EPJ Web of Conferences **214**, 01049 (2019) https://doi.org/10.1051/epjconf/201921401049
CHEP 2018

A Web-based control and monitoring system for DAQ applications

Alexey Anisenkov[1,2] [1] , *Daniil* Zhadan[1], and *Ivan* Logashenko[1,2]

[1] Budker Institute of Nuclear Physics SB RAS, 630090 Novosibirsk, Russia
[2] Novosibirsk State University, 630090 Novosibirsk, Russia

Abstract. A comprehensive and efficient environment and data monitoring system is a vital part of any HEP experiment. In this paper we describe the software web-based framework which is currently used by the CMD-3 Collaboration at the VEPP-2000 Collider and partially by the Muon g-2 experiment at Fermilab to monitor the status of data acquisition and the quality of data taken by the experiments. The system is designed to meet typical requirements and cover various use-cases of DAQ applications, starting from the central configuration, slow control data monitoring, data quality monitoring, user-oriented visualization, control of the hardware and DAQ processes, etc. Being an intermediate middleware between the front-end electronics and the DAQ applications the system is focused to provide a high-level coherent view for shifters and experts for robust operations. In particular, it is used to integrate various experiment dependent monitoring modules and tools into a unified Web oriented portal with appropriate access control policy. The paper describes the design and overall architecture of the system, recent developments and the most important aspects of the framework implementation.

1 Introduction

The data quality monitoring system and the slow control system are the essential parts of data acquisition of any high energy physics experiments. A number of important functions such as monitoring of the status of DAQ, environment conditions, quality of data taken, properly operating the hardware equipments are carried out by monitoring tools. Using such tools, people on shift can timely detect malfunctions of the subsystems, thus preventing accidents. Historical analysis of the slow control data allows experts to evaluate the performance of the subsystems. Ultimately, safety and correct functioning of whole experiment depend on monitoring tools.

The system described in this work was initially developed for the CMD-3 detector [1], installed at the VEPP-2000 e+e- collider at the Budker Institute of Nuclear Physics (BINP, Novosibirsk, Russia). It is a typical small-to-medium scale HEP experiment. The CMD-3

[1] Corresponding author: Alexey.Anisenkov@cern.ch

detector consists of several subsystems such as: cryogenics, superconducting magnet, drift chamber, three calorimeters, muon-range system and others. Overall there are about 1000 slow control channels to be displayed and analyzed by the monitoring system through about a hundred histograms, tables and the data quality plots.

2 Architecture overview

The system relies on a client-server architecture which is crucial for hiding direct dependencies with front-end electronics and data sources. The web-based approach makes the system flexible and independent from the client OS, and allows to access monitoring services anywhere remotely - someone only needs a web-browser and internet connection to use it.

The system meets the goals stated earlier. There is access to the real-time and archive monitoring data, it allows to control detector subsystems, it provides a unified and user-friendly access to the diverse pool of monitoring and control data required for DAQ operations. One of the important requirements addressed by the system is the ability to facilitate operations as well as the extension of the system itself by physicists with minimum knowledge in programming. The web-based approach meets well with all of them.

The monitoring system developed follows a modular approach, which helps to divide the implementation into shared components and the modules, specific for given experiment. With proper configuration a generic functional module can be tuned and adapted for certain experiment. Thanks to this feature, apart from the CMD-3 Experiment, some parts of the implemented monitoring framework are used at two independent experiments: Muon g-2 (Fermilab) [2] and MRT (BINP).

The software of the CMD-3 DAQ [3] and slow control systems is based on the program package MIDAS [4]. It is a rich data acquisition software developed at PSI and TRIUMF. The MIDAS toolkit contains a set of useful services and program, in particular it includes the native web-interface and the hierarchical Online database (ODB). MIDAS uses a shared-memory buffer for event collection and distribution and supports ROOT for producing data quality histograms. Native MIDAS API is implemented with C/C++; we've developed a python library to access ODB and Buffer modules named *pymidas*. The python library allowed for easy integration of web applications with the DAQ services. Figure 1 illustrates the overall architecture of the CMD-3 monitoring framework, lists examples of various data sources including MIDAS, involved DAQ services and web applications implemented.

It is worth mentioning the details of implementation of client-server architecture. The server part is based on the high-level Python web framework Django [5]. The specific templates of Django are used to customize functional modules for the needs and conditions of a certain experiment. Python is a modern programming language with large community, this fact makes development more friendly and rapid. The client part was implemented using JavaScript and sets of tools such as Bootstrap [6] and jQuery [7]. The last two provide a number of plugins, which help to make the web interfaces responsive, interactive, mobile- and user-friendly.

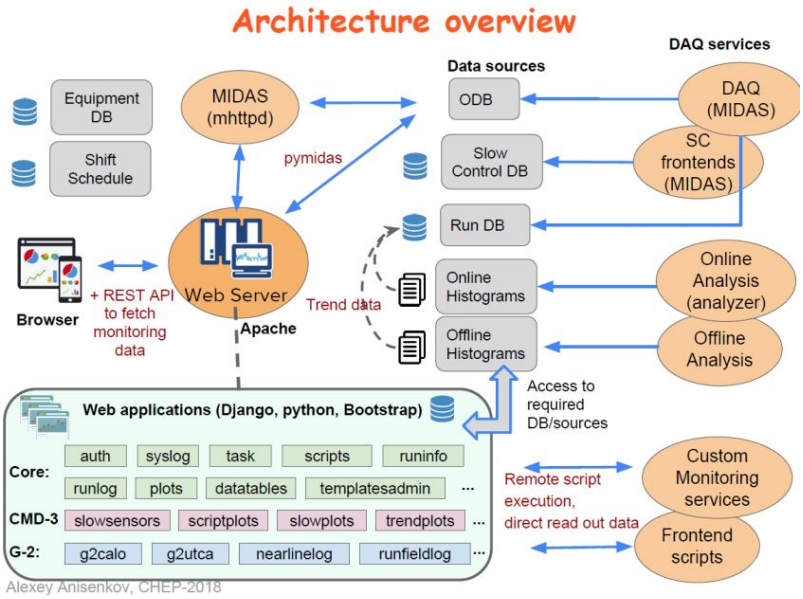

Figure 1. Architecture overview.

3 Components

As it is described in the previous chapter the system implements modular approach. Modularity implies constructing shared blocks that can be reused by different components which allows to optimize the development process as well as isolate logic into functional dedicated Django applications. Each project (experiment specific monitoring instance) configures the list of applications to be used out of the box, extends the logic according to required use-cases and overwrites default look and feel of user interfaces, if needed, using template settings. Below we describe examples of basic monitoring components of the system.

3.1 Module «slowplots»

The slowplots module is used to monitor critical and important parameters of the experiment, or, in general, any slow control data. Its view is realized in the form of graphs, organized as a set of pads with one or more graphs, each responsible for certain sensor.
Shared x-axis for all the pads within same canvas provides convenient way for end-users to apply interactive zoom or change axis range to required one. Above the main canvas there is a panel that allows to choose predefined presets of the sensors, to select the data source and to display the status of the data. At the bottom of the view there is a navigation panel for the time range of the loaded data. The implementation of our own shared graphical widget is based on HTML5 vector graphics using D3.js library [8]. Figure 2 shows how this module is displayed on the web site page.

Data visualization is fully interactive and dynamic. A user can receive the pop-up window with detailed information of the point on graph, subject to certain X scale. All graphs share the common x-axis with single slider responsible for the displayed X range,

while the y-axis is common only inside one pad; there is a possibility to add a second y-axis to the pad, as well as draw data in logarithmic scale, useful when one needs to display two graphs having different Y scale on one pad.

Figure. 2. Main view of module «slowplots» on the website page.

The data are loaded from the server with JSON API. The source of data depends on the chosen mode of operation of the module. There are two modes: *online* and *archive*. The first mode is more suitable for people on shift, because it is constantly updated and shows actual data. In this case the online database is used as data source. At CMD-3 the online database stores only last few (e.g. a hundred) values so the access to the older history requires the change of the module mode. In the archive mode the module allows the user to choose a time range through the datepicker for loading data. In this case the data source is the archive (slow control) database, which stores the full history of values of all sensors. If being loaded for a long time range, the data are filtered server-side for faster access. There is a possibility to improve the accuracy of the data displayed in range chosen by the slider.

The client-side html widget of this module has been implemented as standalone jQuery plugin, which makes its use more convenient and allows easy integration into a custom web page at the site by end-users. The module is independent from the experiment details and uses configuration data, which indicates the sources of the data to be displayed. Thanks to this fact the module is also used at other experiments.

3.2 Module «trendplots»

The appearance of the view of the module «trendplots» is similar to the previous one. In more details, these two modules use the same graphic engine plugin implementing pads with graphs. The main view of module «trendplots» is shown in Figure 3. There are some minor differences, for example, for X-axis the run numbers are used rather than timestamp values. The main difference comes in its purpose, the trendplots are aiming to monitor the data quality rather than environmental conditions.

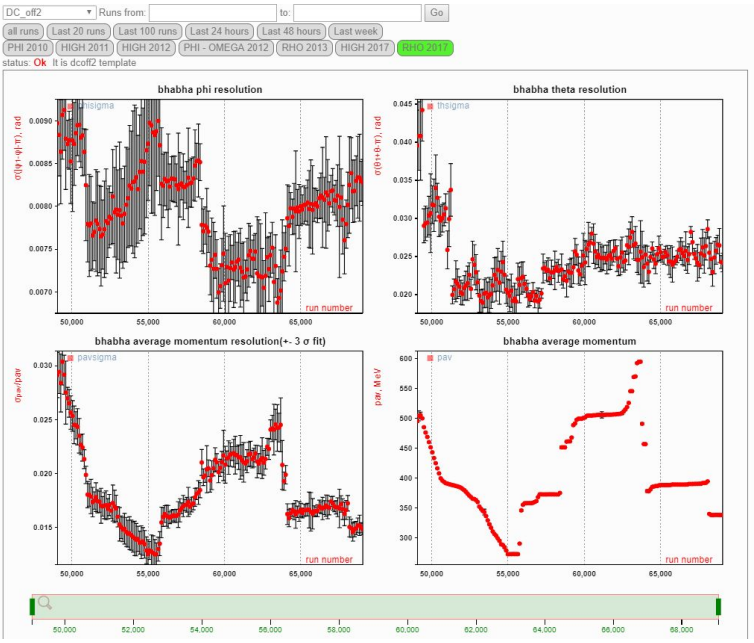

Figure. 3. Main view of module «trendplots» on the website page

The module allows data having different acquisition logic to be shown on same pad. To have this kind of flexibility, the central logic element determines with help of configuration file what kind of data source should be used for each sensor, e.g. which script should be started.

At CMD-3 the data source for this module is the quality control database based on MySQL DBMS. The unit of information is the run. The run parameters and characteristics stored in the database are divided into two types: «online» and «offline». The first are calculated online during data taking, for example, the number of bad channels of the detector electronics. While the values for the the second type of attributes will be recorded in the database only after the full scale data production.

3.3 Module «runlog»

The role of the module «runlog» is to provide a list of collected runs with primary information exposed. The view is configurable. At CMD-3 the view was implemented as the table with the following columns: Run – number of a run; Q – quality (type) of the run; Start time; End time; Events – number of collected events during the run; Size; Shift; Run comment; Additionally values of magnetic field, energy, online and offline luminosity. The given module provides an extensive search in the run database and quickly presents the run information needed. All stated above was implemented using the DataTables plugin of JavaScript library. An example of the main view for the runlog module is shown in Figure 4.

The module automatically highlights “suspicious” runs when online data processing logic detects a deviation for critical parameters from expected values. In this case the operator should apply special attention to the run data and investigate the source of problem. The module allows one to open additional windows showing detailed information about the run, including the values of parameters, data quality histograms etc. The operator

can update run details using provided user forms, for example to change the quality value of the run or complement the run comment with additional observed information.

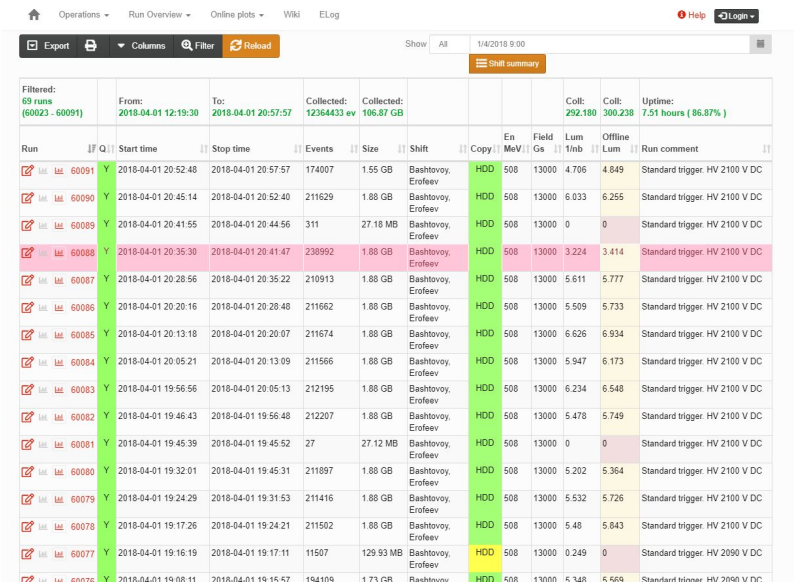

Figure. 4. Main view of module «runlog» on the website page

4 Conclusion

During operation, the DAQ system and related systems produce a lot of information, for experts and shifters, that need to be monitored and taken into account. Modern Web 2.0 technologies and open source tools can be effectively used to build functional, handy and attractive applications for slow control and monitoring systems.

The CMD-3 web-based monitoring system has been developed with the help of various tools and frameworks such as Django, Bootstrap, SVG, jQuery plugins and others. It provides a unified and user-friendly access to a whole set of monitoring and control data as well as the functionality to configure hardware equipment. The use of a modular approach is quite important, making the system more flexible and extensible. Some parts of the system framework implemented are actually used at other experiments.

The work is supported in part by the Russian Foundation for Basic Research, grant RFBR 16-02-00873-a.

References

1. B. Khazin, *Physics and detectors for VEPP-2000,* Nucl. Phys. Proc. Suppl. **181-182**, 376 (2008)
2. W. Gohn, *Data Acquisition for the New Muon g-2 Experiment at Fermilab,* J.Phys.Conf.Ser. **664,** 082014 (2015)
3. A.N. Kozyrev et al., *The CMD-3 TOMA DAQ infrastructure*, JINST, **9**, C10016 (2014)

4. S. Ritt, P. Armaudruz and K. Olchanski, MIDAS (Maximum Integration Data Acquisition System), http://midas.triumf.ca (accessed 2018-12-01)
5. Django project, https://www.djangoproject.com/ (accessed 2018-12-01)
6. Bootstrap toolkit, https://getbootstrap.com/ (accessed 2018-12-01)
7. JQuery library, https://jquery.com/ (accessed 2018-12-01)
8. Bostock M, Ogievetsky V, Heer J: D^3 Data-Driven Documents. IEEE T Vis Comput Gr. 2011, 17: 2301-2309.