JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Reproducible and Shareable Data Science in Distributed Clouds

Randal Burns
Professor and Chair
Department of Computer Science

23 January 2020

Unpacking the Title

- **Reproducible**: anyone should be able repeat your analysis and produce the exact same result
- **Shareable:** reproducible and
  - Customize, extend, challenge, interrogate
  - Collaborate: in git verbs, branch, fork, pull, and push
- **Data Science**: Open platforms, such as Jupyter Lab and Rstudio
- **Distributed Clouds**: uniform experience on your laptop, an enterprise compute cluster, and a cloud service provider (AWS, GCE, Azure)

Disclaimer and a Brief History

- This talk describes the Gigantum data-science environment
  - Need came from my experience trying to build and distribute tools for science
  - Frustration that distributing code was hard an error prone
  - And, that reproducibility was poor
- Small Business Innovative Research Award from DARPA
  - Founded company 2016

Dr. Burns is a founder of and holds equity in Gigantum, Inc. He also serves as a Board Member of Gigantum, Inc. The results of the study discussed in this presentation could affect the value of Gigantum, Inc. This arrangement has been reviewed and approved by the Johns Hopkins University in accordance with its conflict of interest policies.

Things that are hard in data science

- Reproduce computational environments
  - User pulls a github repository and then goes through the iterative process of finding dependencies, installing packages, and determining code version.
  - Occasionally works. Mostly ends in a configuration conflict.
- Keep track of history
  - User loses track of what code, data, and software was used to make a figure.
  - git log has no record of computation, just of code versions.
- Synchronize large files
  - Have to be stored separately. Lose relationship to experiment.

Gigantum is an Automation Tool

- Best practices of software engineering for data science
  - Extends concept to cover datasets also
- Replaces complex command lines tools with simple UI
  - git, docker, JupyterLab, and RStudio with no expertise
- Models collaboration, sharing, and publishing
- Captures a reproducible history of *every action* in a project

The data scientist works in the tool of their choice and Gigantum makes their work product reproducible and sharable.

Amazon S3

Concept: Gigantum Project

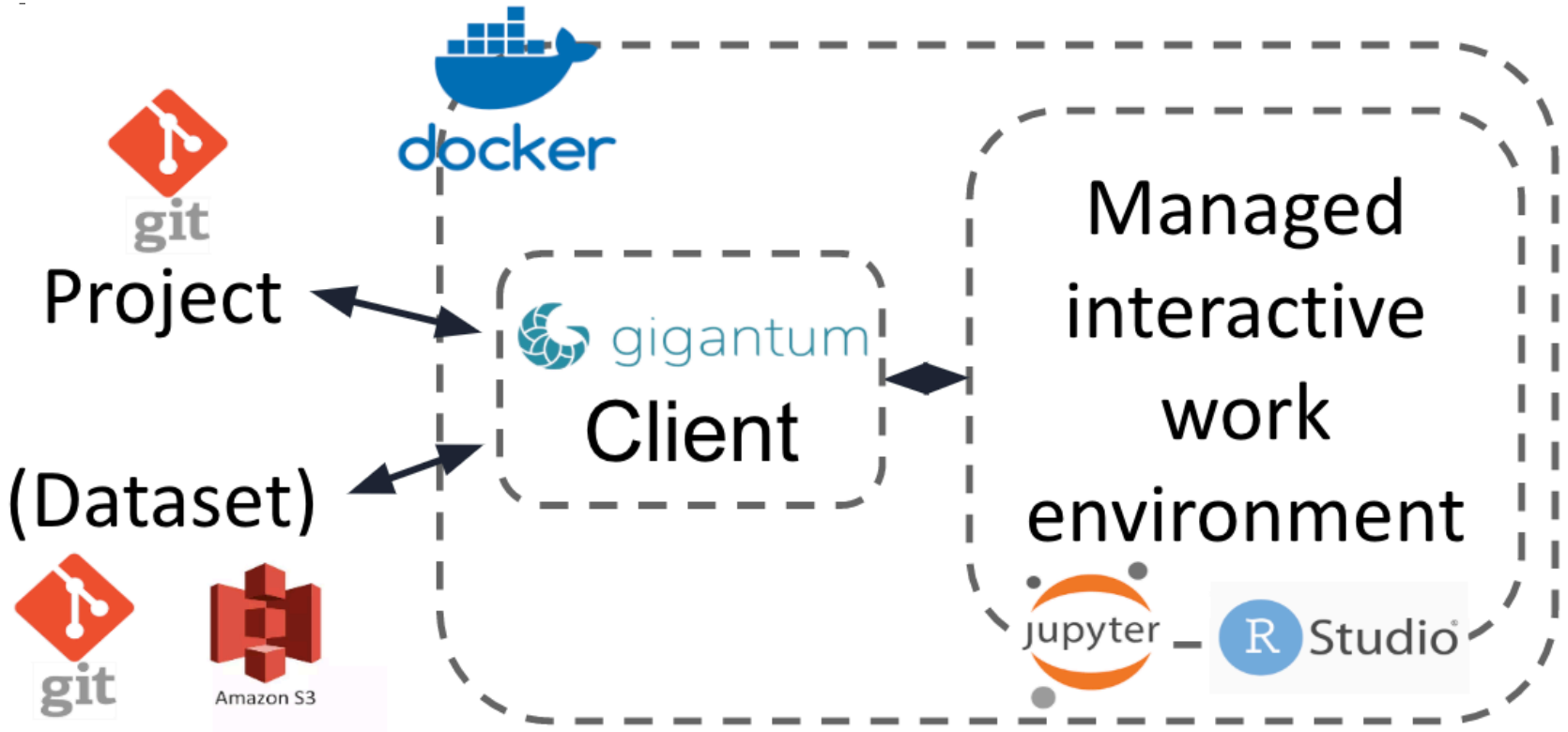
- A managed repository of code, data, and environment
  - A complete description of everything needed to run analyses
  - And a complete history of all activity
- Simplest interaction: launch a data science environment
  - Any work you do is recorded in a rich way.

The Activity Feed: Provenance Data

- Activity feed is a git log linked to an object database
  - Log records repository state, time, and metadata
  - Database keeps code snippets, images, outputs
- Every action leads to a git commit
- Activity feed is a reproducible record
  - Rollback to a good state

Schematic Gigantum Client

- Open-source, free to use

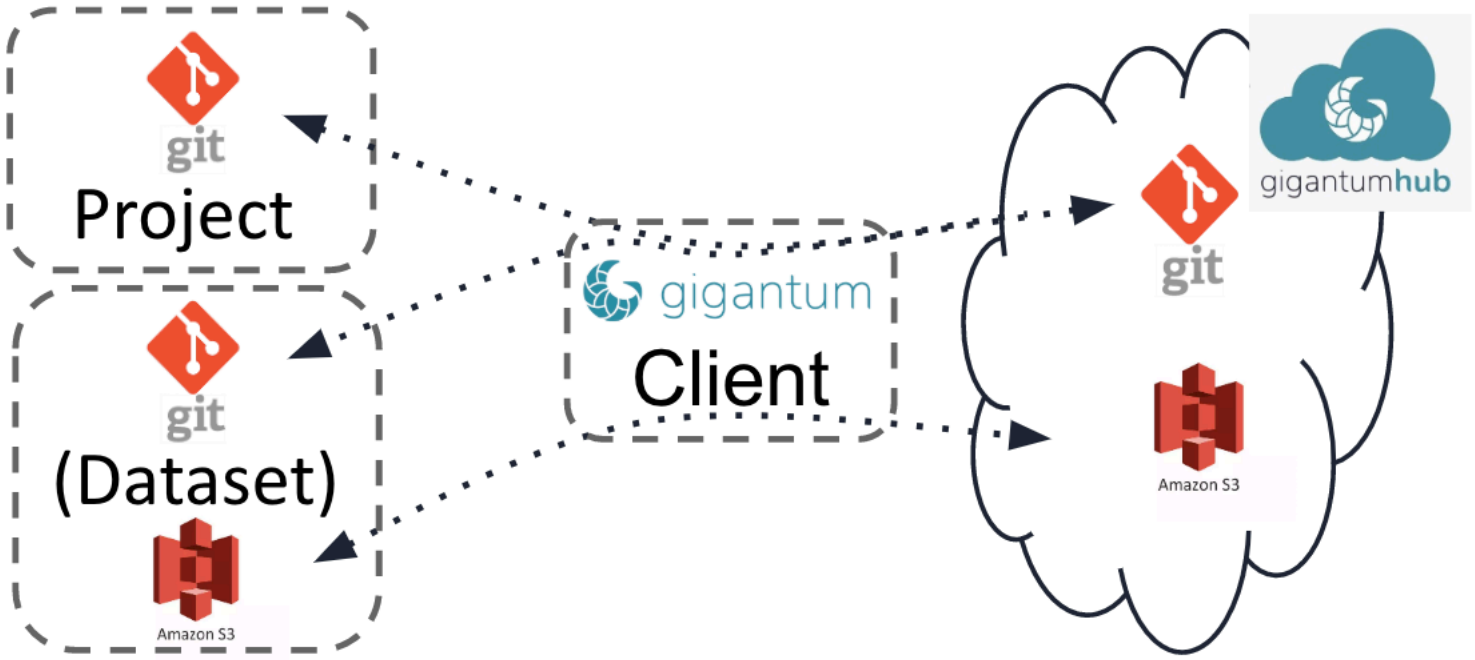


Building and Sharing a Project

- Base image: preconfigured OS, packages, and development tool
  - Can be customized
- Environment: install software from compatible package managers
- Sharing: default model is read/write collaboration
  - Given collaborators permission
  - Push to cloud, pull, modify, push back
- Moving projects to other computers is sharing with yourself

Gigantum Hub

- Project storage
- Dataset storage
- Managed compute
  - Launch projects



Collaboration Model: git workflows

- Gigantum has a sync button
  - This encapsulates the common set of git actions: add/commit/pull/push
  - Prompted to keep yours or take other on conflict
  - No problem on errors easy rollback
- Default sharing is to work within the same repository (like, git clone)
  - Read/write sharing allows push back
  - Read only sharing: change local, no push back
- Can create a copy of a repository (like, git fork)
  - Import/export in client. Copy button on hub.

Working with branches

- Gigantum branch model (again simplified git)
  - Create a named branch from current branch (new version)
  - Create a named branch from the activity log
- Managing branches
  - UI widget to merge branches

Final Thoughts

- Open-source product for open-source users
  - Local compute is always free
  - You own your data
- Cloud services with limits (free tier 10GB and 5 hours compute)
- Tool that I work in every day
  - Solves problem of distributing complex configurations to students
- Users other than data-science collaboration
  - Publishing
  - Teaching