

The pool is implemented as a database. The open-source MySQL system was chosen as it proved to be reliable, performant and light-weight.

The database is used not only to keep the data published by the agents, but also to store the configuration of AMORE as a system. This includes information about the agents such as the machine where they can run and to which detector they belong (agents table) as well as the optional configuration files. When a new agent is created in the system, a row is added to the agents table. The table where published data will be stored is created or recreated when the agent is started.

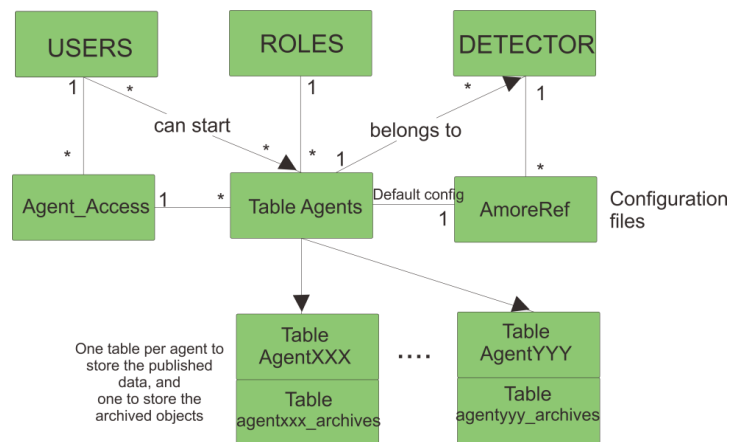


Figure 3. Schema of AMORE Database

3 AMORE Web Administration Tool

The LHC (Large Hadron Collider) is in his first long shutdown until the end of 2014 to prepare it for higher collision energy in 2015. For this purpose, the ALICE Data Acquisition group is also improving several AMORE administration tools that have been implemented using TCL/TK scripts into web application developed in PHP language. The aim of the project is not only to translate from an old language to a modern one but also to integrate the various tools into a more consistent one, to make it easier to develop and to greatly improve the users experience in gathering, displaying and modifying data and information coming from a database.

The web administration tools also improve the security aspect over their previous implementation. Having a web application tool running on a server removes the possibility for users to read and change the script including the credentials of the databases. All they can do now is only to use the application via web-browser.

For the next startup of the LHC, this web administration tool will be used for setting up the configuration of the AMORE system during the ALICE experiment's operations .

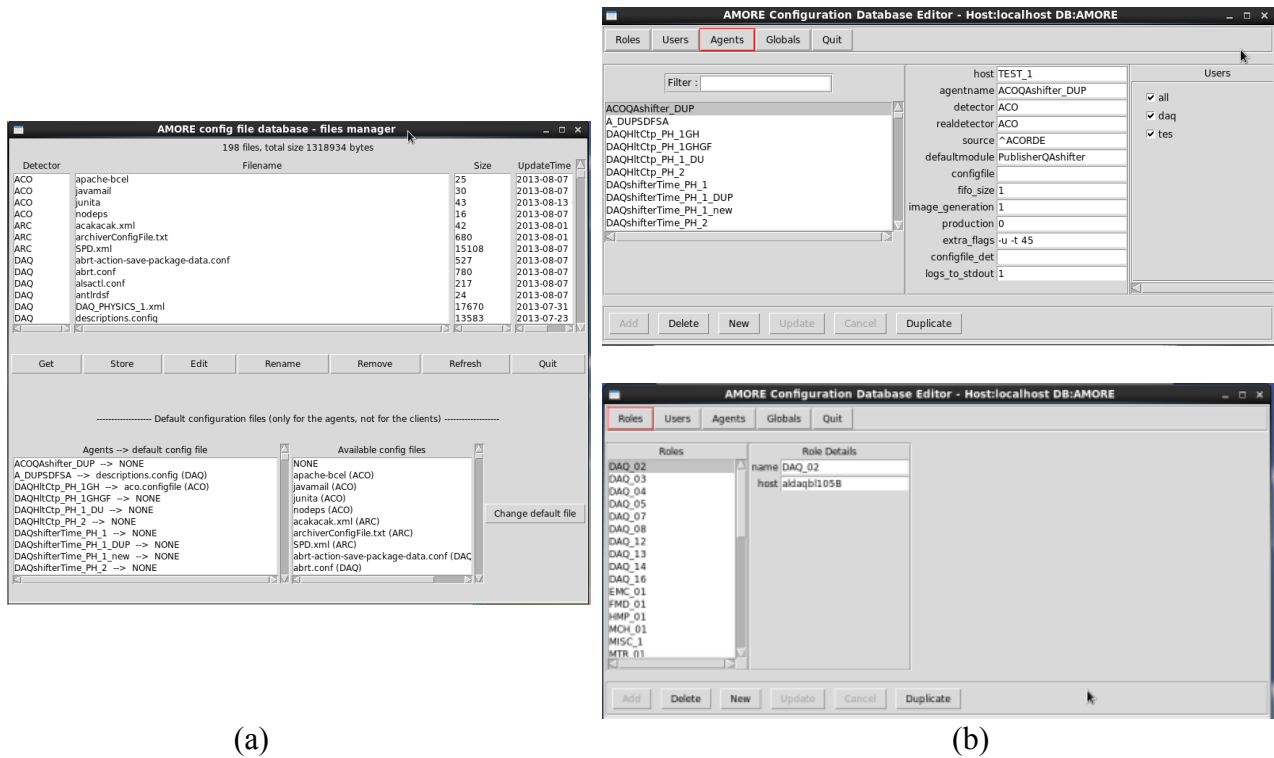


Figure 4. The Old AMORE Administration Tool Developed in TCL/TK Scripts.

(a)AmoreConfigFileBrowser **(b)**AmoreEditDb

3.1 AMORE Configuration Database Editor (AmoreEditDb)

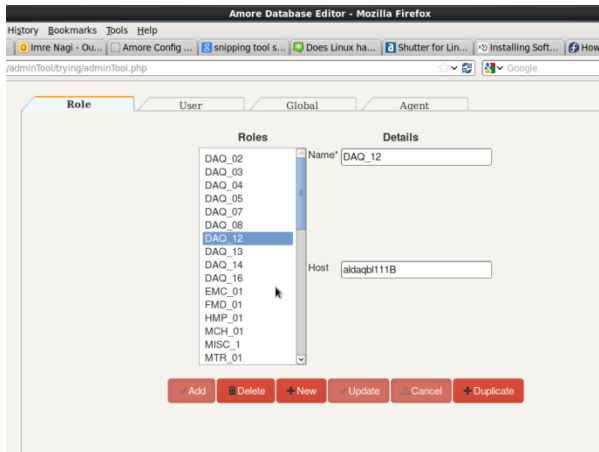
AmoreEditDb is an expert user interface to browse and edit the database. It allows the operator to edit Roles, Users, Globals, and Agents table from AMORE database without manually typing

queries for MySQL. Therefore, this application consists of 4 tabs for editing the records from different tables in AMORE database :

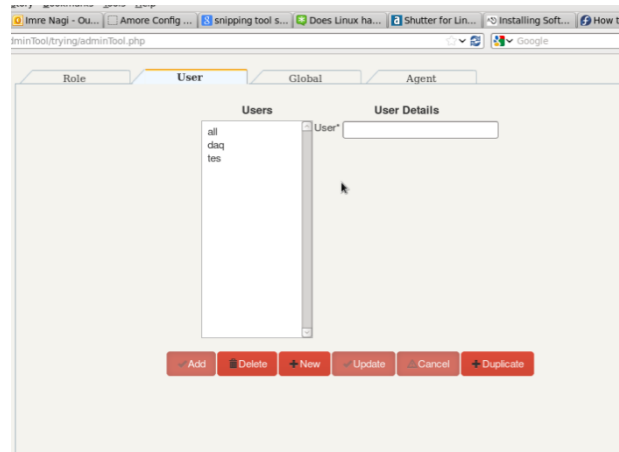
- **Role.** This table maps the real name of the DQM nodes (host) with their role name.
- **User.** This table consists of many users having a privilege to start, stop and restart the agents.
- **Globals.** This table contains various global variable used throughout the DQM system.
- **Agents.** The list of agents and their properties.

As we are working with MySQL database, we need to send several queries from this application into MySQL server. To handle this case, we have created a single file in handling MySQL query (built in PHP and Object Oriented paradigm). On the main page, when operator click a button, it will submit this form to the server through a controller file (written in PHP) to choose which action will be taken (depends on which button is pressed). If the variable which is passed to the server is correct, the database will store its value into the table.

In order to have a neat and user friendly interface for this application, it was built by using DTHMLX library to make tabs, Javascript for operator and browser interaction, and CSS3 for creating a minimalist page style.



(a)



(b)

Figure 5. AMORE Database Editor Running in Web-Browser: (a) Role Tab (b) User Tab

When the application is started, several buttons (Delete, New, and Duplicate) will be enabled automatically while the others will be disabled. Using these features, an operator will be able to delete the selected roles / user / globals / agents, create a new record and duplicate an existing records. The Update and Cancel button will be enabled after the operator does several changes in the textbox and Add button will be enabled for submitting a new record after New Button is pressed. To show the details of the option list in this application, operator needs to click an option from the list and then Javascript will set its value into textboxes.

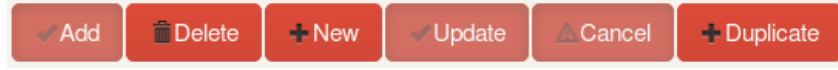
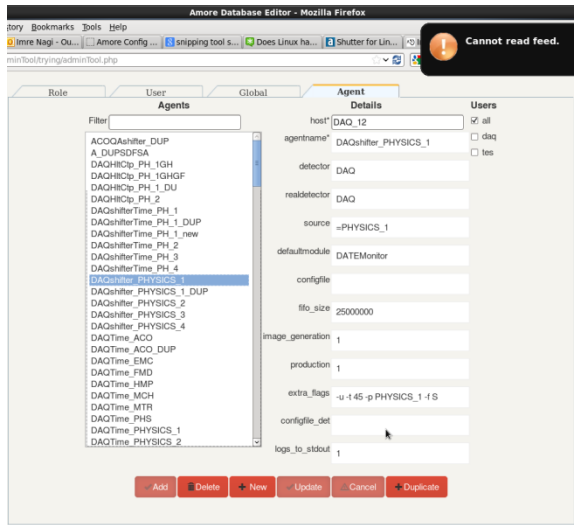
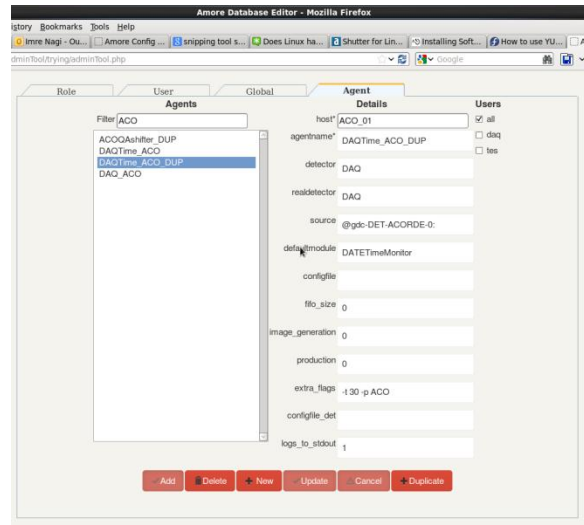


Figure 6. Buttons in Amore Database Editor Web Application

In order to avoid an empty input from operator, this application checks whether the primary key input (based on database) typed by the operator is empty or not. If yes, this application will not record the value into the database and return to last tab opened. In another case, if the operator leaves a non primary key input empty, the system will automatically assign a default value into the database (if it was configured at the table creation).



(a)



(b)

Figure 7. (a) Agents Tab (b) Agents Tab Using Agents Filter

The figures above show several properties describing the agents table in AMORE database. This properties will determine how the agents work. Let's say, one or more users from the table users are associated to every agent. They represent the users allowed to start, stop and restart the agent. If the user "all" is associated to the agent, all users are allowed to start, stop and restart the agent. So, when creating a new agent, the operator must specify the allowed user. If the user is not present in the table or the field is left empty, the user "all" will be used.

3.2 AMORE Config File Database (AmoreConfigFileBrowser)

AMORE Config File Database is an interface to manage the configuration files. Each configuration files belongs to a certain detector and is stored in amoreref table in AMORE database. Using this application, the operator can download the configuration file from AMORE database and save it into specific directory on his local machine, upload a new configuration file for certain detector into the server, edit the content of configuration file, rename the configuration file, or remove it from the table.

If the operator wants to do several actions only on a certain detector, they can choose the detector by selecting it from the dropdown list at the top of the window. Using this feature, the operator will only see the configuration files which belong to this detector and the corresponding agents. This utility allows the operator to upload a new configuration file and to set the default configuration file to use for given agent from amoreconfig (agents table).

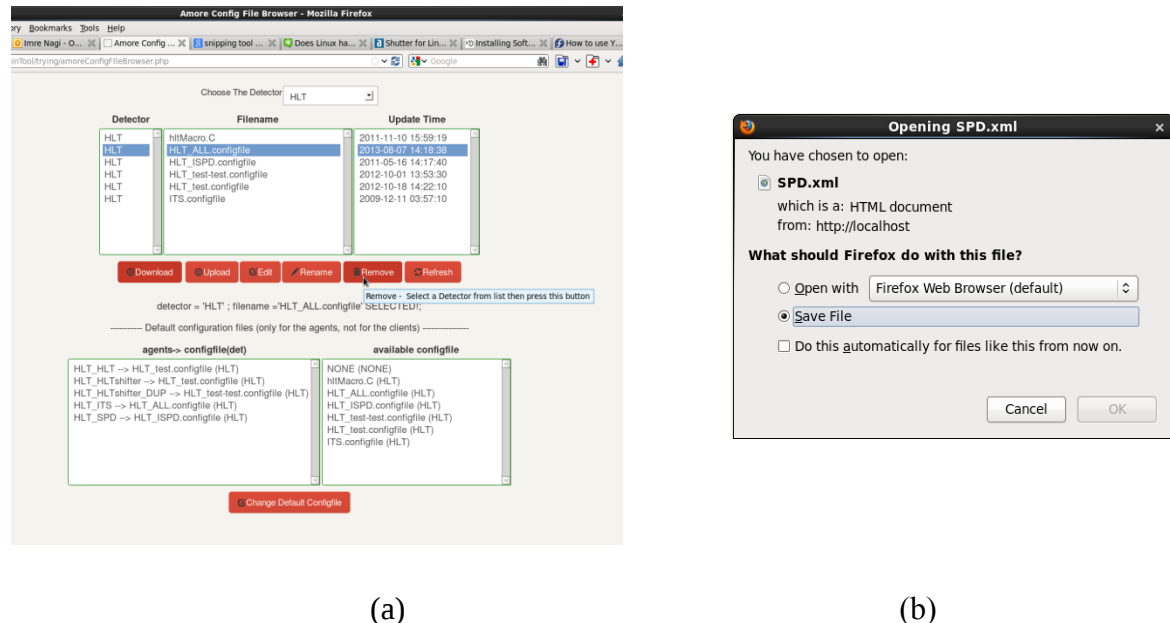


Figure 8. (a) AmoreConfigFileBrowser Web Application (b) Dialog box for download

In order to prevent the user from performing an action without selecting any entry of the table, several buttons (Edit, Rename, Remove) will be disabled until the operator choose a configuration file as well as Change Default Configfile button will be disabled until operator choose an agent and a new configuration file given. To improve the user experience, there are some hints available if the operator doesn't know how to use those buttons.



Figure 9. Buttons for AmoreConfigFileBrowser

As the old AmoreConfigFileBrowser application used a local text editor to edit the content of the configuration file, we implemented simple text editor created using simple PHP and HTML script (by clicking Edit button). As a result, this simple text editor running on the web-browser works perfectly to edit the and store the data into the database as well as the rename function (Rename button) and remove function (Remove button).

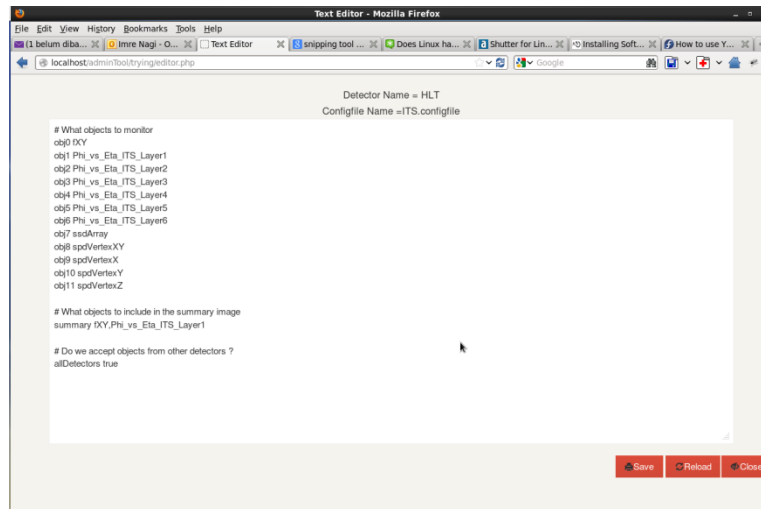


Figure 10. Text Editor in AmoreConfigFileBrowser Web Application

4 Conclusion

A new implementation for AMORE Web Administration Tool developed in PHP, HTML5, Javascript and CSS3, has been successfully used for adding, changing, and removing records in the MySQL database. This application will be ready to configure AMORE database for next start-up of LHC in the beginning of 2015.

**This work is supported by Kalla Group and Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung (STEI ITB), as sponsors of the author in CERN Summer Student Programme 2013.*

References

- [1] The ALICE DAQ Project website [Online] available : <http://aliceinfo.cern.ch/DAQ>
- [2] Nixon, R. (2012) . *Learning PHP, MySQL, JavaScript, and CSS, 2nd Edition : A Step-by-Step Guide to Creating Dynamic Websites*. O'Reilly Media.
- [3] von Haller, B. et al., *The ALICE Data Quality Monitoring System*, J. Phys.: Conf. Ser. 331 022030, 2011.
- [4] Telesca A. et al, *The ALICE Data Quality Monitoring system*, Real Time Conference, 2010. RT '10. 17th IEEE-NPSS.