Easing the Burdens of HPC File Management

Stephanie N. Jones
snjones@cs.ucsc.edu

Christina R. Strong
crstrong@cs.ucsc.edu

Aleatha Parker-Wood
aleatha@cs.ucsc.edu

Alexandra Holloway
fire@soe.ucsc.edu

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

Storage Systems Research Center & Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064

ABSTRACT

While the amount of data we can process and store grows, our ability to find data remains dependent upon our own memories more often than not. Manual metadata management is common among scientific users, consuming their time while not making use of the computing resources at hand. Our system design proposes to empower users with more powerful data finding tools, such as unified search spaces, provenance, and ranked file system search. By returning the responsibility of file management to the file system, we enable scientists to focus on their science without the need for a customized file organization scheme for their work.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms

Human Factors

Keywords

indexing; naming; provenance; ranking

1. INTRODUCTION

We recently had the opportunity to meet with scientists at both Los Alamos National Laboratory (LANL) as well as Pacific Northwest National Laboratory (PNNL). While at LANL we were fortunate enough to also meet a scientist working at the National Metacenter for High Performance Computing (NOTUR) in Norway. We talked with them about the science they do and how they use supercomputers. As the discussions went on, we noticed a trend: the scientists were bookkeeping their data in lieu of using the file system. It wasn't that the file system didn't work; each scientist had his or her own idea of how their data should be organized and why the file system couldn't accommodate them.

Almost every scientist we talked to kept a notebook, electronic or paper, to record additional metadata about their work that wasn't expressed or collected by the file system. The metadata varied based on the scientist and science being performed. Some scientists recorded the workflow of their experiments, while others use their notebooks to remind them what each of their codes does and where they put them.

By keeping their own notebooks to record additional metadata, scientists have set themselves up as an invisible and vulnerable part of the metadata management system. If they need to find related experiments or results, they must search their notebooks. Even when scientists know that related data exists, it is difficult to find using only the file system. Instead, they rely on manual bookkeeping that only they can find or understand, in order to manage their data and understand relationships.

We propose a provenance enabled file system, which will automatically record relationships among files. This provenance information will be integrated with traditional metadata to create a unified search space over both primary and archival storage. The integrated provenance and metadata information will be used to help create descriptive names, relieving scientists from the inconvenience of naming and remembering names. Queries issued over the search space will return a list of results ranked in order of importance to the scientist. Such a system reduces the burden of management on the scientist by automatically relating files and facilitating finding data that is important to him. It returns the task of data management to the file system, allowing the system to work for the scientists in a way that is more efficient for both system and scientist.

2. SCIENCE IN HPC

LANL uses a Network File System (NFS) [19] for "smaller files" that are written sequentially, and contains home directories and project directories. The Panasas Parallel File System (PFS) [28] is shared among supercomputers and used for scratch space, meant for large files and parallel writes. For archival storage, LANL mainly uses HPSS [26], but are looking at transitioning to a GPFS [20] solution.

At PNNL, the major supercomputer's user base is mainly comprised of external users. There are multiple smaller clus-

PDSW'11, November 13, 2011, Seattle, Washington, USA.

Number of Scientists	Type of Metadata Management System
IIII	text file
I	PowerPoint presentation (with embedded visualization)
III	long, complex directory names
I	spreadsheet (with embedded visualization)
II	encoded in the input deck
I	printed out (in binders)

Table 1: Types of personal metadata management systems practiced by scientists.

ters, which are used primarily for analysis. All the clusters run Lustre [5], and PNNL uses HPSS for archival storage as well. NOTUR consists of several different supercomputers in various places. Similar to LANL, the scratch space is a parallel file system while the home directories are on NFS.

While the intended usage of the file systems is that large files that are written in parallel are kept on PFS and smaller, sequentially written files are stored on NFS, this is not always the case. Some scientists use the systems the way they were intended to be used, using PFS for storing checkpoint files and NFS for everything else. Other scientists keep everything in PFS because it is easier. Still others keep the static files on NFS and everything else on PFS.

This can have serious impacts on the performance of the file systems, as they struggle to manage loads they were not designed to handle. In turn, this also can have serious impacts on the performance of the scientists. Since the file system is not meeting the needs of the scientists, they have turned to their own methods of managing information.

2.1 File Organization Methods

In our discussions, we came across many different ways of keeping track of what file was placed where. Many scientists keep a separate file to keep track of what calculation is being run on which machine and the status of that calculation. Some of these are kept by hand; one scientist uses the Mac Stickies application on his personal computer as an electronic notebook. Other scientists have an ASCII file written to their home directory by the code, which contains information such as the time, the file size, and the path to the output. Table 1 shows the different ways scientists have developed to manage their data. There is often redundancy in the schema; the scientist who uses PowerPoint also encodes the information in the input deck.

These data management solutions have been created because the scientists feel the file system is lacking. For some scientists, this is because the information they want recorded is not gathered by the file system. For most, however, managing their own data has become easier than finding it again. While long, complex names are descriptive, short names are faster and easier to type, and maintaining a record allows scientists to relate files based on their own classifications. However, this results in scientists working part-time as scientists and part-time as data managers—something the system could and should do very well.

2.2 An Ideal System

Scientists are acutely aware of time lost on data management. When asked what an ideal file system would be, most discussions revolved around finding their data once it had been stored, and reducing dependency on separate notebook files. Despite ranging from power users to users who viewed the file system as a black box, everyone had an idea of how they would like to be able to use the file system.

Many of the scientists didn't want to worry about *where* their data is being stored; they don't want to have to think about whether it should be written to the parallel file system, the network file system, or to archive. They really only care that their data *is* being stored and that they can retrieve it when needed. This also alleviates the need to remember where the file is stored when looking for it again. One person mentioned that she would like to never have to execute a `du` or `find`. She would like to be able to say "Show me all files related to Experiment A" and the result would be a list of all files related to Experiment A regardless of where they were stored.

The type of queries varied by discipline, but every scientist wanted to be able to query. To some scientists, a file is really just coordinates and the characteristics of those coordinates; to these scientists being able to find subsets of files such as retrieving coordinates with specific temperatures is desirable. Another scientist runs code which stores metadata information in a header of every restart file; he would like to be able to limit a search to just the header of the restart files.

In order to be able to maintain their own classifications, some scientists wanted the ability to create their own tags. Not only did they want the ability to search using tags, they also wanted to be able to query over the values of the tags themselves for more expressive searches. Other scientists, however, wanted the correlations between files to be created automatically. For instance, when a visualization file is created, they want it to be automatically correlated with related files. Since the visualization file is often the first (or only) thing they remember about an experiment, they want the ability to query the system for information related to that visualization file.

In light of these interviews, we believe that better support for search at the file system level can improve matters for both scientists and the file system. By creating a unified search space, with the necessary query capabilities, fewer machine cycles will be spent using slow, brute force tools like `grep` and `find`. Furthermore, scientists will no longer need to manually manage data outside the file system, with all the time and risks that incurs, in order to find their files.

3. BACKGROUND

One way to define the relationship between files is a file's *lineage*. For example, an output file is a descendant of an input deck and the application that created it. Output files used to generate a visualization file are the ancestors to the visualization file. In computer science, *data provenance* refers to capturing and expressing the lineage of files in a system. In addition to providing new query possibilities in its own right, provenance also allows a richer analysis of file system access patterns at the semantic level. This has already proven effective at improving retrieval of related files in a desktop context [21].

There are two types of data provenance: content-based provenance and information-flow provenance. Content-based provenance records the history of the changes made to the contents of a file [18]. It records additional metadata information, including who made the change to a file and when the change was made. Information-flow provenance records the files and processes used to create the current version of a file [16].

Additionally, a new type of provenance was introduced by Jones et al. [14], called *transient provenance*. Transient provenance keeps track of data moving off a provenance aware system by creating a *ghost object*. This ghost object represents the period of time during which a user could have removed data from the provenance system. The inputs to the ghost object are all the files that the user read during that period of time. When the connection ends, you have a record of all the files read by the user from off the system.

Scientists have the ability to track this kind of information themselves, using applications such as Taverna [13], Kepler [2], and VisTrails [6]. These tools allow scientists to create and execute their workflow via a graphical user interface. The downside is that these tools all track *prospective provenance*, the steps that need to be followed for the workflow to generate information correctly [7]. This means the scientists are responsible for correctly recording their own provenance before the workflow is actually executed. Our proposed system automatically collects the workflow provenance as the workflow is running.

Once the provenance information has been collected, the issue becomes finding that information. One way we are working to improve search is through ranking the search results. Attempts to do file system ranking have lagged behind the web, even though the largest scientific file systems contain billions of files, within an order of magnitude of the web.

File system ranking has focused primarily on single-user file systems, and has been only moderately effective. One reason for this is the lack of metadata that is comparable to the rich link structure of the web. Most file system ranking has relied on existing metadata, such as access times [9]. Others have attempted to create new metadata by inferring links [21, 4, 23], or having users insert them manually [3].

There are other solutions for HPC search, such as the XQuery solution for HPSS [15]. However, the authors note that it is a bolt-on solution, and not as high performance or scalable as a dedicated search solution.

4. PROVENANCE IN HPC

We propose a provenance enabled file system, which automatically collects information-flow and transient provenance. Information-flow provenance will automatically create and track relationships among files, allowing a visualization file to be related to the input deck used to create it as well as the calculation that was run. Transient provenance will be used to track files as they migrate to archival storage, maintaining a record of the archive in primary storage. The provenance information will be integrated with traditional metadata information, creating a unified search space. Currently there is no way to issue a query over both the metadata and provenance information; we will integrate them so that any search query will consider both provenance and traditional metadata.

Having provenance to automatically track file relationships will help free scientists from having to track these relationships themselves. The ability to query over these relationships as well as traditional metadata provides scientists a way to find experiments with a common feature with ease, and the unified search space which abstracts where the file is physically located means they will find experiments regardless of where they are stored. Creating descriptive file names using the provenance and metadata information allows us to present the scientist with pertinent information about the file in the name, and by ranking the results by order of importance to them we can increase the speed with which scientists find the information they need to continue doing science.

4.1 Provenance Enabled File System

Currently, there is no implementation of a provenance enabled file system in the HPC community. The main focus of provenance research for scientific computing is in grid computing. Within grid computing, data provenance collection is varied. The system can be generic and useful for all science disciplines, but requires applications to explicitly state provenance information [8]. Conversely, the system can be specific to a science discipline and collect provenance automatically [22].

Instead of creating another provenance collection system from scratch, we will select an existing implementation to generate provenance for our provenance enabled file system. The solutions presented in this paper assume that a provenance collection system exists and is providing provenance information.

4.2 Storing Provenance

Despite the variety of data provenance collection systems, the common method for storing data provenance is in a relational database. Relational databases are chosen because they provide an inherent query language that can be used to access data provenance once it is stored. Using a relational database is fine for proof-of-concept work, but should not be used for a full-scale implementation.

Many in the database community have argued and shown [10, 24] that using a traditional DBMS for a variety of search and indexing applications is often a poor solution and that a customized, application-specific design that considers the technology and workload requirements of the specific problem can often significantly outperform a general-purpose DBMS. Similarly, as evidenced by their contrasting designs, databases are often optimized for either read or write workloads and have difficulty doing both well [1, 11, 12].

We are exploring two ways to create one combined metadata repository in the Ceph file system [27]: including a link to the provenance database as an extended attribute in the i-node and including a piece of the provenance in the i-node. We believe the second option will be the most successful, and are exploring ways to determine which piece of provenance would be most beneficial. We will compare these methods with the current method of using a separate database for provenance storage.

4.3 Unified Search Space

In order to create a unified search space in HPC two things must be true: a query must be able to access all types of metadata information, which we discussed above, and it must span all types of storage. Remembering where you

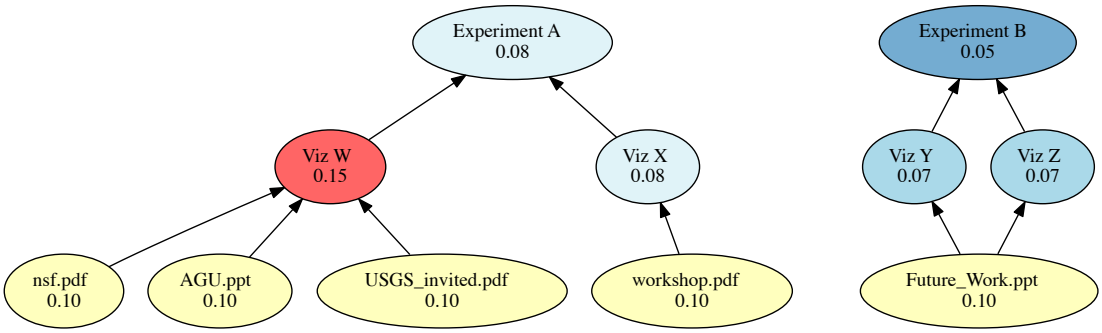

Figure 1: Provenance ranking favors newer popular files, versus the roots, which are often ubiquitous applications like gcc.

put your data is hard enough without having to remember which storage system the data is on. At LANL, for example, there are three types of storage (PFS, NFS, and HPSS), and no way to query across all three. If a scientist has forgotten where his data is, he must query each system individually, resulting in situations like the scientist who keeps everything on PFS, which is not backed up, because it is "easier" [25].

We are looking into using transient provenance [14] to create a record of archived data, retaining the metadata information on the primary storage system. Any query over the provenance will include the transient provenance, and therefore will include information about the archived data as well. We are starting with including the archival storage system in the primary storage system searches; incorporating the parallel file system is future work.

4.4 File Naming

As we have seen, scientific users often use file names to express every possible attribute that applies to a file, or create directory hierarchies many layers deep with a single file at the leaf levels, in order to allow them to disambiguate their files. Rather than forcing the scientist to create and remember file names, we will use the unified set of metadata and provenance information to create file names which are a reflection of what distinguishes a given file. When a query is issued, we will use this information to create a list of expressively named results.

Through statistical analysis of metadata, and techniques derived from faceted browsing, we can determine what attributes distinguish files, as well as which are most meaningful to scientists. At query time, we can select items which are more likely to identify the file, or we can choose meaningful attributes at file creation time. Our preliminary studies suggest that attributes which follow a power law distribution are more likely to convey meaning to the scientist. By choosing the most distinguishing meaningful attributes, we can present a file name which serves its intended purpose: to allow the scientist to identify a file.

4.5 Ranked Search Results

Search results will be ranked by importance, using provenance based ranking algorithms. The provenance graph forms a link structure similar to that of the web. Like web links, provenance allows us to examine what files scientists think are useful, by examining which data and codes are worth deriving from. To do ranking, the provenance graph will be analyzed using eigenvector analysis similar to PageRank [17]. However, naively applying PageRank to a provenance graph simply results in ranking frequently used roots (such as `gcc`) as most important. Instead, by modifying the PageRank transition function, using weighting based on the distance from the provenance leaves, we can favor newer, less ubiquitous, but still frequently used files, as shown in Figure 1.

5. CONCLUSIONS

After discussions with a variety of users, ranging from scientists to developers, we learned that everyone has his or her own system for organization. While their improvisation is admirable, their organizational schemes are not optimized for a file system to manage. For example, creating files with lengthy and cryptic filenames to annotate additional metadata not present makes collaboration difficult, if not impossible.

Scalable search and file management are open research problems. Currently, scientific users are manually organizing directories, naming files, and relying on single-threaded tools such as `grep` to find pertinent files. These techniques are increasingly ineffective, and a point of frustration with scientists [25].

To address their frustrations, we are developing a provenance enabled file system which automatically collects information flow provenance at the file system level. This provenance information will be used to create a unified search space, across both primary and archival storage. When a query is issued, it will consider both traditional metadata as well as provenance information, and a set of results will be created. These results will go through analysis to create a list of results ranked in order of importance to the scientist.

Our system directly addresses the issues identified by scientists related to finding data. The provenance information itself creates correlations among files, automatically relating files created with similar inputs or producing similar outputs. The unified search space allows scientists to find files regardless of where in the system they are stored, an

ability many scientists were interested in having. By identifying which files are more important to a scientist through ranking query results, we can increase the speed with which scientists find the data they are looking for. Our research proposes to improve on the state of the art by freeing scientists from file management, and allowing them to focus on what really matters: science.

Acknowledgements

We would like to thank all the scientists at LANL and PNNL who took time out of their busy schedules to talk with us. And special thanks to Meghan Wingate McClelland, who helped organize the LANL interviews, and John Johnson, who helped organize the PNNL interviews.

This material is based upon work supported in part by: the Department of Energy under Award Number DE-FC02-10ER26017/DE-SC0005417, the Department of Energy's Petascale Data Storage Institute (PDSI) under Award Number DE-FC02-06ER25768, and the National Science Foundation under awards CCF-0937938 and IIP-0934401 (I/UCRC Center for Research in Intelligent Storage).

6. REFERENCES

[1] D. J. Abadi, S. R. Madden, and N. Hachem. Column-stores vs row-stores: How different are they really? June 2008.

[2] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludäscher, and S. Mock. Kepler: An extensible system for design and execution of scientific workflows. 2004.

[3] S. Ames, N. Bobb, S. A. Brandt, A. Hiatt, C. Maltzahn, E. L. Miller, A. Neeman, and D. Tuteja. Richer file system metadata using links and attributes. In *Proceedings of MSST 2005*, Apr. 2005.

[4] D. Bhagwat and N. Polyzotis. Searching a file system using inferred semantic links. In *Proceedings of HYPERTEXT '05*, 2005.

[5] P. J. Braam. The Lustre storage architecture. http://www.lustre.org/documentation.html, 2004.

[6] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: Visualization meets data management. June 2006.

[7] S. B. Davidson and J. Freire. Provenance and scientific workflows: Challenges and opportunites. June 2008.

[8] E. Deelman, G. Singh, M. P. Atkinson, A. Chervenak, N. P. C. Hong, C. Kesselman, S. Patil, L. Pearlman, and M.-H. Su. Grid-based metadata services. *International Conference on Scientific and Statistical Database Management*, 2004.

[9] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. In *Proceedings of ACM SIGIR '03*, 2003.

[10] S. Harizopoulos, D. J. Abadi, S. Madden, and M. Stonebraker. OLTP through the looking glass, and what we found there. June 2008.

[11] S. Harizopoulos, V. Liang, D. J. Abadi, and S. Madden. Performance tradeoffs in read-optimized databases. 2006.

[12] A. L. Holloway and D. J. Dewitt. Read-optimized databases, in depth. *Proceedings of VLDB '08*, 1, August 2008.

[13] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(Web Server issue):729–732, July 2006.

[14] S. N. Jones, C. R. Strong, D. D. E. Long, and E. L. Miller. Tracking emigrant data via transient provenance. In *Proceedings of USENIX TaPP '11*, June 2011.

[15] M. Meseke. Using xml and xquery for data management in hpss. In *Proceedings of MSST 2011*, 2011.

[16] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer. Provenance-aware storage systems. In *Proceedings of USENIX ATC '06*, 2006.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[18] S. Ram and J. Liu. Understanding the semantics of data provenance to support active conceptual modeling. In *Proceedings of Active Conceptual Modeling of Learning '06*, 2006.

[19] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon. Design and implementation of the Sun network file system. In *Proceedings of USENIX ATC '85*, 1985.

[20] F. Schmuck and R. Haskin. GPFS: A shared-disk file system for large computing clusters. In *Proceedings of FAST '02*, 2002.

[21] S. Shah, C. A. N. Soules, G. R. Ganger, and B. D. Noble. Using provenance to aid in personal file search. In *USENIX ATC '07*, 2007.

[22] Y. L. Simmhan, B. Plale, and D. Gannon. A framework for collecting provenance in data-centric scientific workflows. *IEEE International Conference on Web Services*, 2006.

[23] C. A. N. Soules and G. R. Ganger. Connections: using context to enhance file search. In *Proceedings of SOSP '05*, 2005.

[24] M. Stonebraker, C. Bear, U. Çetintemel, M. Cherniack, T. Ge, N. Hachem, S. Harizopoulos, J. Lifter, J. Rogers, and S. Zdonik. One size fits all? part 2: Benchmarking results. January 2007.

[25] C. Strong, S. Jones, A. Parker-Wood, A. Holloway, and D. D. E. Long. Los Alamos National Laboratory Interviews. Technical Report UCSC-SSRC-11-06, University of California, Santa Cruz, Sept. 2011.

[26] R. W. Watson and R. A. Coyne. The parallel I/O architecture of the High Performance Storage System (HPSS). In *Proceedings of MSS '95*, 1995.

[27] S. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn. Ceph: A scalable, high-performance distributed file system. In *Proceedings of OSDI '06*, Nov. 2006.

[28] B. Welch, M. Unangst, Z. Abbasi, G. Gibson, B. Mueller, J. Small, J. Zelenka, and B. Zhou. Scalable performance of the Panasas parallel file system. In *Proceedings of FAST '08*, 2008.