Django Remote Submission

Tanner C. Hobson[1], Mathieu Doucet[1], and Ricardo M. Ferraz Leal[1]

[1]Oak Ridge National Laboratory

24 July 2017

Paper DOI: http://dx.doi.org/10.21105/joss.00366
Software Repository: https://github.com/ornl-ndav/django-remote-submission
Software Archive: http://dx.doi.org/10.5281/zenodo.848749

Summary

The Django Remote Submission (DRS) is a Django (Django, n.d.) application to manage long running job submission, including starting the job, saving logs, and storing results. It is an independent project available as a standalone pypi package (PyPi, n.d.). It can be easily integrated in any Django project. The source code is freely available as a GitHub repository (django-remote-submission, n.d.).

To run the jobs in background, DRS takes advantage of Celery (Celery, n.d.), a powerful asynchronous job queue used for running tasks in the background, and the Redis Server (Redis, n.d.), an in-memory data structure store. Celery uses brokers to pass messages between a Django Project and the Celery workers. Redis is the message broker of DRS.

In addition DRS provides real time monitoring of the progress of Jobs and associated logs. Through the Django Channels project (Channels, n.d.), and the usage of Web Sockets, it is possible to asynchronously display the Job Status and the live Job output (standard output and standard error) on a web page.

Practical case

The Liquids Reflectometer (LR) at the Oak Ridge National Laboratory's Spallation Neutron Source (SNS) (Mason et al. 2006) provides neutron reflectivity capability for an average of about 30 experiments each year. In recent years, there has been a large effort to streamline the data processing and analysis for the instrument. While much of the data reduction can be automated, data analysis remains something that needs to be done by scientists. With this in mind, the Reflectivity Fitting Web Interface has been developed (Doucet, Leal, and Hobson 2017). It provides a smooth data analysis interface, capturing the process of setting up and executing fits while reducing the need for installing software or writing Python scripts.

Currently the fitting routines are written for the software package REFL1D (Kienzle, Krycka, and Patel, n.d.). The management of the fitting routines is performed with the DRS. The DRS transparently submits the jobs to a cluster and provides real-time monitoring of the remote jobs and their associated logs. The user can thus track the status of the jobs and eventually inspect the associated logs. If the job was successful, DRS gathers the output data and transfer them to the webserver. Finally, the web interface provides a user friendly display and visualization of those data.

Acknowledgments

This research used resources at the High Flux Isotope Reactor and Spallation Neutron Source, a DOE Office of Science User Facility operated by the Oak Ridge National Laboratory.

References

Celery. n.d. "Celery: Distributed Task Queue." http://www.celeryproject.org.

Channels. n.d. "Django Channels." https://github.com/django/channels.

Django. n.d. "Django Web Application Framework." https://www.djangoproject.com/.

django-remote-submission. n.d. "Django Remote Submission." https://github.com/ornl-ndav/django-remote-submission.

Doucet, M., R. M. Ferraz Leal, and T. C. Hobson. 2017. "ORNL Reflectivity Fitting Interface," January. doi:10.5281/zenodo.260178.

Kienzle, PA, JA Krycka, and N Patel. n.d. "Refl1D: Interactive Depth Profile Modeler." https://github.com/reflectometry/refl1d.

Mason, T.E., D. Abernathy, I. Anderson, J. Ankner, T. Egami, G. Ehlers, A. Ekkebus, et al. 2006. "The Spallation Neutron Source in Oak Ridge: A Powerful Tool for Materials Research." *Physica B: Condensed Matter* 385: 955–60. doi:http://dx.doi.org/10.1016/j.physb.2006.05.281.

PyPi. n.d. "PyPI - the Python Package Index." https://pypi.python.org/pypi/django-remote-submission.

Redis. n.d. "Redis." https://redis.io.