A Novel Update Propagation Module for the Data Provenance Problem: A Contemplating Vision on Realizing Data Provenance from Models to Storage *

Abed Elhamid Lawabni [1] Changjin Hong [1] David H.C. Du [2] Ahmed H. Tewfik [1]

Digital Technology Center, Intelligent Storage Consortium (DISC) and

[1] *Department of Electrical and Computer Engineering*
University of Minnesota, Minneapolis, MN 55455
{lawabni, hongcj92, tewfik}@ece.umn.edu

[2] *Department of Computer Science and Engineering*
University of Minnesota, Minneapolis, MN 55455
du@cs.umn.edu

Abstract

To date, the systems approach to science, which emphasizes the connections among phenomena studied at different scales and by different disciplines, is causing dramatic changes in how scientific results are communicated. These changes drive a shift on how to propagate data with certain properties so that it can be used intelligently by others. In this work we elaborate on three major factors governing the propagation module of data provenance. The proposed propagation module provides an efficient solution for many critical problems in the management and provenance of scientific data. Unlike previous work, our work aims at realizing data provenance from models to storage. The natural representation of data as objects and its utility for capturing provenance has led us to consider a new storage architecture based on the object-based storage (OSD) technology. An outline of this framework is discussed.

1. Introduction

Provenance is a well-established concept in the art world where the lineage, pedigree or origins of a painting are critical to determining its authenticity and value. It is of equal importance in present and future data-rich environments ranging from computational biology, intelligence information gathering, to high energy physics.

Scientific research is based on exchanging data and conclusions. Data collected by a research group, or conclusions reached by the group, build on prior data and results produced by the group and the entire community. They in turn contribute to other derivative innovations, corrections and data. The integrity of scientific knowledge (its accuracy and reproducibility), the rate at which any scientific community can extend it, and the time elapsed between a new discovery and its widespread use for the greater good of society, all depend on the ability to track the propagation and complex interdependencies of the underlying representations and embodiments of knowledge.

There are numerous forms of provenance. In particular and of importance to this work, is the derivation path of information. The derivation path records the process by which results are generated from input data. This could include a workflow that orchestrates a number of processes and their parameters, or services. Accurately tracking the lineage (origin and subsequent processing history) and the propagation of data in the derivation path is essential for effective management and decision making when an experiment needs to be re-run in light of new or modified information.

The following two motivating use cases illustrate the importance of the issues described above. In molecular biology, where data is repeatedly copied, corrected, and transformed as it passes through numerous genomic databases, understanding where data has come from, its original confidence level and how it arrived in the user's database is of crucial to the trust a scientist will put in that data. Furthermore, if one or more of the data inputs get slightly modified or changed, knowing exactly the contribution of each input on the output variability can have a tremendous saving in terms of computations.

Consider a second case involves analysis of remote sensing data from satellites. Remote sensing data may be processed and reprocessed in many different ways. Examples include need to correct for distortion caused by the atmosphere and to interpolate measured data values onto geolocations. Very large datasets consisting of several years of data are periodically reprocessed to contribute to other derivative data products. Assessing the uncertainty and the influences or relative importance of each input parameters on the output variability, can correct subtle errors and expedite lengthy and complex procedures. The chain or pipeline of processing steps that generate standard *"levels"* of NASA remote sensing data products provides one common example.

Aligned with this vision, in this work, we elaborate on three major factors governing the propagation module of data in the derivation path; namely:

(1) Sensitivity analysis (SA):to ascertain how much a model (numerical or otherwise) depends on each or some of its input parameters
(2) Confidence level and uncertainty: how much the data can be trusted, and

* This work was supported by StorageTek, Veritas, Engenio, and Sun Micro through the sponsorships of DISC.

Proceedings of the 22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 2005)
0-7695-2318-8/05 $20.00 © 2005 IEEE

(3) Complexity: the required time and cost of computation

Unlike previous work, we aim on designing an efficient propagation module which predicts the impact of an input update or slight changes to the model parameters on the downstream derived data. More precisely, our primary objective is to analyze and to point out the vital role that these above mentioned factors play in modeling data provenance, so that we can alleviate the following scenarios:

(1) Propagation of erroneous outcomes, and
(2) Unnecessary rerunning of time consuming and heavily computations

Furthermore, our proposed module considers each piece of scientific data as a data object. Much relevant information related to this data object can be expressed by attributes associated with it. This naturally leads to an implementation based on an emerging storage technology called Object-based Storage Devices (OSD).

Our primary focus in this paper is to establish new and novel models for update propagation and decision making under uncertainty. As a long term vision, we aim to exploit the capabilities of OSD-based storage devices to provide a powerful framework for solving the data provenance problem, by

- Utilizing the extensible attribute mechanism in OSD to enable fast search and content-based queries
- Exploiting the scalability properties of OSD to support "infinite" object versions, and
- Building an OSD-based prototype which allows the seamless management and tracking of provenance

The rest of this paper is organized as follows. Section 2 presents the proposed update propagation module in details. Illustrative theoretical example is presented in section 3. Section 4 discusses prior work of data provenance. In section 5, we elaborate on compatibility with OSD implementation. Finally, conclusions are drawn in section 6.

2. Provenance Modeling and Update Propagation

2.1. Basic Definitions and Assumptions

Most applications used in collaborative scientific field usually share common procedures. The scientists generate a meaningful source data and several algorithms or analysis tools are design to process that data. In general, the information or the data to be processed can have several formats and attributes describing its characteristics and physical meaning. Let's denote our input data by o_i. Assuming that a numerical mapping function is given interpreting the meaning of the data, this input data can be represented as a numerical value in the form of either a scalar or a vector. We also define the "process" $f_k^{v_i}$ (v_j represents the jth version) as a function of application to represent any type of mathematical models, an analysis tool or searching engine on distributed database system. $f_k^{v_i}$ can have a different configuration on model parameters ($\beta_k^{v_i}$) and on the application algorithm ($m_k^{v_i}$). It is worth noting that $f_k^{v_i}$ is not constrained by any linear property. A generic type of workflow is described in our provenance modeling framework in figure 1.

Two possible sources of errors are considered. The first source of error can be induced by either source data or derived object through the derived path and the second one is that of the process, $f_k^{v_i}$ itself.

2.2. Motivation of Propagation Decision

As figure 1 illustrates, a single data is obtained through multiple computation steps in which innumerable inputs and parameters get involved in generating it. Furthermore, the derived data may be fused with any other data generated by other processes. This paradigm of the generic

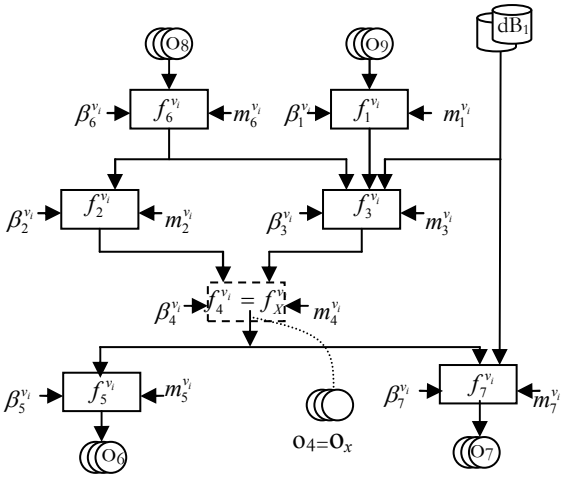

Figure 1. A generic workflow

The $f_k^{v_i}$, $\beta_k^{v_i}$, and $m_k^{v_i}$ denote a process, parameter, and algorithms respectively. o_x is represented an object derived by unknown process

workflow can be interpreted as three phases, namely, sourcing, transforming, and recording.

The information or resources to have a specific outcome derived is not always transparent. While a tiny modification of one source, even from indirect step in the workflow, may have an influential impact on one of the corresponding derived data, a significant change on a direct input can give an infinitesimal update in the output. In the former scenario, the situation can be even worse if there is a significant fault on either the input or the process

model causing an expensive and lengthy computation. In computational biology, we often face a problem of tremendous computation time (e.g. it can be a day or a week) in the homology search among different creatures' genomes when updated amino acid sequence is obtained. In huge digital libraries, most searching engines are highly sensitive to queries by clients.

As a result of this, the design issue of data provenance arises.

- After a model performs the computation, what additional information should be included in the derived data to maximize utilization of the resources by the next users?
- Before processing input data, how efficiently can models capture and analyze useful pieces of information on that input?

The answer to these questions underpins the fundamentals of our proposed propagation decision module. In the next subsections, we will explore three major factors, namely; sensitivity, uncertainty, and complexity. These factors are considered as the main pillar of our proposed propagation module. Therefore, a thorough investigation of these factors to formulate the final decision will alleviate scenarios such as propagation of erroneous outcomes, unnecessary rerunning of time consuming and heavily computations, and provide a significant solution of efficient data management system in terms of performance and quality.

2.3. Novel Decision Factors

2.3.1. Sensitivity Analysis (SA)

From this motivation, we need to determine the *relevance* of an object update to its ancestors and descendants. More dependent objects on the update we have, more objects can be affected. Notification of update ought to occur in the order of relevancy of the update to each object. Clearly, not all updates will affect an ancestor or descendant object. For example, a descendant may be relatively insensitive to the characteristics or data that changed. In our model, *sensitivity* reflects "which update of inputs contributes the most to output variability so that how many relevant objects can be triggered by this affected object". Each individual input has different sensitivity to its related objects and also the objects can be affected by interaction among updates of objects with uncertainty.

We will base our derivation using the *variance-based methods*. Those methods consist in the computation of sensitivity indices, which apportion the sensitivity of model output variance to model inputs. For a model

$$Y = f(X_1, \ldots\ldots, X_p)$$

Assuming independent inputs, first order sensitivity indices are defined by

$$S_i = \frac{V(E[Y|X_i])}{V(Y)} \quad (1)$$

and express the part of variance of model output Y due to model input X_i. Higher order indices are also defined, to express effect of input interactions and total indices for total effect of one input. An important property, which enables us to easily interpret sensitivity indices values, is that the sum of all these indices is equal to 1, when inputs are independent (for more details on this property, the reader is referred to [31]) i.e. $\sum_{i=1}^{p}(S_i) = 1$. The variance-based methods are classed as *global SA* in the sense that sensitivity assessment on the output to each input parameter is carried out by considering the combined variability of all the parameters simultaneously.

Since the independence assumption of the model inputs is sometimes difficult to justify in practice, and as usual sensitivity indices are not meaningful when inputs are non-independent, in what follows we will present the problem of sensitivity analysis for real models with correlated inputs.

Consider the following model

$$Y = f(X_1, \ldots\ldots, X_p)$$

where

$$(X_1, \ldots\ldots, X_p) = (\underbrace{X_1, \ldots, X_i}_{x_1}, \underbrace{X_{i+1}, \ldots, X_{i+k_1}}_{x_{i+1}}, \underbrace{X_{i+k_1+1}, \ldots, X_{i+k_2}}_{x_{i+2}}, \underbrace{X_{i+k_{l-1}+1}, \ldots, X_p}_{x_{i+l}})$$

$(X_1, \ldots, X_i) = (x_1, \ldots, x_i)$ are independent inputs, and $(x_{i+1}, \ldots, x_{i+l})$ are l groups of intra-correlated inputs (x_i are independent of x_j, for all $1 \le i, j \le l$).

We wrote mono-dimensional non independent variables $(X_1, \ldots, X_p)$ like multidimensional independent variables $(x_1, \ldots, x_{i+l})$. Thus, we define first order sensitivity indices

$$S_j = \frac{V(E[Y|x_j])}{V(Y)} \qquad \forall j \in [1,\ i+l]$$

To connect this to mono-dimensional variables, if $j \in [1, \ldots\ldots, i]$, we have well defined the same indices:

$$S_j = \frac{V(E[Y|x_j])}{V(Y)} = \frac{V(E[Y|X_j])}{V(Y)}$$

and if $j \in [i+1, \ldots\ldots, i+l]$, for example $j=i+2$:

$$S_j = S_{\{i+k_1+1, \ldots\ldots, i+k_2\}} = \frac{V(E[Y|X_{i+k_1+1}, \ldots\ldots, X_{i+k_2}])}{V(Y)}$$

Finally, it is very important to note that if all input variables are independent, those sensitivity indices are identical to those given in equation (1). And so, multidimensional sensitivity indices can well be interpreted like a generalization of usual sensitivity indices (1).

Like in classical analysis (Sobol [31]), Monte-Carlo estimations are possible. We estimate mean and variance of *Y* by:

Mean of *Y*: $\hat{f}_0 = \frac{1}{N}\sum_{k=1}^{N} f(x_1^k, \ldots\ldots, x_{i+l}^k)$,

Variance of *Y*: $\hat{D} = -\hat{f}_0^{\,2} + \frac{1}{N}\sum_{k=1}^{N} f^2(x_1^k, \ldots\ldots, x_{i+l}^k)$

And first order indices by $\hat{S}_j = \frac{\hat{D}_j}{\hat{D}}$, where

$$\hat{D}_j = \frac{1}{N}\sum_{k=1}^{N} f(x_1^k, \ldots x_{j-1}^k, x_j^k, x_{j+1}^k \ldots, x_{i+l}^k) f(x_1^k, \ldots, x_{j-1}^k, \underline{x}_j^k, \underline{x}_{j+1}^k, \ldots, \underline{x}_{i+l}^k) - \hat{f}_0^{\,2}$$

where $(x_1^k, \ldots, x_{i+l}^k)_{k=1,N}$ and $(\underline{x}_1^k, \ldots, \underline{x}_{i+l}^k)_{k=1,N}$ are two independent sets of *N* (multidimensional) inputs simulations.

Another important challenge related to both our propagation model and SA, often encountered in practice, is the case on which sensitivity analysis have been made, undergoes a transformation, or, a minor mutation. In such case, is it possible to obtain information about sensitivity analysis of the mutated model, without doing a new complete analysis, by only using sensitivity results from the original model?

In the following, we will present an outline of the methodology which we used to answer this question. For some possible mutations, we will mathematically relate sensitivity indices of original model with those of mutated model. Following the nature of the mutation, some assumptions are necessary. Quit often the independence of the model inputs is a valid assumption.

Assume that a sensitivity analysis has been made on a model $M: Y = f(X_1, \ldots\ldots, X_p)$ where $(X_1, \ldots\ldots, X_p)$ variables are independent inputs. Let us suppose that new information about the model, new measurements, or even changes in the modeled process, oblige us to consider a new model M^{New} that is also a mutation of the original model *M*. For instance, consider a model $M: Y = f_1(X_1) + f_2(X_2, \ldots\ldots, X_p)$ where $(X_1, \ldots\ldots, X_p)$ are independent random variables, and suppose that *M* undergoes a mutation, and is also transformed in a new model M^{New} where X_1 is fixed to its mean $\mu_1 = E[X_1]$. Thus, this new model is $M^{New}: Y^{New} = f_1(\mu_1) + f_2(X_2, \ldots\ldots, X_p)$. First order sensitivity indices S^{New} can be expressed from the sensitivity indices S of *M* by:

$$S^{New} = S \times \frac{V(Y)}{V(Y^{New})}$$

Finally, let us present another type of mutation. Assume that two analysis have been made on two models $M_1: Y_1 = f_1(X_1, \ldots\ldots, X_p)$ and $M_2: Y_2 = f_2(X_{p+1}, \ldots\ldots, X_{p+q})$, and also the sensitivity indices S^1 for M_1 and S^2 for M_2 have been computed. We suppose that input variables of the two models are different and independent. Let us create a new model $M^{New}: Y^{New} = Y_1 + Y_2$. Sensitivity indices of M^{New} are obtained by

$$S^{New} = S^1 \times \frac{V(Y_1)}{V(Y_1) + V(Y_2)} + S^2 \times \frac{V(Y_2)}{V(Y_1) + V(Y_2)}$$

To conclude, if an original model (on which sensitivity analysis has been made) is transformed, it is possible to deduce sensitivity indices of the mutated model, without starting again heavy calculation, in a given number of cases. Those cases are principally deletion of variables or introduction of new independent variables.

2.3.2. Uncertainty Analysis and Confidence Level

An update may be erroneous as it may have been produced by faulty data of a faulty experimental procedure, or by a breakdown in the process. This raises the issue of how much to trust a particular update within a specified range or interval of possible error (e.g. uncertainty probability). In our model, a *confidence level* given to an object and a process is defined as one minus uncertain probability.

More importantly, we would like to investigate the law of propagation of uncertainties, and combining uncertainties due to different sources with respect to our sensitivity. Consider the following Model: $Y = f(X_1, \ldots\ldots, X_p)$, and suppose that each input X_i is associated with u_i uncertainty, then, there will be an uncertainty in the output result due to each of uncertainties of measured quantities X_i, given by

$$u_{Y,i} = \left|\frac{\partial f}{\partial X_i}\right| u_i \qquad , (1 < i < p)$$

Note that the first term in the above equation is expressed by magnitude of the corresponding partial derivative (based on the first-order Taylor series coefficients) which is another classic definition of sensitivity.

Since there are many different sources of uncertainties for the same measurement, they will increase the total uncertainty of that quantity. It is unlikely for many uncertainties (due to their random nature) to have the same sign, so it would be inappropriate to combine them by adding their magnitudes, since many of them will be in opposite directions and cancel each other to some extent. Instead we will combine uncertainties of the same quantity (say *Y*) into a *combined uncertainty*, $u_{Y,combined}$ using the so-called Root-Sum-of-the-Squares (RSS) rules:

$$u_{Y,combined} = \sqrt{\sum_{i=1}^{p} (u_{Y,i})^2} = \sqrt{\sum_{i=1}^{p} \left(\left|\frac{\partial f}{\partial X_i}\right| u_i\right)^2} = \sqrt{\sum_{i=1}^{p} (S_Y^i u_i)^2}$$

where S_Y^i is the sensitivity of *Y* to input *i*. It is worth noting that the initial confidence level is assumed to be given by experts (best-estimated).

2.3.3. Complexity

When a workflow can be viewed as one lumped chained process including a large number of input sources with parameters, where the final outcome may be only of our interest (e.g. when fine-grain recording on intermediary result is not allowed to conduct), the *complexity* of the process become another crucial factor. Complexity denotes how long it will take to complete running a whole process. We will denote it by ***T***. Intuitively a process with a high complexity should be more concerned with the other mentioned factors.

2.4. Sequential Hypothesis Testing

Once we obtain relevancy or sensitivity to changed/updated data and the degree of trust to associate with it, we will pose the update problem as a sequential hypothesis testing problem.

Specifically, for a given update, we will ask the following questions

- Is this update valid and relevant to an output object of our interest?
- Is it valuable to rerun the process in the given computation time?

The answer is positive if a decision module that evaluates the effect of all factors represents a certain decision value exceeding a threshold.

The integration of all three factors is done by simply combining them using a weighting factor as follows

$$w_S S_Y^i + w_u u_{Y,combined} + w_C T \begin{array}{c} > \\ < \end{array} \delta \quad \begin{array}{l} Run \\ Do\ \ not \end{array}$$

where w_S, w_u, w_C are the weighting factors of the sensitivity, uncertainty, and complexity factors respectively.

It worth noting that the most critical factor to evaluate the propagation decision is the sensitivity factor, as it helps in predicting the estimated output or accurate discrepancy between the current version of the output and the new version to be generated.

3. Illustrative Example

As we mentioned in the previous subsection, the most critical factor to evaluate the propagation decision is the sensitivity factor. In this section we will spot light on this particular factor and present a theoretical example which emphasizes the usefulness and the important role that sensitivity analysis plays on a model with correlated inputs.

Consider the following model

$$Y = aX_1X_2 + bX_3X_4 + cX_5X_6,$$

where $X_i \sim N(0,1)$, for $i = 1,\ldots,6$ and where X_3 and X_4 are correlated $(\rho_{X_3,X_4} = \rho_1)$. Similarly, X_5 and X_6 $(\rho_{X_5,X_6} = \rho_2)$. The sensitivity indices are given by:

$$S_{12} = \frac{a^2}{a^2 + b^2(1+\rho_1)^2 + c^2(1+\rho_2)^2}$$

$$S_{\{3,4\}} = \frac{b^2(1+\rho_1)^2}{a^2 + b^2(1+\rho_1)^2 + c^2(1+\rho_2)^2}$$

$$S_{\{5,6\}} = \frac{c^2(1+\rho_2)^2}{a^2 + b^2(1+\rho_1)^2 + c^2(1+\rho_2)^2}$$

and all the other indices are equal to 0. Notice that the value of the numerator of the interaction sensitivity indice S_{12} is a function of the coefficient *a*. The values of numerators of the non-zero sensitivity indices $S_{\{3,4\}}$ and $S_{\{5,6\}}$ are function of the model coefficients *b* and *c*, and the correlation coefficients ρ_1, ρ_2, as well. To illustrate this, let us elaborate about the impact of different numerical values of the correlation coefficients and the model coefficients (*a*, *b* and *c*), on the sensitivity indices (see table 1).

Table 1. Numerical values.

	a	b	c	ρ_1	ρ_2	S_{12}	$S_{\{3,4\}}$	$S_{\{5,6\}}$
(i)	1	1	1	0.8	0.8	0.2336	0.3832	0.3832
(ii)	3	1	1	0.8	0.8	0.7329	0.1336	0.1336
(iii)	1	1	3	0.8	0.8	0.0575	0.0943	0.8483
(iv)	1	1	1	0.8	0.3	0.2881	0.4397	0.2922
(v)	1	1	3	0.8	0.3	0.0803	0.1317	0.7880
(vi)	1	1	3	0.3	0.8	0.0593	0.0647	0.8760

First of all, let us underline that as X_1 and X_2 are independent variables, indices S_1, S_2, and S_{12} are usual sensitivity indices, and can also be computed without our multidimensional method. In situation (ii), as X_1 and X_2 are independent variables, usual sensitivity indices allows us to conclude that variance of *Y* is essentially (73%) due to interaction between X_1 and X_2. But in the others situations, when X_1 and X_2 are less important, we need multidimensional sensitivity indices to apportions effect to the two couple (X_3, X_4) and (X_5, X_6). These multidimensional indices allow us to know that couple (X_3, X_4) and (X_5, X_6) have the same importance in situation (i), and that (X_5, X_6) is the most important in situation (iii). Effectively, in situation (i) couples (X_3, X_4) and (X_5, X_6) are symmetric in the model, and so they have same importance. In (iii) a coefficient equal to 3 is multiplying the product X_5X_6, that's why the couple (X_3, X_4) is most important than (X_3, X_4).

Situations (iv), (v) and (vi) illustrate that indices $S_{\{3,4\}}$ and $S_{\{5,6\}}$ are function to the correlation (S_{12} is also function to the correlation, but it's due to its denominator, which is the variance of *Y*). As couples (X_3, X_4) and (X_5, X_6) are in the model in a product form: X_3X_4 and X_5X_6, the greater is the correlation, the greater is the importance of the couple, and so the greater is the value of the sensitivity indices. In (iv) the correlation of (X_3, X_4) is greater than correlation of (X_5, X_6), and so $S_{\{3,4\}}$ is greater than $S_{\{5,6\}}$. In situations (v) and (vi), we can see similar behavior.

4. Prior Work

Many research groups have made some progress on aspects of this problem, e.g., [1, 2], but a complete solution has been elusive. As genealogical charts reveal successive generations of parents for an individual, data provenance generally refers to the sources and derivation of a data set or product [3, 4]. It also captures dependencies across different data object types, e.g., a file description of an experiment, a journal paper and a micro-array image. Much of the research involving data provenance has focused on two areas. The first area deals with the proliferation of scientific data transfers between different groups and systems. In this situation data provenance is used to protect downstream data users from unintended consequences resulting from these transfers. The second area studies how providers of scientific data can themselves benefit from tracking the provenance of their computational work [5, 6]. For example, the researchers in [7] study tracing the provenance of the integrated data in a data warehouse. Individual transformation steps in a schema transformation pathway are used to trace the derivation of the integrated data in a step-wise fashion. A framework for computing and verifying approximate fine-grain provenance of data items was proposed in [3]. The work presented in [8] described an algorithm called SUB-pushdown to trace provenance of array data. GOOSE [9] is a prototype system which uses data object attributes to track object versions and trace varying resolutions of provenance in a graphical interface. Data provenance is also referred to as data pedigree [10, 11], data lineage [5], derivation history [12], data set dependence [13], filiation [14], data genealogy [15], data archeology, and audit trail [16].

The authors in [17] discussed some of the technical issues that have emerged in scientific databases. The work in [3, 18, 19, 20, 21] studies the provenance of database queries and draws a distinction between *why-provenance,* which describes why a view or data object exists, and *where-provenance,* which captures the ancestor data objects that led to the creation of an object of interest. How to estimate original detail data from a summary was formulated as an inverse problem in [22]. This work also proposed a solution based on the optimization of a well-defined cost function under constraints.

Several preliminary attempts to integrate elements of provenance in scientific data have also been reported [23, 24, 25, 26, 27]. PASOA [28] aims to investigate the concept of provenance and its use for reasoning about the quality and accuracy of data and services in the context of e-Science. The Chimera Virtual Data System [29] uses a workflow "recipe" to create the data when required by transforming other virtual or real data. It combines a virtual data catalog that contains data derivation procedures and derived data with a virtual data language interpreter that translates user requests into data definition and query operations on the database. Collaboratory for the Multi-Scale Chemical Science (CMCS) project [30] is using advanced collaboration and metadata-based data management technologies to develop a chemical sciences portal providing support for distributed research, community communications, and data discovery, management, and annotation capabilities. The portal assists in documenting and browsing data pedigree and in communicating cross-scale dependencies between data produced at one scale and the results of computations using it at the next. In general, these pioneering efforts have focused on narrow applications and provide limited functionality.

Another related project "network-attached secure disks" (NASD) [32, 33, 34, 35] is defining a scalable storage interface characterized by four properties. First, direct storage-device-to-client transfers. Second, secure interfaces (e.g. via cryptography). Third, asynchronous oversight, whereby file managers provide clients with capabilities that allow them to issue authorized commands directly to devices. Fourth, an interface that provides variable-length objects with separate attributes, rather than fixed-length blocks, to enable self-management and avoid the need to trust client operating systems.

5. Compatibility with OSD Implementation

In our OSD-based modeling, our module can be deployed in existing processes without much difficulty and also it will be able to outperform others in accomplishing a facile integration of heterogeneous data which is a major design requirement. The simple task is to tag an identification indicating which process an object undergoes on extended attributes with execution environments including all decision factors during its job. Tracking to the lineage of the object can be simply performed through this tag to be extracted by our model. However, the potential problem may occur when our module fails to extract the tag since someone builds their own tool which is not known in public. The newly derived object meant to be unique turns out to be already used by other parties or groups. By looking in figure 1, for example, process 5 may

misunderstand the object, o_x as a primary or raw data. A possible approach is to search similar objects with o_x. The optimal searching algorithm should offer objects including o_2 and o_3 as its best similarities since an object is represented in multi-resolution of vector presentation. Also, as many people lookup o_x, they also search o_2 and o_3 most likely sharing same query. If we define query set of an object i as $q(o_i)$, o_j's which share a certain part of query set with $q(o_i)$ can be obtained. Once we succeed to find which objects get involved into deriving o_x from two methods, we can approximate the unknown process, f_X or we are potentially able to identify the unknown process with the selected candidates of object sources from all lists of unregistered processes.

In order to support the knowledge patch or extension described above, we plan to adapt the state-of-art of automatic metadata generation into our module. By observing user behavior of interaction with a process, our extended module can generate useful hints of object's descriptive information which enhances context analysis among objects. The inference of aggregation is recorded into our attributes automatically and it thereby breaks through scalability issue in the information management.

It is worth mentioning that the provenance problem is often addressed by a database system where the data and their relationships are maintained in tables. A database system is good for processing queries. However, the data provenance relationship has to be in a pre-determined fixed format (database schema). This may not be flexible enough for supporting multiple types of data objects and variable forms of data provenance relationships. Since data are scattered over several systems, this also add difficulty to archiving and long-term data preservation. As an alternative, and as future work, we plan to develop a data provenance architecture based on an emerging storage technology, Object-based Storage Device (OSD). In an OSD-based storage system, a file-level understanding of data objects is added to the storage device to exploit the increasing intelligence and capabilities of storage devices. This requires metadata and data to be stored together on OSD-based storage devices. In contrast the traditional storage devices only support block-level data accesses via SCSI commands and metadata is stored separately in the file system. In addition, OSD-based storage systems can support variable number of extended attributes. These attributes can be application-dependent and stored in the storage devices. Many of OSD-based storage devices are directly connected to IP network and form a global file system with metadata servers to facilitate the search and identification of particular data objects through keywords or Globally Uniquely IDentifications (GUID).

Instead of the traditional implementation based on database, we will rely solely on the underlying OSD objects and metadata service support. Especially, we will take advantage of the extended attributes of data objects by storing data object relationships and provenance information related to an object as extended attributes. We plan to develop an OSD-based architecture to support data provenance. The proposed architecture will be implemented as a prototyping system based on the Lustre code to demonstrate the feasibility and strength of OSD-based storage system to solve the data provenance problem of functional genomics. The detailed description of the implementation work will be reported elsewhere.

6. Conclusion

This work addresses several important aspects of the provenance problem and outlines a novel storage-based solution for them. Data provenance tracks the interdependencies between heterogeneous data objects as they get created or are derived from existing objects. The proposed propagation module provides an efficient solution for many critical problems in the management and provenance of scientific data. Unlike previous work, our work aims at realizing data provenance from models to storage.

In the modeling side, three major factors were integrated as a sequential hypothesis testing problem to form a unique decision module.

The natural representation of data as objects and its utility for capturing provenance has led us to consider new storage architectures that represent data as objects on the device itself, object-based storage (OSD).

Finally, future directions will be devoted to exploit the capabilities of OSD-based storage devices to provide a powerful framework for solving the data provenance problem. In particular, we are currently working on designing and implementing a flexible framework to store extended attributes for file system objects. Our data provenance function relies on recording provenance information for each object in its extended attributes. In comparison, the existing approaches store all provenance records in a central relational database within the provenance server. One obvious limitation of this centralized solution is scalability. As more and more provenance information is recorded into the relational database, the overheads of performing access, queries and managements increase accordingly. Furthermore, when there are a lot of concurrent submissions of provenance records from concurrent clients, the provenance server becomes a bottleneck due to its limited processing power and buffer space. To this end, we are also exploring the design space of a highly-scalable approach that distribute the provenance information to the MetaData server and every related OSD.

References

[1] Workshop on Data Derivation and Provenance, (Chicago),Oct.17-18,2002, http://people.cs.uchicago.edu/~yongzh/position_papers.html

[2] Workshop on Provenance and Annotation, (Edinburgh, U.K), Dec. 1-3, 2003, http://www.nesc.ac.uk/index.html

[3] A. Woodruff, M. Stonebraker, "Supporting fine-grained data lineage in a database visualization environment," In *Proc. of the Thirteenth International Conference on Data Engineering (ACDE'97)*, (Birmingham, U.K), IEEE Computer Society, pp 91-102, 1997.

[4] D. G. Clarke and D. M. Clark, "Lineage," In *Elements of Spatial Data Quality, S. C. Guptill and J. L. Morrison, Eds. Oxford: Elsevier Science*,13-30, 1995.

[5] R. Bose, "Composing and Managing Lineage for Scientific Data: A Review," Technical report, University of California, (Santa Barbara, CA), 2002.

[6] R. Bose, " A Conceptual Framework for Composing and Managing Scientific Data Lineage," In *IEEE Proc. of the 14th International Conference on Scientific and Statistical Database Management (SSDBM'02)*, Pages 15-19, July 2002.

[7] H. Fan and A. Poulovassilis, "Tracing data lineage using schema transformation pathways," In *Workshop on Knowledge Transformation for the Semantic Web (with ECAI'02)*, 2002.

[8] A. P. Marathe: "Tracing Lineage of Array Data," *SSDBM 2001*, Pages 69-78, 2001.

[9] G. Alonso and A. E. Abbadi, "GOOSE: Geographic Object Oriented Support Environment," In *Proc. of the ACM Workshop on Advances in Geographic Information Systems*, (Arlington, Virginia, USA), Pages 38-49, 1993.

[10] P. Buneman, D. Maier, and J. Widom, "Where was your data yesterday, and where will it go tomorrow?" *Data Annotation and Provenance for Scientific Applications,* White Paper, Feb. 28, 2000.

[11] J. C. French. "What is Metadata?" *In Proc of the SDM-92 Workshop*, (Richland, WA), Pages 3-8, 1995.

[12] N. I. Hachem, M. Gennert, and M. Ward. "The Gaea System: A Spatio-Temporal Database System for Global Change Studies," *In Proc of the AAAS Workshop on Advances in Data Management for The Scientist and Engineer*, (Boston, MA), Pages 84-89, 1993.

[13] G. Alonso, C. Hagen, H.-J. Schek, and M. Tresch, "Towards a Platform for Distributed Application Development," In *Workflow Management Systems and Interoperability*, Vol. 164, NATO ASI Series, (Berlin), Springer, Pages 195-221, 1997.

[14] L. Spery, C. Claramunt, and T. Libourel. "A lineage metadata model for the temporal management of a cadastre application," In *Proc of the Tenth International Workshop on Database and Expert Systems Applications*, (Florence, Italy), Pages 466-474, 1999.

[15] B. R. Barkstrom, "Digital Archive issues from the Perspective of an Earth Science Data Producer," In *Proc of the International Standards Organization (ISO) Archiving Workshop Series: Digital Archive Directions (DADs) Workshop*, (College Park, MD), 1998.

[16] P. Brown and M. Stonebraker, "Big Sur: A system for the management of Earth science data", In *Proc. of the 21st International Conference of Very Large Data Bases*, (Zurich, Switzerland), pp 720-728, 1995.

[17] P. Buneman, S. Khanna, and W. C. Tan, "Data Provenance: Some Basic Issues," *FSTTCS 2000*, pp 87-93, 2000.

[18] P. Buneman, S. Khanna, and W. C. Tan. "Why and Where: A Characterization of Data Provenance," In *International Conference on Database Theory,* 2001.

[19] P. Buneman, A. Deutsch, and W. Tan, "A Deterministic Model for Semistructured Data," In *Proc. of the Workshop on Query Processing for Semistructured Data and Non-standard Data Formats*, Pages 14-19, 1999.

[20] Y. Cui and J. Widom, "Practical lineage tracing in data warehouses," In *ICDE*, Pages 367-378, 2000

[21] Y. Zhuge, H. Garcia-Molina, J. Hammer, and J. Widom, "View maintenance in a warehousing environment," In *SIGMOD*, Pages 316-327, 1995.

[22] C. Faloutsos, H. V. Jagadish, and N. Sidiropoulos, "Recovering Information from Summary Data," *VLDB Journal,* Pages 36-45, 1997.

[23] myGrid project. http://www.mygrid.info/

[24] M. Greenwood, C. Goble, R. Stevens, J. Zhao, M. Addis, D. Marvin, L. Moreau, and T. Oinn, "Provenance of e-science experiments - experience from bioinformatics," Poster, UK e-Science All Hands Meeting, (Nottingham), Sep. 2003.

[25] J. Zhao, C. Goble, M. Greenwood, C. Wroe, and R. Stevens, "Annotating, linking and browsing provenance logs for e-science," In *ISWC 2003 Workshop: Semantic Web Technologies for Searching and Retrieving Scientific Data*, (Sanibel Island, Fl.), Oct. 2003.

[26] J. D. Myers, C. Pancerella, C. Lansing, K. L. Schuchardt, and B. Didier, "Multi-scale science: supporting emerging practice with semantically derived provenance," In *ISWC 2003 Workshop: Semantic Web Technologies for Searching and Retrieving Scientific Data*, (Sanibel Island, Florida), Oct. 2003.

[27] C. Pancerella, *et al*. "Metadata in the Collaboratory for Multi-scale Chemical Science," In *Proc. of DC-2003: the 2003 Dublin Core Conference*, (Seattle, Washington), Sep. 2003.

[28] PASOA project, http://cmcs.org/index.php
[29] I. Foster, J. Voeckler, M. Wilde, and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," In *14th Conference on Scientific and Statistical Database Management*, 2002.
[30] Collaboratory for the Multi-Scale Chemical Science (CMCS), http://cmcs.org/index.php
[31] I.M. Sobol, *Sensitivity Estimates for Nonlinear Mathematical Models*. Mathematical Modelling and Computational Experiments, 1993, 1: 407-414.
[32] Garth A. Gibson and Rodney Van Meter, "Network Attached Storage Architecture", *Communications of the ACM*, November 2000, Vol.43, No.11.
[33] Gibson, G.A., Nagle, D.F., Courtright II, W., Lanza, N., Mazaitis, P., Unangst, M. and Zelenka, J., "NASD Scalable Storage Systems", *USENIX99*, Extreme Linux Workshop, Monterey, CA, June 1999.
[34] Gobioff, H., Nagle, D.F. and Gibson, "Integrity and Performance in Network Attached Storage", G.A. CMU SCS Technical Report CMU-CS-98-182, December 1998.
[35] NASD, http://www.pdl.cmu.edu/NASD/