Developing EASE
A remote EPICS monitoring tool

Nolan W. Brown[1,2,+], Alex Wallace[2+], Murali Shankar[2]

[1]College of Arts & Sciences, L.C Smith College of Engineering and Computer Science, Syracuse University, Syracuse, NY 13244, USA

[2]Linac Coherent Light Source, SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA.

[+]Contact: nobrown@syr.edu

Introduction

EASE, the Easiest Alarm System Ever, is a web service that enables engineers, users and scientists to remotely monitor EPICS PVs in emergency states. This system frees individuals from manually monitoring PVs by contacting users when defined conditions are reached. Despite the extensive configuration required to monitor large arrays of PVs, EASE presents users with an easy-to-use GUI built with Django, a Python web framework. The development of EASE has been a multi-year endeavor that offers a long overdue solution for a persistent issue.

Development Process

The creation of EASE began last summer by the intern Jacob Rudolph under the direction of Alex Wallace. During this time, several of EASE's fundamental components were prototyped. This summer, many features were finished and further functionality has been added.

The first tools to be finished were the Epics Archive data retriever and the emailing procedures. Once these utilities were functional, tools were given to the users to improve their control over the data their alarms would send.

Three significant features still needed completion. When a PV varied rapidly near the border for a specified alarm, EASE would send messages logging all trips. This could net hundreds of emails per minute ensuring that any important notifications were quickly lost. To block excessive emails without losing important information, an alarm filter has been implemented. While building the filter, a number of database transactions were noted to contribute little to the system. This presented no issue at the given operational size but it opened the possibility for scaling issues during release. To prevent such issues from occurring EASE's interactions with the database were rewritten to use fewer queries without sacrificing important information. The last significant feature of EASE's pre-release development is the now-implemented multi-user support. Individual users maintain their own protected alarms and configurations but are also able to share their tools for others to use.

The most recent efforts on EASE have been devoted to preparing this web service for internal release.

EASE v.1.0

The EASE web-service is composed of two primary sections. One half is built upon the Django web-framework and on the other utilizes SLAC-built packages. (Fig.1) The Django-based side manages users' accounts and provides a web-based GUI. Each user with an account can subscribe to and modify alarms created by themselves or others. Alarms dedicated to specific systems can be bundled for easier management. (Fig.2) Individuals who manage projects can be given superuser status for more management tools.

All of the users' configurations are stored in a MySQL database. This dictates which PVs the monitor watches and how to respond to incidents. The email generator then pulls data from the Epics archiver to generate plots. (Fig.3)

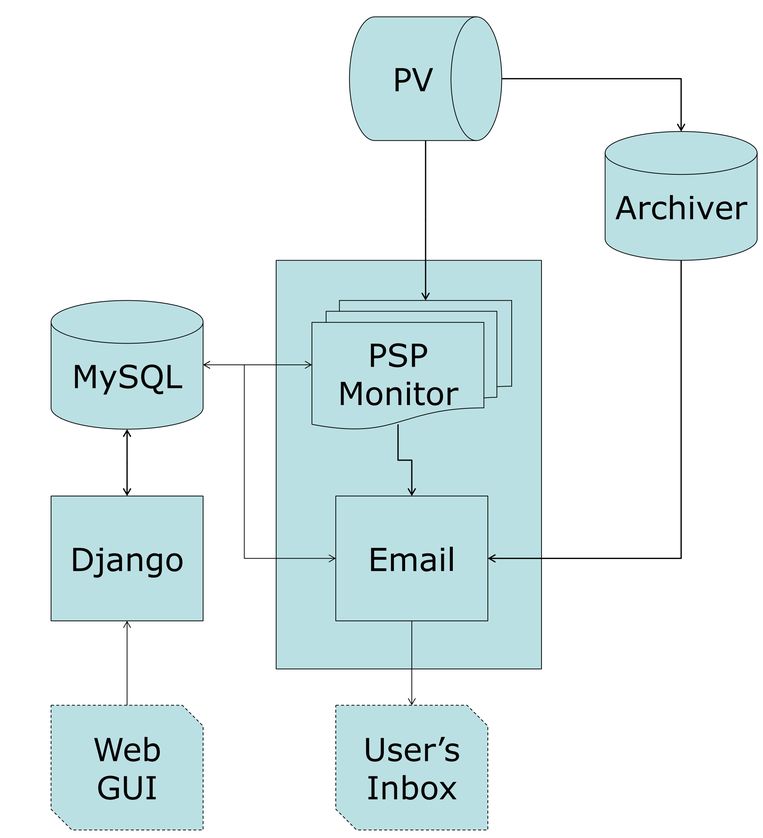

Fig 1. EASE's underlying structure.

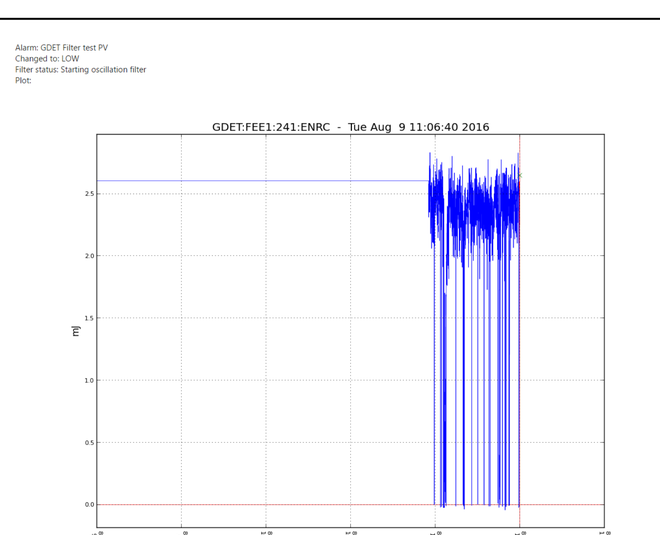

Fig 2. A screenshot of the EASE administration pages. Listed here are individual alarms, each monitoring a PV. This screenshot was taken during early testing of EASE and represents disproportionately few alarms. The other administration pages follow a similar form.

Fig 3. An alarm notification received via email. The plot logs activity leading up to the triggering incident. Included in the email are fields documenting the filter state and notes on managing the alarm.

Further Development

While EASE is ready for its first release, there are several features and improvements the system would benefit from.

EASE's PSP based monitor was originally designed for small scale use. While this model scales acceptably, modifying this component's thread handling should substantially lower underlying processor demands.

Better web performance could be achieved by more clearly establishing the boundaries between the PSP-based monitors and Django. This would allow multiple Django instances to be daemonized in parallel improving the web-service's responsiveness and ability to accommodate large numbers of users.

Further Dev. cont.

EASE would also benefit from a number of new features. Tools to make multi-PV alarms could apply the same limits to a large number of PVs quickly. Logical alarms could trip under more carefully defined conditions and perform custom actions. The possibilities for further development are nearly endless.

Acknowledgments

Use of the Linac Coherent Light Source (LCLS), SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515.

Date: 08/09/2016