Examining Scientific Data for Scalable Index Designs

Aleatha Parker-Wood Brian A. Madden Michael McThrow Darrell D. E. Long
Storage Systems Research Center, University of California, Santa Cruz
{*madden, aleatha, mmcthrow, darrell*}*@cs.ucsc.edu*

1 Introduction

The largest modern file systems contain billions of files. Faced with these kinds of volumes, manual file navigation and management is no longer feasible and users have turned to search as an alternate method of finding files.

Consider scientists working on a shared computing system. An astrophysicist might search for data files with a certain peak brightness. A biologist might search for files about a specific watershed area. These are semi-structured searches, and the scientist may even refer to them as metadata searches, but rather than relying on universally present system generated metadata, they rely on data that is domain specific, and embedded in content.

Scientific metadata can outstrip the data it describes. In many cases the line between data and metadata is blurry. What matters is whether the data is semi-structured, and queried. When we use the term *metadata* we are referring to POSIX and extended metadata as well as metadata embedded in file contents. We discuss *fields*, a single dimension of metadata such as temperature or author. And we refer to *items*, a single data object and its associated metadata fields.

Previous works in this field, such as Spyglass [2], SmartStore [1], Loris [6], and Pantheon [3] have focused entirely on testing with POSIX metadata. Rather than focusing on POSIX metadata as a surrogate for other metadata, we examine a variety of scientific metadata directly, in order to better understand the design space. We find that scientific metadata is unlike POSIX metadata, which is homogenous, low-dimensional, mostly numeric, and has no missing values. Scientific metadata is sparse, even within a single object type. It is heterogenous. It is high-dimensional. And it is a mix of numeric, textual, and categorical data. Approaches used by previous systems, such as spatial trees and row major databases, which are effective for POSIX metadata, will perform poorly when faced with the high dimensional, sparse nature of scientific metadata. In our current work [4], we suggest that column stores may be better suited.

1.1 Sparsity

Many indexing systems assume all fields are present for all items. In contrast, we find that even within a single discipline fields tend to be sparse, as shown in Figure 1. Spatial trees use estimation to fill in missing values, creating a large amount of spurious data [5]. A naive row-based index also wastes space. Even with tables for each data type, it must store a null for each missing value. A column store, which only stores data when data is present for that field, can index sparse data without any wasted space.

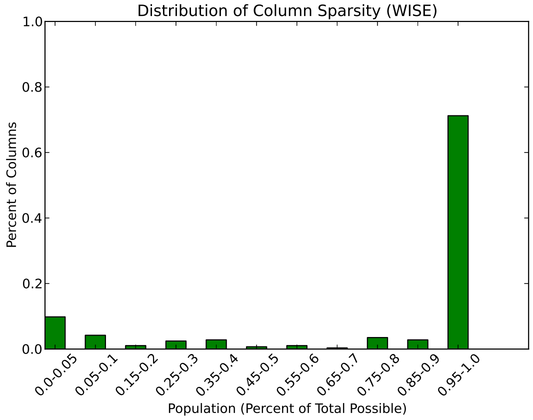

Figure 1: Data from the Wide-field Infrared Survey Explorer data is 20% sparse on average, over 285 fields.

1.2 Cardinality

Many indexing schemes assume a tabular, one to one relationship, where each field is present only once. In biology, we find many fields with high cardinality for a single item, as shown in Figure 2. For instance, biology data sets list every species seen during data collection. Any system which supports a variety of scientific metadata must handle many to one relationships. We also saw data with range values. Both multi-valued entries and range entries would thwart spatial tree approaches. With careful schema design an RDBMS can support many-to-one relationships and range values. Column stores can support high cardinality natively.

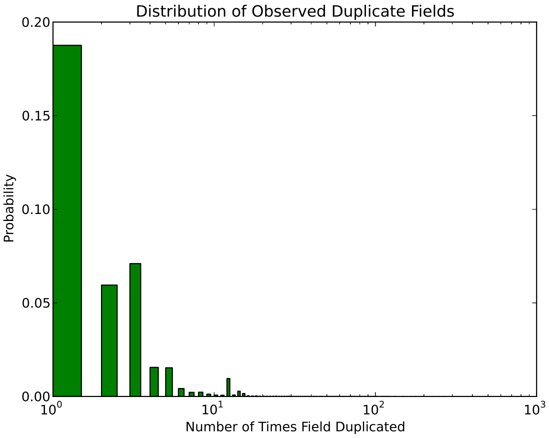

Figure 2: Cardinality of biology data, with log-scaled frequency. Only 18% of fields have a single entry.

1.3 Data Types

In Table 1, we show the distributions of raw data types and semantic types. Scientific data is a mix of numeric types, strings, and more specialized types such as dates and geospatial data. Spatial trees are excellent at indexing numeric data, but cannot index strings. Relational databases can do very well at indexing strings, and many databases have a variety of specialized features for efficient string searching. Some column stores index strings well.

Table 1: Data types in scientific data. We examine both storage types and semantic types that can have specialized indexes.

	Distribution (out of 345 fields)
Storage Type	18% strings, 82% numeric
Semantic Type	9% spatial, 4% dates, 16% flag sets, 71% native storage types

2 Design Implications

Examined together, our findings suggest that previous approaches to metadata indexing will not scale to scientific metadata. Spatial trees must fill in inferred values to index sparse data, and row based indexes must index nulls, wasting space. Spatial trees do not handle high arity data, and row based indexes require multiple tables and manual table designs. Finally, spatial indexes are not well suited for text fields. Previous approaches will degrade or fail when presented with scientific metadata, and our findings suggest column stores are a better choice.

References

[1] Hua, Y., Jiang, H., Zhu, Y., Feng, D., and Tian, L. SmartStore: A new metadata organization paradigm with semantic-awareness for next-generation file systems. In *Proceedings of SC09* (Nov. 2009).

[2] Leung, A., Shao, M., Bisson, T., Pasupathy, S., and Miller, E. L. Spyglass: Fast, scalable metadata search for large-scale storage systems. In *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST)* (Feb. 2009), pp. 153–166.

[3] Naps, J., Mokbel, M., and Du, D. Pantheon: Exascale file system search for scientific computing. In *Scientific and Statistical Database Management*, J. Bayard Cushing, J. French, and S. Bowers, Eds., vol. 6809 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2011, pp. 461–469.

[4] Parker-Wood, A., Madden, B., McThrow, M., and Long, D. D. E. Examining extended and scientific metadata for scalable index designs. Tech. Rep. UCSC-SSRC-12-07, University of California, Santa Cruz, Dec. 2012.

[5] Talbert, D. A., and Fisher, D. An empirical analysis of techniques for constructing and searching k-dimensional trees. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2000), KDD '00, ACM, pp. 26–33.

[6] van Heuven van Staereling, R., Appuswamy, R., van Moolenbroek, D., and Tanenbaum, A. Efficient, Modular Metadata Management with Loris. In *Networking, Architecture and Storage (NAS), 2011 6th IEEE International Conference on* (July 2011), pp. 278 –287.