# Examining Extended and Scientific Metadata for Scalable Index Designs

Technical Report UCSC-SSRC-12-07
December 2012


Aleatha Parker-Wood
aleatha@soe.ucsc.edu

Brian A. Madden
madden@soe.ucsc.edu

Michael McThrow
mmcthrow@soe.ucsc.edu

Darrell D.E. Long
darrell@soe.ucsc.edu

Storage Systems Research Center
Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
http://www.ssrc.ucsc.edu/

# Examining Extended and Scientific Metadata 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*

## Abstract

The sheer volume of modern data makes manual file management impractical. Search-oriented file systems, where data and metadata are indexed for fast search, are increasingly viewed as a necessity, everywhere from desktops to HPC. However, current techniques have been designed and tested for file system metadata, such as POSIX metadata, and fail to account for the wide variety of metadata users would like to search.

In particular, the scientific world has been vocal about a desire to search extended and content metadata. While file system metadata is well characterized by a variety of workload studies, scientific metadata is much less well understood. We characterize scientific metadata, in order to better understand the implications for index design. We demonstrate that previously suggested index structures, such as k-d trees, R-trees, and row major databases, are not well suited to scientific metadata. Finally, we provide suggestions for a system design based on our findings.

## 1 Introduction

The largest modern file systems contain billions of files, as many as the Web of only a few years ago. Faced with these kinds of volumes, manual file navigation and management is no longer feasible, and similarly to the Web, users have turned to search as an alternate method of finding files.

However, search means different things to different types of users. A system administrator might want to search for files in order to manage quotas and migrate files within storage hierarchies, something that can be done with POSIX and other system generated metadata. Users, on the other hand, may want to search files by size or age, but are more interested in searching metadata about content [26].

Consider scientists working on a shared computing system, such as is common in HPC. An astrophysicist might wish to look for data files with a certain peak brightness. A biologist might want files about a specific watershed area. And a geologist might want files where some set of minerals are present. Each of these searches is a metadata search, but rather than relying on universally present system generated metadata, it relies on metadata that is domain specific, and may be embedded in content.

Scientific metadata can easily outstrip the data it describes. In many cases the line between data and metadata is blurry. For instance, in astronomy a single dataset may contain raw data such as pixel brightness and location, and metadata such as a catalog identifier or a star name. In these instances file content and extended metadata are indistinguishable. In some sense, the difference is moot. What matters is whether the data can and will be queried, and how quickly it can be queried. When we use the term *metadata* we are referring to both the typical 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 [19], SmartStore [17], Loris [28], and Pantheon [21], even when purporting to provide good performance for extended metadata, have focused entirely on testing with POSIX metadata. Rather than focusing on POSIX metadata as a surrogate for other metadata, we examine scientific metadata directly, in order to better understand the design space of scientific metadata and content indexing systems. We find that scientific metadata is very unlike POSIX metadata, which is homogenous, low-dimensional, mostly numeric, and has no missing values. Scientific metadata can be very sparse, even within a single discipline and object type. It is heterogenous, with different fields for different disciplines. It is very high-dimensional. It is a mix of numeric, textual, and categorical data. And in the aggregate, it is large. We will demonstrate that approaches used by previous systems, which have proven highly effective for POSIX metadata, will perform poorly when faced with the high dimensional, sparse nature of scientific metadata.

One recent approach to indexing metadata is to use a spatial tree, such as a k-d tree [19], or an R-tree [17]. However, these trees have a number of limitations which can impair their ability to index metadata. First, they have poor performance on high dimensional data sets [8]. Second, they handle missing values poorly, or break when confronted with them [27]. Last, these structures are ill-suited for many-to-one data, where a single item has multiple values for a field.

A second popular approach is to simply throw everything into an external database and index it. However, this approach also has problems, for example, consistency issues arise when managing metadata outside of the file system. Additionally, if a naive schema is chosen, such as a single table in a traditional RDMBS, or single B-tree index [21, 15], not only will space be wasted by the indexing of null values, but the system will have difficulty indexing multi-valued fields. Many researchers have previously made compelling arguments for integrating search deeply into the file system [19, 24]. Integrated indexing reduces consistency issues, eliminates duplication of metadata, and can improve performance.

In this paper we characterize scientific metadata from a variety of different disciplines, and demonstrate that it is very unlike file system metadata. We provide enough information to create a realistic scientific metadata snapshot which can be used to design and test systems designed for scientific systems. And lastly, we describe issues with previous indexing systems, and provide guidelines for future indexing systems designed to handle such metadata.

## 2 Data and Experimental Design

In this section, we describe the data sets which we analyze, and the types of tests we apply. Our analysis is focused on the twin goals of narrowing index design space and making it possible to generate realistic synthetic data sets. In order to get a representative sample of scientific metadata, we selected a variety of different types of scientific data that are likely to be found on a large computing installation, drawing from biology, astronomy, and climate science. Our goal is to characterize what could be expected in a system where one or more of these data types is resident. While some scientific computing installations may handle only a single kind of data, the largest installations support a wide range of scientific research, and will have indexes which must support multiple types of scientific data.

### 2.1 Data sets

As representative samples of scientific data, we chose files from Dryad [2], the Wide-field Infrared Survey Explorer (WISE) All-Sky Release [5], and a historical data set of carbon-14 observations from Oakridge National Laboratories [16]. These files cover a wide range of scientific fields which one might find in a large computing installation, and have very different metadata characteristics.

Dryad is an online repository for biology data, designed for data sharing after papers are published. Researchers upload their data, along with metadata about it, to allow other researchers to examine the data and replicate results. We sampled approximately 31,000 dataset records. Each record had two associated collections of metadata in two XML formats, one in The Open Archives Initiative (*OAI*) Protocol for Metadata Harvesting (a Dublin Core vocabulary) [1], and one in the Metadata Encoding & Transmission Standard (*METS*) [4]. There was some overlap in the metadata, but there were a number of distinct fields available in each. We chose to use both to get full coverage, but analyzed them separately for clarity. The Dryad data was presented as 400 MB of XML data containing 44 unique fields; 14 of the fields came from OAI, and 30 from METS.

The WISE All-Sky Release is a NASA infrared digital imaging survey of the entire sky. It is available in its entirety as CSV data from Caltech. It contains a total of 285 unique fields. These are a mix of observation data, statistical analysis, and descriptive fields. The entire dataset consists of approximately 564 million records, or 1 TB of CSV data. For statistical purposes, we took a uniform random subsample of 10,000 records from the first part of the catalog, which is sufficient for fitting distributions.

The ORNL Historical C-14 dataset is a scientific data set consolidated by Oakridge National Laboratories, drawn from historical observations of oceanic carbon-14. It contains 14 unique fields. This was the smallest data set, containing 1478 records, and 154 kB of CSV data; we did not subsample.

### 2.2 Statistical Tests

To characterize the data, we look at the following aspects. First, we compare the frequency distribution of data values to common statistical distributions. Second, we examine the sparsity of fields. Third, we analyze the *arity* of fields, *i.e.* whether a field can be present multiple times in a single item. Fourth, we look at the overall distribution of data types for fields, both in terms of storage and semantic data types. Each of these tests allows us to reason about how large the data is, how compressible it is, and how much space a naive index can waste. This information is useful to guide choices when building a scientific data index or indexing system. It can also be used to generate synthetic data for testing index prototypes.

The first aspect we examine is frequency distributions. Frequency distributions are agnostic to data type, facilitating direct comparisons of string and numeric data. The frequency distribution of values within a field is useful when designing and testing indexing systems for performance. For instance, spatial trees can perform quite well on power law distributed data under certain circumstances, as noted by Leung [19]. Frequency distributions also impact the compressibility of indexes. Long-tailed data compresses well, while uniform is less compressible.

After visual inspection of frequency distributions, we select two distributions as being most representative of the data. These are the uniform distribution, and the Zipfian, or power law, distribution. We then apply goodness of fit algorithms appropriate for the distribution (Clauset's method [12] for power law, Anderson-Darling for the uniform distribution) to verify our visual intuition, and select those with negative log likelihood as a match.

Second, we examine sparsity. Sparsity of data is critical to understand when designing indexes, as very sparse data can impact the size and behavior of indexes. For instance, spatial trees such as k-d trees are not designed to deal with missing values. Row based indexing can handle missing values, but must waste space to store them. We determine the total number of fields possible for any item, and then calculate what number of those fields were actually present in each of the individual items.

Third, we examine arity. Arity, or how often a field can be present in an item, is important in an index choice. For instance, tabular data can only support one value per column in a row. Depending on the schema of the data, a field may be present zero, one, or multiple times. The data may represent an array of items, such as a list of authors, or a range, such as a geographic area. Data with an arity higher than one requires an index design that can represent multiple values or ranges, and match them to queries. We determine the distribution of arity in data sets which support more than one value in a column.

Fourth, we examine the data types of the fields, based on the data set documentation. The distribution of field types is useful for index choices, benchmarking and testing. Different index designs are better for text versus numeric data, and there exist specialized indexes for data which is geospatial or time based. However, semantic types such as dates or latitudes cannot be easily distinguished from integers and require human intervention to discover. Knowledge about these can characterize how much human time is required to maintain indexes. To examine type distributions, we initially categorize the fields as numeric or string data. We then further examine them manually to determine if they are geospatial, categorical, free text, and so on.

# 3 Results and Analysis

Here, we describe our findings, and discuss their implications for index design. We describe the statistical distributions of the data, the sparsity of the data, the arity, and the distribution of types. In conjunction with one another, these findings can be used to create realistic benchmarks, and to make informed decisions when building indexing systems.

## 3.1 Distributions

When examining the distribution of value frequencies, we find that by far the majority of fields were power law distributed, as shown in Table 1. This suggests that indexes and compression techniques optimized for long tailed data will be most effective for scientific data sets. Very few distributions do not fit either a power law or uniform distribution. In Figure 1 we show the average distribution of the power law data frequencies, with error bars to demonstrate variance. However, a handful of fields do not fit any known statistical distribution. These generally have very few values, so might have a recognizable distribution with more data.

**Table 1:** Frequency distributions in scientific data by percentage. Zipfian data will compress well if stored column-wise. Some data did not fit a known distribution.

| Distribution | WISE | ORNL | OAI | METS | Total |
|---|---|---|---|---|---|
| Power law | 83% | 93% | 87% | 77% | 83% |
| Uniform | 5% | 7% | 0% | 20% | 6% |
| Unrecognized | 12% | 0% | 13% | 3% | 11% |

## 3.2 Sparsity

Many indexing systems assume a fixed schema, where all fields are present for all items. In contrast, we find that even within a single discipline fields tend to be sparse. Even in the WISE data set, which is primarily observational data, on average 20% of fields are missing from objects, as shown in Figure 4. The ORNL C-14 data set is similar, with about 20% of fields missing, as shown in Figure 5. Dryad is the most sparse. In Figure 3, we show that over half of fields are blank for any given object in the METS metadata. Only a few fields are present in all items, and these fields tend to be unique identifiers for

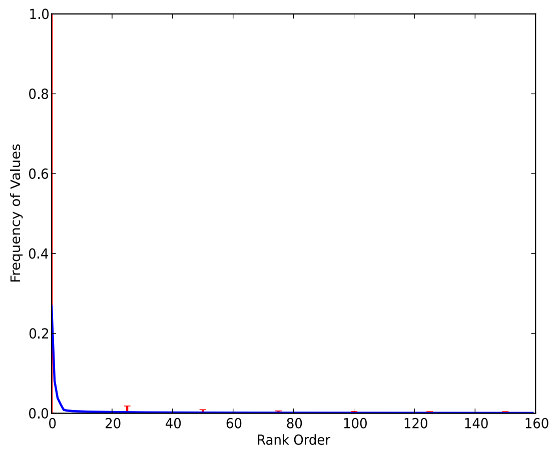


**Figure 1:** To show goodness of fit, we take the average of all the power law distributions, then overlay error bars to show the variance between maximum and minimum values. The variance is quite low, except for the beginning of the distribution. This graph has been truncated due to the long tail, but the variance is minimal after this point.

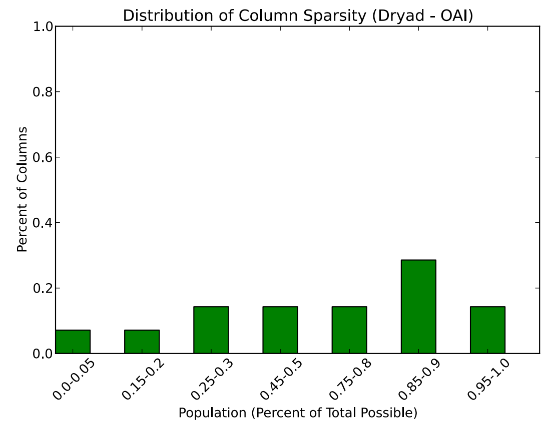


**Figure 2:** Dryad OAI is over 25% sparse on average, out of 16 fields.

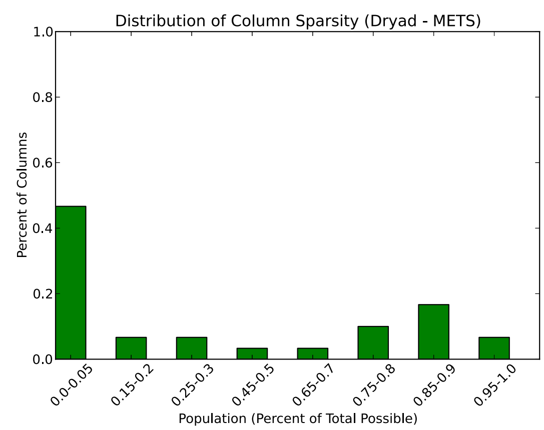


**Figure 3:** Dryad METS is over 50% sparse on average, out of 30 fields.

the system's use. Of the Dryad metadata, OAI is slightly less sparse, at 25%.

Sparsity is challenging for spatial tree based indexing schemes, which handle missing values poorly, if at all. Most spatial trees cannot place data with missing values in the tree, and must use estimation techniques to fill in missing values, creating a large amount of fake data that must then be stored [27]. A naive row-based index, such as most databases default to, will also have space implications. Even if tables are built separately for each data type, it will still need to store a null for each missing value [11].

A column store, which only stores data when data is present for that field, can index sparse data without any wasted space. Due to the sparse nature of the data we examine, we believe that a column store would make a better choice as storage substrate than traditional row stores when designing an indexing system for scientific metadata.

## 3.3 Arity

Many indexing schemes assume a tabular, one to one relationship, where each field is present only once. This holds true for astronomy data, but in biology, we find many fields with high cardinality for a single item, as shown in Figure 6. For instance, biology data sets list every author on the paper, and often list every species seen during data collection. One item in our data has over eight hundred species. In addition, we find that one of the fields in the ORNL metadata is numerical data, but mixes ranges and point values. Any system which supports a variety of scientific metadata must handle many to one relationships, and should have a strategy for dealing with range values. Both multi-valued entries and range entries would thwart most spatial tree or naive row based approaches, which expect point values. With careful schema design an RDBMS can support many-to-one relationships and range values, but still suffers from the sparse nature of the data, and requires a human curator if schema updates or optimizations are necessary. Column stores can support high arity natively, and are a better choice.

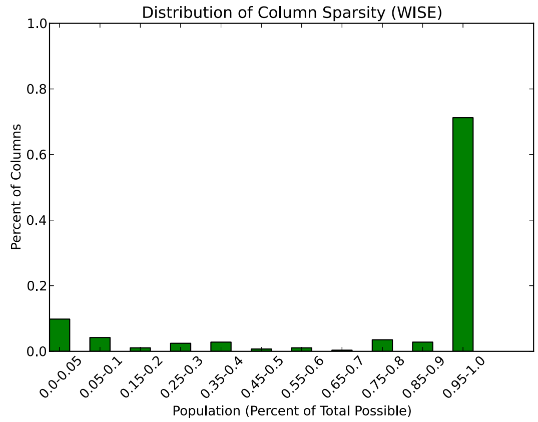


**Figure 4:** WISE is 20% sparse on average, out of 285 fields, but that sparsity is concentrated in a few fields.

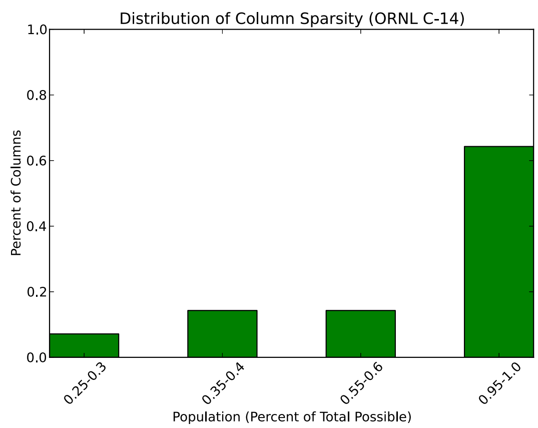


**Figure 5:** ORNL C-14 is 20% sparse on average out of 14 fields.

## 3.4 Data Types

The majority of biology data consists of categorical and free text data. The only numeric field is a date field. However, some of the free text fields describe geospatial locations or ranges. Astronomy data is dominated by numeric data, but some data describes spatial locations, or consists of a set of flags encoded as a number. Some flag sets are also encoded as strings. The ORNL C-14 data is a fairly even mix of strings and numeric data, with some geospatial data, and a date field encoded as a string. In aggregate, this suggests that having native index support for time and space can significantly speed up queries, but knowledge of the data format is needed, requiring human intervention. In Table 2, we show the distributions of raw data types and semantic types.

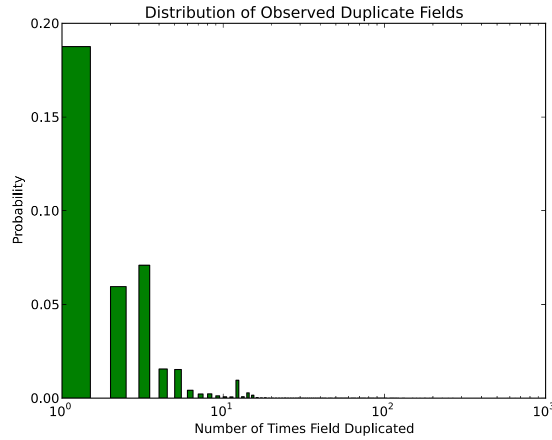


**Figure 6:** Dryad arity, with log-scaled frequency. Only 18% of Dryad fields have a single entry. The majority of fields have multiple entries, running as high as eight hundred.

**Table 2:** 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 |

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. Neither approach takes advantage of the long tailed nature of data values to offer index compression, and spatial indexes are not well suited for text fields. In addition to showing that previous approaches will degrade or fail when presented with scientific metadata, our findings suggest column stores are a better choice of storage substrate for sparse, long tailed, and high arity data.

# 4 Related Work

Here we describe related work in the areas of file system indexing, indexing for sparse and semi-structured data, and metadata studies.

## 4.1 File System Indexing

File system indexing is a wide field. Here we focus specifically on systems for searching metadata, rather than text search. Inversion [22] was the first system to propose integrating indexes into the file system, and focused on both system and user-supplied metadata. They proposed replacing the file system with a POSTGRES database, and defining tables for user-supplied data types. This ignores sparsity within a given data type, and the many to one nature of metadata fields.

Spyglass [19] and Smartstore [17], were the first to suggest using spatial indexes for metadata. While these performed well on their test data, they focused strictly on POSIX metadata.

Loris [28] and Pantheon [21] were both indexing systems tested for system metadata only. Pantheon used B-trees, which are row-based, and will face challenges with sparse data. Loris used log-structured merge-trees [23].

BeFS [15] was designed to handle both system metadata and extended metadata. In BeFS, all metadata was stored in a B+-tree, using row-major order. This technique was effective at desktop scales, but it suffers from problems with sparse, heterogeneous data.

### 4.2 Metadata studies

There have been a number of previous metadata studies. However, they have focused exclusively on file system metadata and file types, rather than scientific metadata. For instance, Douceur's large-scale study of file-system contents [14], and Agrawal's five-year study of file-system metadata [7], also did detailed statistical analysis of distributions. Both focused on desktops. Impressions [6] extended Agrawal's work, and used it to generate realistic file system workloads. Our data can be used with tools like Impressions to benchmark indexing using a realistic scientific file system.

On a larger scale, we note Leung's large scale network file system study [20], which tracked behavior and file system metadata for corporate file servers. Perhaps the closest to our research are Dayal's study of HPC at rest [13] and Wang's study of HPC workloads [29]. However, they focused on file system metadata.

### 4.3 Other indexing

Indexing shares many challenges with databases as well as file systems. Column stores such as C-store [25], HBase [3], or Cassandra [18], are one popular approach to dealing with sparse data. These have some advantages for scientific data, since they are well organized for tasks such as computing maximums, minimums, and averages.

WideTable [11] was specifically designed to meet the challenges of extremely sparse high-dimensional indexes. Like us, they note the high prevalence of Zipfian and long tailed distributions in sparse data.

Patil et al. [24] also explored the question of appropriate architectures for searchable metadata in file systems. They suggested using BigTable [10] as the underlying storage for a file system. BigTable has good support for sparse indexes, and is highly scalable.

## 5 A Scalable Architecture

Based on our findings, we believe that a column store has many advantages for indexing scientific metadata, and that the scale of scientific metadata lends itself to hierarchical index caching strategies. We propose an architecture that uses a column store as a basis for a *lazy index*. In a lazy index, fields are stored, unprocessed, embedded in the data until the first query is issued. Once a query has been issued, a simple index is built, to facilitate faster querying, using a parallel technique such as SciHadoop [9], and stored on disk. As demand goes up further, the index is optimized, and consolidated with co-queried fields. In the near future, we intend to explore the implications of our findings in a fully fledged indexing system which incorporates file system metadata, scientific metadata, and content.

## 6 Conclusions

Scalable searchable file systems will be crucial as users demand the ability to search ever more of their data, and understanding the characteristics of data can assist in making good design decisions. In this paper, we have examined scientific metadata, and demonstrated that it is sparse, heterogenous, long tailed, and high dimensional. Based on our findings, existing approaches to file system indexing, such as spatial trees and row major databases, will perform poorly for indexing scientific metadata. We use our findings to inform a new architecture based on column stores, which uses lazy indexing to save space while still offering acceptable query performance.

## Acknowledgements

This work has been supported by grants from NASA, the DoE, and the NSF. We also thank our industrial sponsors.

This publication makes use of data products from Oakridge National Laboratories, Dryad, and the Wide-field Infrared Survey Explorer, which is a joint project of the University of California, Los Angeles, and the Jet Propulsion Laboratory/California Institute of Technology, funded by the National Aeronautics and Space Administration.

## References


[1] The Open Archives Initiative Protocol for Metadata Harvesting. `http://www.openarchives.org/OAI/openarchivesprotocol.html`.

[2] Dryad. http://www.datadryad.org/, September 2012.

[3] Hbase. http://hbase.apache.org/, September 2012.

[4] Metadata Encoding & Transmission Standard. http://www.loc.gov/standards/mets/, November 2012.

[5] Wide-field Infrared Survey Explorer (WISE) All-Sky Release. http://irsadist.ipac.caltech.edu/wise-allsky/, September 2012.

[6] AGRAWAL, N., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. Generating realistic *impressions* for file-system benchmarking. In *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST)* (Feb. 2009), pp. 125–138.

[7] AGRAWAL, N., BOLOSKY, W. J., DOUCEUR, J. R., AND LORCH, J. R. A five-year study of file-system metadata. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)* (Feb. 2007), pp. 31–45.

[8] BERCHTOLD, S., BÖHM, C., KEIM, D. A., AND KRIEGEL, H.-P. A cost model for nearest neighbor search in high-dimensional data space. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Sytems* (New York, NY, USA, 1997), PODS '97, ACM, pp. 78–86.

[9] BUCK, J. B., WATKINS, N., LEFEVRE, J., IOANNIDOU, K., MALTZAHN, C., POLYZOTIS, N., AND BRANDT, S. SciHadoop: array-based query processing in Hadoop. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis* (New York, NY, USA, 2011), SC '11, ACM, pp. 66:1–66:11.

[10] CHANG, F., DEAN, J., GHEMAWAT, S., HSIEH, W. C., WALLACH, D. A., BURROWS, M., CHANDRA, T., FIKES, A., AND GRUBER, R. E. Bigtable: A distributed storage system for structured data. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)* (Seattle, WA, Nov. 2006).

[11] CHU, E., BECKMANN, J., AND NAUGHTON, J. The case for a wide-table approach to manage sparse relational data sets. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 2007), SIGMOD '07, ACM, pp. 821–832.

[12] CLAUSET, A., SHALIZI, C. R., AND NEWMAN, M. E. J. Power-Law Distributions in Empirical Data. *SIAM Review 51*, 4 (Nov. 2009), 661–703.

[13] DAYAL, S. Characterizing HEC storage systems at rest. Tech. rep., Carnegie-Mellon University, 2008.

[14] DOUCEUR, J. R., AND BOLOSKY, W. J. A large-scale study of file-system contents. In *Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (1999), SIGMETRICS '99.

[15] GIAMPAOLO, D. *Practical File System Design with the Be File Sstem*, 1st ed. Morgan Kaufmann, 1999.

[16] GRAVEN, H., KOZYR, A., AND KEY., R. M. Historical observations of oceanic radiocarbon conducted prior to GEOSECS. http://cdiac.ornl.gov/ftp/oceans/Historical_C14_obs/, 2012.

[17] 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).

[18] LAKSHMAN, A., AND MALIK, P. Cassandra: a decentralized structured storage system. *SIGOPS Oper. Syst. Rev. 44*, 2 (Apr. 2010), 35–40.

[19] 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.

[20] LEUNG, A. W., PASUPATHY, S., GOODSON, G., AND MILLER, E. L. Measurement and analysis of large-scale network file system workloads. In *Proceedings of the 2008 USENIX Annual Technical Conference* (June 2008).

[21] 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.

[22] OLSON, M. A. The design and implementation of the Inversion file system. In *Proceedings of the Winter 1993 USENIX Technical Conference* (San Diego, California, USA, Jan. 1993), pp. 205–217.

[23] O'NEIL, P., CHENG, E., GAWLICK, D., AND O'NEIL, E. The log-structured merge-tree (LSM-tree). *Acta Inf. 33*, 4 (June 1996), 351–385.

[24] PATIL, S., GIBSON, G. A., GANGER, G. R., LOPEZ, J., POLTE, M., TANTISIROJ, W., AND XIAO, L. In search of an API for scalable file sys-

tems: under the table or above it? In *Proceedings of the 2009 conference on Hot topics in cloud computing* (Berkeley, CA, USA, 2009), HotCloud'09, USENIX Association.

[25] STONEBRAKER, M., ABADI, D., BATKIN, A., CHEN, X., CHERNIACK, M., FERREIRA, M., LAU, E., LIN, A., MADDEN, S., O'NEIL, E., O'NEIL, P., RASIN, A., TRAN, N., AND ZDONIK, S. C-Store: A column oriented DBMS. In *Proceedings of the 31st Conference on Very Large Databases (VLDB)* (Trondheim, Norway, 2005), pp. 553–564.

[26] STRONG, C., JONES, S., PARKER-WOOD, A., HOLLOWAY, A., AND LONG, D. D. E. Los Alamos National Laboratory interviews. Tech. Rep. UCSC-SSRC-11-06, University of California, Santa Cruz, Sept. 2011.

[27] 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.

[28] 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.

[29] WANG, F., XIN, Q., HONG, B., BRANDT, S. A., MILLER, E. L., LONG, D. D. E., AND MCLARTY, T. T. File system workload analysis for large scale scientific computing applications. In *Proceedings of the 21st IEEE / 12th NASA Goddard Conference on Mass Storage Systems and Technologies* (College Park, MD, Apr. 2004), pp. 139–152.