OHP: an Online Histogram Presenter for the ATLAS experiment

A. Dotti, Università di Pisa and INFN Sezione di Pisa, Pisa, Italy
M. Della Pietra, INFN Sezione di Napoli, Napoli, Italy
R. Ferrari, G. Gaudio, INFN Sezione di Pavia, Pavia, Italy
P. Adragna, Queen Mary, University of London, London, United Kingdom
D. Salvatore, Università della Calabria and INFN Gruppo di Cosenza, Cosenza, Italy
P. F. Zema, Università della Calabria, Cosenza, Italy and CERN, Geneva, Switzerland
W. Vandelli, Università degli Studi di Pavia and INFN Sezione di Pavia, Pavia, Italy
D. Cimino, Università di Pisa, Pisa, Italy
C. Roda, Università di Pisa and INFN, Sezione di Pisa, Pisa, Italy

Abstract

The Online Histogram Presenter is a general purpose, highly configurable, interactive presenter developed in the context of the Online Monitoring System of the ATLAS experiment. The GUI is based on the Qt® framework, and ROOT is the underlying technology for histogramming. In this paper, a complete description of OHP requirements, design and features is presented, as well as a description of its workflow.

INTRODUCTION

ATLAS [1] is one the four experiments being assembled at CERN at the Large Hadron Collider (LHC), the new accelerator that will provide proton-proton collisions with an energy of 14 TeV in the center-of-mass system of reference.

The ATLAS detector includes about 140 millions of electronic channels. The LHC collides proton bunches at a frequency of 40 MHz, and the output of the first level trigger will be about 10^5 Hz. This frequency will be further reduced by the higher trigger levels, and finally some hundreds of events will be selected and stored every second.

In view of the huge number of channels and the high event rate, a monitoring system is an essential tool to assess the status of the hardware and the quality of the data while they are being acquired.

THE ATLAS ONLINE MONITORING SYSTEM AND THE OHP

The Atlas Online Monitoring System is a highly distributed environment in which different processes run at the same time on different machines. All these processes produce information that must be delivered from providers to receivers.

Two different kinds of online monitoring data producers are foreseen in ATLAS. The low-level monitoring (GNAM) [2] will be devoted to check the hardware conditions (i.e. dead/noisy channels, electronic mappings) as well as the data quality (e.g. synchronizations among different detectors). The high-level monitoring (Athena) [3] will run mainly on the full events, and will check physics related quantities, such as momentum spectra, invariant masses and so on. Both these monitoring processes produce histograms summarising the results.

The ATLAS Trigger and Data Acquisition group (TDAQ) [4] offers a number of services to share this information among processes. These services are exploited to implement online monitoring applications. In particular, the Online Histogramming Service (OHS) is of interest to the OHP.

The Online Histogramming Service [5] manages histograms, providing a transient storage between histogram producers and displays. Histograms are published in the OHS server, where they are available to the entire system. The OHS also allows commands to be routed. Actually, any application can issue a command related to a particular histogram: the OHS takes care of sending the command to the appropriate histogram provider.

If a process wants to receive histograms, it must set up a histogram receiver and subscribe to the plots it needs. Subscription allows the process to be informed whenever the subscribed histogram change status, e.g. when the histogram is published or updated.

Two mechanisms are currently implemented in the OHS to inform processes. In the first one, the full histogram is sent to the subscriber every time its status changes (push mode). In the second one, only a notification is sent to the subscriber; the notified subscriber may then decide whether to retrieve the histogram from the OHS (pull mode). As we will see, this second mechanism is the key feature to minimise the network traffic, as we will see in the Requirements section.

The OHP acts as a general purpose interactive presenter, to display online monitoring histograms produced and published through the Online Histogramming Service. From this point of view, the OHP is a client of the OHS and not of the histogram producer. It unifies the approach to histogram visualisation inside the ATLAS online environment.

OHP REQUIREMENTS

A list of requirements the OHP has to fulfil has been prepared. Requirements are divided into functional and non-functional ones.

Functional requirements

1. OHP must allow the user to select a configurable set of histograms to be displayed. The possibility of choosing particular histograms is known as shifter mode;
2. OHP must allow the user to browse the whole content of the OHS. The possibility of browsing all the histograms is known as expert mode;
3. OHP must allow the people on shift to interact with the displayed histograms. This involves the possibility of zooming, fitting and changing the graphical appearance of the plots;
4. OHP must automatically display histograms when they are available;
5. OHP shall manage reference histograms;
6. OHP shall interact with histogram producers. This implies the possibility of changing histogram scales, resetting histogram content, turning on/off histogram production.
7. OHP shall allow the user to set the graphical attributes of the plots. This includes, for example, the possibility of setting the axis to a logarithmic scale.

Non-functional requirements

1. OHP must be integrated into the ATLAS TDAQ system;
2. OHP shall be independent from the histogram producer, since many different kinds of producers, developed independently, are expected.
3. OHP shall minimize the network traffic.

DESIGN

OHP is composed of the following subsystems (fig. 1):

the Core Subsystem which provides access to the histograms from the Graphical User Interface. It also controls all the other components.

the Histogram Receiver which is responsible for receiving notifications and retrieving histograms from OHS on request. Multiple subscriptions are possible: each subscription creates a dedicated receiver.

the Cache which is used to store the received notification and histogram, each Histogram Receiver owns a cache. The aim of the cache. The cache is a key component to minimise the network traffic, and the description of its usage is described below.

the Filtering Subsystem which filters the notifications that are related to the active canvas. The active canvas is the one on focus, i.e. the one that the user actually sees.

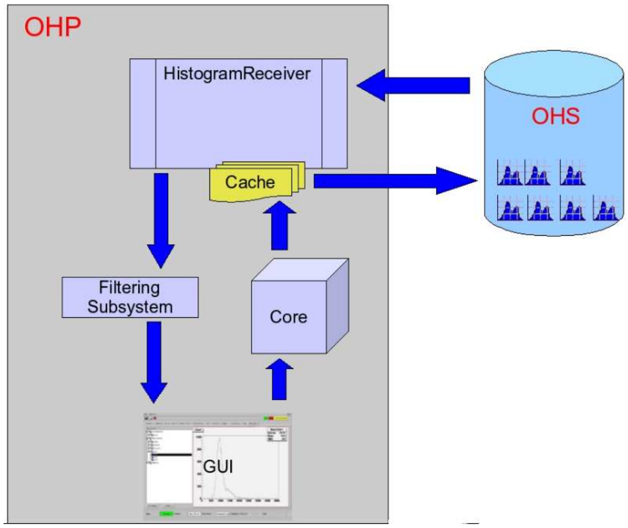
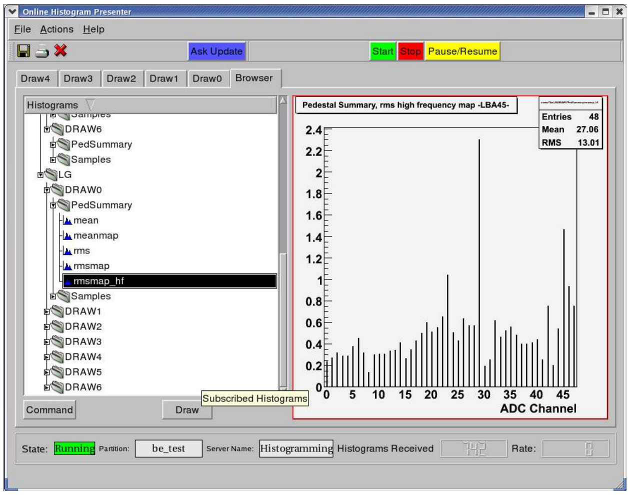


Figure 1: Schematics of the Online Histogram Presenter with all subsystems (Core, Cache, Histogram Recevier, Filters and GUI). The arrows indicate interactions between subsytems or with OHS.

the Graphical User Interface (GUI) which is based on the Qt® (Trolltech Inc.) framework. The underlying technology for histogramming is ROOT [6], which provides the user with all the functionalities to interact with the histograms, such as zooming, fitting, changing the graphic aspect etc. Figure 2 shows a screenshot of the GUI as it appears when used to browse the OHS content.



State: Running Partition: be_test Server Name: Histogramming Histograms Received 742 Rate: 0

Figure 2: OHP Graphical User Interface. The screenshot shows how the GUI appears when used to browse the OHS content, on the left side there is the list of histograms organized in a tree structure while on the right side the selected histogram is displayed.

As previously mentioned, the OHS provides a mechanism of notification by subscriptions. Histograms are identified by their names, which are strings of the form ProviderName/HistogramName (e.g. ProviderNumber1/TotalEnergy_histo). Subscriptions may be done either with the histogram name or with regular expressions (e.g. ProviderNumber2/.*). Whenever a histogram is published or updated, the OHS notifies the event to the OHP, which will decide whether to retrieve the plot and draw it on the screen.

To better understand how the different subsystems operate together, let us examine the OHP workflow in more detail.

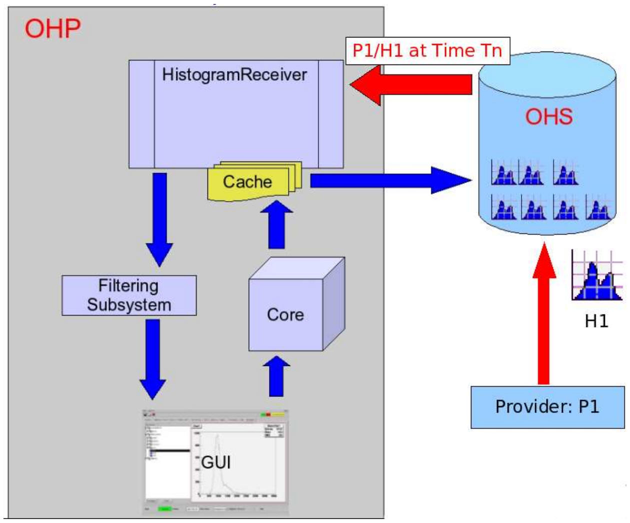


Figure 3: A notification is sent to the Histogram receiver.

Suppose a provider P1 publishes (or updates) a histogram H1 in the Online Histogramming Service server. If the OHP is subscribed to that histogram, the OHS sends it a notification containing the time-stamp of notification T_n (fig. 3). This notification is sent to the Filtering Subsystem and the cache is updated with the histogram name P1/H1 and with the time-stamp T_n. If the histogram P1/H1 must be drawn in the active window, the notification is also propagated to the GUI; otherwise it is discarded. Note that no histogram is transferred over the network during this operation (fig. 4).

Let us now suppose the histogram P1/H1 must be drawn in the active window. After receiving the notification, the GUI asks the Core for the histogram P1/H1. The histogram is searched for in the correct cache. If it is not found, the histogram P1/H1 is withdrawn from OHS, stored in the cache together with the time-stamp T_r (where r stands for retrieval) and sent back to the GUI that displays it on the screen (fig. 5). On the contrary, if the histogram P1/H1 is found in the cache, the notification time-stamp T_n is compared with the retrieval time-stamp T_r . If $T_r < T_n$ the cached histogram is replaced with the more recent one available in the OHS. Otherwise the cached histogram is displayed immediately.

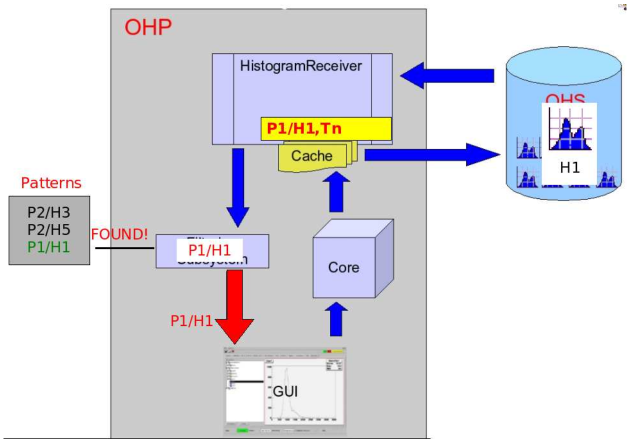


Figure 4: The Filtering Subsystem compares the notification with the names of histograms to be drawn in the active window.

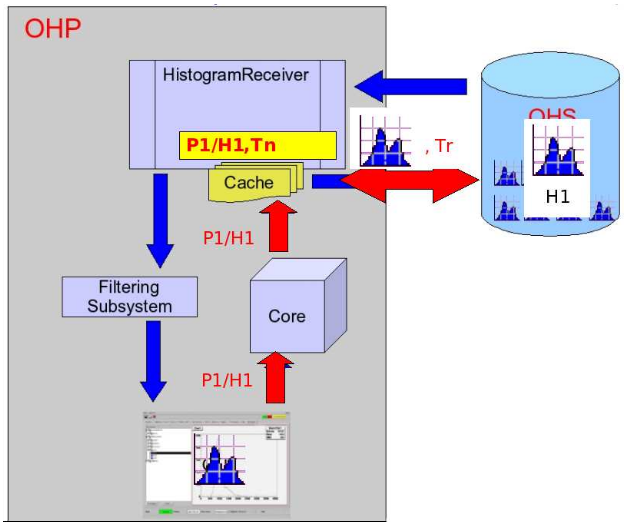


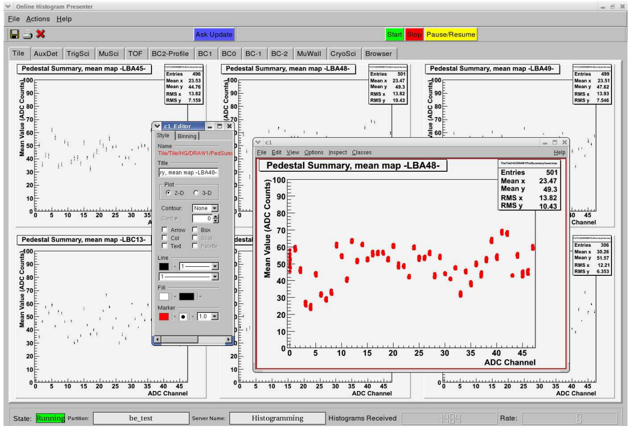
Figure 5: When a histogram is not found in the cache, it is withdrawn from the OHS.

When the user changes the active window, the histograms to be drawn are searched in the caches and, only if needed, retreived from OHS.

The use of the notification mechanism together with a caching system effectively contributes to reducing the network traffic, since histograms are transferred only when actually needed.

The histograms in the active windows are continuosly updated with the same frequency they are received. In order to be able to interact with a particular histogram, for example to perform a fit, when a histogram is double-clicked the histogram is copied and displayed in a pop-up window. This allows both to work on the selected histogram and to keep the active window continuously updating. A screenshot of the OHP in this situation is shown in figure 6.

The complete set of received notifications can be browsed to allow the users for a more detailed inspection. The browser index, organized in a tree structure, is updated as the notifications are received. The histogram names are used to define a folder structure using the '/' character as a



State: Running Partition: be_test Server Name: Histogramming Histograms Received Rate:

Figure 6: Screenshot of the OHP as it appears in the shifter mode, a preconfigured set of histogram is displayed in the window when one of the grey tabs is clicked . The pop-up window allows to perform operation on the selected histogram while keeping active window continuosly updating as described in the text.

delimiter. The histograms can be displayed on request following the same retreival mechanism described above (see fig. 2).
A reference histogram is associated to each retreived histogram. The default association logic is to search for the reference histogram with the same name as the retreived one in a local ROOT file or a dedicated provider. It is possible to change the association logic by specifying the reference histogram name in the configuration file.

CONFIGURATION

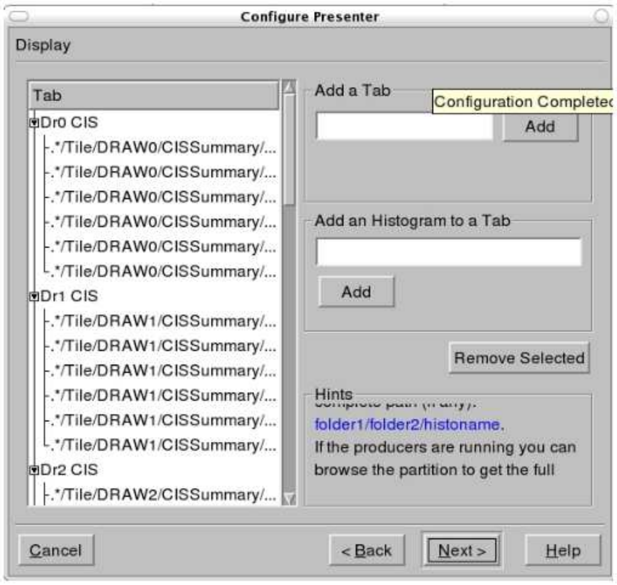
The Online Histogram Presenter can be fully configured either through an ASCII file or via a graphical panel (see fig. 7).
The following properties can be configured:

- the set of histograms to be received;
- the names of the tabs and their content;
- the graphical aspect of each histogram;
- the reference histogram sources (local ROOT files or a histogram provider).

SENDING COMMANDS

Histogram Producers can receive commands from OHP. The command modifies the histogram production mechanism. At the moment four commands are supported:

Reset which forces a clean up of the histogram content.
Rebin which changes the number and/or width of the bins.
Update which forces the publishing of a histogram in the OHS.
Custom This is a producer dependent command. The user must fill in the command content.


Configure Presenter
Display
Tab
Dr0 CIS
-.*/Tile/DRAW0/CISSummary/...
-.*/Tile/DRAW0/CISSummary/...
-.*/Tile/DRAW0/CISSummary/...
-.*/Tile/DRAW0/CISSummary/...
-.*/Tile/DRAW0/CISSummary/...
-.*/Tile/DRAW0/CISSummary/...
Dr1 CIS
-.*/Tile/DRAW1/CISSummary/...
-.*/Tile/DRAW1/CISSummary/...
-.*/Tile/DRAW1/CISSummary/...
-.*/Tile/DRAW1/CISSummary/...
-.*/Tile/DRAW1/CISSummary/...
-.*/Tile/DRAW1/CISSummary/...
Dr2 CIS
-.*/Tile/DRAW2/CISSummary/...
Add a Tab
Configuration Complete
Add
Add an Histogram to a Tab
Add
Remove Selected
Hints
Cancel
< Back
Next >
Help

complete path (if any):
folder1/folder2/histoname.
If the producers are running you can browse the partition to get the full

Figure 7: The GUI panel for the configuration, in figure the "Display" panel is shown. Through this panel tab names and content can be specified.

The commands Reset and Rebin may be addressed only to one specific histogram or to all the histograms managed by a producer.

TEST RESULTS AND CURRENT APPLICATIONS

A prototype version of OHP with minimal functionalities was used during the test beam of a slice of the ATLAS detector in 2004 and is currently being used for the commissioning of various ATLAS subsystems. This prototype version has driven the design of the new OHP version to cope with the need for a network minimization system (cache and filters).
The result of this has been described in this paper.
Each OHP component has been successfully tested in standalone mode. We are now performing a detailed test of the entire application inside a more realistic environment.

A new development phase to optimise the code and add new functionalities will be started soon after.

REFERENCES

[1] ATLAS collaboration, ATLAS Technical Proposal, CERN/LHCC/94-43, LHCC/P2, CERN, Geneva, Switzerland, 1994

[2] P. F. Zema et al., The GNAM monitoring system and the OHP histogram presenter for ATLAS, 14th IEEE-NPSS RTC 2005, Stockholm, Sweden, 2005

[3] M. Bosman et al., Development and Tests of the Event Filter for the ATLAS experiment, 14th IEEE-NPSS RTC 2005, Stockholm, Sweden, 2005

[4] ATLAS collaboration, ATLAS High Level Trigger, Data Acquistion and Controls, CERN/LHCC/2003-022, CERN, Geneva, Switzerland, 2003

[5] S. Kolos et al., Online Monitoring software framework in the ATLAS experiment, CHEP 2003, La Jolla, Ca, USA, 2003

[6] R. Brun, F. Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.ROOT.

[7] I. Antcheva ed., ROOT User Guide, CERN, Geneva, Switzerland, 2005