

Monitoring tools of COMPASS experiment at CERN

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2015 J. Phys.: Conf. Ser. 664 082054

(<http://iopscience.iop.org/1742-6596/664/8/082054>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 188.184.3.56

This content was downloaded on 08/03/2016 at 22:15

Please note that [terms and conditions apply](#).

Monitoring tools of COMPASS experiment at CERN

M Bodlak⁵, V Frolov^{3,4}, S Huber², V Jary¹, I Konorov², D Levit²,
J Novy¹, R Salac¹, J Tomsa¹ and M Virius¹

¹ Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Czech republic, Brehova 7, 115 19 Prague 1, Czech Republic

² Physik-Department E18, Technische Universitat Munchen, Germany, James-Franck-Strasse, 85748 Garching

³ Joint Institute for Nuclear Research, Russia, Joliot-Curie 6, 141980 Dubna, Moscow region

⁴ European Organization for Nuclear Research - CERN, Switzerland, CH-1211, Geneve 23

⁵ Department of Low-Temperature Physics, Charles University, V Holesovickach 2, 180 00 Praha 8, Czech Republic

E-mail: jan.tomsa@cern.ch, josef.novy@cern.ch

Abstract. This paper briefly introduces the data acquisition system of the COMPASS experiment and is mainly focused on the part that is responsible for the monitoring of the nodes in the whole newly developed data acquisition system of this experiment. The COMPASS is a high energy particle experiment with a fixed target located at the SPS of the CERN laboratory in Geneva, Switzerland. The hardware of the data acquisition system has been upgraded to use FPGA cards that are responsible for data multiplexing and event building. The software counterpart of the system includes several processes deployed in heterogenous network environment. There are two processes, namely Message Logger and Message Browser, taking care of monitoring. These tools handle messages generated by nodes in the system. While Message Logger collects and saves messages to the database, the Message Browser serves as a graphical interface over the database containing these messages. For better performance, certain database optimizations have been used. Lastly, results of performance tests are presented.

1. Introduction

Nowadays, all modern high energy physics experiments are substantially dependent on fast and reliable data acquisition systems that are able to collect large quantities of data supplied by various detectors. This paper is focused on the new data acquisition system of the COMPASS experiment at CERN. Such a complex system requires thorough surveillance, therefore implementing monitoring tools is a necessary step.

At the beginning, the scientific program of the experiment is shortly introduced. Then the new DAQ system of COMPASS is described from both hardware and software point of view. The following section will describe the main processes involved in the DAQ as well as the DIM library used for network communication. The paper is concluded with performance test results.

2. The COMPASS experiment

COMPASS (COmmon Muon Proton Apparatus for Structure and Spectroscopy) is a high-energy particle physics experiment with fixed target situated on the M2 beamline of the Super Proton Synchrotron (SPS) particle accelerator at CERN laboratory in Geneva, Switzerland. The



scientific program of the COMPASS experiment was approved in 1997. It's goal was to study the structure of gluons and quarks and the spectroscopy of hadrons using high intensity muon and hadron beams. By the year 2010 the experiment entered it's second phase COMPASS-II ([4]) focusing on the Drell-Yan effect, the Primakoff scattering, and the Deeply Virtual Compton Scattering.

3. The Hardware of the DAQ

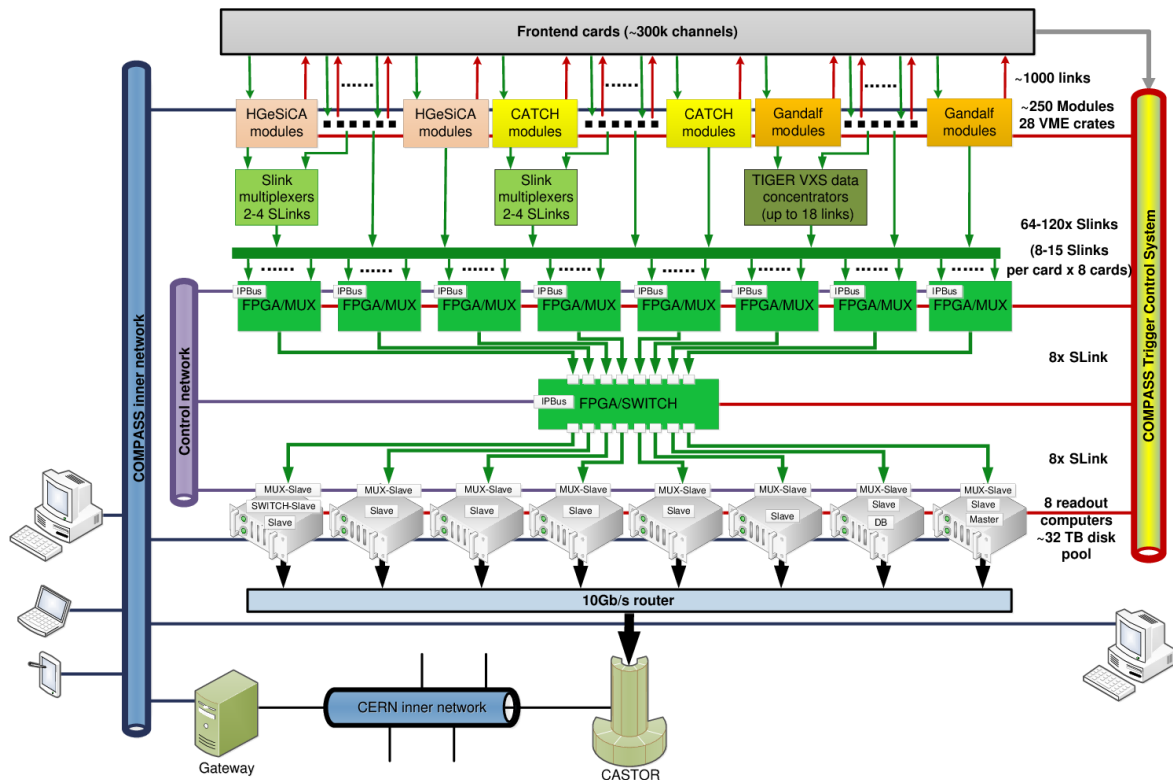


Figure 1. The new hardware configuration of the DAQ. The FPGA modules perform buffering and load balancing and complete event building process.

The hardware of the original Data Acquisition System consisted of several layers of different electronics. The layer closest to the detector is called frontend electronics. It's task was to capture signals directly from the detectors and convert them to digital values. There were approximately 300 000 of data channels coming from the first layer. This data was readout by roughly 250 of CATCH, GeSiCA, and Gandalf concentrator modules based on VME standard and grouped into subevents. Subevents were then sent over the S-Link network to 30 readout buffers (ROB) servers that were used to distribute the load during the whole SPS cycle. The next step was a layer of 20 event builder (EVB) server assembling subevents into final events. Data taking process was synchronized by the TCS (Trigger Control System). Full events were stored locally on hard disks and afterwards transferred to the central CERN storage facility CASTOR. [7] The trigger rate was up to 30 kHz with event size approximately 36 kB. Maximum in spill data rate of this system was 1.2 GB/s and maximum sustained data rate was 500 MB/s. [8]

Due to the increasing amount of incoming data, the old hardware is becoming insufficient. The obsolete parts that performed readout of subevents and buffering have been replaced with

new and modern devices. The first and the second layer (frontend electronics and concentrator modules) remain intact. The readout buffers and event building computers have been exchanged with modern FPGA (Field Programmable Gate Array) cards that take care of load balancing and event building on hardware level. In order not to affect data processing chain, the format of output data of the new system remains the same. The new setup has sufficient performance even in the long term view due to simple scalability [8]. The schema of the new system is shown in figure 1.

4. The software of the DAQ

The old DAQ software was based on a modified version of the DATE (Data Acquisition and Test Environment) software package originally developed for the LHC experiment ALICE ([6]). The new software is conceptually inspired by the ALICE software, however it is more lightweight and thus easier to use and to maintain. According to ([1], [2]), the new software consists of five main types of processes:

- **Master process** - a Qt console application. It is the most important part, almost all application logic is concentrated here [2]. It serves as a mediator between Slaves and GUI.
- **Slave process** - an application that controls and monitors a custom hardware. It is controlled and configured by the Master, it informs the Master process about the state of the hardware it is deployed on.
- **GUI** - a Qt GUI application designed for controlling and monitoring of the whole DAQ. It sends commands to the Master. Master sends back monitoring data about hardware controlled by Slaves. GUI can run in many instances, only one has the rights to change configuration and to execute control commands, the others are only allowed to monitor the status of DAQ.
- **Message Logger** - a console application that receives informative and error messages and stores them into the MySQL database. It is directly connected to the Master and to the Slave processes via the DIM service
- **Message Browser** - a GUI application that provides an intuitive access to messages from system (stored in the database) with an addition of online mode (displaying new messages in realtime). Equipped with filtering and sorting capabilities, it is able to run independently from the whole system in case of emergency.

5. DIM library for communication

DIM library (Distributed Information management) is a communication system originally developed for Delphi experiment at CERN [5]. It is based on the client/server paradigm, that is further extended by the concept of DNS (DIM Name Server). A service is a basic building block for communication and data transfer. Servers publish their services to the DNS and clients may subscribe to them. Clients receive data either periodically or once the data on the server side change. Communication in reverse direction is also possible, clients can send commands to servers and thus control their behaviour. The DIM library is heavily used in the communication between nodes of the new DAQ of COMPASS. The interaction of the DIM components is shown in figure 2.

6. Message Logger

Message Logger is a console application that serves for gathering and storing informative and error messages from the other nodes of the DAQ. It relies on the DIM library to secure communication with the master and the other nodes (it subscribes to all INFO services available in the system). If a relevant message arrives, it is stored into the MySQL database [3]. The schema can be seen in figure 3.

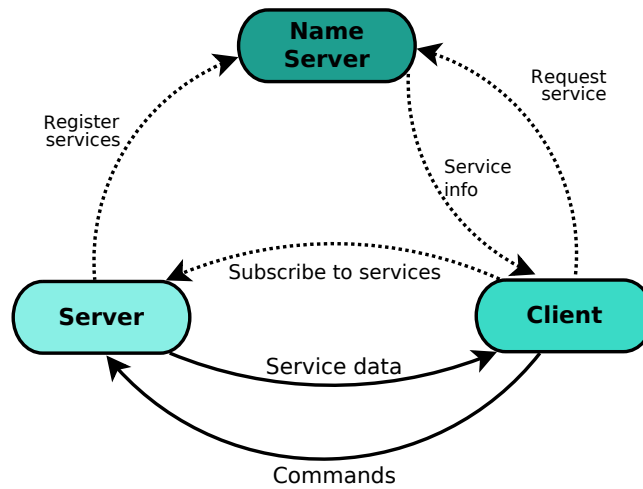


Figure 2. The interaction of DIM components. [5]

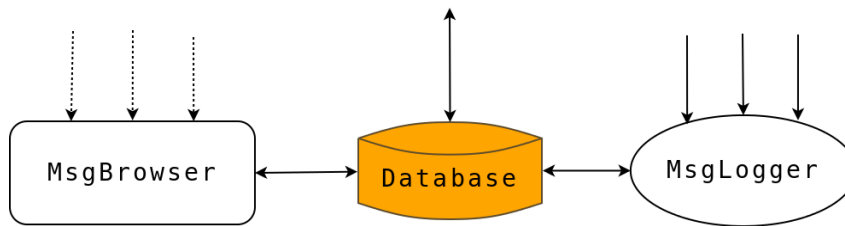


Figure 3. The schema of the monitoring tools.

7. Message Browser

The Message Browser is a GUI (Graphical User Interface) application developed using Qt framework. Its purpose is to display both messages already stored in the MySQL database by the Message Logger and the currently incoming messages. It offers rich and intuitive filtering and sorting options. It is intended as a replacement of the infoBrowser application from the DATE software package used in the old COMPASS DAQ setup. The screenshot of the Message Browser application can be seen in figure 4.

User can use Browser’s customisable filtering to display messages that are relevant at a given time. Filtering can be done according to the date and time (with precision of seconds) both from upper and lower limit. Another numerical restrictions (such as run number, spill number, and event number) can be applied as well. The severity (figure 5) (Info, Warning, Error, Fatal Error) of given message and it’s sender are also criteria for filtering. Besides ordering by columns values, user can choose which columns should be displayed and which hidden. Message Browser is also equipped with a “Default” button which resets all filter settings. There is also an option to hide all filters to maximize the screen usage for data. All messages are colored according to their severity to allow users to spot potential problems much quicker.

The Message Browser is based on the standard MVC (model-view-controller) software architecture. The model part of the architecture is represented by a grid-like underlying data structure containing all messages fetched from the database and received via the online part.

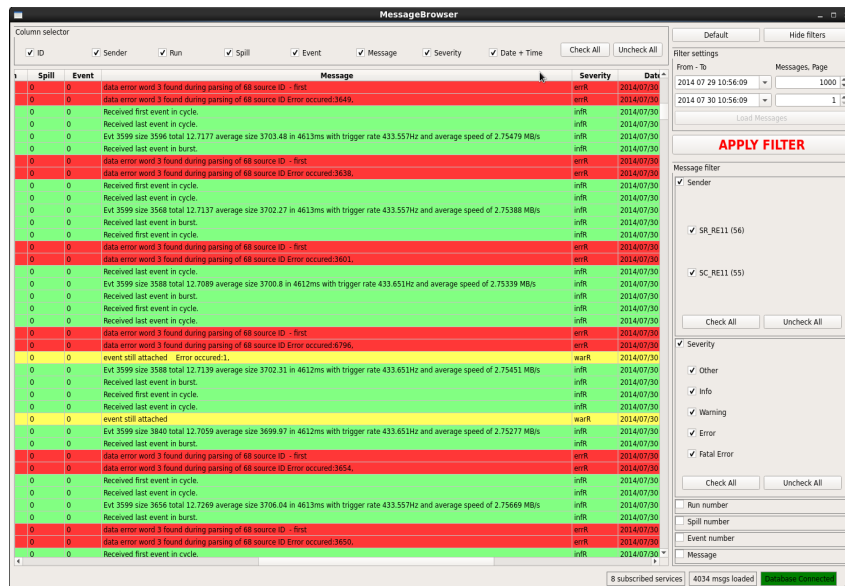


Figure 4. The interface of the Message Browser application.

The view is a table in the graphical interface displaying those messages. Controller stands for the filtering mechanism, allowing only the messages meeting the chosen criteria to be displayed.

Upon startup, the application loads messages from the MySQL database from the last 24 hours (this time period might be refined later) with maximum amount of 1000 messages. User can adjust the settings once the application is started. Then it tries to subscribe to the same services as the Message Logger does. If it succeeds, it begins to receive informative and error messages from system as they supervene. Therefore it is not necessary to poll the database periodically for new messages. It is worth mentioning that it does not overtake the job of the Message Logger of storing the messages into the database. If the Message Browser is unsuccessful in connecting to the DIM services (that probably means the whole system is down), the database-related part is still untouched and working. This approach allows efficient usage of system resources (no unnecessary polling), while ensuring independence from the rest of the system [3].

During the 2014 and 2015 run, Message Browser has been updated with new requested features. It is now equipped with a status bar showing current number of messages it has in cache, the number of INFO services it is connected to (INFO services are DIM services published by DAQ nodes, informing about their state), and the database connection status. User can choose how many messages should be loaded from the database in a single query. Online part can be switched on or off, message cache can be cleared.

8. The database

The database for the logging system is a part of the database setup for the new DAQ of the COMPASS experiment. The table `Message_log` contains all relevant data for a given message (sender information, run, spill and event numbers, text of the message, severity and stamp). As this will be the largest table inside the DAQ database, retrieval of messages needs to be optimised. A MyISAM storage engine has been used as it handles heavy read operations faster. The rest of the database uses InnoDB storage engine because of it's foreign key support (maintaining referential integrity). Since the Message Browser offers the possibility of advanced filtering and ordering of data, several indices have been created to speed up the retrieval. The index on the time stamp column is so far the most important as ordering by the time is the



Figure 5. The Severity filtering option, colorful messages.

most demanded. Because of the `Message_log` size (millions of rows), a separate database for this table has been created. Keeping it separate will make it easier to backup the main configuration database.

9. Performance tests

The browser has been tested for the speed of loading messages from the database. It was set to load 100, 200, 500, 1000, 2000, 5000, 10000, 20000, and 50000 messages from the database, each set was loaded 100 times in a row. Two times have been measured: time that was consumed by storing the messages into the internal data structure used by the Browser, and time it took to complete the whole task of loading the given amount of messages 100 times (including loading from the database, updating the filter model, etc.) Each test was rerun at least 10 times. The averaged time is drawn in figure 6. The time needed only to store the messages grows linearly with the amount of messages, whereas the total time is not so strictly linear. With higher amount of messages selected from the database, there is some loss of time caused by the increased overhead required to process such amounts of data. Moreover, cache effect of the database is used only for smaller amounts of data. Average time to load 100 messages is roughly 2.88ms, the storing part takes only 1.17ms.

All tests were run in virtualized 64-bit SLC 6.5 Linux with 2GB of RAM and 2 cores available in Virtualbox, hosted in Ubuntu 14.04 64-bit. The processor used is Intel Core i5-2410m (2 cores, 4 threads, 2.3-2.9 GHz).

10. Conclusion

This paper introduced the new DAQ system of the COMPASS experiment at CERN. Both hardware and software parts were described. From the software point of view, Message Logger and Message Browser applications were outlined. They are useful tools for logging and displaying informative messages about the DAQ system. They are used on the daily bases during data taking shifts. Tests show that these tools are able to meet performance requirements. Depending on the demand of the operators, the database indices and other parameters will be fine-tuned to assure smooth operation.

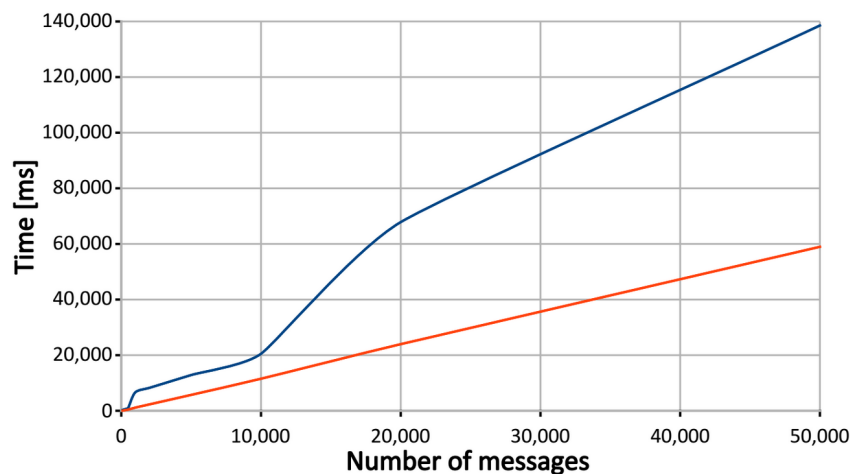


Figure 6. The time needed to load 100 times the given number of messages. The lower line describes time needed only to store the messages, the upper line shows the total time.

11. Acknowledgments

The work on this project has been supported in part by the following grants: MSMT LA08015 and SGS 11/167. It is also being supported by the Maier-Leibnitz-Labor, Garching and the DFG cluster of excellence Origin and Structure of the Universe.

References

- [1] Bodlak M 2012 *COMPASS DAQ – Database architecture and support utilities* (Prague: CTU in Prague)
- [2] Novy J 2012 *COMPASS DAQ – Basic Control System* (Prague: CTU in Prague)
- [3] Tomsa J 2014 *Monitoring tools for the data acquisition system of the COMPASS experiment at CERN* (Prague: CTU in Prague)
- [4] Adolph Ch et al. (The COMPASS Collaboration) *2010 COMPASS-II Proposal* CERN-SPSC-2010-014 SPSC-P-340
- [5] Gaspar C Donszelmann M Charpentier Ph 2000 DIM, a Portable, Light-Weight Package for Information Publishing, Data Transfer and Inter-process Communication (Padova, Italy) *Int. Conf. on Computing in High Energy and Nuclear Physics*
- [6] Anticic T. et al. 2006 *ALICE DAQ and ECS Users Guide* CERN, EDMS 616039
- [7] Bodlak M Frolov V Jary V Huber S Konorov I Levit D Mann A Novy J Paul S and Virius M 2012 New data acquisition system for the COMPASS experiment (Oxford, UK) *Topical Workshop on Electronic for Particle Physics*
- [8] Novy J Bodlak M Jary V Konorov I Virius M Steffen D Tomsa J Kveton A Levit D Huber S and Frolov V 2015 Pilot run of the new DAQ of the COMPASS experiment (Okinawa, Japan) *Int. Conf. Proc. CHEP2015*