Integrating Provenance Information in Reservoir Engineering

Jing Zhao, Na Chen
Computer Science Department
University of Southern California
Los Angeles, CA
{*zhaoj, nchen*}@usc.edu

Karthik Gomadam, Viktor Prasanna
Ming Hsieh Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA
{*gomadam, prasanna*}@usc.edu

***Abstract*—Data management and analysis has become an integral component in the area of reservoir engineering. An important metric that determines the overall effectiveness of data analysis is data quality. Data provenance, the metadata that pertains to the derivation history of data objects, has emerged as an invaluable asset in evaluating data quality. The reservoir facilities and software systems that collect provenance information are often distributed, thus making it difficult to analyze provenance data. Our primary contribution in this paper is an approach for provenance information integration in reservoir engineering.**

***Keywords*-Data provenance, Data integration, Reservoir engineering**

I. Introduction

Data provenance information has become an invaluable asset in evaluating data quality in reservoir engineering. Data provenance is the metadata that pertains to the derivation history of data objects [1], [2]. Information about how, when, and by who a piece of data is created and modified, coupled with knowledge about domain processes, allows scientists and engineers to estimate the accuracy and the currency of data. For example, understanding the physical and chemical properties of a rock is an important aspect of reservoir engineering. Rock properties can be obtained by using different techniques such as wireline logging [1] and logging while drilling (LWD) [2]. Designed to replace wireline logging, LWD captures a wealth of additional information, including properties related to the resistivity [3] and borehole caliper[4]. The technique used for obtaining the rock properties is a part of the provenance information and will aid the design of approaches that can exploit the additional measurements (such as using the right kind of drilling fluid) for optimized drilling. Analysis of provenance information has become a critical requirement of data analysis in reservoir engineering.

The primary contribution of this paper is an approach for integrating provenance information stored in distributed environments. To obtain the provenance information of a data object, one must consider the applications and data objects that were involved in its derivation. The information about these applications and data objects can be distributed, since the facilities and systems that execute the applications and collect the data are distributed themselves. This necessitates the need for integration of provenance information.

In our approach, we consider provenance integration as an outcome of a user query. The first step in integrating provenance information is to use the provenance index service to identify the target repositories. The index service, described in Section III-D has a mapping between the provenance metadata and the provenance repositories. The metadata is captured in the IAM framework, described in Section III-A. We have extended the IAM framework with a semantic provenance model, discussed in Section III-B that captures the causal relationships between the data objects and applications. In Section III-C, we present a graph based data structure, derived from the Open Provenance Model [3], to represent provenance information. Section IV describes the algorithm for provenance integration. A discussion on the performance of our algorithms is presented in Section V. We discuss related research in Section VI and present our conclusions in Section VII.

II. Motivating Example: Reservoir Forecasting Process

We present the reservoir forecasting process as our motivating example. A reservoir forecasting process predicts reservoir performance under different scenarios, during the lifetime of a reservoir and involves different sub-processes. An important sub-process is a complex simulation application. The input data for the simulation includes data about the reservoir deliverability, capacity description about the ability of the reservoir to produce oil, the historical production data collected from production history, the surface facility constraints of the facility, and the export system capacities over the life of the reservoir. Each input is further generated using different techniques such as lab tests, seismic and production simulations, and fluid property analysis. Each of these techniques involve applications that are complex processes in themselves and are often executed in various locations. Figure 1 illustrates an overview of the reservoir forecasting process.

[1] http://en.wikipedia.org/wiki/Wireline_(cabling)

[2] http://en.wikipedia.org/wiki/Logging_while_drilling

[3] http://en.wikipedia.org/wiki/Resistivity_logging

[4] http://en.wikipedia.org/wiki/Caliper_log

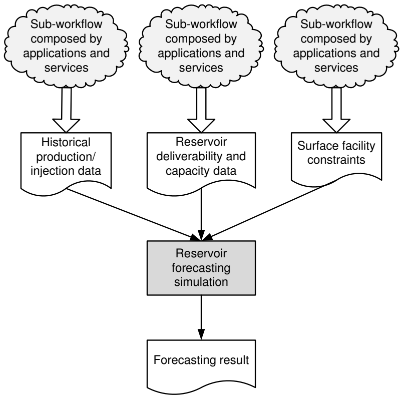
Sub-workflow composed by applications and services
Sub-workflow composed by applications and services
Sub-workflow composed by applications and services
Historical production/ injection data
Reservoir deliverability and capacity data
Surface facility constraints
Reservoir forecasting simulation
Forecasting result

Figure 1. A big picture of a reservoir forecasting process.

The outcome of the forecasting process plays an important role in decisions such as detailing the development strategies of a reservoir. The quality of the input data is a significant factor in determining the quality of the forecasting and this in turn is affected by the data generated using different sub-processes. The provenance information of the input to the forecasting process can be used as a reliable estimator of the data quality. However, to obtain the provenance information of the input, we need to integrate the provenance information collected from the sub-processes. Considering the fact that data flows across sub-processes and applications that are distributed, the provenance information is usually captured in heterogeneous data objects that are stored in multiple repositories. This paper discusses approaches for integrating provenance information in such a scenario.

III. System Overview

In this section, we present a brief overview of our system. The two main components of the system , illustrated in Figure 2, are: 1) a provenance index service and 2) the IAM framework. The provenance data that is collected from various processes is stored in provenance repositories.

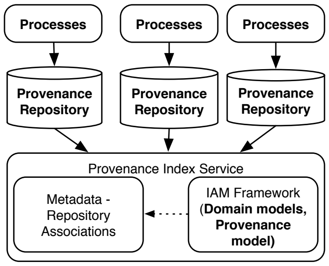

Figure 2. Overview of System Architecture

A. Integrated Asset Management Framework

Data heterogeneity is a serious impediment to any data integration effort. To address this problem in the reservoir engineering domain, we have developed an Integrated Asset Management (IAM) framework. The IAM framework is a collection of semantic models that capture the domain knowledge, including assets, their properties, and the relationships [4], [5]. A key component of the IAM framework is the metadata index and catalog. Metadata is collected from different domain applications and is published into the metadata catalog of the IAM framework. The catalog contains information about domain entities and is linked to the semantic models in the IAM framework, thereby facilitating data integration and exploration.

B. Provenance Model

The provenance model allows for capturing the derivation history of data objects containing their creation, consumption, and modification information. The basic building blocks of the model are derived from the Open Provenance Model (OPM). Three primary entities and their relationships defined in OPM are: *Artifact*, *Process*, and *Agent* [3]. The *Artifact* entity represents the data objects, which may be a physical object or a digital object in a computer system. The *Process* entity represents an action or a series of actions belonging to a process or a standalone application. A process may consume one or several artifacts to generate new artifact(s). The *Agent* usually represents users in control of the *Processes*. The provenance model is integrated with the IAM framework and the provenance information is annotated using the domain ontologies in the IAM framework. The semantic annotations allow us to integrate provenance information obtained from different models. The information in the provenance model is organized into three levels:

1) The **metadata level** maintains references to data objects and applications.
2) Each data/application instance contained in the metadata level is represented as an artifact/process instance at the **provenance level**, and the provenance level captures the causal relationships among metadata instances.
3) Each metadata instance is annotated by a domain entity, defined at the **domain level**.

C. Provenance Graph

The provenance information can be represented as a directed graph. We first define a *process* or an *artifact* as a *Provenance Node* and the *agent* as a property of the *process*. We map the causal relationships between processes and artifacts as edges between provenance nodes. E.g., if an artifact *A* was generated by a process *P*, we define two provenance nodes A and P, connected using a directed edge originating at P. If an artifact *A* is consumed by a process *P*, the directed edge will have A as its origin. A process node may have multiple incoming/outgoing edges since a process may have multiple input/output items. Similarly an artifact node may also have multiple outgoing edges since the artifact can be used as input by multiple processes. When tracking the provenance information of an artifact, the provenance node associated with the artifact is used as the origin for provenance graph exploration. This node is defined as a *sink node*.

An *Abstract Provenance Graph* of an artifact is a template of the provenance graph of the artifact. In an abstract provenance graph, each provenance node is represented by its category and domain concept, and does not refer to a detailed data/application instance. Therefore the provenance model for nodes in an abstract provenance graph only contains information from the provenance and domain levels. Provenance graphs generated by executing a process P with different constraints and parameters, will be instances of the the same abstract provenance graph, the one that is derived from the process template of P.

D. Provenance Index Service

The provenance index service is used in integrating provenance information from distributed provenance repositories. In our current implementation, we have incorporated the domain models into the index service for the sake of computational efficiency. The index service maintains a mapping between the provenance metadata captured in the IAM framework and the repositories in which provenance information pertaining to the metadata is stored. When a new repository is added, it is registered with the index service using the metadata information of the provenance data that is stored in that repository.

IV. Provenance Integration

In this section, we discuss our algorithm for integration of provenance information for provenance queries. The algorithm is a centralized algorithm although queries need to be processed in distributed repositories.

In a provenance query, a user is interested in the provenance information of an artifact. For example, in Figure 3, the user submitted a query Q1 for the provenance information of the artifact F1, which is a forecasting result. To process this query, the first step is to identify the process that created the artifact. This process is defined as the parent process of the artifact. In our example illustrated in Figure 3, the parent process of F1 is a simulation application S1. Once the parent process is identified, we use the provenance index service to identify the repository that has information on the parent process. The provenance information of the artifact is obtained by querying this repository and is represented as a provenance graph. In Figure 3, when the provenance information about F1 is requested, we identify repository R1 and fetch the provenance information. As illustrated in the figure, F1 was created by process S1. The inputs to S1 are the historical data H1, the reservoir deliverability data D1, and the surface facility constraints C1.

In the next step, we explore the artifacts that are associated with the parent process. In our example, in this step we will explore H1, D1, and C1. For each of these artifacts, we check if the parent process of the artifact is present in the same repository. If it is present, we further explore the parent process. Going back to our example, the process HC1, which is the parent process of H1, is present in the repository R1 and will be explored in this step. If the parent process is not present, we do a look up on the provenance index service to identify the repository that contains the parent process and continue to explore the newly identified repository. In our example, we explore repository R2 for D1 and R3 for C1. We repeat this process until all the artifacts are explored or if the provenance information of an artifact is not available.

Each time an artifact is explored, its provenance information is obtained as a provenance graph, illustrated within dotted lines in Figure 3. Once all artifacts are explored, the provenance information requested by the user, is generated by making an union of the provenance graphs obtained from each repository. In Figure 3, after exploring repository R1, we explore the repositories R2 and R3, containing provenance information about artifacts D1 and C1. The provenance information for H1 can be obtained from R1 and is a part of the provenance graph obtained from R1. The graphs obtained from R2 and R3 are combined with the one obtained from R1 to create the provenance information of F1.

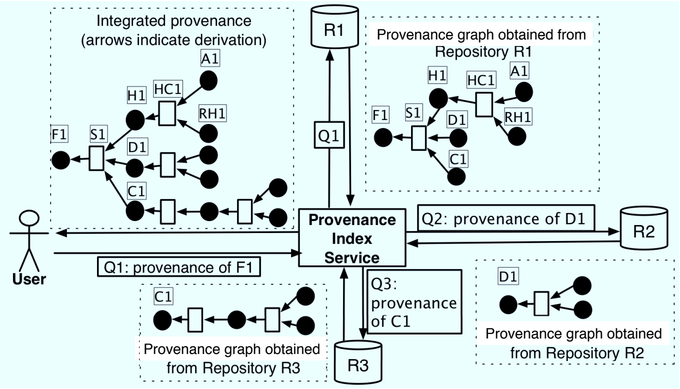

Figure 3. Integration for Provenance Query

V. Evaluation

To evaluate our approach, we created two kinds of synthetic *workflows*. These workflows were based on the patterns of reservoir engineering workflows. The first kind of synthetic workflow, W_1, was created based on an implementation of the integrated forecasting workflow (IFT), introduced in Section II. This workflow has 2000 provenance nodes, including both artifacts and processes. The second workflow, W_2, was derived from the reservoir management workflow that calculates the original oil in place [5] (OOIP). Around 400 provenance nodes are contained in this kind of workflow. Provenance information was collected from

[5] http://en.wikipedia.org/wiki/Oil_in_place

multiple executions of the workflows (which can be seen as a series of executions of processes).

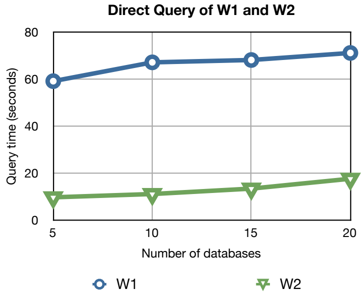

Figure 4. Time for provenance queries for W_1 and W_2.

Figure V illustrates the time performance of provenance queries for both W_1 and W_2. One hundred executions of the workflows generated around 1.5 million triples of RDF data for W_1 and around 500,000 triples for W_2. The number of databases used to store the triples was varied between five and twenty. As illustrated in Figure V, increasing the number of databases causes an increase in the query time. This is because, increasing the number of databases increases the number of times the provenance index service is accessed. Workflow W_1, being more complex than W_2, has a higher query time.

VI. Related Work

The challenges in the area of data provenance has been outlined in [6] and [7]. They further provide an approach for computing provenance information from database queries in [1]. Zhao et. al discussed the application of semantic Web techniques for managing and querying provenance information as a part of the MyGrid project [8]. Miles, in [9], defines a provenance query and describes techniques for scoped execution of provenance queries. Y. Zhao et al. [10] and Holland [11] have proposed approaches for expressing provenance queries. Distributed provenance query has been studied by Groth in [12]. Heines et al. [13] focused on mechanisms for efficient storage and querying of provenance information. Szomszor and Moreau in [14] addressed the problem of recording provenance information in Grid and Web service environments.

VII. Conclusions

In this paper, we have discussed our work in integrating distributed provenance information, with a focus on the domain of reservoir engineering. Two building blocks of our approach are: 1) a semantic provenance model that utilizes semantic domain models to address the heterogeneity of data objects, and 2) a provenance index service that integrates provenance information from distributed repositories.

Acknowledgment

We acknowledge the contributions of CiSoft (Center for Interactive Smart Oilfield Technologies), a Center of Research Excellence and Academic Training and our collaborators at Chevron Corporation. CiSoft is a joint venture between the University of Southern California, Los Angeles, CA and Chevron Corporation.

References

[1] P. Buneman, S. Khanna, and W. Tan, “Why and where: A characterization of data provenance,” in *In ICDT*, 2001, pp. 316–330.

[2] Y. L. Simmhan, B. P., and D. Gannon, “A survey of data provenance in e-science,” *SIGMOD Record*, vol. 34, no. 3, pp. 31–36, September 2005.

[3] L. Moreau and et. al., “The open provenance model (v1.01),” http://eprints.ecs.soton.ac.uk/16148, University of Southampton, Tech. Rep., 2008.

[4] R. Soma, A. Bakshi, and V. Prasanna, “A semantic framework for integrated asset management,” in *Proceeding of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid)*, 2007.

[5] R. Soma, A. Bakshi, and V. K. Prasanna, “An architecture of a workflow system for integrate asset management in the smart oil field domain,” in *Proceeding of the 1st IEEE International Workshop on Scientific Workflows (SWF)*, July 2007.

[6] P. Buneman, S. Khanna, and W. chiew Tan, “Data provenance: some basic issues,” in *In Foundations of Software Technology and Theoretical Computer Science*, 2000, pp. 87–93.

[7] R. Ikeda and J. Widom, “Panda: A system for provenance and data,” in *Proceedings of the 2nd USENIX Workshop on the Theory and Practice of Provenance (TaPP ’10)*, February 2010.

[8] J. Zhao, C. Goble, M. Greenwood, C. Wroe, and R. Stevens, “Annotating, linking and browsing provenance logs for e-science,” in *Proceedings of the 2nd International Semantic Web Conference (ISWC2003) Workshop on Retrieval of Scientific Data*, 2003.

[9] S. Miles, “Electronically querying for the provenance of entities,” in *Proceedings of the International Provenance and Annotation Workshop (IPAW’06)*, 2006.

[10] Y. Zhao and S. Lu, “A logic programming approach to scientific workflow provenance querying,” in *Proceedings of the International Provenance and Annotation Workshop (IPAW’08)*, 2008.

[11] D. A. Holland, U. Braun, D. Maclean, K.-K. Muniswamy-Reddy, and M. I. Seltzer, “Choosing a data model and query language for provenance,” in *Proceedings of the International Provenance and Annotation Workshop (IPAW’08)*, 2008.

[12] P. Groth, “A distributed algorithm for determining the provenance of data,” in *Proceedings of the fourth IEEE International Conference on e-Science (e-Science’08)*, 2008.

[13] T. Heinis and G. Alonso, “Efficient lineage tracking for scientific workflows,” in *SIGMOD’08*, 2008.

[14] M. Szomszor and L. Moreau, “Recording and reasoning over data provenance in web and grid services,” 2003, pp. 603–620.