The Message Reporting System of the ATLAS DAQ System

M. Caprini

National Institute of Physics and Nuclear Engineering,
Bucharest, Romania

I.Fedorko*

CERN,
Geneva CH-1211, Switzerland
**Corresponding author E-mail: ivan.fedorko@cern.ch*

S.Kolos

University of California,
Irvine, California, USA

The Message Reporting System (MRS) in the ATLAS data acquisition system (DAQ) is one package of the Online Software which acts as a glue of various elements of DAQ, High Level Trigger (HLT) and Detector Control System (DCS). The aim of the MRS is to provide a facility which allows all software components in ATLAS to report messages to other components of the distributed DAQ system. The processes requiring a MRS are on one hand applications that report error conditions or information and on the other hand message processors that receive reported messages. A message reporting application can inject one or more messages into the MRS at any time. An application wishing to receive messages can subscribe to a message group according to defined criteria. The application receives messages that fulfill the subscription criteria when they are reported to MRS. The receiver message processing can consist of anything from simply logging the messages in a file/terminal to performing message analysis. The inter-process communication is achieved using the CORBA technology. The design, architecture and the used technology of MRS are reviewed in this paper.

Keywords: ATLAS, DAQ, Online software, Message Reporting System

1. Introduction

The ATLAS experiment [1] is designed to observe the outcome of collisions between protons of 7 TeV provided by the Large Hadron Collider (LHC), which will start to operate in 2008 at CERN and deliver collisions at a rate of 40MHz. The designed luminosity value is $10^{34}cm^{-2}s^{-1}$. ATLAS experiment covers huge spectrum of physics research with main focus on Higgs boson and supersymetry search and precise measurement of W-boson and t-quark. Atlas records about 2MB of information per event at rate of ~200 events per second. The event selection is performed via a three-stage trigger system. The first level is based on custom-made electronics. The second and third level are software triggers. The ATLAS DAQ [2] system is composed of $\sim$ 25000 applications running on more then 2600 interconnected computers. The Online Software as a part of the DAQ is responsible for configuring, controlling and monitoring the data taking system, provides many services to DAQ and detectors system, is responsible for synchronization of entire system and supervision of processes. For diagnostic purpose applications inform about their states or report error condition and Online Software has to provide transport mechanism for such messages. Due to fact that ATLAS DAQ is a large distributed system the transport system has to offer also sufficient scalability. The Message Reporting System facility, as a part of Online Software, is providing the means to exchanging unique information and error messages across different DAQ applications.

2. System architecture

2.1. *General view*

The MRS system is responsible for transport, distribution and filtering of messages. It has been designed following the client-server paradigm. The processes of the DAQ requiring service of the MRS are on one hand applications that report specific information or error conditions and on the other hand message processors that receive reported messages. A message reporting application can inject messages into the MRS at any time. The message is composed of the message name, the severity, optional text, parameter values, optional parameter values, qualifiers (time stamp, process ID, etc.) and optional qualifiers. An application wishing to receive messages can subscribe to a message group according to criteria based on logical expression of the message name, severity and qualifiers. The application receives messages that fulfil the subscription criteria when they are reported to MRS. The MRS server can receive messages from one or more MRS senders. It searches the MRS message database (MD) for the message name. After a successful search the corresponding database information is added to the message instance. If the message is not found in the database, the information as received from the sender is passed to the client. Using of MD provides centralized message content management and reduce amount of data sent over the network. The MRS server keeps a list of subscribed applications, the receivers, and their corresponding subscription criteria. In the server can be set a filter to stop reporting messages of a set of messages, according to the filter criteria. If the message has passed the filter then it is distributed to those receivers in this internal list, which have matching subscription criteria. The client library, which is used by processes to send and to subscribe to messages, which has been issued by other processes, is implemented in C++ and Java while the server side has been implemented in C++. The inter-process communication is achieved using the CORBA technology [3].

2.2. *System implementation*

The MRS server has been initially implemented as a single Linux process, dealing with the reception of all messages and their redistribution to any subscribers. Nevertheless, the large number of applications involved in the DAQ, combined with the need of publishing many messages during the debugging phase of the experiment, has led recently to a complete redesign of this part of the MRS. The server functionality has been distributed over a publicly accessible MRS server process and a variable number of private MRS workers which are performing the actual work of receiving each a sub-set of the messages and to forwarding them to the subscribers. The number of installed private servers can be adjusted according to the size and complexity of the DAQ system and the required message throughput. Having been started up the private MRS servers join the public MRS server sending a handle (self handle) which can be used later to access the methods of the private MRS servers.

The sender API provides the MRS stream interface and offer possibility to create MRS message by user. In the ATLAS DAQ the error reporting is done using the Error Reporting System (ERS) package. Output from ERS can be linked to MRS stream and the MRS is used as transport mechanism of the ERS messages. A user application wishing to report messages has to connect to the public MRS server via the MRS sender API and get handle to a particular private MRS server. Therefore the message will automatically be sent directly to the private MRS server. For the case where the MRS server is not present in the system, the MRS sender writes the message to a local log file. An application wishing to receive messages must subscribe to the public MRS server with the help of the MRS receiver interface. Subscription includes the request to receive messages, implemented as a callback mechanism, and the subscription criteria. The public MRS server then transmits this subscription to all private servers. All private MRS servers report any corresponding message directly to the MRS receiver application interface (MRS receiver API). When an application wants to unsubscribe from a particular message group it sends this request in the same way to the public MRS server, which transmits it to all the private MRS servers. The DAQ expert can initiate set of command via the MRS command API, e.g. command can adjust the filter criteria for messages on their way through the private MRS servers.

3. Performance

The functionality tests showed that the implementation of the MRS library fulfils the test requirements and that the component functionalities correspond to the specification. The MRS system is regularly used during commissioning phase and proved that the MRS is able to fulfil expected functionality and reliability. With latest version of system we measured mean message reporting time for a distributed configuration. The servers, senders and receivers are running at different machines to emulate conditions as used in final DAQ system. For configuration when one public server and one private server are running on the same machine (4-core Intel Xeon CPU 3GHz, 2GB RAM), no receiver subscribed, and applications sending typical message at maximum rate we measured average message reporting rate at the server with a value of ∼9kHz. For more parallel sending applications the saturation value was observed at ∼17KHz. For the scalability test the private servers have been distributed on separate machine of the same type (2-core Intel Xeon CPU 2GHz, 500MB RAM). Applications sending messages at maximum rate were running on PCs with 2-core Intel Xeon CPU 3GHz and 500MB RAM. For the receivers were used PCs with 1-core Intel Xeon CPU 3GHz and 500MB RAM and reported message was written to file. The good scalability in expected range of active receivers was observed, see Fig 1.

4. Conclusion

We reported the design and implementation of the Message Reporting System as a part of the DAQ Online Software. The aim of the MRS is to provide a facility which allows all software components in the ATLAS DAQ system and related processes to report messages to other components of the distributed DAQ system. The MRS provides the functionality to transport,

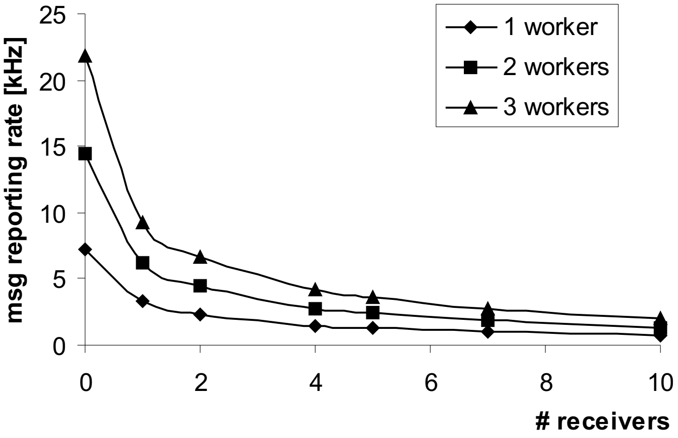

Fig. 1. The average message reporting rate as a function of number of subscribed receivers. The throughput of the system growths with number of the active private servers.

to filter and route messages. Usage of the MRS during detector and trigger commissioning phase demonstrated expected system functionality and usage of design with public and private servers provide requested scalability.

References

1. ATLAS Collaboration, Detector and Physics Performance Technical Design Report, CERN/LHCC 99-14/15,CERN, (Geneva, Switzerland, 1994).
2. High-Level Trigger Data Acquisition and Controls, CERN/LHCC/2003-022, (CERN, Geneva, Switzerland, 2003).
3. R. Jones *at al*, Applications of CORBA in the ATLAS prototype DAQ, IEEE Transactions on Nuclear Science, volume 47, Issue 2, Part 1, (April 2000), Page(s):244 - 249.