Nuclear Instruments and Methods in Physics Research A 623 (2010) 549–551

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Configuration and control of the ATLAS trigger and data acquisition

Giovanna Lehmann Miotto [a,*], Igor Aleksandrov [b], Antonio Amorim [c], Giuseppe Avolio [d], Elisabeta Badescu [e], Mihai Caprini [e], Alina Corso-Radu [d], Georgiana L. Darlea [a], Andre dos Anjos [f], Ivan Fedorko [a], Andrei Kazarov [a,g], Serguei Kolos [d], Vladislav Kotov [b], Andrew J. Lankford [d], Marius Leahu [a], Livio Mapelli [a], Raul Murillo Garcia [d], Yuri Ryabov [g], John E. Sloper [a], Igor Soloviev [a], Lourenco Vaz Gil Lopes [h]

[a] European Laboratory for Particle Physics (CERN), Geneva, Switzerland
[b] Joint Institute for Nuclear Research, Dubna, Russia
[c] Laboratorio de Instrumentaao e Fisica Experimental, Lisboa, Portugal
[d] Department of Physics and Astronomy, University of California, Irvine, CA, USA
[e] National Institute of Physics and Nuclear Engineering, Bucharest, Romania
[f] Department of Physics, University of Wisconsin, Madison, WI, USA
[g] Petersburg Nuclear Physics Institute (PNPI), St. Petersburg, Russia
[h] Laborat–rio de Instrumentaao e Fisica Experimental de Particulas, Coimbra

ARTICLE INFO

Available online 10 March 2010

Keywords:
LHC
ATLAS
Data acquisition
Online software
Run control

ABSTRACT

ATLAS is a general purpose experiment aimed at studying high-energy particle interactions at the Large Hadron Collider (LHC). This paper describes the evolution of the Controls and Configuration system of the ATLAS Trigger and Data Acquisition from the Technical Design Report to the first events taken with circulating beams. We present the lessons learned during the development.

1. Introduction

The ATLAS [1] Trigger and Data Acquisition (TDAQ) is a large computing system running on 3000 computers. Its aim is to readout, assemble, select and store interesting collision data generated within the detector. TDAQ control occurs over a dedicated Gigabit Ethernet network.

In this paper we describe the evolution of the controls and configuration system of the TDAQ from the Technical Design Report (TDR) [2] in 2003 to the first events taken at CERN with circulating beams in autumn 2008. We will present the status of the initial system and the reasons for launching a new development project; then we will highlight the areas in which major upgrades were performed. We will conclude with an overall assessment of the project results.

2. The system after the ATLAS TDAQ TDR

The TDR included a complete design of all aspects of control and configuration of the TDAQ. Validation of the existing implementation was done on small-scale systems with real detector readout [3] and on large computing farms with simulated data [4]. All in all the functionality and performance assessment was positive—except for the fault tolerance of the system, which was deemed insufficient.

Nevertheless, we identified a set of missing features and areas of concern, most of which had not been formulated as requirements to the original design. Hence a project was launched in autumn 2004, with the goal of reviewing all configuration and control aspects and of upgrading the design and implementation in time for the experiment start-up. As major constraints, since the lifetime of this project was overlapping with the overall commissioning of the TDAQ and of the detectors, the project had to deliver working software at any TDAQ software release (every 4–6 months), and changes to public API changes had to be minimized.

3. The controls and configuration system

The system encompasses all the software required to configure and control the data taking of the experiment. It is designed following a layered component model: at the very bottom are common base libraries, which include an in-house developed

[*] Corresponding author. Tel.: +41 22 7672369.
E-mail address: Giovanna.Lehmann@cern.ch (G.L. Miotto).

0168-9002/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.nima.2010.03.066

database—the *Object Kernel Support* (OKS) [5]—and the libraries for the CORBA-based *Inter Process Communication* (IPC) [6]. Higher up is a service layer, including the *Process Manager* and the *Message Reporting Service*. The application layer contains the *Run Control*, the *Diagnostics and Verification System* and the *Expert System*. On top there is a set of graphical user interfaces.

In the following sections we will show how the introduction of new requirements has reshaped the original design. For clarity, we have subdivided them into five distinct areas: security, auditing, scalability, error recovery and operability.

4. Security

Computing security was never really considered an issue in previous high energy physics experiments and had not been taken into account in the TDR. However, information security does not only address concerns about malicious intrusions but also about unintentional mistakes. In a collaboration of more than 2000 people this second aspect can easily become the main threat to the experiment's operation.

Requirements in this area remained confused until very late, thus leaving very little time for the deployment of a coherent, experiment wide strategy.

Traceability is the first step for the introduction of any kind of security measures: thus service accounts were discouraged in favor of personal ones. The need for preserving the identity of an actor also when he launches processes via the *Process Manager* required a complete re-design of this component [7].

The second step was to grant each user the minimum set of permissions requested to perform his job. To this purpose we developed an *Access Management* system [8], based on the role based access control paradigm. Authorization for a specific action is granted to a user based on his expertise, on his present function and on the experiment status.

We upgraded most of the critical services to use the *Access Manager* in order to accept or refuse user requests. Furthermore we introduced the so-called *OKS Server* [9]: its main task is to grant the permission to modify different parts of the configuration database according to the active roles of the requester. In addition, it performs a series of consistency checks on the database before accepting a modification and it allows any change to be traced and rolled back.

Finally, mechanisms had to be put in place to allow collaborators who are not on site to monitor the status of the detector. Information on data taking sessions is thus replicated to a read-only mirror copy, accessible from the public network.

The access management model developed within the project was embraced, besides TDAQ, by the detector control system and by the overall computing infrastructure at the experiment site.

5. Auditing

Under auditing we include all activities about post mortem analysis of a data taking session. No tools were foreseen in the original design for this purpose.

As a first measure we introduced a common *Error Reporting* package to provide a uniform message format and a common reporting policy, and a *Log Service* [10] component, in order to store all messages produced by the applications into a relational database. This change had a deep impact for the overall TDAQ software, which had to be modified accordingly. On the other hand it was a necessary step to allow for efficient message retrieval and for any kind of automated error analysis. Based on this novel ground we developed two new tools: a Java application which allows to browse through the messages and search through them according to a set of criteria and a MATLAB [11] based tool for automated analysis.

We also introduced the possibility of browsing through application log files, which are still maintained for low level debugging purposes, via a graphical user interface [12], without the need for logging onto the different hosts. For the management of the sizeable amount of log files that are continuously produced, we developed the *Farm Tools* component, which regularly removes them from the data taking machines after archiving them to a central storage area.

Finally, in order to be able to retrieve the configuration used for a specific run, we established a *Run Number Service* to assign a unique identifier to each run, and the *OKS Archiver* [14] that stores the configuration of each run into a relational database.

6. Scalability

At the time of the TDR the system was expected to be able to control and configure some 5000 software processes. The introduction of multi-core technologies in the high level trigger farms progressively scaled up the number of processes by one order of magnitude.

This evolution could be addressed without a change in the overall architecture, which would have caused major disruption to the experiment's commissioning. Nevertheless, we had to re-design low level components such as the *Message Reporting Service* [13] to cope with the increased load. Furthermore, we developed and deployed IPC proxies on the multi-core machines in order to limit the number of TCP connections to be handled by the central services. We revised the configuration service [14] in all its parts: in particular, the database schema was optimized to reduce any information duplication and a hierarchical tree of *Remote Database Servers* was put in place to serve configuration data to all clients. Finally, even at the application and graphical interfaces layer we had to re-implement part of the software to deal with the augmented demand on scalability and performance.

Dedicated tests have proven that no scalability issues appear for 30 thousand processes. Larger configurations could not be measured yet, due to lack of hardware resources.

7. Error recovery

While fault tolerance had to be embedded in the whole dataflow software, improved error recovery was one of the project targets. Besides the capability of ignoring or restarting some non-essential failing processes, the TDR TDAQ had no means of recovering from complex errors. In order to allow for a more flexible evolution of the error recovery, we re-designed and re-implemented the *Run Control* and *Expert System* components—originally combined—with a clear separation of duties. This new structure has allowed to progressively introduce more and more sophisticated error recovery scenarios [15]. Of course the *Expert System* will still evolve as the experiment will reach a mature state and the number of actions that do not need a decision by the operator will increase.

8. Operability

The commissioning runs at the experiment's site, which started in 2006, showed the importance of simplifying the working environment for the operator, having easy to use graphical interfaces and providing an online help to the tools.

The effort in this area led to the re-implementation of most of the graphical user interfaces as well as to the development of the so-called *Control Room Desktop*, a working environment based on Kiosk [16], which exposes via simple menus and icons all the tools and web sites that the operator needs to access.

It is difficult to measure the success of our efforts in this area. The fact that more and more collaborators are capable of operating the TDAQ without expert's help is surely a positive indicator, but it is likely that the development of graphical interfaces will continue, fostered by new ideas and users' feedback.

9. Results and outlook

The controls and configuration project delivered a complete system in time for the first circulation of protons at the LHC in September 2008, thus achieving its main goal.

Despite the fact that the design of most components was upgraded and that several new components were introduced, the overall structure of the original software and its main design guidelines could be preserved. In particular, the substantial change of performance requirements in terms of scalability, which could have caused the invalidation of the complete communication model, could be addressed by localised software optimisations and by an increase of computing and networking resources, thanks to the excellent performance of the CORBA based IPC.

While some of the new features were introduced by adding standalone components, other aspects proved to be quite problematic and required the reworking of many packages. The introduction of a common error reporting package at an advanced stage of the TDAQ and detector software development was quite painful: since not all developers modified their software to comply to the new guidelines, automated error analysis is still very complicated and only partially useful. Similarly, we would like to emphasize the importance for security aspects to be part of the software development cycle from the very beginning.

We are now in the consolidation phase and further development is only expected in the areas of error recovery and, possibly, graphical user interfaces. Nevertheless, the project will be kept active at least until the complete trigger farms will be commissioned at the experiment site, as only at that moment we will be able to fully asses the achievement of the project goals.

References

[1] ATLAS Collaboration, J. Instr. 3 (2008) S08003, 437pp.

[2] ATLAS Collaboration, ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report, CERN/LHCC/2003-022, June 2003, See 〈http://cdsweb.cern.ch/record/616089/〉.

[3] S. Gadomski, et al., IEEE Trans. Nucl. Sci. NS-53 (2006) 2156.

[4] D. Burckhart-Chromek, et al., Testing on a large scale: running the ATLAS data acquisition and high level trogger software on 770 PC nodes, ATL-DAQ-2006-002, CHEP2006, Mumbai, Maharashtra, India, 13–17 February 2006.

[5] R. Jones, IEEE Trans. Nucl. Sci. NS-45 (4) (1998) 1958.

[6] R. Jones, et al., IEEE Trans. Nucl. Sci. NS-47 (2) (2000) 331.

[7] G. Avolio, et al., IEEE Trans. Nucl. Sci. NS-55 (1) (2008) 399.

[8] M.C. Leahu, et al., IEEE Trans. Nucl. Sci. NS-55 (1) (2008) 386.

[9] I. Soloviev 〈twiki.cern.ch/twiki/bin/view/Atlas/DaqHltOks〉.

[10] R. Murillo Garcia, G. Lehmann Miotto, IEEE Trans. Nucl. Sci. NS-54 (1) (2007) 202.

[11] 〈www.mathworks.com/products/matlab/〉.

[12] A. Kazarov, et al., IEEE Trans. Nucl. Sci. NS-54 (3) (2007) 604.

[13] I. Fedorko, et al., The Message Reporting System of the ATLAS DAQ System, ATL-DAQ-CONF-2007-029, Presented at 10th ICATPP Conference on Astroparticle, Particle, Space Physics, Detectors and Medical Physics Applications, Como, Italy, 8–12 October 2007.

[14] J. Almeida, et al., J. Phys. Conf. Ser. 119 (2008) 022004.

[15] J.E. Sloper, et al., IEEE Trans. Nucl. Sci. NS-55 (1) (2008) 405.

[16] Kiosk 〈http://en.wikipedia.org/wiki/Kiosk_(KDE)〉.