Redivis: A Scalable Web Platform for Business Research

Alex Storer*
astorer@stanford.edu
Stanford University Graduate School of Business
Stanford, California, USA

Ian Mathews*
Erin DeLaney
ian@redivis.com
erin@redivis.com
Redivis Inc.
Oakland, California, USA

ABSTRACT

Redivis is a web platform specifically tailored to the needs of research computing in the context of large, restricted datasets. At the Stanford Graduate School of Business, Redivis has been leveraged to distribute terabyte-scale datasets to the research community. The platform addresses longstanding challenges in the administration of large, high-risk datasets, including scalable data ingest and curation tools, integrated access management, and robust audit trails. In turn, it also provides an accessible, high-performance compute environment to end users, allowing them to collaboratively build reproducible analytical workflows in SQL, R, Python, Stata, and SAS.

CCS CONCEPTS

• **Information systems** → **Computing platforms**; **Information integration**.

KEYWORDS

big data, business research, hpc, cloud computing

ACM Reference Format:
Alex Storer, Ian Mathews, and Erin DeLaney. 2024. Redivis: A Scalable Web Platform for Business Research. In *Practice and Experience in Advanced Research Computing (PEARC '24), July 21–25, 2024, Providence, RI, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3626203.3670604

1 INTRODUCTION

For academics working in business research, the growth in size and complexity of datasets has led to a host of opportunities to perform groundbreaking studies. Researchers across business disciplines are using public data, commercially licensable data, and proprietary datasets to test hypotheses using real-world data that cannot be collected in a laboratory setting [3]. While this real-world data provides a remarkable glimpse into behavior, it frequently comes with additional burdens due to its size and terms of use. Single datasets can easily extend into the terabyte range, spanning tens of billions of records, provided to researchers in thousands of compressed files. Additionally, these datasets are managed by licenses that can restrict who can use the data, on which systems. Providing systems that can effectively support the governance of these datasets while simultaneously allowing for practical solutions for researchers is a major goal of research computing organizations supporting business researchers.

2 BACKGROUND

High Performance Computing (HPC) environments at academic institutions are routinely built to allow researchers to log on to a Linux-based cluster and submit jobs to run on a large networked collection of compute nodes. For many workloads, these systems work effectively, providing a mechanism to request specific resources to process data and perform computations. While these systems aim to provide a general interface to compute resources, some workloads are less natural candidates for traditional HPC environments.

Many researchers, particularly in academic business schools, are trained using tools like R and Stata and rely heavily on graphical interfaces to data. Once a dataset can no longer be "loaded" into this software, researchers are faced with a paradigm shift regarding how to analyze their data. Powerful big data platforms like Spark can provide one solution, and Spark itself has interfaces to both R and Python. Unfortunately, distributed systems like Spark require extensive configuration to launch on multiple nodes in HPC systems. Even when these platforms are properly configured and launched, the scale of large datasets can strain HPC systems. Reading from large distributed datasets in parallel can produce problematic I/O patterns, particularly on systems with storage arrays that are not configured to handle high throughput from simultaneous users. Supporting the use of large datasets can prove challenging for users, as home directories, temporary directories and local disks often have size limitations that are orders of magnitude smaller than the data in question.

At business schools, a further challenge comes from expectations regarding data governance and compliance with strict terms of use [4]. For companies that collect and license data for commercial use, this underlying data is a valuable asset, and protecting this asset is paramount. Academic licensees need to verify that the data is stored securely, and that only approved researchers are able to access data. Unfortunately, on shared clusters, users can traditionally copy data to other locations, and set permissions on files that they own. This opens the possibility for nefarious or inadvertent exposure of this data to other system users, or even making the data public to external users by using popular tools like rclone or Globus. Furthermore, auditing tools such as `auditd` can report if a file is read by an approved user, but do not make clear

*Both authors contributed equally to this research.

PEARC '24, July 21–25, 2024, Providence, RI, USA

ACM ISBN 979-8-4007-0419-2/24/07
https://doi.org/10.1145/3626203.3670604

if a written file by that user contains the same data (e.g., it was copied to an unapproved location). Additionally, audit tools can be extremely verbose and technical, and the librarians and academic staff who track data use are not well positioned to use them directly.

From the standpoint of large datasets with monitoring requirements, neither researchers nor data administrators are well served by academic HPC systems. These general-purpose systems enable a broad range of diverse workflows, but leave gaps for scholars in business schools engaged with big data. Specifically, a preferred solution should enable investigators to query large datasets without complex configuration requirements, provide a clear and simple audit trail for non-technical data managers, and provide the ability to restrict sharing and exfiltration of data.

3 SOLUTION AND CAPABILITIES

Over the past five years, Redivis and the Stanford Graduate School of Business have closely collaborated to develop a platform that meets these needs in administering research data. Today, data administrators can ingest large datasets onto the platform, document them, assign access controls, and make them available to researchers. In turn, researchers can explore this data, build analysis pipelines in SQL, and execute code directly on the data to run models and create visualizations. Taken together, this functionality supports the entire research data lifecycle, from initial data ingest, to access control, data analysis, and research output management.

3.1 Application architecture

To support the evolving scale of data and to best align with the needs of burstable research workloads, as well as to enable us to focus our efforts on implementing the software requirements, we opted to leverage the commercial cloud for infrastructure. However, care was taken to choose technologies that were either a) open-source; or b) utilized open and standard interface, so as to prevent provider lock-in.

Google Cloud was chosen as the commercial cloud provider, with the Redivis application running on containerized services within a kubernetes cluster. Core components include a Node.js application layer, PostgreSQL metadata store, Google BigQuery as a highly-scalable tabular data store, and Google Cloud Storage for unstructured data. While Google BigQuery and Cloud Storage are both proprietary products, they use standard interfaces. Data in BigQuery is queried via ANSI SQL, and Cloud Storage provides a simple API for reading and writing objects, which could be trivially substituted for other object storage solutions, such as AWS S3, or a traditional POSIX filesystem.

3.2 Data ingest and version control

Data ingest is a precursor to any solution for research data management and analysis. It is also an immensely challenging problem when working with real world data and its myriad file formats and inconsistencies, particularly at scale.

Our backend tabular infrastructure, BigQuery, only supports well-formatted files in comma-delimited, newline-delimited JSON, Avro, and Parquet formats, with limited and error-prone type inference. To better meet the realities of the data provided by vendors, we developed a system of task runners to run data preprocessing

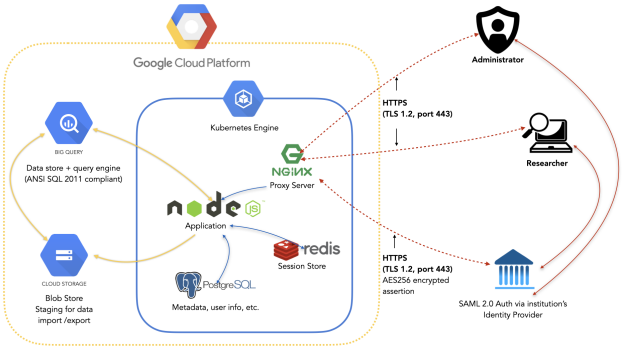

Figure 1: Core services and network diagram for the Redivis platform

pipelines, which are horizontally scaled within a kubernetes cluster and communicate via Redis pub/sub messaging. These task runners perform operations such as inferring schema, converting various file formats into a type supported by BigQuery, and fixing common idiosyncrasies found in source data files. This processing is all done in a streaming manner with minimal memory overhead, allowing us to easily process and convert individual files up to 5TB in size.

Additionally, many real-world datasets are an ever-evolving product, and data is often updated after the initial release. With datasets that are in continuous use, immutability of existing data is key. To handle this need, we implemented a version control system that is tuned to large tabular datasets and aims to avoid duplication and minimize storage costs. Our implementation computes a row-level difference between versions, only adding or deleting those records that have changed. All rows for all versions of a table are stored in a single "base" table, with a partition key that is unique to the minimum and maximum versions where the row is present. For a given version of a table, a logical view is defined that only selects the rows from the relevant partitions. To the end user, it appears that each version of a table is a distinct entity with the performance characteristics of a materialized table, but through this system we are able to substantially reduce storage costs for rapidly-changing datasets.

3.3 Access management and auditability

To meet the needs of managing access to high-risk datasets, we implemented a tiered, attribute-based access control (ABAC) system used by administrators to define the rules for accessing specific datasets. Importantly, different rules can be applied for access to a dataset's overview (existence and general documentation), metadata (variable names and univariate summary statistics), 1% sample, and full dataset. These rules can include direct approval from an administrator for a given access level, as well as the completion of a collection of requirements, implemented as customizable forms to be submitted by the researcher and approved by the data administrator.

Upon approval of access, it is important for many datasets that researchers be restricted from downloading the data to a personal computer, as this may violate data usage agreements and generally presents a major risk for unauthorized distribution. Therefore, the

access controls in Redivis were designed to allow administrators to define export restrictions based on the size and target environment. For example, rules can be configured to only allow export to another secure compute environment, or following administrator approval of the particular data derivative, or only for exports up to a certain size.

Finally, all access-related events are logged to an easily-navigable and searchable interface. Data administrators can track changes to users' access, as well as any data query or export events, with comprehensive details around the event, such as the acting user, their IP address, and information about the variables that were exported or queried.

3.4 Analysis

The analysis functionality on Redivis provides an accessible yet powerful interface for working with terabyte-scale datasets. Expressed as a Directed Acyclic Graph (DAG), researchers can join across any dataset that they have access to as they build data pipelines and analytical workflows.

When working with these massive datasets, researchers often only require a subset or aggregate of observations for their final analysis. The BigQuery query engine is highly optimized for this manipulation, with the ability to scale to thousands of computational nodes while performing complex filters, joins, and aggregations. Through the interface, authorized users can execute SQL queries to better understand the data and develop their analytical subset. With many of our users unfamiliar with SQL, we also developed a no-code graphical interface that can be used to perform most SQL operations, expressed as a series of consecutively applied "steps". Importantly, users can view and download the generated SQL at any time, ensuring reproducibility and serving as a pedagogical tool for those who want to learn SQL.

Once researchers have leveraged BigQuery to reduce their data to an appropriate subset, they require a mechanism to perform various analytical tasks, such as applying statistical methodologies, training and evaluating machine-learning models, and creating final figures. We chose JupyterLab [2] as the frontend interface for these analytical notebooks, given its web-native architecture, language agnostic design, and broad adoption amongst the academic community. Researchers can provision these notebooks with a base image offering R, Python, Stata, or SAS. These images come preinstalled with common data science packages, and researchers can run custom install scripts to further augment their computational environment. To prevent exfiltration of high-risk data, we automatically apply firewall rules that prevent data egress immediately after the install scripts are executed, and before any data can be present on the machine. These machines can be provisioned with the resources of any available virtual machine on Google Cloud; at the time of this writing, this includes up to 416 CPUs, 11.50TB RAM, and 16 NVIDIA Tesla A100 GPUs.

3.5 Reproducibility and research output management

The combination of data transformations, notebooks, and their derivatives are collectively referred to as a "project" on Redivis. The project is designed as a collaborative and reproducible environment.

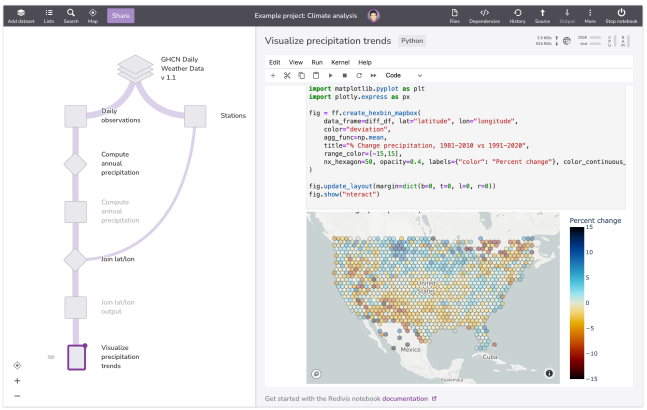

Figure 2: Screenshot of a Redivis project, with a notebook node currently active. Projects are represented as a directed acyclic graph made up of datasets, their tables, and a series of transforms, notebooks, and their derivatives.

All code is fully version controlled, with the ability to view and revert to a state at any previous point in time. Additionally, project owners can easily add their peers and work together in real-time, though each individual must possess appropriate access to view data and any derivatives.

The inherent reproducibility of these projects unlocks a significant opportunity to better manage derivative outputs. In many shared HPC systems, the size and storage cost of these outputs can eclipse that of the original source data. However, with full code history in Redivis, alongside version controlled datasets, we maintain the ability to recreate any output by re-executing the upstream code, and the decision to persist outputs becomes an optimization problem between ongoing storage costs vs. the computational cost to re-materialize an output. Based on historic access patterns to a given table, Redivis continuously evaluates whether the data for a given table should be deleted, while providing researchers a simple pathway to "undelete" a historic output if it does become relevant in the future.

4 UTILIZATION

At the time of this writing, Redivis has been deployed for several years at Stanford's Graduate School of Business, and its usage has been expanding – the bytes processed on the platform has expanded four-fold in the last year to 4.2PB. Informal feedback from researchers has been positive, and Stanford Graduate School of Business administrators are currently supporting multiple requests to port workflows from on-prem HPC systems to Redivis. This positive reaction is in line with other Stanford groups using Redivis as a central platform for hosting sensitive data [1].

5 CONCLUSION

The implementation of Redivis at the Stanford Graduate School of Business has allowed the school to significantly scale its research impact. The platform allows for data administrators to more effectively apply their resources, enabling a larger corpus of datasets and research use cases to be supported by the same team. In turn,

Table 1: Metrics for the Stanford Graduate School of Business on Redivis, April 24 2023 – April 24 2024

Data stored	96TB
Bytes processed	4.2PB
Transform queries	18,191
Notebook sessions	2,462
Total CPU time	414M seconds
Active users	151
Active projects	188

the platform increases the accessibility of these data for the school's research community, allowing for a broad population of researchers to engage with the data and develop novel findings. We are excited to continue to grow the implementation, learn from user feedback, and apply these results to similar research data settings across the academy.

REFERENCES

[1] Isabella Chu, Rebecca Miller, Ian Mathews, Ayin Vala, Lesley Sept, Ruth O'Hara, and David Rehkopf. [n. d.]. FAIR Enough: Building an Academic Data Ecosystem to Make Real-World Data Available for Translational Research. *Journal of Clinical and Translational Science*. ([n. d.]).

[2] Brian E. Granger and Fernando Perez. 2021. Jupyter: Thinking and Storytelling with Code and Data. *Computing in Science and Engineering* 23, 2 (2021). https://doi.org/10.1109/MCSE.2021.3059263

[3] Alice Kalinowski and Todd Hines. 2020. Eight things to know about business research data. *Journal of Business & Finance Librarianship* 25, 3-4 (10 2020), 105–122. https://doi.org/10.1080/08963568.2020.1847548

[4] Alex Storer and Julie Williamsen. 2021. The Research Hub: Providing Cross-functional Data Services. In *IASSIST*. Global Virtual Conference. https://doi.org/10.5281/zenodo.6754632