1. The ATLAS Online Monitoring System

The ATLAS experiment

ATLAS is one of the five major experiments on the Large Hadron Collider accelerator at CERN.

The ATLAS Trigger(DAQ) System

The ATLAS Detector Interface(DI) system transports event data from the 1000 detector read-out boards to the main energy.

The ATLAS Online Monitoring System

The Online Monitoring system is responsible for managing the ATLAS performances and operational conditions of the ATLAS data taking including physics event analysis and data quality assessment. Online monitoring system is operating in tightly distributed multi partition computing environment.

The Web Monitoring Interface

The efficiency of the system can be improved so allowing system experts on site to follow the ATLAS operations in real-time, spotting potential problems which otherwise may not be assessed for a non-trivial time. Taking into account the wide geographical spread of the ATLAS collaboration, the solution of this problem is to have an online monitoring system with minimal access latency available worldwide.

For histograms there is the dedicated plugin, with a number of any HEP specific software on the client computer, used where IS resources URLs have been put to the monitoring interface of the ATLAS private network from a Web-Browser via HTTP.

2. Web Monitoring Framework Architecture

The Flow of Data

The Web Monitoring interface is implemented by a number of homogeneous servers written in Python using REST technology. The servers are stateless, which facilitates information caching and load-balancing. The latency to access an item is determined by the HTTP access, typically in the order of 100 ms if one is outside of CERN.

The API

Web Monitoring interface provides generic RESTful API to the Monitoring Services. Each monitoring resource has a unique URL associated with it, e.g. the following is URL of the object in the Information Service.

https://atlasop.cern.ch INFORMATION/ATLAS/Services/RunParams

A GET request with each URL returns the value of the respective IS object in XML format. This design allows users to program their preferred forms. It is used to construct complex WEB based GUIs using CSS, Java,XML or standalone Java applications.

The Framework

The Web Monitoring server is implemented as a framework which can load and maintain an arbitrary number of specific data handling modules. When the monitoring receives a request it parses it, finds an appropriate data module and passes parameters of the request to this module. The module is supposed to return some relevant data either in JSON or XML format, which framework sends back to the client of the original request.

3 Data Plugins for Run 1

Information Service module

For displaying an arbitrary piece of information from IS the HTML page has been used where IS resources URLs have been put to the IS as URLs and tags have been introduced to escape XML content. When such HTML page is loaded in a browser, the dedicated IS script parses through the HTML elements and sends HTTP GET request to the server using element ID as URLs and extract the necessary information from the XML data which is returned.

Log Files module

The module, which supports this interface, can return a list of log files for a particular machine. The module returns the results of this request for the particular machine.

Online Histograms module

For histograms there is the dedicated plugin, which performs rendering of histograms using the ROOT libraries and return those in a number of standard formats (PNG, JPG, GIF, SVG, etc.). Therefore there is no need for any specific software on the client computer.

Monitoring Archive module

This HTML page looks similar to the one which shows online histograms but instead of talking from the online monitoring system, the corresponding program will instead read them from the Monitoring Archive. One can use a run number and a histogram name for getting back specific histograms from any given run.

4. New Data Plugins for Run 2

Event Monitoring module

A new module can return either a full ATLAS event or an individual fragment from any given read-out model in XML format. Any modern Web Browser can display each XML data in a reasonable way without any development on the client side. Alternatively one can use XML data in a JavaScript to customize this view. This functionality has been proved to be very useful for the commissioning and debugging period which precedes the Run 2 operation period, when detector experts debug read-out electronics and new versions of the firmware.

Data Quality Archive module

For Run all OI data, produced online, is stored in the IS. The interface for retrieving data from the DQM archive is implemented as another module for the Web Monitoring interface. Generic DQM archive browser application, which can retrieve a Web Browser, has been implemented using ExtJS framework.

5. Extended Web Interface to IS

The Information Service provides asynchronous way of getting information, the application may subscribe for the information changes to be notified when they happened. The Extended Web Interface uses this functionality while communicating with external world through Server-Sent Event(SSE) technology introduced by HTML5.

Client API is provided in JavaScript

Function installPlugin(data, name) {
  document.form.func.value += name + "=" + data + ";"
}

Function ErrorValueClick(msg) {
  document.form.errors.value += msg + "=" + data + ";"
}

Function set_subscription() {
  web_is_is.subscribe(all+ation, server, reg, map, installPlugin, ErrorValueClick);
}

Function common_subscription() {
  web_is_is.subscribe(all+ation, server, reg, map, installPlugin, ErrorValueClick);
}

Function openSSE Session()

IS Extended Web Interface (Python)

Web Server (Apache)