Dredging a Data Lake: Decentralized Metadata Extraction

Tyler J. Skluzacek
University of Chicago
skluzacek@uchicago.edu

ABSTRACT

The rapid generation of data from distributed IoT devices, scientific instruments, and compute clusters presents unique data management challenges. The influx of large, heterogeneous, and complex data causes repositories to become siloed or generally unsearchable—both problems not currently well-addressed by distributed file systems. In this work, we propose Xtract, a serverless middleware to extract metadata from files spread across heterogeneous edge computing resources. In my future work, we intend to study how Xtract can automatically construct file extraction workflows subject to users' cost, time, security, and compute allocation constraints. To this end, Xtract will enable the creation of a searchable centralized index across distributed data collections.

CCS CONCEPTS

• **Information systems → Computing platforms**; *Search engine indexing*; • **Applied computing → Document metadata**.

KEYWORDS

data lakes, serverless, metadata extraction, file systems

ACM Reference Format:
Tyler J. Skluzacek. 2019. Dredging a Data Lake: Decentralized Metadata Extraction. In *Middleware '19: 20th International Middleware Conference Doctoral Symposium (Middleware '19), December 9–13, 2019, Davis, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3366624.3368170

1 INTRODUCTION

The rapid generation of data from IoT devices, scientific instruments, simulations, and myriad other sources presents unique data management challenges. Currently data are stored across multiple machines, are often siloed, and require significant manual labor to create metadata that promote usability and searchability. Some [4, 5] have created data catalogs from user-submitted metadata. These, however, are not scalable to current and future storage systems, as humans cannot possibly label billions of heterogeneous files. In order to better organize, discover, and act upon distributed big data, we first require automated methods to crawl file systems and extract metadata for each file therein. While others have developed end-to-end automated metadata extraction systems, they require that data be moved to a central service [7–9, 11] or lack built-in scaling capabilities [6]. In this work we strive to create a flexible, scalable, and decentralized metadata extraction system that can be employed centrally or at the edge.

We present our prototype and vision for Xtract, a decentralized middleware that provides high-throughput and on-demand metadata extraction, enabling the automated creation of rich, searchable data lakes from unsearchable data silos. We leverage a Function as a service (FaaS) model for managing the invocation of many short-running extractors on an arbitrarily large number of files. The current Xtract implementation uses the *funcX* serverless supercomputing platform [3] to execute functions across diverse and distributed computing infrastructure. The advantage of using *funcX* is that it allows us to explore a novel distributed FaaS model that overcomes the need to move large amounts of data to the cloud. Instead, Xtract is able to push metadata extractors to the edge systems on which the scientific data reside. We envision that Xtract could also use other edge computing fabrics, such as Amazon Web Services IoT Greengrass or Google Cloud IoT. The primary contributions of Xtract are:

- Scalable across distributed computing resources, including laptops, clusters, and edge devices.
- Flexible extraction model that can be deployed centrally or at the edge, facilitating decentralized metadata extraction.
- Supports dynamic construction of customized extractor pipelines for diverse file types.
- Intelligently executes data staging decisions based on user-supplied constraints, including time, cost, compute allocation availability, and security.

2 APPROACH

Xtract is a decentralized middleware that provides distributed metadata extraction capabilities over heterogeneous compute resources. The Xtract service plans dynamic metadata extraction pipelines, comprised of specialized metadata extractor functions, and coordinates the execution of those extractors either locally or at the edge, subject to various constraints. An example two-site deployment of Xtract is shown in Figure 1. The remainder of this section details Xtract's core components and design goals.

Metadata extractors are functions that input a file or group of files, and output a metadata dictionary. Each metadata extractor runs in a given container runtime with all required dependencies (i.e., files and libraries). Xtract currently provides a number of built-in extractors, including those to identify null values in tabular files, nesting patterns in structured XML files, topics and keywords from free text, and location tags from map images, among others [10]. In future work, we plan to support user-submitted metadata extractors, automatically generate (and potentially share) runtime containers based on inferred dependency requirements, and train Xtract to recognize when it is appropriate to apply user-submitted extractors in each file's metadata extraction workflow.

Middleware '19, December 9–13, 2019, Davis, CA, USA

ACM ISBN 978-1-4503-7039-4/19/12…$15.00
https://doi.org/10.1145/3366624.3368170

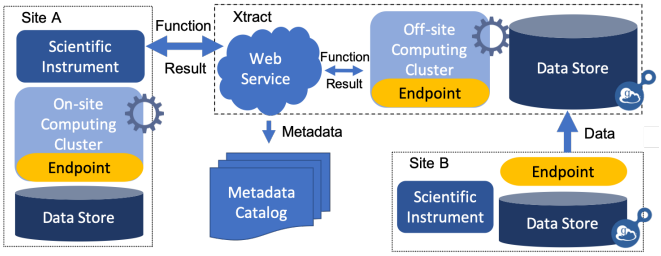

Figure 1: Overview of the Xtract architecture. For *Site A* extractors are transmitted to the remote resource for execution on local computing resources, returning metadata to the Xtract service. *Site B* lacks suitable local computing capabilities, requiring data to be staged to Xtract for extraction.

The **Xtract service** dynamically applies a set of metadata extractors to each file in the repository. First, Xtract sends a crawler function to the data and populates a metadata dictionary for each file containing its file system properties (e.g., path, size, extension). Once the initial dictionary is created, Xtract invokes a file type extractor on each file that is used to select downstream extractors. We have shown that feeding the first n bytes of a file as features into a trained model can predict the appropriate first extractor functions to apply to a file, in significantly less time than attempting to execute incorrect extractors on each file [11]. Once this first extractor function returns metadata, the Xtract service dynamically selects additional extractors to apply based on the output. For instance, a tabular file with a multi-line free text header (e.g., describing experimental setup) is identified by Xtract as first requiring a tabular extractor, but the resultant metadata will denote free text parts of the file that can benefit from keyword and topic extractors. Xtract protects requests to the web service using Globus Auth [13], and stores metadata to a Globus Search index.

Endpoints are the edge computing fabric that provision compute resources and execute functions in container runtimes for files on the local file system. Endpoints can currently be deployed across myriad compute providers such as IoT devices, cloud instances, and clusters. Endpoints deployed in *funcX* utilize the Parsl [1] parallel programming library to provision compute resources, and to manage the execution of functions in containers on provisioned resources. Endpoints enable Xtract to execute metadata extraction functions at any registered and accessible endpoint. We have shown that deploying *funcX* endpoints on HPC systems allows Xtract to reliably scale to deploy millions of metadata extraction functions across thousands of nodes spanning multiple compute locations [3].

3 EVALUATION PLAN

In future work we plan to study how Xtract can optimize the creation of extraction workflows and deployment of extractors subject to user-defined constraints. Specifically, we plan to explore how metadata extraction workflows can be augmented to place extractors on, or stage data to, idle or under-utilized resources. We will also investigate globally optimal extraction strategies with respect to diverse user constraints of financial cost, computing time, security, and compute resource allocation availability or volatility.

We intend to evaluate Xtract's performance and optimization strategies across a diverse set of datasets. These include the Carbon Dioxide Information Analysis Center (330+ GB, 10,000+ unique file extensions of carbon dioxide data); the Materials Data Facility [2] (30+ TB, tens of millions of materials science files); Petrel (4+ PB, 50,000+ files of cross-disciplinary data at Argonne National Lab); and Globus-accessible endpoints (20,000+ unique endpoints containing hundreds of billions of files).

4 CONCLUSION

Xtract is a metadata extraction middleware that addresses data locality and scalability challenges by deploying metadata extractors to edge devices and constructing extraction workflows subject to a number of user constraints. Xtract will enable researchers, companies, and individuals alike to more easily discover, organize, and understand increasingly large, complex, and distributed data, leading to enhanced scientific and industrial progress.

ACKNOWLEDGMENTS

This research is conducted under the guidance of Dr. Ian Foster and Dr. Kyle Chard, and with contributions from Dr. Ryan Chard, Dr. Zhuozhao Li, Yadu Babuji, and Ryan Wong. We gratefully acknowledge the use of compute resources from the Jetstream cloud for science and engineering [12].

REFERENCES

[1] Yadu Babuji, Anna Woodard, Zhuozhao Li, Daniel Katz, Ben Clifford, Rohan Kumar, Lukasz Lacinski, Ryan Chard, Justin Wozniak, and Ian Foster. 2019. Parsl: Pervasive parallel programming in python. In *Proceedings of the 28th Int'l Symposium on High-Performance Parallel and Distributed Computing*. ACM, 25–36.

[2] Ben Blaiszik, Logan Ward, Marcus Schwarting, Jonathon Gaff, Ryan Chard, Daniel Pike, Kyle Chard, and Ian Foster. 2019. A Data Ecosystem to Support Machine Learning in Materials Science. (apr 2019). arXiv:1904.10423

[3] Ryan Chard, Tyler J Skluzacek, Zhuozhao Li, Yadu Babuji, Anna Woodard, Ben Blaiszik, Steven Tuecke, Ian Foster, and Kyle Chard. 2019. Serverless Supercomputing: High Performance Function as a Service for Science. *arXiv preprint arXiv:1908.04907* (2019).

[4] MP Egan, SD Price, KE Kraemer, DR Mizuno, SJ Carey, CO Wright, CW Engelke, M Cohen, and MG Gugliotti. 2003. VizieR Online Data Catalog: MSX6C Infrared Point Source Catalog. The Midcourse Space Experiment Point Source Catalog Version 2.3 (October 2003). *VizieR Online Data Catalog* 5114 (2003).

[5] Gary King. 2007. An introduction to the dataverse network as an infrastructure for data sharing.

[6] Chris Mattmann and Jukka Zitting. 2011. *Tika in action*. Manning Publications.

[7] Smruti Padhy, Greg Jansen, Jay Alameda, Edgar Black, Liana Diesendruck, Mike Dietze, Praveen Kumar, Rob Kooper, Jong Lee, Rui Liu, et al. 2015. Brown Dog: Leveraging everything towards autocuration. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 493–500.

[8] Gonzalo P Rodrigo, Matt Henderson, Gunther H Weber, Colin Ophus, Katie Antypas, and Lavanya Ramakrishnan. 2018. ScienceSearch: Enabling search through automatic metadata generation. In *2018 IEEE 14th International Conference on e-Science (e-Science)*. IEEE, 93–104.

[9] Tyler J Skluzacek, Kyle Chard, and Ian Foster. 2016. Klimatic: a virtual data lake for harvesting and distribution of geospatial data. In *2016 1st Joint International Workshop on Parallel Data Storage and data Intensive Scalable Computing Systems (PDSW-DISCS)*. IEEE, 31–36.

[10] Tyler J. Skluzacek, Ryan Chard, Ryan Wong, Zhuozhao Li, Yadu Babuji, Logan Ward, Ben Blaiszik, Kyle Chard, and Ian Foster. 2019. Serverless Workflows for Indexing Large Scientific Data. In *5th Workshop on Serverless Computing (WoSC '19)*. ACM, New York, NY, USA, 6. https://doi.org/10.1145/3366623.3368140

[11] Tyler J Skluzacek, Rohan Kumar, Ryan Chard, Galen Harrison, Paul Beckman, Kyle Chard, and Ian Foster. 2018. Skluma: An extensible metadata extraction pipeline for disorganized data. In *2018 IEEE 14th International Conference on e-Science (e-Science)*. IEEE, 256–266.

[12] Craig A Stewart, Timothy M Cockerill, Ian Foster, David Hancock, Nirav Merchant, Edwin Skidmore, Daniel Stanzione, James Taylor, Steven Tuecke, George

Turner, et al. 2015. Jetstream: a self-provisioned, scalable science and engineering cloud environment. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*. ACM, 29.

[13] Steven Tuecke, Rachana Ananthakrishnan, Kyle Chard, Mattias Lidman, Brendan McCollam, Stephen Rosen, and Ian Foster. 2016. Globus Auth: A research identity and access management platform. In *2016 IEEE 12th International Conference on e-Science (e-Science)*. IEEE, 203–212.