A Layer Based Architecture for Provenance in Big Data

Rajeev Agrawal
Department of Computer Systems Technology
North Carolina A&T State University Greensboro, USA ragrawal@ncat.edu

Ashiq Imran
Department of Computer Science
North Carolina A&T State University Greensboro, USA aimran@aggies.ncat.edu

Cameron Seay
Department of Technology
North Carolina A&T State University Greensboro, USA cwseay@ncat.edu

Jessie Walker
Department of Mathematics and Computer Science
University of Arkansas at Pine Bluff Pine Bluff, USA walkerjj@uapb.edu

Abstract—Big data is a new technology wave that makes the world awash in data. Various organizations accumulate data that are difficult to exploit. Government databases, social media, healthcare databases etc. are the examples of the big data. Big data covers absorbing and analyzing huge amount of data that may have originated or processed outside of the organization. Data provenance can be defined as origin and process of data. It carries significant information of a system. It can be useful for debugging, auditing, measuring performance and trust in data. Data provenance in big data is relatively unexplored topic. It is necessary to appropriately track the creation and collection process of the data to provide context and reproducibility. In this paper, we propose an intuitive layer based architecture of data provenance and visualization. In addition, we show a complete workflow of tracking provenance information of big data.

Keywords—Provenance; Big data; Query; Visualization

I. INTRODUCTION

Big data is a prevalent term but not wholly new area in information technology. Many of the foremost research based agencies such as NASA, NSF, NIH along with various intelligence agencies have been involved with big data. Social networking websites such as Facebook, Twitter, Linked-In, Google+, Youtube etc. are getting popular day by day. These sites deal with huge amount of data every day. Big data is getting importance not for its quantity of data rather the data can be used to extract a lot of information from it.

Whenever we find some data, do we think what the source of data is? It is quite possible that data is copied from somewhere else. It is also possible that information is incorrect. The data we usually see on the web, such as, rating of a movie, smart phone, or news story, do we think about it, how much legitimate is the information? We have also learned to be careful before relying on the information extracted from the Internet. For scientists, they need to have confidence on accuracy and timeliness on the data that they are working on.

Data provenance refers the description of the origin, creation and propagation process of data. Data provenance is the lineage and derivation of the data. It stores ownership and process history about data objects. Provenance has been studied extensively in the past and people usually use provenance to validate physical objects in arts, literary works, manuscript etc. Recently, the domain for provenance has gained significance attention in digital world and e-science. The provenance of data is crucial for validating, debugging, auditing, evaluating the quality of data and determining reliability of data. In today’s period of Internet, complex ecosystems of data are even more ubiquitous. It is not a surprise that data provenance has become a major research topic for conferences and workshops. Provenance capture is often incorporated into a workflow system [1].

Provenance information has a broad range of critical application areas. For example, scientific data processing needs to keep track of data ownership and action performed in workflow to ensure the trust assigned to the output data. In business environments, provenance of documents is even more important, e.g., for regulatory compliance and litigations.

Let us illustrate following scenario to get a better idea about provenance. Bob is a doctor of a hospital. Paul is a patient who visits to Bob for checkup. After diagnosis of Paul’s health, Bob passes the prescription of Paul to another doctor Peter. After reviewing the prescription and Peter makes some changes, he sends it to another doctor James for doing some test. The provenance of the document should contain the chain (Bob, Peter, James, along with some information about the process names that prescribed in the document. So, provenance information is for keeping tracking the changes. Important information can be verified using previously collected provenance records.

But provenance in big data is relatively unexplored topic and has not got much attention so far. Big data is expensive to manage and hard to extract value from. It is important to manage big data because more data may lead to more precise analysis and more precise analysis may lead to more confidence in decision making.

Applying provenance in a different layer can provide various level of useful information [2]. In our paper, we propose a layer based architecture to use the provenance data. The main goal of the proposed architecture is to improve robustness and flexibility for the collection and management of provenance data. A lot of factors have been
considered for designing the architecture. Each layer has
different role to play. We introduce new component
visualization toolkit to visualize provenance data. Both
administrator and user can access visualization toolkit to get
a graphical overview in provenance information. We impose
authorization for security and privacy on provenance data.
Each user can only be able to view the changes that user
made. Administrator can have access to all the users’
provenance data. Precisely, administrator has the flexibility
to view provenance information of different user. It provides
protection on provenance data.

We use MongoDB [3] as a storage which will be
discussed in later parts.

The main contributions of this paper are as follows:

- A technique for capturing provenance in big data is
  provided.
- We present a layer based architecture of provenance
  collection and visualization.
- We introduce an access control mechanism in
  provenance data.

The rest of the paper is organized as follows. Section II
provides a discussion of related works on provenance.
Section III describes detailed overview of our proposed
layer based architecture, and the various terms associated
with it. Section IV describes the overview of components of
the proposed architecture. We summarize our architecture
and discuss some future improvements of it and the various
security and privacy issues related to trustworthy
provenance in section V. Finally, the paper is concluded in
section VI.

II. RELATED WORKS

Big data concept refers to a database which is
continuously expanding that makes difficult to control and
manage. The difficulty can be related to data capture,
storage, search, sharing, analytics and visualization etc.
Provenance in big data has been identified by a recent
community whitepaper on the challenges and opportunities
of big data [4].

Provenance has been studied extensively in the past, in
particular in archival theory, and for the purpose of asserting
authenticity of literary work, arts, manuscripts etc.
Provenance has traditionally been collected at the workflow
level often making it hard to capture relevant information
about resource characteristics and is difficult for users
easily incorporate in existing workflows. Multiple
provenance tools have been established for workflow e.g.
VisTrails [5] and Kepler [6]. In recent years, we have got
systems RAMP [7] that uses provenance collection from
MapReduce workflows [8]. The use of provenance in
determining the quality of scientific data and data
provenance has also been shown [9].

Some applications have used provenance data in
debugging (e.g., to track an erroneous data item back to the
source from which it was derived), trust (e.g., by merging
trust scores for the data in a data item's provenance),
probabilistic data (the probability of a query result can be
computed from the probabilities of the data items in its
provenance [10, 11].

Provenance may be practiced across various domains
where different standards are used, driving needs for
interoperability which is currently satisfied by the Open
Provenance Model (OPM) [12].

PROB [13], a toolkit for tracking provenance data has
been recently proposed for Big data. This paper provides a
diagram on scientific workflow on big data. Initially, from
the subjects or phenomena, raw data is collected. After
processing of the raw data, preprocessed data is made. This
preprocessed data is ready to load into the cluster file
system. After doing analysis, derived data is produced. This
derived data can be visualized or described in publications.
Finally, publications are distributed and shared.

Provenance information can be used for security and
privacy issues. Some challenges and solutions has been
discussed about secure provenance in [14]. A framework for
a family of Provenance-based Access Control (PBAC)
models with extensions has been introduced that can handle
traditional access control issues [15].

Big data provenance can be used to detect and evaluate
performance blockages, to figure out performance metrics,
and to test a system's ability to exploit commonalities in
data and processing [16]. Provenance data can be useful to
play the role of benchmarking. In addition, they show an
example of monitoring and profiling using provenance
information. Log files contain a lot of useful information
that generate data objects. A model for detecting and
accumulating data from log files has been used in [17]. A
universal data provenance framework is proposed for data
provenance information for real-world applications without
any code modifications [2]. The ORCHESTRA [18],
collaborative data sharing system describes provenance
model and architecture.

Our work tries to capture and represents from multiple
layers. From the big data storage, it captures the provenance
data. From the collection of provenance data, user and
administrator can perform query in an intuitive manner. In
addition, user and administrator can also visualize captured
provenance information.

III. LAYER BASED ARCHITECTURE

Before starting to describe our layer based architecture,
it is important to know what the advantages of layer based
architecture are. The main advantage is that this architecture
is possible to change the contents of any one of layers
without having to make corresponding changes in any of the
others. This enables parallel development of the different
layers of the application. Complex application rules are
easy to implement in application server. Any future
changes can be easily incorporated.

Figure 1 shows our proposed architecture for provenance
collection, querying and visualization for scientific
applications. We propose a layered architecture for
exploring and collecting provenance. Our proposed
architecture is divided into 3 layers.
1) **Storage Layer:**
Vast number of information stores in Big data storage. This layer is for capturing provenance data and save it to provenance database. Storage layer collection component is responsible for capturing provenance data. Provenance information is extracted from big data storage. Provenance data itself is not big data.

2) **Application Layer:**
Application layer is composed of two components: Query and Visualization toolkit. Query component is for accessing query from the provenance data. Visualization toolkit component supports graphical view of provenance data which may be useful for getting more knowledge about provenance information.

3) **Access Layer:**
Access layer is for security and privacy purposes. This layer will provide privacy among users. If a user wants to check provenance record that user needs to login to verify her authorization. A user can only get access of her own provenance information whereas administrator can check every user’s provenance information for security purposes.

The amount of provenance data may differ according to the application and also the implementation done by the user. Provenance is captured in all forms and sizes, our architecture does not store provenance in a well-structured manner, focuses instead on the ability to write the captured provenance in a fast and efficient manner. It provides a way to collect all raw provenance data from the systems. Provenance data can be queried directly or further processed and transformed into different formats and more structured provenance. By separating the access layer, application layer, storage layer, optimizations on provenance collection is possible.

**A. Use case**
We consider a use case for provenance collection and storage system. The use case is for user provenance data which is stored into database. Whenever, user makes any changes on data, we keep track of the changes. For example, on each user session, user commits data into database. We need to capture user data submission. In our context, we capture user id, output of submission, execution time, user notations. Additionally, user can visualize his provenance information.

**B. Design Goals**
- **Less overhead:** We need to process huge volume of data, so high performance is critical. It is necessary that provenance collection has minor impact on application’s performance [19].
- **User annotations support:** It is important to capture user notes or metadata. This is applicable for database and workflows as well as for big data. Thus, we need an interface that allows users to add their notes about the experiment and output.
- **Visualization support:** Provenance data can be used for security and privacy reasons. If we are able to visualize the provenance data, we will clearly retrieve the ownership and access history of data.
- **Scalability:** Scalability comes into play for big data. The volume of big data is growing exponentially. Provenance data is also rapidly increasing and therefore making it necessary to scale up provenance collection.
- **Various data models support:** So far data models for provenance system have been in structured databases. It is important to not only support for structured as well as semi structured provenance data models from users and systems because big data may not follow a particular structure.
- **Secure Provenance:** There is huge volume of data and information. It is important to maintain privacy and security of provenance information.
- **Flexibility:** A typical approach to deal with velocity of the data is to introduce data cleaning and integration in pay per use fashion. This may reduce the cost and consume less time [16].

C. **Procedure**
Figure 1 shows the design of layered architecture. Our motive is to capture, query and visualize provenance information. After collecting provenance from big data storage, we keep our provenance information in provenance data storage.

- Initially, user needs to login to verify her identity whether she is an administrator or just a user. A user is an individual user who is not allowed viewing provenance information of other users. Administrator can view provenance information of all users. This is the only difference between an administrator and a user.
- Moreover, user submits query to provenance database. User gets the result of the query. However, administrator will get different result than a user.
- User can submit request to visualization toolkit and get chart of changes the user made in a timely manner.
- Administrator can also view provenance information using visualization toolkit.
<table>
<thead>
<tr>
<th>Access Layer</th>
<th>Application Layer</th>
<th>Storage Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provenance data of all users</td>
<td>Administrator</td>
<td>User</td>
</tr>
<tr>
<td></td>
<td>Submit</td>
<td>View</td>
</tr>
<tr>
<td>Result</td>
<td>Result</td>
<td>Result</td>
</tr>
<tr>
<td>Query</td>
<td>Visualization Toolkit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Big Data Storage</td>
</tr>
</tbody>
</table>

Figure 1. Overview of Layer based Architecture

Figure 2. A workflow diagram of tracking provenance in big data
IV. OVERVIEW OF COMPONENTS

A. Provenance Collection

Figure 2 shows the workflow diagram of tracking provenance. In the beginning, we have big data subject or phenomenon such as the data of social networking site. We collect some portion of this paradigm of data. We get the raw data which may be structured or semi-structured. Then we process data using big data analytics and load the preprocessed data into databases.

Databases can be of various types such as graph database and relational database. Relational database can be used for handling structured data. Graph database is useful for handling semi-structured or unstructured database since it contains graph like characteristics. Each node has a link with adjacent node. This allows graph database systems to utilize graph theory to very rapidly examine the connections and interconnectedness of nodes – and how Netflix can recommend videos for you.

After carefully analyzing and recognizing patterns or structure, we come to a decision that this data may be meaningful. Then we finally publish the data and distribute data to High Performance Computing (HPC) system. To clarify the shapes in the following figure 2, the parallelogram shape denotes data in different stage and rectangular shape denotes the process. Provenance data is captured whenever any changes occur from the beginning to the end.

B. Storage

The captured provenance data is sundry and wide-ranging. So, we need a data storage that must have capability of supporting semi-structured provenance document. In our case, we use an open source NoSQL data store that is MongoDB. It is document oriented and scalable. It was developed in C++. It is appropriate for our needs since we need to deal with semi-structured data. It also supports sharding. Sharding allows storing data records across multiple machines. MongoDB stores all data in documents, which are JSON-style data structures composed of field-and-value pairs [3]. In our context, captured provenance data will be stored into MongoDB data store. MongoDB stores documents on disk in the BSON serialization format. BSON is a binary representation of JSON documents, though it contains more data types than JSON. Figure 3 shows the storage format of MongoDB.

For file management, we use the GridFS specification for storing the actual file contents since it provides metadata and large file support. GridFS uses two collections for storing data. Large data objects are divided into chunk that are stored in chunk collection and metadata is stored in files collection. Figure 4 describes the prototype of a document files collection.

C. Query

In our implementation, we consider building query interface to access the resulting provenance data. The query language is designed to abstract the underlying language used at the storage layer. This storage layer allows the user to query in an instinctive manner without knowing in depth knowledge.

In our implementation, we have both command line and user interface for the users and administrators. The command line interface is meant to be used by users who are comfortable with the command line. On the other hand, user interface is meant supporting fairly open-ended provenance queries on the collection.

D. Visualization

Our approach includes visualization toolkit to provide an interactive and graphical view of provenance data. Visualization toolkit is different for administrator and user.

For administrator, we consider two types of visualization:

1) User based visualization: This technique is only applicable for administrator. Administrator can have a view and select desired user to check provenance information. Figure 5 shows the user based visualization.
2) **Time based visualization:** It is applicable for both administrator and user. Figure 6 shows the time based visualization for administrator in a typical day. Figure 7 shows the time based visualization for user in monthly manner. In our design, administrator has the flexibility of choosing time in hourly, daily, weekly, monthly basis. User can only view his provenance record in weekly or monthly basis.

**Figure 5.** Provenance information (user based).

**Figure 6.** Provenance information in a typical day (time based).

**Figure 7.** Provenance information in a year by a user (time based).

E. **Administrator**

Administrator is a super user who has the authorization for checking global provenance information. Administrator has the capability to individually select user and view that user’s provenance record.

F. **User**

User is an authorized person who has the capability to query and view his provenance information. Provenance data is a sensitive information. Without authorization, user can’t check his provenance information.

V. **DISCUSSION**

In this paper, we describe layer based storage management architecture and workflow of tracking provenance. This supports both structured and semi-structured provenance collection.

Provenance captured in storage layer is a mixture of data and process provenance. It can be used for a number of use cases. Our approach is useful in some cases and falls short in others. We consider structured and semi-structured data.

Our proposed architecture covers all of the design goals as discussed in section III. It is simple so it may not take much overhead. We provide multiple data model support to handle structured and semi structured data. We introduce visualization to have a clear view on provenance information. Since we use MongoDB database which support scalability.

We include user annotation support which gives every user flexibility to add some custom notation in provenance information. Each user can write suitable notation according to his need. Provenance initiation is user controlled but entire process is automated. Our aim is to capture provenance close to raw format.

We have introduced access control to preserve security and privacy of provenance data of each user. Administrator can visualize provenance information of all the users. Users can only visualize provenance of their own. Though for protecting provenance information only access control is not enough, we may need some additional policies. But, proper authorization can give protection of provenance information up to some extent. In future, we will impose other regulations and policies to make provenance data more secure. We may use digital signature and encryption method to increase security.

The big data explosion has seen a number of NoSQL data stores for scalability. In our paper, we use MongoDB which support both structured and semi structured provenance document.

The way we manage provenance data, it will be easy for the user to know where data objects are stored. Users can verify the NoSQL data store using regular expressions to search for previous job runs for a variety of information.

VI. **CONCLUSION**

In this paper, we discuss some design goals of provenance in big data. We present the design of a layer based architecture for provenance in big data. We divide the
architecture into 3 layers. Each layer has a significant role to play. We introduce access control mechanism to serve security aspect.

Our design is applicable for simple queries. We have just proposed this design as capturing provenance record. We are in the process of implementing our architecture. In future work, we will add additional components to handle complex queries. We may use provenance information as a benchmark of performance measurement. We will measure the performance of our proposed architecture in High Performance Computing (HPC) systems.

ACKNOWLEDGMENT

This research work was funded by the Office of National Consortium of Data Science (NCDS).

REFERENCES


