Easing the Burden of HPC File Management

Stephanie N. Jones, Christina R. Strong, Aleatha Parker-Wood, Alexandra Holloway, Darrell D.E. Long
Storage Systems Research Center, University of California, Santa Cruz

File Organization Methods

Number of Scientists	Method of Metadata Management
4	Text File
1	PowerPoint Presentation with embedded visualization
3	Long, complex directory and file names
1	Spreadsheet with embedded visualization
2	Encoded in the input deck
1	Printed out (in binders)

Many scientists maintain a separate file to keep track of what calculation is being run on which machine, as well as the parameters associated with that calculation.

Some are kept by hand; one scientist uses the Stickies application on his personal computer to remind him of what his file names mean, since they are usually names like ProbA123.

There is often redundancy in the management schema; the scientist who maintains a PowerPoint presentation encodes the same parameters in his input deck.

These solutions have arisen because scientists feel the file system is lacking: either it does not collect the information they need, or because it has become easier to manage their own data than it is to find it.

Ideal File System

What Scientists Want

Scientists care that their data is stored, but don't want to have to know where or how it is being stored. They just want to be able to find the data again.

The ability to create correlations among files, both automatically and manually with tags, was a common request. Being able to associate similar files is incredibly helpful when looking back at a calculation.

The scientists are aware that their ad-hoc solutions at data management are suboptimal, and would like to be able to get away from it. They just don't have a better solution available.

With the amount of data that can be generated, finding what you are looking for is difficult. Finding the information important to you is even more difficult.

Our Proposed Solution

A unified search space that spans the primary storage system and the archival storage system will help abstract the storage systems from the scientists. When a query is issued, results from both systems will be returned.

Provenance information creates correlations among files, automatically relating similar files. By keeping the provenance information closely related with other metadata information, similar data will be much easier to find.

By automatically generating file names, we help eliminate the need for both long, unwieldy naming schemes and short, unintelligible naming schemes. Names are created based on unique, useful metadata.

Ranking query results will decrease the amount of time scientists must spend looking for the right data by identifying files important to the scientist

Provenance Enabled File System

Data provenance is the history or lineage of a piece of data. The two most common types of provenance are workflow provenance, which records the files and programs used to create the data, and content based provenance, which tracks changes made to the data itself. While there is no known implementation of a provenance enabled file system for high performance computing, there has been work done in grid computing on collecting provenance information.

Scientists have the ability to track this kind of information themselves, using applications such as Taverna, Kepler, and VisTrails. 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. This means the scientists are responsible for correctly recording their own provenance before the workflow is actually executed. The Provenance Aware Storage System (PASS) collects workflow provenance automatically, but requires a modified Linux kernel.

Since the majority of the provenance work done focuses on how to collect it, our work focuses on how we can use provenance information to assist the scientists. Therefore, our work assumes that provenance information has been collected in some way and is available for us to use. For our preliminary work, we are looking at using provenance information from existing provenance collection systems such as PASS or Taverna.

Storing Provenance

The most common storage method for data provenance is a relational database. Since relational databases provide an inherent query language that can be used to access provenance once it is stored, they are a natural choice for proof-of-concept work.

There are several problems with using a database solution for provenance storage. First and foremost, scientists are not going to learn a query language in order to gain the benefits of using provenance. Secondly, databases are often optimized for either read or write workloads, and therefore have difficulty doing both well. Lastly, there has been work done showing that using a traditional database management system for a variety of search and indexing applications is often a poor solution. This work showed that an application specific design that takes into account the technology and workload requirements of the specific problem is far superior. However, creating a customized provenance database for each application for every problem is infeasible.

To address the issue of where provenance information belongs in a HPC environment, we are exploring two methods of storing provenance. Using the Ceph file system as a test bed, we are exploring implementations which include a link to the provenance graph as an extended attribute in the i-node, and including an important piece of the provenance in the i-node. We believe this second option will be most successful, and are experimenting with ways to determine which piece of provenance is most beneficial. We will compare both approaches, and contrast them with the current method of using a separate database of provenance.

Unified Search Space

Remembering where you 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 systems. If a scientist has forgotten where his data is, he must query each system individually.

Transient provenance is an extension to workflow provenance which tracks data as it leaves the system. We are looking into using transient provenance information to create a unified search space over the primary storage system (NFS) and the archival storage system (HPSS). When data is archived, transient provenance creates a record of it, retaining metadata information on the primary storage system. Thus, any query over the provenance will include information from archival storage as well as the primary storage system.

Currently this approach only deals with presenting a unified search space over NFS and HPSS, as including the parallel file system presents several unusual difficulties. At LANL, it is possible to have information stored across multiple distinct scratch spaces, thus necessitating a merge of the parallel file system scratch spaces themselves first. Unique to PNNL was the challenge of geographic disparity: since the large simulations are often run at other national laboratories, being able to include information from different geographical locations becomes important. These issues are beyond the scope of our current work, but we are working on solutions with these problems in mind.

File Naming

Scientists 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. This often results in misplaced information when an attribute is forgotten, ordered incorrectly, or files are named similarly. Rather than forcing the scientist to create and remember file names, we will use a unified set of metadata and provenance information to create file names that 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.

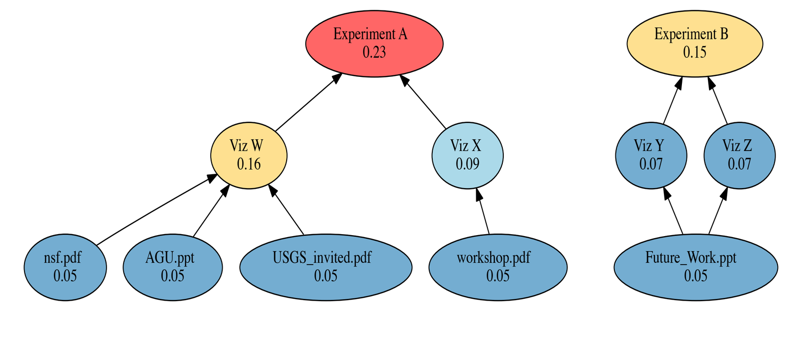
While it is easy to create very unique file names, creating unique and *expressive* file names is not. A file's i-node number could be used for an extremely unique name, but is completely meaningless to a scientist. Through statistical analysis of metadata, and techniques derived from faceted browsing, we can determine what attributes distinguish files, as well as which are most meaningful to scientists. At query time, we can select items that are more likely to identify the file, or we can choose meaningful attributes at file creation time.

We are looking into borrowing techniques from linguistics to determine which attributes are meaningful. 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.

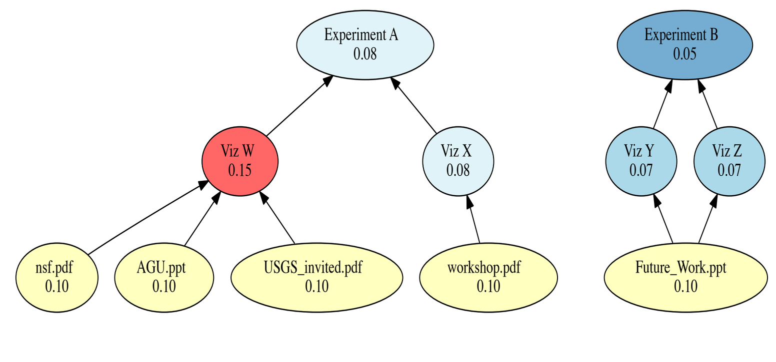
Query Ranking

Query 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 pieces of code are most frequently used. To do ranking, the provenance graph will be analyzed using eigenvector analysis similar to Google's PageRank.

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.



PageRank importance ranking of a provenance graph



Provenance Rank importance ranking of a provenance graph

Many thanks to the developers and scientists who took the time to talk with us at both the Los Alamos National Laboratory and the Pacific Northwest National Laboratory. We greatly appreciate their willingness to answer our questions and explain their work! A very special thanks to Meghan Wingate McClelland, who coordinated everything at LANL for us, and John Johnson, who coordinated our visit to PNNL. This would not have been possible without them!

I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Luda¨scher, and S. Mock. Kepler: An extensible system for design and execution of scientific workflows. 2004.
S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: Visualization meets data management. June 2006.
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.
K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer. Provenance-aware storage systems. In Proceedings of USENIX ATC '06, 2006.
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.