Capturing Relationships Based on Structure Similarity for Self-describing Scientific Data Formats

Chenxu Niu
Texas Tech University
Lubbock, Texas
Chenxu.Niu@ttu.edu

Wei Zhang
Texas Tech University
Lubbock, Texas
X-Spirit.Zhang@ttu.edu

Suren Byna
Lawrence Berkeley National Laboratory
Berkeley, California
sbyna@lbl.gov

Yong Chen
Texas Tech University
Lubbock, Texas
Yong.Chen@ttu.edu

ABSTRACT

Many scientific data sets use self-describing files for storing data, and files within a data set are often isolated without any definition of relationships among them. Because of the isolated management of scientific data files, locating, assimilating, and utilizing relationships for a given query remains a long-standing problem in data discovery. Many relationships are often hidden in complex file structures where different kinds of relationships are not explicitly categorized. To build relationships among scientific data files that are stored in self-describing formats, we propose a relationship capturing method based on structure similarity determination of the files. Similarity is a fundamental concept in computer science. Crucially, "similarity" is not a typical relationship that reflects correlations of properties in different scientific files. But determining similarity has been proven useful in a variety of application areas. We can use structure similarity determination to capture important relationships. We have evaluated our approach on six sets of real-world scientific files. Our approach demonstrates effective relationship capturing and efficient relationship building performance.

1 INTRODUCTION

In the context of large-scale scientific data sets, numerous self-describing files formats - such as HDF5 [4], netCDF [7], FITS [10], ASDF [5], ADIOS-BP [6], Zarr [2], N5 [9], PnetCDF [8], and ROOT [3] - have become increasingly popular formats for storing immense amounts of experimental, observational, and simulated data from various scientific applications. However, most self-describing files store data in an isolated manner. Specifically, while metadata contained in self-describing files provide information in regards to internal data objects, it does not describe potential relatedness to other files in the data set. In order to avoid the productivity bottleneck of discovering new data sources, we need a more general method that organizes the data sources, captures linkages and represents their relationships to allow relationship querying for data discovery; otherwise, data discovery becomes a restrictive process that hampers the overall productivity of organizing and managing data.

Most definitions of scientific datasets have four features: *grouping, content, relatedness, purpose.* Descriptions such as "observations" indicates content of a propositional sort, while "values" or "records of values" might be understood as being representations of propositional content. In general, data in a scientific dataset are typically expected to have the same syntactic structure and similar metadata structure (i.e., records of the same length, field values in the same places, etc). Data in a dataset are "sharing a structure". Exemplifying this, all records of a dataset from Baryon Oscillation Spectroscopic Survey (BOSS) [1] are assumed to have a common structure, with each position having its specific meaning, which is common to all values appearing in it. It should specify the context in which the observations or measurements were obtained. The context may include, for example, the place and the time of observation or measurement, and the object or group of objects observed. Many scientific data formats are designed to include internal metadata, which is used to describe the data together using attributes. Some scientific datasets only give a partial view of entities, with several attributes that can be missing or sparsely populated. Instead of capturing relationships among entities, we need to capture high-level relationships based on data objects. Unfortunately, some relationships are often hidden in complex structures where different kinds of relationships are not explicitly categorized. The objective of this research investigation is to discover relationships based on structure similarity, which can benefit many applications such as expert finding and experimental analysis.

2 METHODOLOGY

Figure 1 shows an overview of our proposed approach. The ideology behind our method is to extract the tree structures of scientific files and reformulate the structure similarity determination problem of finding common sub-trees among trees for self-describing file formats. In the first phase, we recursively scan the groups, data objects and their attached metadata attributes of each sub-directory within the root directory. During this process, we actively extract the file contents into a tree-like structure. After the tree structure extraction, we can propose the key component of our methods. The similarity between two scientific files can be reformulated to the problem of finding all common sub-trees between two trees. A detailed description of the process will be explained in next paragraph. Given a base file, we will find all common sub-trees and calculate the structure percentage of the base file as a similarity "score". Then we can capture a novel relationship based on similarity to link self-describing data files. The percentage constitutes our

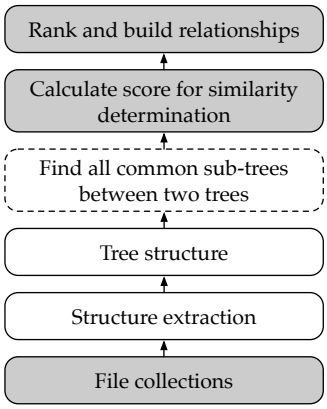

Figure 1: Overview of similarity determination and relationship capturing process in our method

similarity "score" to represent the structure similarity and metadata similarity. We will rank the "score" and link the scientific files with structure similarity relationships.

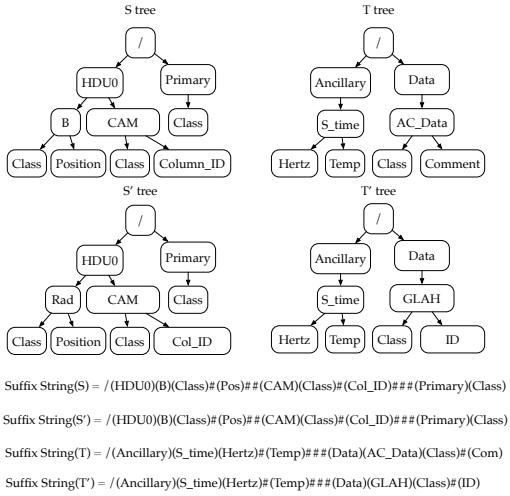

Figure 2: Four HDF5 files and suffix strings of their tree structure patterns.

As shown in Figure 2, we illustrate our tree similarity determining method with four HDF5 files as a concrete example. First, we extract the tree structure from four HDF5 files. In this concrete example of S tree, "HDU0" and "Primary" are two groups in "/" root group. Different from the original "group" name or "data object" name, the new suffix string is labelled with pairs of integers (starting position and length) representing sub strings. The concatenation of the labels on a path from the root to an attribute leaf represents the suffix string starting at position. Then we mark all the nodes of the suffix string, which have at least one red leaf node and one blue leaf node as a descendant. Given an internal node of one tree, we denote the concatenation of the sub-strings. In the next step, we compute the value for each blue leaf node of one tree. After that, we try to find the deepest marked node along the path from the root to the leaf node. If it exists, we take $d(l)$ as the length and the depth of l as $deep(l)$. From the property of the suffix tree, we have that $len(l) \leq deep(l)$ if and only if there exists a red sub-tree in the other tree with the same code of the blue sub-tree. Then we can get all common sub-trees if they satisfy the property.

3 EVALUATION

We chose HDF5 files and netCDF files as two typical examples of hierarchy organizations and flat organizations, respectively.

We collected three sets of real-world HDF5 files for evaluation from the following data sets: the National Snow and Ice Data Center (NSIDC) dataset, the Sea Ice Altimetry Data Center (SIADC) data set, and the Baryon Oscillation Spectroscopic Survey (BOSS) dataset. To prove the generality of our method, we also collected three sets of real-world netCDF files from the following data sets: 100 Stratospheric Aerosol and Gas Experiment (SAGE) files from Earth System Science Data, 100 Arctic sea-ice data (ASID) files from Pangea Earth and Environmental Science, and 100 Climate Projection (RCP) files from Hanze organizations.

For each base file, we calculate all similarity scores between it and any single file from these three sets of files. As shown in the poster, we calculated the similarity score between any two files and recorded the average percentage on each base file. The experimental results show that the average similarity score was higher if the files came from the same organization than from other sets of files. This indicates that our method effectively differentiates some files from a set of files if they are original from the same organization or share similar structure and attribute keys. For example, if we select a NSIDC file as a base file, the average percentages from NSIDC files are much higher than SIADC and BOSS files. In addition, the experimental results prove the generality of our method, which means our methods can be applied in both hierarchical and flat structures.

In our evaluation, we expect an larger structure similarity score if the base file and testing files are from the same organization or used to describe similar observations or simulating data. It has been proved by our experimental results.

4 CONCLUSION AND FUTURE WORK

In this paper, we have presented the design details of a relationship catapulting and similarity determination method. Our method determines structure similarity by finding common sub-trees among trees and builds relationships with these files. We demonstrate the potential of this approach with an evaluation of real-world scientific files. The experimental results demonstrate that our method can differentiate some files from a set of files by its data structure and attribute key and the relationship capturing process is efficient. We will continue improving the capturing efficiency by optimizing the common sub-trees' determination and better storage for linked scientific files. Our goal is to search and discover scientific files more efficiently in the HPC community.

REFERENCES

[1] 2020. http://www.sdss3.org/surveys.

[2] Francesc Alted, Martin Durant, Stephan Hoyer, John Kirkham, Alistair Miles, Mamy Ratsimbazafy, Matthew Rocklin, Vincent Schut, Anthony Scopatz, and Prakhar Goel. 2018. Zarr. https://zarr.readthedocs.io.
[3] R. Brun and F. Rademakers. 1997. ROOT: An Object Oriented Data Analysis Framework. *Nucl. Instrum. Meth.* A389 (1997), 81–86. https://doi.org/10.1016/S0168-9002(97)00048-X
[4] Mike Folk, Albert Cheng, and Kim Yates. 1999. HDF5: A File Format and I/O Library for High Performance Computing Applications. In *Proceedings of supercomputing*, Vol. 99. 5–33.
[5] P Greenfield, M Droettboom, and E Bray. 2015. ASDF: A New Data Format for Astronomy. *Astronomy and Computing* 12 (2015), 240–251.
[6] Jay F Lofstead, Scott Klasky, Karsten Schwan, Norbert Podhorszki, and Chen Jin. 2008. Flexible I/O and Integration for Scientific Codes through the Adaptable I/O System (ADIOS). In *Proceedings of the 6th international workshop on Challenges of large applications in distributed environments*. ACM, 15–24.
[7] Russ Rew and Glenn Davis. 1990. NetCDF: An Interface For Scientific Data Access. *IEEE computer graphics and applications* 10, 4 (1990), 76–82.
[8] Rob Latham, Rob Ross, Rajeev Thakur, Kui Gao, Alok Choudhary, Wei-keng Liao, Jianwei Li and Bill Gropp . 2010. Parallel NetCDF. http://trac.mcs.anl.gov/projects/parallel-netcdf.
[9] S Saalfeld. 2017. N5: Not HDF5. https://github.com/saalfeldlab/n5.
[10] Donald Carson Wells and Eric W Greisen. 1979. FITS: A Flexible Image Transport System. In *Image Processing in Astronomy*. 445.