Experiment Explorer: Lightweight Provenance Search over Metadata

Delmar B. Davis, Hazeline U. Asuncion
Computing and Software Systems
University of Washington, Bothell
Bothell, WA USA

Ghaleb Abdulla
Lawrence Livermore National Laboratory

Abstract

Scientific experiments typically produce a plethora of files in the form of intermediate data or experimental results. As the project grows in scale, there is an increased need for tools and techniques that link together relevant experimental artifacts, especially if the files are heterogeneous and distributed across multiple locations. Current provenance and search techniques, however, fall short in efficiently retrieving experiment-related files, presumably because they are not tailored towards the common use cases of researchers. In this position paper, we propose Experiment Explorer, a lightweight and efficient approach that takes advantage of metadata to retrieve and visualize relevant experiment-related files.

1. Introduction

The growing field of eScience leverages computational resources to conduct scientific research. Being able to trace the origin of the data and its connections to the eventual results, referred to as *data provenance*, is crucial in supporting the repeatability of analyses and experiments. Usually, raw data, intermediate data, experimental results, and other experiment-related files reside within a researcher's file system or in a database. After a period of time, these files accumulate to the point where it is difficult to determine which files correspond to which experiments. Moreover, if the files are image-based, simple text search becomes infeasible.

For example scientists at LLNL conduct experiments and collect data from image files. They often find themselves needing to cross-reference materials from multiple experiments or from experiments performed by their colleagues. Searching for files that are heterogeneous and distributed across multiple locations can be a time-consuming task.

Such search scenarios, however, are not easily facilitated by current techniques. Provenance tools enable searching through provenance logs produced in a scientific workflow by issuing query commands [1, 13] or searching through metadata [15]. Another technique ties workflow instances with data inputs or outputs [11]. However, these techniques do not provide topical search across experiments. Methods to search through a file system exist [3, 7], but these require the ability to specify a search phrase that matches the contents of the file [3] or to navigate through previously examined documents [7]. A project called PODD also provides search capabilities, but only for information that resides inside a database [12].

In order to address these challenges, we propose a novel provenance search technique, Experiment Explorer (EE). EE is a technique that combines an enterprise level indexing tool, Apache Solr [2], with a machine learning technique called topic modeling [6]. We use Solr to index the metadata associated with experiment files and we use topic modeling to provide additional semantic meaning to experiment files based on the metadata. Topic modeling is an unsupervised statistical approach for learning semantic topics from a set of documents. Finally, search results can be sorted, filtered, and visualized along different dimensions (e.g., time, author, and topics).

We discuss our approach in the next section. Section 3 covers tool support, and Section 4 discusses related work. Finally, we conclude with ideas for future work.

2. Approach

The main elements of our approach are metadata search, topic modeling, and visualization of results. These elements complement the experiment process followed by researchers at LLNL. Each step in the experiment process produces artifacts accessible through Experiment Explorer's search facilities.

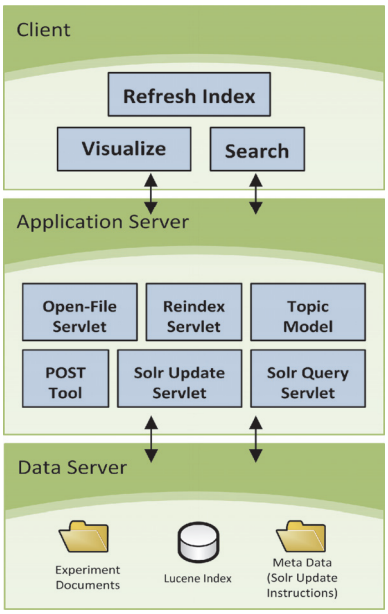

Figure 1. EE Architecture

For each experiment, scientists perform the following steps. First, an experiment design and a provenance template, containing information regarding the experiment, are created. Workflows are produced based on the design. The experiment is run using the workflow, producing several images and feature data. Finally, the output files are analyzed, and this analysis is stored in a spreadsheet. The provenance information is recorded in the metadata for all of the artifacts involved in the experiment, along with document specific fields such as title, description, and keywords.

Metadata provides a connection between files based on provenance-specific fields. While the basic search functionality is topical, faceting support provides a means to limit results to research-related documents. In Fig. 2, we see results that are connected based on the experiment author, keywords, and category. Specifying a filter indicates a narrow query. In this case we see that a publication on computer vision and an image used within the publication are returned as results.

In order to provide a more meaningful search through the various experiment files, we associate each file with its provenance metadata. We then save the files and metadata to a central network location (i.e., Data Server) where it can be indexed (see Fig. 1). Once the metadata are indexed, users can proceed to search for files. In the meantime, our topic model algorithm classifies the metadata in real-time as the users perform their search.

Topic modeling (based on a Bayesian model known as Latent Dirichlet Allocation (LDA)) is a widely-used machine learning technique that automatically learns topics from a text corpus (see [4, 6, 10]). Once an LDA model is learned on a corpus, one can use the learned probability distribution over words, P, to display a list of W words, sorted by decreasing probability, for each topic t. For instance, if topic 1 has high-probability words "ball score players win", one can assume this topic to be about sports. LDA also associates each document to these topics. In our case, topic modeling can be performed on the metadata fields.

Finally, we provide different views of the search results, such as the ones shown in Fig. 2 and 3. When a user selects a search result, the associated experimental file is then displayed. Fig. 2 shows that selecting a document link from one of the search results displays the associated image file.

Users may also choose to visualize the search results. Fig. 3 shows a sample visualization which displays the topics that were automatically learned on the left. The files that correspond to the topics are color-coded on the right. The files are also associated with the re-

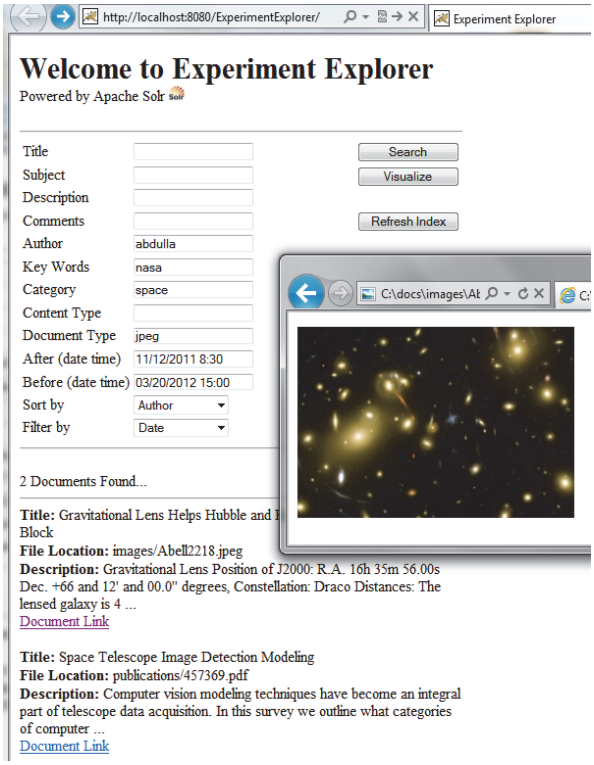

Figure 2. EE Search Page with a searched image

searchers who created the files. The date adjustment bar at the bottom left allows users to view which files were created at a given point in time, which is useful for finding relevant files. The display mode at the top left allows users to view icons or filenames (or both).

3. EE Tool Support

We designed the tool to be accessible to researchers, to facilitate efficient search of experiment files, and to support cross-referencing of files across various experiments, researchers, and projects. To support accessibility, we have a web-based user interface that allows users to search for metadata fields used by the researchers. To support efficient search of experiment files, we use Apache Solr to index the metadata offline. Finally, to support the cross-referencing of experiment files, we propose topic modeling to automatically classify files -- the topic model results can be visualized in a graph that is easily navigated by users. When users click on a file in the graph, the file is shown within the native editor.

3.1 Use Cases

We envision the following use cases. After Researcher A runs his experiments, he creates metadata with the generated image file. He saves the file along with the metadata to the Data Server. After a week, Researcher B wishes to see the experiment results of Researcher A. She uses EE and searches under the author field. She then filters the results to the previous week's date range. She also visualizes the search results to find the related image files.

Another week passes and three other researchers complete their experiments. They all save their files and metadata to the Data Server. Researcher A now wishes to see which experiments have been completed. He runs the index update and then searches for the experiment title. All the image files that were added by his three colleagues are now shown in the search results.

Researcher A processes several image files into an Excel data file and extracts features using an image processing tool. He records the metadata associated with the Excel file which includes the name of the image processing tool, the version, the extracted features, date and time among other attributes. The Excel file is analyzed by both researchers A and B and the analysis produces other Excel sheets. Researcher C would like to know whether the original image files were analyzed, and if so, the time of analysis and the name of researcher who performed the analysis. She also would like to use the reduced data in a publication. Thus, she searches for all files related to the original experiment and filters by analysis type.

3.2 Implementation

We have implemented the search and topic model components. The search component is implemented with Apache Solr and the topic model is implemented with Perl Scripts and Lucene.

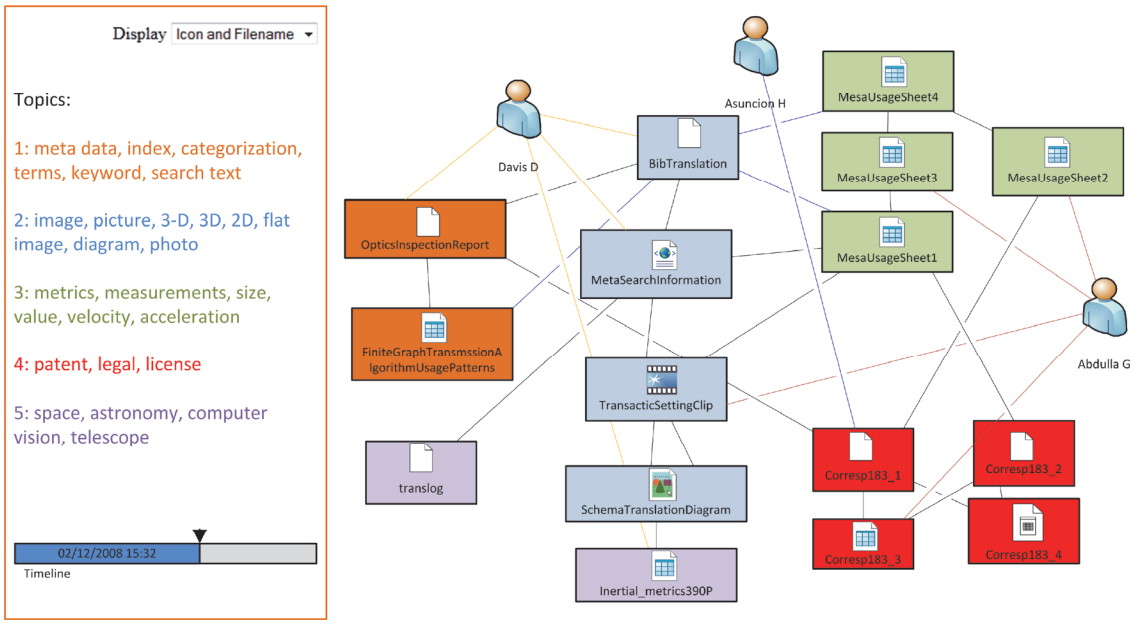

Figure 3. Visualization which relates the files to authors and classifies them according to topics on the left.

Search requests are based on input fields that match the Solr schema. Responses are returned in JavaScript object notation and are used to build document objects. This collection of objects is used to display information about, and provide links to, the research documents. EE runs on an Apache Tomcat container. The search page is dynamically updated using JavaScript and server functionality is implemented with the J2EE web components shown in the application server area of Fig. 1. These components are used to access network accessible experiment documents and manipulate the Lucene index. Metadata fields for the experiment documents are stored in the form of index update instructions known to the Apache community as Solr documents. These documents are formatted in XML and constitute the final component of the data server shown in Fig. 1.

Index updates are handled on the server via HTTP POSTs to Solr's update servlet, where the update instructions constitute the POST body. Currently, the reindex servlet shown in Fig. 1 launches a command line POST tool when the user clicks the refresh index button (see Fig. 2). The POST tool gathers all update instructions on the data server and POSTs them to Solr's update servlet. Given the size of the production metadata set, indexing may take over 10 minutes. At deployment, indexing will be performed offline to ensure efficient search.

Files are opened from links displayed in each result. Unlike a normal link that navigates to a new page or file mapped to a web server location, these links invoke the open-file servlet shown in the application server area of Fig. 1. The file location, which is part of the metadata, and consequently the document result, is passed to the servlet. The servlet opens the file on the server and streams it back to the browser using the appropriate MIME type.

4. Related Work

Searching for a document on a given machine is traditionally performed using desktop search techniques [9]. These techniques, however, require that the search phrase match the contents of the full text. An improvement over this technique is to use user activity mining along with the search to help the user retrieve previously opened files [7]. This technique, however, does not aid in searching for files that others may have created.

To enable efficient querying of data, other techniques require using a database [5, 12, 15], a scientific workflow [1, 11, 13], or a specific technology such as map-reduce [8]. While map-reduce has been shown to be scalable when compared to techniques that take hours to get results, searching map-reduce data can still take over 10 minutes on large data sets. Web crawlers have also been used to search through files in different locations [14]. However, it still falls short of automatically classifying files based on semantic similarity.

5. Conclusion

We have proposed Experiment Explorer, a lightweight and efficient approach that takes advantage of metadata to retrieve and visualize relevant experiment-related files. Our next steps in this project are to integrate the topic model into the application server and to implement the visualizations based on the topic model. We then plan to evaluate our techniques within a research group at LLNL. We anticipate that our approach will allow researchers to find relevant experimental information more efficiently.

6. Acknowledgement

This work is supported in part by the University of Washington Royalty Research Fund No. A65951.

7. References

[1] ALTINTAS, I., BARNEY, O., AND JAEGER-FRANK, E. Provenance collection support in the Kepler Scientific Workflow System. In *Proc of Int'l Provenance and Annotation Workshop (IPAW)* (2006).

[2] APACHE SOFTWARE FOUNDATION. Apache Solr. http://lucene.apache.org/solr/.

[3] APACHE SOFTWARE FOUNDATION. Lucene. http://lucene.apache.org/.

[4] ASUNCION, H. U., ASUNCION, A. U., AND TAYLOR, R. N. Software traceability with topic modeling. In *Proc of Int'l Conf on Software Engineering* (2010).

[5] BENABDELKADER, A., SANTCROOS, M., MADOUGOU, S., VAN KAMPEN, A., AND OLABARRIAGA, S. A provenance approach to trace scientific experiments on a grid infrastructure. In *Proc of Int'l Conf on e-Science* (2011).

[6] BLEI, D., NG, A., AND JORDAN, M. Latent dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 993–1022.

[7] CHEN, J., GUO, H., WU, W., AND WANG, W. iMecho: an associative memory based desktop search system. In *Proc of Conf on Information and Knowledge Management* (2009).

[8] DEDE, E., FADIKA, Z., GUPTA, C., AND GOVINDARAJU, M. Scalable and distributed processing

of scientific XML data. In *Proc of Int'l Conf on Grid Computing* (2011).

[9] DUDA, C., KOSSMANN, D., AND ZHOU, C. Predicate-based indexing for desktop search. *VLDB Journal 19*, 5 (2010), 735–758.

[10] GRETARSSON, B., O'DONOVAN, J., BOSTANDJIEV, S., HÖLLERER, T., ASUNCION, A., NEWMAN, D., AND SMYTH, P. TopicNets: Visual analysis of large text corpora with topic modeling. *ACM Trans. Intelligent Systems & Tech 3*, 2 (2012), 23:1–23:26.

[11] KOOP, D., SANTOS, E., BAUER, B., TROYER, M., FREIRE, J., AND SILVA, C. T. Bridging workflow and data provenance using strong links. In *Proc of Int'l Conf on Scientific & Statistical DB Mgmt* (2010).

[12] LI, Y.-F., KENNEDY, G., DAVIES, F., AND HUNTER, J. PODD - towards an extensible, domain-agnostic scientific data management system. In *Proc of Int'l Conf on e-Science* (2010).

[13] MISSIER, P., SOILAND-REYES, S., OWEN, S., TAN, W., NENADIC, A., DUNLOP, I., WILLIAMS, A., OINN, T., AND GOBLE, C. Taverna, reloaded. In *Proc of Scientific and Statistical Database Mgmt*. 2010.

[14] PANDEY, S., KANE, M., AND SPRINGER, J. GLASS: Genomic literature area sequence search. In *Proc of Int'l Conf on Bioinformatics and Biomedicine Workshops* (2011).

[15] PLALE, B., ALAMEDA, J., WILHELMSON, B., GANNON, D., HAMPTON, S., ROSSI, A., AND DROEGEMEIER, K. Active management of scientific data. *IEEE Internet Computing 9*, 1 (2005), 27–34.