Client-side plug-ins for Tukey
OSDC PIRE Research Challenge

Eric Griffis   Joshua Eisenberg

June 19, 2014

Abstract

OSDC exposes resource to data scientist through a console. This arrangement divides administrative tasks into browser-based Tukey portal interactions and secure console interactions. Experts and power users may be comfortable in the console, but many data scientists are neither, which causes friction between data scientists and the OSDC platform. We propose a small extension to the Tukey portal service that enables support client-side Tukey plug-ins authored by the user or. By reducing this friction, we expect to broaden the OSDC user base by reducing the effort required for mundane tasks and providing automating solutions to advanced problems. If we enable programmable access to Tukey services, we can free data scientists from systems-level programming tasks so they can get more science done.

1 The current state of Tukey

Figure 1 outlines the Tukey system design. It contains three components: portal, proxy, and server. The portal is a user-facing, Django-powered, MVC-style Web application. Responsibilities of the portal include handling authentication requests, rendering HTML templates, and making server requests through the proxy. The proxy is a Python WSGI reference implementation Web server that exposes the complete OpenStack Nova REST API to authenticated users. The portal contains a Python library that wraps calls to a select few Nova API functions. Exposing part of the Nova API to end users benefits both sides of the connection. The proposed changes enable client-authored plugins and optional features for the Tukey Web portal.

2 A client-side plug-in system

Figure 2 illustrates the Tukey design as prescribed by the proposed solution. Our proposal calls for the following modifications to the existing Tukey design:

1. Port the existing Nova API wrapper from Python to JavaScript.
2. Extend portal templates to support manipulation by client-side JavaScript.
3. Add plug-in management pages to portal.
4. Add optional functionality configuration toggles to portal.

Note that plug-ins can be managed by the user like any other resource, such as images or snapshots, through the Web portal.

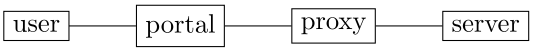

Figure 1: Current Tukey system

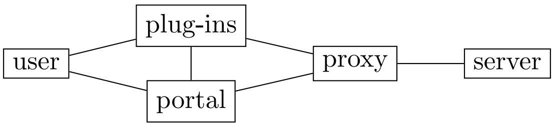

Figure 2: Extended Tukey system

2.1 Example plug-ins

The impetus for extending Tukey with plug-ins is a tool to aid reproducible research. Existing work calls for static linking of automatically-detected libraries and supporting tools [1]. The existing OSDC snapshot functionality provides a far simpler alternative. A client-side Tukey plug-in could provide a consistent framework for recording snapshots appropriate for reproducibility and sharing these snapshots with other scientists.

We would also like to provide an optional feature for simple browser-based file transfers directly to running instances. This feature could set up and tear down the necessary SSH ports as needed.

3 Evaluation

Our goal in extending Tukey is to change the way people interact with the cloud. We want to lay down a framework with more ways to compute on the OSDC than just the command line interface. Server-side plug-ins, written by admins, can be selected and removed at will by the users. Client authored plug-ins can be managed and shared by the community of researchers on the cloud. These additions will allow data scientists to customize the way they analyze data on the cloud with convenience and efficiency. The proposed design requires modification to the Tukey portal templates. If our enhancements do not function correctly, the users will notice quickly. We also propose exposing the Nova API to the client. Though the proposed system could be made no less secure than the current design, we need to be mindful of potential vulnerabilities.

4 Conclusion

To get the "datascope" of the OSDC into the hands of as many scientists as possible, we believe the OSDC interface should be made more extensible. Making Tukey extensible will enable researchers to interact with the cloud more productively. Not every science researcher is a confident programmer, and plug-ins provide a means for scientists without programming expertise to analyze data sets with less use of the terminal.

References

[1] Robinson, Casey, and Douglas Thain. Automated Packaging of Bioinformatics Workflows for Portability and Durability Using Makeflow, 98105. ACM Press, 2013. doi:10.1145/2534248.2534258.