Genomics Across Clouds with Galaxy

Enis Afgan[1,2], John Chilton[3], Dannon Baker[1], Nate Coraor[3], The Galaxy Team, James Taylor[1]

[1]Department of Biology
Johns Hopkins University
Baltimore, MD, USA
{enis.afgan, dannon, jxtx}@jhu.edu

[2]Centre for Informatics and Computing
Rudjer Boskovic Institute (RBI)
Zagreb, Croatia

[3]Department of Biochemistry and Molecular Biology
Penn State University
University Park, PA, USA
{jxc755, ndg1}@psu.edu

ABSTRACT
As Cloud resources continue to dominate and provide apparent benefits in terms of flexibility, accessibility and scalability, individual applications often lack the ability to integrate those benefits for their users. In this paper, we describe seamless integration of Cloud resources into a pool of dedicated cluster resources for the Galaxy application. This integration works toward increasing adoption of cloud resources and expanding the capacity of the core application.

Introduction
With genomic analyses growing increasingly complex, a thriving community of domain-specific data analysis tools has emerged. To simplify the user interactions with those tools, tool management and workflow frameworks have materialized. One such framework is Galaxy [1] - a web-based platform for biomedical research. Galaxy transforms command-line tools into web forms and allows those to be easily chained into composite workflows. Integrated with a range of data providers, it makes it easy for domain researchers to import data from multiple sources, utilize appropriate tools from an expansive toolset, visualize desired datasets, and ultimately share the results with collaborators or the community. Such ease of interaction with the tools and (increasingly big) data results in ever-growing need for adequate compute and storage capacity. To that end, Galaxy integrates with a number of cluster job managers and distributes the workload to compute clusters.

The adopted execution model and the overall Galaxy ecosystem have also been made available in the Cloud via the CloudMan application [2]. The CloudMan application makes it easy to deploy a virtual compute cluster atop cloud resources, configure those for use via Galaxy, and provide a management interface for the running applications and deployed resources [3]. CloudMan has been made compatible with a number of Cloud middleware technologies, easing the ability to deliver a complete genomics data analysis platform in a variety of settings.

However, individual Galaxy instances (whether dedicated or deployed via CloudMan) represent silos - once a user initiates an analysis on a given instance of Galaxy, transitioning to a different instance can prove to be difficult. A researcher may wish to transition to a different instance because of an alternate toolset, to move beyond possible usage quotas, or to move closer to their data. While Galaxy supports the ability to export and import user-facing artifacts (e.g., analyses, workflows), the size of the associated data, required application configuration compatibility as well as the time and effort on behalf of the user all represent challenges.

To alleviate these challenges, we plan on (1) integrating the ability to support cloud-bursting from a given Galaxy instance and (2) leveraging cloud data storage to benefit from data locality. Cloud bursting will enable expansion of the user-facing toolset (see below) as well as user

quotas. Particularly in the context of academic clouds, researcher will be able to apply their personal merit allocation to expand the compute capacity of a given Galaxy instance. Cloud bursting will be enabled with Galaxy's Pulsar application as part of a CloudMan cluster. Pulsar is a standalone server application that allows Galaxy jobs to be run on remote systems; it handles all aspects of job execution, including data staging, job preparation and submission to a local resource manager, job monitoring, and output data transfer to the Galaxy instance that initiated the job. One challenge in this context is the availability of required tools on the remote systems. Galaxy prides itself in data analysis reproducibility aspects [4] and hence every job should be run using the exact version of the underlying tool. To ensure the required tool is available on the remote system, we will be leveraging Docker containers. By providing a Docker container for each tool and each tool version where the container incorporates all the tool dependencies, a given data analysis can become truly reproducible. Because Docker containers are self-contained, it is also possible to allow a researcher to utilize tools that may otherwise not be installed on a given Galaxy instance.

Although Pulsar performs job data staging, continuously transferring increasingly large datasets is hardly desirable. Instead, we will leverage Cloud's object storage. For this purpose, Galaxy implements a notion of an Object Store - a pluggable file management interface that acts as a layer between Galaxy and any user datasets. Implementing the Object Store interface for various storage mediums (an abstract hierarchical store, Amazon S3, iRODS, and various local disk object stores are currently implemented) allows datasets to be 'physically' disconnected from a particular instance of Galaxy while the application can still access and interact with them. Pulsar can hence leverage the collocation of the data while Galaxy schedules consecutive (workflow) jobs on the same Pulsar instance (or even the same Cloud), reducing the amount of required data transfer.

Materializing above described functionality requires a Cloud supporting a number of features; these include: API access, support for custom machine images, instance user data, and an object store with support for per-object ACLs. To provide a cloud-contained solution (i.e., where the master Galaxy instance and all the Pulsar instances are cloud-based), the Cloud should also support persistent data volumes and shareable volume snapshots. We believe the listed features represents a reasonable set of requirements any Cloud should support.

Overall, as physically disparate compute resources continue to increasingly rely on each other, we strive to provide a unified resource view to the user. Simultaneously, we seek to optimize individual components of a growing puzzle. Providing support for cloud bursting in the context of the Galaxy application ecosystem while leveraging data locality to reduce the amount of data transfer represents a step in this direction.

References

[1] J. Goecks, A. Nekrutenko, and J. Taylor, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biol.*, vol. 11, no. 8, p. R86, Jan. 2010.

[2] E. Afgan, D. Baker, N. Coraor, B. Chapman, A. Nekrutenko, and J. Taylor, "Galaxy CloudMan: delivering cloud compute clusters," *BMC Bioinformatics*, vol. 11 Suppl 1, p. S4, 2010.

[3] E. Afgan, D. Baker, N. Coraor, H. Goto, I. M. Paul, K. D. Makova, A. Nekrutenko, and J. Taylor, "Harnessing cloud computing with Galaxy Cloud," *Nat. Biotechnol.*, vol. 29, no. 11, pp. 972–974, Nov. 2011.

[4] A. Nekrutenko and J. Taylor, "Next-generation sequencing data interpretation: enhancing reproducibility and accessibility," *Nature Reviews Genetics*, vol. 13, no. 9. pp. 667–672, 2012.