Online remote monitoring facilities for the ATLAS experiment

S. Kolos¹, E. Alexandrov², E. Feng³, R. Hauser⁴, A. Iakovlev² and A. Zaytsev⁵

¹University of California, Irvine, USA
²Joint Institute of Nuclear Research, Dubna, Russia
³Enrico Fermi Institute, University of Chicago, USA
⁴Michigan State University, USA
⁵Budker Institute of Nuclear Physics, Novosibirsk, Russia

E-mail: Serguei.Kolos@cern.ch

Abstract. ATLAS is one of the four LHC experiments which started to be operated in the collisions mode in 2010. The ATLAS apparatus itself as well as the Trigger and the DAQ system are extremely complex facilities which have been built up by the collaboration including 144 institutes from 33 countries. The effective running of the experiment is supported by a large number of experts distributed all over the world. This paper describes the online remote monitoring system which has been developed in the ATLAS Trigger and DAQ(TDAQ) community in order to support efficient participation of the experts from remote institutes in the exploitation of the experiment. The facilities provided by the remote monitoring system are ranging from the WEB based access to the general status and data quality for the ongoing data taking session to the scalable service providing real-time mirroring of the detailed monitoring data from the experimental area to the dedicated computers in the CERN public network, where this data is made available to remote users through the same set of software tools as being used in the main ATLAS control room. The remote monitoring facilities have been put in place in 2009 to support the ATLAS commissioning and have been improved in face of the first collisions runs based on the feedback which was received from the users. Now the remote monitoring system are in mature state and being actively used by the ATLAS collaboration for running the experiment.

1. Introduction

ATLAS is one of the four major LHC experiments at CERN, which was put into production in 2010. ATLAS construction has been started more than 15 years ago and involve more than 3000 people from all over the world. In 2010 the ATLAS collaboration has put a lot of efforts to minimizing the dead-time of the experiment in in order to acquire as much data as LHC can deliver. The first experience quickly shown that the efficient utilization of the LHC beam time strictly depends on the fast availability of the ATLAS detector, Trigger and DAQ systems experts.

Unfortunately there are some difficulties with providing expertise directly at the ATLAS experimental aria, as many of the ATLAS collaborators are involved in other activities at their home institutes. Even those experts, who are based at CERN, are not able to access monitoring information directly unless they are physically present in one of the ATLAS control rooms,
since the access to the ATLAS Technical Control Network (ATCN) is cut during data taking operations due to security considerations. In order to provide support for remote monitoring of the experiment several services have been put in place for exporting data from the ATLAS online monitoring system to the CERN Global Public Network(GPN) where it can be non-restrictively used by all members of the ATLAS collaboration.

2. Online monitoring system overview
The online monitoring system of the ATLAS experiment is organized as a distributed modular hierarchical facility which includes several applications, ranging from low-level information sharing services up to high-level analysis frameworks and graphical interfaces. This organization offers high flexibility in terms of accommodating various types of monitoring information as well as in configuring analysis algorithms and managing their outcome. The overall structure of the monitoring system is shown in Figure 1.

Figure 1. The Event Monitoring service (Emon) [1] provides Event Analysis Frameworks with statistical samples of physics events which are selected according to the given physics properties, like stream, trigger or sub-detector type. Event Analysis Frameworks produce histograms out of those events and publish them to the Information Service (IS). The Data Quality Monitoring Framework (DQMF) [2] retrieve those histograms from the IS, analyzes them with the pre-configured data quality algorithms and publishes the Data Quality results produced by those algorithms back to the IS.

At any given moment the Information Service contains a snapshot of the most up-to-date monitoring information produced by the experiment. The ATLAS experts and members of the shift crew can use various monitoring GUI applications to retrieve and display specific information from the IS in real time.

2.1. The Information Service implementation
The Information Service is using the client-server architecture with the Information Repository acting as a server to hold information provided by ATLAS software applications. Currently the Information Repository is implemented by a number of processes, called IS Servers, which are distributed over several computing nodes in a location transparent way. Each server has a unique identifier, which a client application is using to communicate with this server. The IS Application Program Interface (API) is available in C++, Java and Python. The Information Repository supports three main types of client interactions which are shown in Figure 2.

2.1.1. The Object Model The IS is using three level object model for the stored information. It provides access both to information values and to the classes that describe the types of those values. The IS type description is provided in a form of XML. IS also provides tools for generating C++ and Java classes to be used for the information publication and retrieval.
2.1.2. The Object Identification
Each information object has a unique name in the IS repository. In the current implementation, the name is a character string which must have the following format:

\[
\text{InformationName}::=\text{ServerName}<.>\text{ObjectName}
\]

The ServerName must be a valid name of one of an existing IS server application. The ObjectName must be unique for each information object in this IS server.

2.1.3. Inter-process Communication Technology
The current IS implementation is based on the Common Object Request Broker Architecture (CORBA) [3] standard. The C++ implementation is done on top of the omniORB [4] CORBA broker and Java implementation is using the JacORB one [5]. However both C++ and Java IS APIs are fully independent of the underlying communication layer, thus allowing to change to a different communication technology without affecting the IS client applications.

2.2. The Monitoring Information: total amount and update rate
Currently the ATLAS TDAQ system consists from about 10000 processes, each of them publishing some information to the IS. The minimal information, which is published by any single process, is the one that reflects the process’ state. In addition to that many applications are publishing a lot of extra information which represent for example the status of the hardware those processes are controlling or the the quality of the collected physics data. In the end the total amount of information available in the IS for an average data taking session is at the order of few tens Gigabytes. This information is periodically updated with the update period varying from 5 to 100 seconds for different information types.

3. Remote Monitoring Services
The total amount and the update rate of the information produced for online monitoring in the ATLAS experiment is too high to be made non-restrictively accessible by every member of the ATLAS collaboration. On the other hand for the majority of experts observing a small sub-set of the monitoring data would be sufficient if the experiment is running under regular conditions. Only in case of issues, experts from the problematic sub-systems would need to request some additional monitoring information, but in this case the number of such requests will be limited. Based on those considerations it has been decided to split the ATLAS Remote Monitoring system into three distinct facilities with the following characteristics:

- the Public Remote Monitoring is dedicated for providing small predefined sub-set of the monitoring data for the whole ATLAS community;
the Expert Remote Monitoring facility is designed to provide any single piece of monitoring data on request to an ATLAS sub-system expert;

- the Remote Shifter Monitoring facility was developed to be used by a few people (shifters) for permanent real-time monitoring similar to what is being done by the ATLAS sub-systems shifters in the ATLAS Control Rooms.

Those facilities are dedicated to different groups of the ATLAS collaborators and are essentially different from each other with respect to the amount of information they are providing and the mode, in which they are interacting with their users. That explains the divergence in the architecture and technologies which have been used for implementing those facilities.

### 3.1. Public Remote Monitoring facility

The top level monitoring data for all members of the ATLAS collaboration is provided by the ATLAS Web server, which is connected to the ATCN. This Web server hosts a set of HTML files which are periodically updated by the Web Monitoring Interface (WMI) service which is running on one of the dedicated monitoring nodes inside ATCN. This service is periodically executing a number of plug-ins which select the appropriate monitoring information to be made available via Web and also defines the layout of the generated HTML pages. The architecture of the WMI service is shown in Figure 3.

![Figure 3. Static Web monitoring architecture](image)

Different plug-ins are responsible for providing different types of monitoring information while the WMI framework itself is taking care of converting this information into HTML and storing it in a form of files at the appropriate place on the Web Server. For the moment there are 4 WMI plug-ins which are providing the following information for the ongoing ATLAS data taking session:

- the overall state of the run, including run number, run type, run time, etc.;
- the trigger configuration and instantaneous trigger rates together with their history for the ongoing run;
- the instantaneous data quality status for the ATLAS sub-systems;
- and the accumulated run efficiency.

Each plug-in is executed in a separate instance of the WMI process, thus achieving two different goals: improving robustness against plug-ins failures and allowing for easy scalability.

### 3.2. Expert Remote Monitoring facility

The expert remote monitoring facility gives access to every single item in the IS on demand via the HTTP protocol. It’s architecture is shown in Figure 4. The online ATLAS web server
provides the interface to the outside world while internally it interfaces to multiple SCGI based servers written in Python. The latter ones are talking to the online IS Repository.

The system provides authentication, caching and proxying of the information via widely used web industry standards and can be accessed via normal browsers or from any programming language that supports HTTP. For histograms the server side can do the rasterization of images using the ROOT libraries and return those in a number of standard formats (PNG, JPEG, GIF). Therefore there is no need for any HEP specific software on the client machine.

This design allows users to present the information they are interested in their preferable form. It strikes a balance between having to copy all available information to the outside world, even if it is not used, and having only a predefined static sub-set available. The latency to access an item is determined by the HTTP access, typically in the order of 200 ms if one is outside of CERN. There are limits on the maximum number of concurrent requests, enforced by the servers to control the load on the online monitoring system, which might impact data taking.

Client usage can be as easy as including an IMG tag into a web page where the source is a URL pointing to the web service. More complicated scenarios include full browsers for all information written in JavaScript. Scripts are used to regularly access information and post-process it outside of the ATLAS online network. That is very useful for providing sub-system specific monitoring which is operating with a sub-set of monitoring data available for the whole experiment. Different sub-systems may have different ways of post-processing and presenting their specific information. That is a complimentary functionality to the one provided by the Public Remote Monitoring facility, which is exporting a predefined set of some generic monitoring information, which is common for all sub-systems. WMI is performing the necessary pre-processing of the information before converting it to HTML which eliminates the necessity of having this processing being done by every client accessing this information.

### 3.3. Remote Shifter Monitoring facility

While the Web pages are available to all members of the ATLAS collaboration the real-time monitoring system provides restricted access for a limited number of users only. The ATLAS remote monitoring system architecture is shown in Figure 5.

This facility provides the real-time copy of the online monitoring information from the master IS Repository at the ATCN to its mirror counterpart which is running at CERN GPN. The information is always passed one-way, from ATCN to GPN, in order to preserve the security of the ATLAS data taking. Special network configuration restrictions are applied to prevent any kind of network connections from the ‘mirror’ monitoring nodes to the nodes inside ATCN. The delay of information transfer is at the order of few milliseconds, so remote users practically see the monitoring information at the same time as it become available in the experimental area. The ‘mirror’ Information Service is providing the same API as the master one in the experimental area and due to that users of this facility are working with the same monitoring GUI applications which are being used by the ATLAS shifters and experts in the ATLAS control...
room. This makes this facility very attractive for been used for remote shifts since it involves no additional learning curve for the shifters and at the same time is having zero maintenance cost for the monitoring GUI developers.

3.3.1. Remote desktops Since remote users run Monitoring GUI applications on the machines located at CERN, an efficient and reactive way of passing screen information from CERN to remote sites is required. However, traditional X11 display export is extremely slow, and its high latency does not provide a viable desktop environment for this kind of work. There are several GPL and commercial software packages which can be used for speed up X11 connections as well as for providing some tools for remote session management, for example, the NX Server implementation provided by NoMachine, and Secure Global Desktop (SGD) provided by Sun Microsystems (since acquired by Oracle). After a systematic evaluation of different remote desktop technologies based on the requirements of the ATLAS remote monitoring system, the NX software [6] was finally chosen.

3.3.2. NX technology NX is a protocol defined on top of X11 that allows running remote X11 sessions even across slow or low-bandwidth network connections. NoMachine provides implementations of both NX Server and Client that use X11 protocol compression techniques and an integrated set of proxy agents that transparently run graphical desktops and X Window applications through the network, improving the performance by reducing round-trips and implementing strict flow-control of data traveling through low-bandwidth links.

The clients, which want to use the Remote Shifter Monitoring facility, have to install the NX Client software on their local machines. There are a number of free NX Client implementations available for most popular platforms, in particular Windows, Linux, Mac OS X, and Solaris. The NoMachine commercial license for the server software is available for Linux and Solaris, but the Remote Shifter Monitoring facility is using the free GPL implementation of NX server called FreeNX that is available for most flavors of Linux [7]. The Free NX server software has been installed on the remote monitoring user nodes, allowing to have a significant gain in performance of the X11 sessions, opened on those nodes, due to the on-the-fly compression and extensive caching which are provided by the NX on top of the standard X11 protocol.

3.4. Evaluation criteria
Alternative remote desktop applications exist, including Sun Secure Global Desktop (SGD) [8]. This software provides secure access to centralized Windows, UNIX/Linux, Mainframe and Midrange applications from a wide range of popular client devices, including Microsoft Windows.
PCs, Solaris OS Workstations, and thin clients. The client is required to have a Java-technology enabled web browser, e.g. Mozilla, Firefox, IE, Opera, etc with suitable Java plug-in installed. The SGD commercial license for the server software is available for Linux and Solaris.

The two software systems were tested against a set of criteria that were developed based on the requirements for the ATLAS remote monitoring system. These included the cost of a license and technical support, remote desktop functionality, performance and resource overhead, user configuration and session management facility, and finally compatibility with CERN software.

From the evaluation, the NoMachine NX and Sun SGD technologies were both found to provide the basic functionality necessary for ATLAS remote monitoring. NX-based systems are more responsive and less resource-intensive, while SGD provides more flexibility for applications and account management.

Due to the absence of experience with the final remote monitoring system during real data-taking at the time of the evaluation, as well as some uncertainties with regard to the scope of its application, the following strategy was taken during 2009. FreeNX Server was installed on the remote monitoring nodes with the aim of having it widely used across all ATLAS sub-systems. Based this experience from the broad ATLAS community, a technical review was performed. It was chosen to continue using FreeNX (GPL license) since this product was found to satisfy the requirements of ATLAS remote monitoring. Alternatives were considered including the purchase of a commercial NoMachine license and use of the commercial Sun SGD system in order to obtain additional features and proper technical support. These were found to be unnecessary and cost ineffective.

4. Status and Conclusions
In order to optimize the usage of the computing resources, the Remote Monitoring system in ATLAS consists from three distinct facilities, which are different from each other by the amount of information, they are providing, as well as by the number of users they are able to serve. All those facilities have been in place at the start of the experiment early 2010 and have been used through the first year of the experiment by many ATLAS users from remote institutes as well as by those people who are actually staying at CERN but are not located in the ATLAS control room at their working time. Currently all together the Remote Monitoring facilities are using 9 computing nodes, which in 2010 were enough to satisfy all remote monitoring needs of the ATLAS collaboration:

- 2 nodes are used to run 4 different WMI plug-ins
- 2 nodes are used to run the IS Python SCGI servers
- 5 nodes are used for the remote shifter monitoring

All remote monitoring facilities are very well scalable so it would be straightforward to expand them for the larger number of users if remote monitoring requirements will evolve in the following years.

References
[1] Scholtes I The ATLAS Event Monitoring Service - Peer-to-Peer Data Distribution in High Energy Physics (IEEE Trans. Nucl. Sci. vol 55) pp 1610 - 1620.
[2] Kolos S 2007 Data Quality Monitoring Framework for the ATLAS Experiment at the LHC 15th IEEE Real Time Conf. IEEE NPSS (Fermilab, Batavia, IL, USA)
[3] Common Object Request Broker Architecture home www.corba.org
[4] omniORB home omniorb.sourceforge.net
[5] JacORB home www.jacorb.org
[6] NX home www.nomachine.com
[7] http://freenx.berlios.de/info.php
[8] http://www.sun.com/software/products/sgd/get.jsp