Streaming Analytics and Workflow Automation for DFS

Yasith Jayawardana and Sampath Jayarathna
Department of Computer Science
Old Dominion University
Norfolk, VA, USA
{yasith,sampath}@cs.odu.edu

ABSTRACT

Researchers reuse data from past studies to avoid costly re-collection of experimental data. However, large-scale data reuse is challenging due to lack of consensus on metadata representations among research groups and disciplines. Dataset File System (DFS) is a semi-structured data description format that promotes such consensus by standardizing the semantics of data description, storage, and retrieval. In this paper, we present *analytic-streams* – a specification for streaming data analytics with DFS, and *streaming-hub* – a visual programming toolkit built on DFS to simplify data analysis workflows. Analytic-streams facilitate higher-order data analysis with less computational overhead, while streaming-hub enables storage, retrieval, manipulation, and visualization of data and analytics. We discuss how they simplify data pre-processing, aggregation, and visualization, and their implications on data analysis workflows.

CCS CONCEPTS

• **Information systems** → **Information systems applications**; • **Applied computing** → *Document management and text processing*.

KEYWORDS

Streaming Analytics, Workflow Automation, Data Analysis

ACM Reference Format:
Yasith Jayawardana and Sampath Jayarathna. 2020. Streaming Analytics and Workflow Automation for DFS. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20), August 1–5, 2020, Virtual Event, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3383583.3398589

1 INTRODUCTION

With advancements in digital technology, researchers have access to vast amounts of data already collected from past studies. Such data is being reused to avoid costly re-collection of experimental data. However, factors such as the heterogeneity of data collection procedures, file storage formats, and regulatory requirements curtail the expansion of this practice beyond a certain research group or discipline. This problem is exacerbated by the risk of encountering outdated, erroneous, or even incompatible data when exploring the unknown. Studies that depend on data reuse, thus require extra time and effort to carefully pick, analyze and transform data into a workable form. Given such circumstances, adopting a cross-disciplinary metadata standard for data discovery, versioning, provenance, and linking, would substantially improve the quality of large-scale data reuse and promote reproducible research.

DFS [5] is a semi-structured dataset description format for replacing text-based documentation with rich, consistent metadata. It provides three abstractions: *data-stream*, *meta-file*, and *meta-stream*. A data-stream carries data from datasets (non real-time) or sensory data sources (real-time). A meta-file describes the files, fields, collection procedures, etc. in a dataset, while a meta-stream describes the device specifications, channel information, etc. of a sensory data source. Together, they disambiguate the field-level relationships in data-streams, which is beneficial for large-scale data reuse. It also provides a formal mechanism to cite datasets immutably in scholarly publications.

Visual analytics is the science of analytical reasoning supported by interactive visual interfaces [8]. While visualizations often help to communicate findings from analysis, they could also be used to improve exploratory analysis [1] and data pre-processing workflows [6]. Studies [2] claim that researchers without a computer science background prefers visual programming over writing code. Thus, visual analytics could be utilized to streamline data analysis workflows, regardless of domain.

We hypothesize that DFS could be expanded into a full-scale metadata framework, given the right tools for data analysis, data analytics reuse, and visualization. To achieve this, we introduce *analytic-streams* – a metadata specification for streaming data analytics with DFS, and *streaming-hub* – a visual programming toolkit built on DFS to simplify data analysis workflows.

2 DESIGN

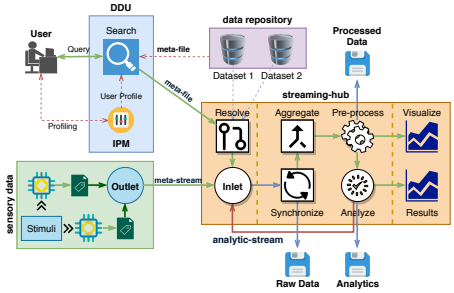

Figure 1: Proposed data analysis workflow with DFS

We propose a data analysis workflow (see Figure 1) based on four components: 1) **DDU** – to search and discover data using metadata, 2) **streaming-hub** – to build visual workflows for data analysis,

JCDL '20, August 1–5, 2020, Virtual Event, China

ACM ISBN 978-1-4503-7585-6/20/08.
https://doi.org/10.1145/3383583.3398589

3) **analytic-streams** – to stream data analytics, and 4) **workflow automation** - to automate data analysis workflow generation.

2.1 Data Discovering Users (DDU)

DDU is the dataset search, discovery and recommendation component of our design. Here, datasets are stored in repositories. DDU servers index metadata to facilitate user queries. They also maintain an interest profile model (IPM) [4] on each user for personalized recommendation. Thus, DDU provides an ecosystem for users to discover data, and for data to reach prospective users.

2.2 Streaming-hub

Streaming-hub is the visual analytics component of our design. We implement streaming-hub using *Orange* [3], which is an open-source data visualization, machine learning, and data mining toolkit with a visual front-end for exploratory data analysis and interactive data visualization.

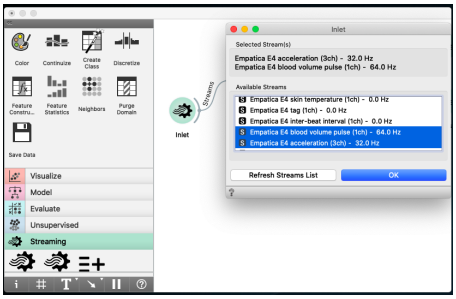

Figure 2: Inlet widget from streaming-hub in Orange canvas

We created input–output widgets to connect DFS into Orange. One such widget is the *Inlet* (see Figure 2), which provides the ability to subscribe to one or more *data-streams*. Each widget was designed to be inter-operable with existing Orange widgets, or with each other. Collectively, they enabled to build workflows using DFS abstractions, and to visualize how data flows in real-time.

2.3 Analytic-streams

Analytic-streams are a specification for streaming data analytics with DFS. Their objective is to carry analysis results analogous to how data-streams carry data. Using analytic-streams brings several advantages to subsequent data analysis workflows: First, each analytic-stream could be immediately reused, or persisted in a dataset for later reuse. This promotes reproducible research, and facilitates computationally-efficient higher-order analysis for complex data analysis tasks. If analytic-streams are indexed in datasets (using meta-files), users can selectively consume data based on their need, and ignore the rest to save bandwidth.

2.4 Workflow Automation

The objective of workflow automation is to intelligently place widgets in the Orange canvas, based on data-streams that a user subscribes to. Figure 3 shows a sample workflow generated for the streams selected in Figure 2, with analytic-stream outlets. Each generated workflow provides seven functionalities: **Resolve:** Fetch data represented by each meta-file from their repositories and make them available for analysis. **Inlet:** Send data streams, meta-files, meta-streams, and analytic-streams into the workflow, using LSL [7]. **Synchronize:** Synchronize data-streams in time using an in-memory cache, and handle missing or redundant packets. **Aggregate:** Combine multiple data-streams where applicable, and return them for analysis. **Pre-process:** Infer pre-processing directives from metadata and apply them onto incoming data-streams. Each directive will have its own widget, and will only be applied if inferred at a high confidence. **Analyze:** Here, users append new widgets to the generated workflow to serve their need. Results from analysis can be published through an analytic-stream, and persisted for later replay. **Visualize:** Data can be visualized on-screen at any point in the workflow, by adding visualization widgets.

Figure 3: Streaming-hub workflow with analytic-stream outlets

3 CONCLUSION AND FUTURE WORK

In this work, we show that DFS could be extended into a full-scale metadata framework through analytic-streams and streaming-hub. Our contribution lays groundwork for large-scale data reuse and visual data analysis by simplifying data discovery, storage, pre-processing, aggregation and visualization. To encourage reproducible research, our implementation of streaming analytics[1] and workflow automation[2] have been made publicly available. In the future, we will explore two variants of analytic-streams: 1) analytics *parallel* to data-streams (current version), and 2) analytics *embedded* into data-streams. We also plan on formulating a concrete evaluation plan for our proof-of-concept.

REFERENCES

[1] A. Batch and N. Elmqvist. 2018. The Interactive Visualization Gap in Initial Exploratory Data Analysis. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan. 2018), 278–287.
[2] G.H. Brimhall and A. Vanegas. 2001. Removing Science Workflow Barriers to Adoption of Digital Geologic Mapping by Using the GeoMapper Universal Program and Visual User Interface. In *Digital Mapping Techniques*. U.S. Geological Survey Open-File Report 01-223, Tuscaloosa, AL, USA, 103–115.
[3] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, et al. 2013. Orange: data mining toolbox in Python. *The Journal of Machine Learning Research* 14, 1 (2013), 2349–2353.
[4] S. Jayarathna and F. Shipman. 2017. Analysis and Modeling of Unified User Interest. In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, San Diego, CA, USA, 298–307.
[5] Y. Jayawardana and S. Jayarathna. 2019. DFS: A Dataset File System for Data Discovering Users. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, Urbana-Champaign, IL, 355–356.
[6] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, et al. 2011. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization* 10, 4 (2011), 271–288.
[7] C. Kothe. 2014. Lab streaming layer (LSL). *https://github.com/sccn/labstreaminglayer* 26 (2014), 2015.
[8] J.J. Thomas and K.A. Cook. 2006. A visual analytics agenda. *IEEE computer graphics and applications* 26, 1 (2006), 10–13.

[1] https://github.com/nirdslab/dfs
[2] https://github.com/nirdslab/streaminghub