Big Data Processing in the Cloud: a Hydra/Sufia Experience

Collin Brittle, Virginia Tech, rotated8@vt.edu
Zhiwu Xie, Virginia Tech, zhiwuxie@vt.edu

Abstract

This presentation addresses the challenge of processing big data in a cloud-based data repository. Using the Hydra Project's Hydra and Sufia ruby gems and working with the Hydra community, we created a special repository for the project, and set up background jobs. Our approach is to create the metadata with these jobs, which are distributed across multiple computing cores. This will allow us to scale our infrastructure out on an as-needed basis, and decouples automatic metadata creation from the response times seen by the user. While the metadata is not immediately available after ingestion, it does mean that the object is. By distributing the jobs, we can compute complex properties without impacting the repository server. Hydra and Sufia allowed us to get a head start by giving us a simple self deposit repository, complete with background jobs support via Redis and Resque.

Audience

Repository managers will be interested in our approach to creating metadata and extracting features from large research datasets. Developers will be inspired by the distributed use case and the techniques used to address it, especially those to build and configure a cloud-based distributed Redis application. Data scientists will be informed of a unique approach to combine data repository and data processing in the cloud.

Background

This proposal address the following conference themes:

- Unconventional approaches to repository-like services
- Researcher-centered design for scholarly workflows
- Positioning repositories closer to (local, consortial, or cloud-based) cyberinfrastructure for data processing

The project reported here is a collaboration between the Smart Infrastructure Laboratory and the University Libraries, both at Virginia Tech, on building a computing infrastructure to deposit, process, and disseminate the large amount of sensor data generated from the Virginia Tech Signature Engineering Building (SEB). Equipped with approximately 140 sensor locations and up to 429 data channels, SEB will be the world's most instrumented public building for vibrations [1]. When fully operational, it will generate more than 30TB of sensor data per year, continuously exposing the acceleration,

temperature, acoustics, strain, wind and flow, and load conditions of the building.

The SEB dataset requires a number of distinctive features not typically found in a digital library repository.

First, the rate of data creation is exceptionally high, yet the metadata generation and feature extraction algorithms can be highly complicated and computationally intensive. This makes it difficult to process the data and generate metadata in place while it's being ingested. In other words, the existing local computing resources may not be sufficient for the conventional ingestion workflow, yet building a new local IT infrastructure from scratch requires expensive investment that we suspect is neither feasible nor necessary. We believe it's more advantageous to build a cloud-base system that takes advantage of the low cost, seemingly unlimited, and elastic on-demand computing power.

Second, the rate of data growth will soon exhaust our local data storage but our existing campus IT infrastructure does not provide a sufficiently economical and sustainable solution to handle the long-term storage. On the other hand, in contrast to the high volume of the data set, we expect the data will quickly turn cold, and the access rate to stale data should be very low if we can properly characterize these data to address most of the use cases. This indicates that a cheaper, near-line type of storage solution such as Amazon Glacier may be more suitable for this type of application.

Third, the metadata needs for such datasets are much more domain specific and harder to be predetermined, therefore requires asynchronous online metadata creation. For example, even after the data ingestion and the creation of the common metadata fields such as title, creator, timestamp, and persistent URI, the end user may still be interested to screen the dataset for a specific acceleration pattern that indicates a potential earth movement. If properly marked, only the stale data segments that are marked positive on this metadata filed will be of interest to earthquake researchers and most of the other segments will be left untouched. Otherwise the researchers may have to repeatedly download large amount of stale data and process them offline. This would not only be inefficient and wasteful of computing cycles and network bandwidth, but also add significant load and cost to the data repository. Therefore, the quality and efficiency of such asynchronous metadata creation and feature extraction will strongly affect the repository access pattern therefore the cost of maintenance of the data repository.

Fourth, given the above three characterization, we consider it a crucial component for the data repository to optimize the asynchronous metadata generation and feature capture to minimize storage I/O. This requires a sophisticated and scalable cloud-based big data processing job-scheduling framework, which is the focus of this proposal.

The SEB data challenge represents one type of big data use case [2] that is become more and more common in today's data intensive scientific research environment. However there is little known experience and few open source projects that address the needs of these use cases. We therefore hope the experience reported here would be useful to other similar endeavors.

Presentation content

Our presentation will cover the background and scope of the project, our solution, the technology stack that makes the solution possible, and the community that helped us along the way.

As an introduction, we will explain the library's involvement, and the requirements for the project. Next, we will dive into our solution. Our focus will be on the parts of the standard Hydra/Sufia [3, 4] technology stack that we customized for our application. Components like Redis [5], Resque [6], and background jobs will feature prominently, while others like Rails, Solr, and the controllers will not. Figure 1 schematically shows the components of the job-scheduling framework.

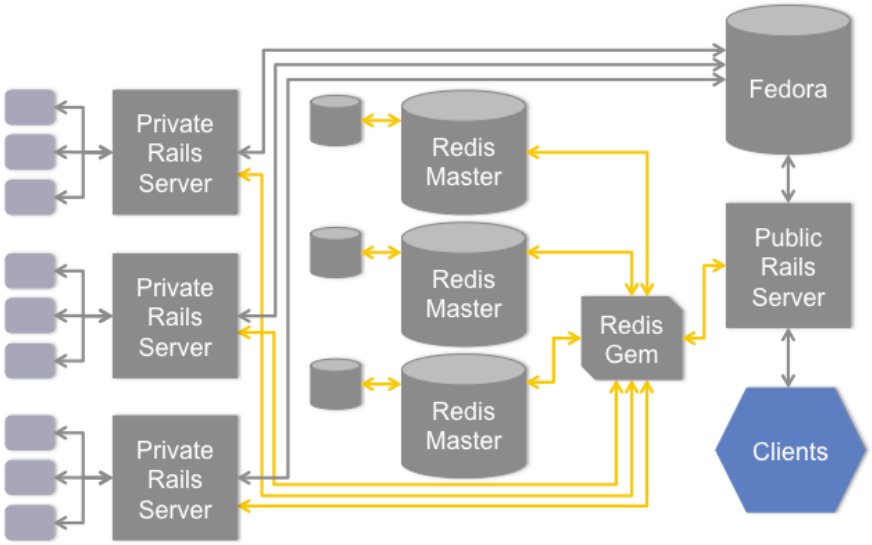

Figure 1. Cloud-based job scheduling for metadata generation and feature capture

Since the Hydra Project is a major foundation to our approach, we will also talk about what we gained by choosing their framework, and how the community has helped us along the way. From the beginning, the project has planned to be scalable. The presentation will address how we planned to accommodate that kind of growth.

Conclusion

Our presentation should inspire the audience to examine their data processing workflow, or consider adding one to their repository. They should be convinced that we benefited from participating in the Hydra community. The problems we encountered while dealing with remote infrastructure should help them when they design their research data ecosystem.

References

1. Pablo Tarazaga and Mary Lasarda. “The Signature Engineering Building’s Instrumentation Program: A Living Laboratory for Research and Education”, 2013. http://www.eng.vt.edu/sites/default/files/pageattachments/SEBinstrumentation.pdf

2. Zhiwu Xie. “Facilitate Cross-Repository Big Data Discovery and Reuse”, Research Data Management Implementations Workshop, March 13-14, Arlington, VA, 2013.

3. Hydra project. http://projecthydra.org/

4. Sufia. http://github.com/projecthydra/sufia

5. Redis. http://redis.io

6. Resque. http://github.com/resque/resque