HHS Public Access

Author manuscript

Proc IEEE Int Conf Big Data. Author manuscript; available in PMC 2018 February 02.

Published in final edited form as:

Proc IEEE Int Conf Big Data. 2015 ; 2015: 2509–2516. doi:10.1109/BigData.2015.7364047.

Big Data Provenance: Challenges, State of the Art and Opportunities

Jianwu Wang,
Department of Information Systems, University of Maryland, Baltimore County, Baltimore, MD, U.S.A

Daniel Crawl,
San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA, U.S.A

Shweta Purawat,
San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA, U.S.A

Mai Nguyen, and
San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA, U.S.A

Ilkay Altintas
San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA, U.S.A

Abstract

Ability to track provenance is a key feature of scientific workflows to support data lineage and reproducibility. The challenges that are introduced by the volume, variety and velocity of Big Data, also pose related challenges for provenance and quality of Big Data, defined as veracity. The increasing size and variety of distributed Big Data provenance information bring new technical challenges and opportunities throughout the provenance lifecycle including recording, querying, sharing and utilization. This paper discusses the challenges and opportunities of Big Data provenance related to the veracity of the datasets themselves and the provenance of the analytical processes that analyze these datasets. It also explains our current efforts towards tracking and utilizing Big Data provenance using workflows as a programming model to analyze Big Data.

Keywords

Big Data; provenance; workflows; distributed data-parallel programming models

I. Introduction

Generally speaking, provenance in the digital context is about the origin and various transformations of data [1]. In the context of scientific computation and workflows, provenance usually means the lineage and processing history of a data product, and the record of the processes that led to it [2, 3]. Provenance is critical to many capabilities including experiment reusability [4] and reproducibility [5], fault tolerance [6], process optimization and performance prediction[1].

Big Data provenance [7, 8] is a type of provenance to serve scientific computation and workflows that process Big Data. In the Big Data era, the *volume*, *velocity* and/or *variety* of the data to be processed increase tremendously, bringing fundamental changes to data provenance tracking and usage [9] which is often referred to by a fourth V, the *veracity* of Big Data. [10] defines veracity as the quality and provenance of Big Data. In this paper, we focus on the second part of this veracity definition: the challenges posed by capturing the provenance of Big Data and some technical opportunities to tackle these challenges.

The ability to capture a holistic view of Big Data provenance depends on our ability to understand the Big Data ecosystem including the platforms and tools used for Big Data management and analysis. First, Big Data processing platforms, like Hadoop[2] and Spark[3], and their associated stacks of tools become indispensible components in the overall system to provide efficient and robust processing. A provenance-aware view over these Big Data platforms is needed for in-depth tracking of data being processed in these platforms. Secondly, workflows and other platforms that process Big Data often require distributed execution with high-volume and/or high-velocity data, making continuous tracking and integration of provenance in a centralized fashion inefficient. So Big Data provenance often needs to be recorded and retrieved in a distributed environment. Thirdly, the variety of security policies and quality evaluations on the analyzed Big Data sets poses challenges to collection of in-depth provenance tracking, requiring a black-box approach to data provenance.

The contributions of this paper are three-fold. First, we analyze the challenges of Big Data provenance and the gaps identified in existing systems based on these challenges (Section III and IV). Second, we layout a set of opportunities of Big Data provenance (Section V). Third, to better address the challenges and opportunities, we propose a reference architecture and present our in-progress efforts (Section VI and VII). We hope that this paper starts a further discussion on Big Data provenance, an important area of Big Data research for quality, validation and reproducibility of the products of Big Data analysis.

II. Big Data Workflow Systems

We focus on provenance that can be recorded by workflow systems. In this section, we survey several workflow systems that support Distributed Data-Parallel (DDP) patterns, such as Map, Reduce, Match, CoGroup and Cross [11]. We call them *Big Data workflow systems* since DDP is the core mechanism for Big Data processing.

Several systems have been developed on top of the Hadoop stack to support workflows by the Hadoop community. Pig Latin is a dataflow programming language for expressing data processing tasks [12]. Pig Latin programs are compiled into MapReduce jobs for Hadoop. The Nova workflow system is designed for batched, incremental processing of large datasets [13], and is built on Pig and Hadoop. Oozie[4] is a workflow scheduler system to manage

[1] http://hpc.pnl.gov/IPPD/
[2] http://hadoop.apache.org/
[3] http://spark.apache.org/
[4] http://oozie.apache.org/

Hadoop jobs, where each workflow is a Directed Acyclic Graph (DAG) of actions. The Cascading project[5] supports dataflow using DDP patterns such as Each, GroupBy, and CoGroup, and transforms user-generated dataflows into MapReduce jobs for Hadoop.

Newer generations of Big Data systems with further support for efficient data-parallel processing patterns were introduced, among which Spark and Flink [6] are two representative open source platforms. Both Spark and Flink (prior name Stratosphere) are suitable for scalable batch and stream Big Data processing. Spark system expresses each spark job as a DAG of operations. The data is split and computed across computing nodes to achieve distributed data-parallel execution. Flink provides a streaming dataflow engine that provides data distribution and communication for distributed computations over Big Data streams.

Evolving from traditional scientific workflow systems, Kepler supports Big Data by running Kepler workflows within Hadoop, Spark and Stratosphere platforms [11]. Kepler defines a set of higher-order components (called *actors* in Kepler) representing DDP patterns for users to build sub-workflows as UDFs. Each DDP actor corresponds to a particular DDP pattern. By expressing DDP sub-workflows in Kepler, the same workflow can be executed on top of different Big Data engines, i.e., Hadoop, Spark or Stratosphere.

III. Challenges of Big Data Provenance

There are many challenges for data provenance at large. Although the provenance of workflow-driven analysis is related to this global view of data provenance, there are specific challenges posed by the complex nature of workflows including data and compute systems, application specific legacy tools, and distributed large datasets. We first analyze some scenarios on Big Data provenance usage. Then we focus on a subset of these challenges posed by such scenarios.

A. Usage Scenarios

From our experience with real projects, we present four scenarios requiring Big Data provenance:

Scenario 1: Many scientific programs have adopted MapReduce programming models to process large volumes of data where each execution in Map/Reduce function only processes a fraction of data. The overall execution numbers of user defined Map/Reduce function (UDF) could easily go above millions. For instance, our experiments on a MapReduce based bioinformatics tool show one application execution could have up to 11 billion UDF executions [23]. Scientists find a very valuable record from the overall result. They want to know the input data of the UDF function execution generated this record and reproduce this specific execution.

Scenario 2: A number of online streaming Big Data sets are used in an analytical workflow application with many steps. Faulty datasets could lead to incorrect scientific conclusion or even catastrophic results [31]. In addition, the pre-processing

[5] http://www.cascading.org/
[6] https://flink.apache.org/

steps to reduce the high-dimensionality of the big datasets potentially introduce errors. So the application builder is trying to assess the quality of the integrated dataset and the final products. Since the input data quality might changes over time, how can application provenance help assessing data quality changes of final products from data qualities of input datasets?

Scenario 3: Cloud resources are becoming increasingly popular for scientific Big Data applications as their execution environments [25, 26]. The workload for these applications varies over time. Because of monetary costs of the Cloud recourse usage, the resource manager wants to adopt Cloud resource provisioning based on application requests without violating their execution deadlines. How can provenance be utilized to calculate the minimal resource requirement for each application request?

Scenario 4: The CAMERA project hosts 800 data sets over 800 metagenomic and genomic data sets (>48 billion base pairs, 120 million reads) and around 20 workflows based scientific applications [24]. These scientific applications run on a computer cluster against the huge reference dataset. Each application will be executed repeatedly by many scientists using different query datasets. Scientists want to know the estimated time for their submitted execution requests. How can the provenance recorded from existing executions be utilized to predict the execution time for a new execution?

These four scenarios demonstrate both computing and data oriented nature of provenance information related to Big Data analysis. The first two are more concerned by the data lineage of big datasets. The last two are more concerned by the performance issues using provenance.

B. Challenges

Several challenges of Big Data provenance arise from the above scenarios:

Challenge 1: The provenance data from Big Data workflows is too large. To get a fine-grain provenance tracking of a workflow execution like above scenario 1, the recorded provenance could easily be several times larger than the original data to be processed [7]. This large provenance data should either be saved efficiently, or reduced without comprising its targeted capabilities.

Challenge 2: Provenance collection overhead during workflow execution is too much. There is always an execution overhead when recording provenance on top of the computation cost related to the analysis. This overhead problem often gets worse for Big Data workflows due to their distributed nature. A challenge is to minimize the provenance collection overhead. This challenge is related to all four scenarios in previous sub-section.

Challenge 3: It is hard to store and integrate distributed provenance. The provenance of UDFs running on Big Data systems is often initially saved on distributed non-permanent nodes. The information collected needs to be either communicated as the analysis is happening or stitched together in the end. The first choice generates a lot

of communication overhead, but is useful to monitor the application progress. The second choice is more efficient, but requires an additional step to upload the information before freeing the computation nodes. The stitching of the data to be centralized in both choices requires additional integration steps. This challenge is related to all four scenarios in previous sub-section.

Challenge 4: It is hard to reproduce an execution from provenance for Big Data applications. Many existing provenance systems only record intermediate data generated during execution and their dependencies. Execution environment information, which is also important for reproducibility, is often neglected. Execution environment information includes the hardware information and parameter configurations of Big Data engines. We have found this information is not only critical to execution performance but also could affect the final results. For instance, our application in [14] partitions data across multiple nodes and ensemble the results in the end using a voting mechanism. The same voting mechanism might get different results depending on how data is partitioned. This challenge is related to the first scenario in previous subsection.

IV. State of the Art in Big Data Provenance

There have been a few key studies to explore modeling and capturing provenance information for Big Data workflows. In this section, we provide a short summary of the efforts that tackle the challenges summarized in previous section. Some other studies on Big Data provenance [8, 27, 28, 29] are not detailed here since we focus on those that are in the context of DDP and have experimental analysis.

Kepler Distributed Provenance Framework [7] is our previous work on Big Data provenance. The paper proposes a data model that is able to capture provenance inside MapReduce jobs as well as the provenance of non-MapReduce workflow tasks. It utilizes the Kepler DDP architecture to record and query provenance in a distributed fashion on a MySQL Cluster. It also provides an API to query the collected provenance. The scalability of collecting and querying provenance is evaluated using the WordCount application and a bioinformatics application called BLAST.

RAMP (Reduce And Map Provenance) [15, 16] is an extension to Hadoop that supports provenance capture and tracing for MapReduce workflows. RAMP captures fine-grained provenance by wrapping Hadoop APIs. This automatic wrapper-based approach is transparent to Hadoop and users. RAMP imposes some time and space overhead during provenance capture and enables efficient backward tracing.

HadoopProv [17] modifies Hadoop to implement provenance capture and analysis in MapReduce jobs. The target is to minimize provenance capture overheads. It traces treats provenance in Map and Reduce phases separately. It also defers construction of the provenance graph to the query stage by joining intermediate keys of the Map and Reduce provenance files.

Pig Lipstick [18] proposes a provenance framework that combines database-style (fine-grained dependencies) and workflow-style provenance (coarse-grained dependencies) on top of Pig Latin. It proposes a comprehensive and compact graph-based representation of fine-grained provenance for workflows which yields a richer graph model than the OPM standard [19] used for workflows. It defines three graph transformation operations to enable novel workflow analysis queries.

We summarize the above related work in Table I, from which we can see that there are still a few limitations. First, these studies focus on a specific Big Data engine. It is not easy for applications that want to track provenance with another Big Data engine. Second, none of the systems record environment information in provenance so far, which is difficult or impossible to reproduce workflow execution. Third, the usages of provenance are limited to querying and data lineage building. We will explain more provenance utilization opportunities, such as provenance mining, in the next section.

V. Opportunities for Big Data Provenance

From our analysis of the scenarios and challenges presented in Section III and related work summary in Section IV, we have formulated several research opportunities in the area of Big Data provenance. In this section, we categorize those opportunities into seven sub-sections. To illustrate how these opportunities could help with day-to-day applications, Table II lists which Big Data provenance capabilities are required for the scenarios described in Section III. The following paragraphs identify some of the research opportunities to start building these capabilities.

A. Big Data Provenance Model

Current provenance models and standards like OPM [19] might need to be extended for Big Data provenance. Additional information could be part of provenance model includes data quality, data compression and execution environment information. Also, provenance data for tasks inside of DDP patterns need to be linked with provenance for tasks outside of DDP patterns. Further, provenance model needs to be flexible or extensible to fit different specific environments and/or heterogeneous datasets.

B. Big Data Provenance Recording

Current provenance recording approaches mainly listen to the notifications of internal state changes. We argue that this internal provenance should be combined with external provenance to be more complete and useful for Big Data provenance. External provenance includes static and runtime dynamic software and hardware information, parameter configurations of Big Data engines. For instance, Hadoop has more than 200 parameters, many of which will determine not only an execution's performance but also whether it can finish successfully. The external provenance often can be collected by third-party tools, such as software profiling tools like gprof[7] and valgrind[8], and system monitoring tools [30]. The external provenance is often at external locations, such as Big Data configuration and log

[7] https://sourceware.org/binutils/docs/gprof
[8] http://valgrind.org

folder. The internal provenance recording components need to integrate third-party tools and provenance data to get holistic provenance information.

Provenance recording also needs to address the size, overhead and storage challenges listed in Section III. It needs to know whether and how to record the whole input data at provenance storage, whether and how to record the data partitions distributed across computing nodes, how to minimize the provenance recording overhead. Further, real-time data ingested for analytics may be large and ephemeral. Novel techniques are needed for provenance recording of real-time data that are not possible for batch data.

C. Big Data Provenance Query and Sharing

Additional provenance query APIs might be needed for Big Data provenance. For example, we need either different query APIs for different levels of capture and granularity, or query APIs that can specify granularity level. From performance perspective, distributed file and database systems can be utilized to achieve scalable provenance query. For provenance sharing, service based queries will be more efficient than direct data transfer for large provenance data.

D. Big Data Workflow Execution Reproducibility based on Provenance

We need to know the minimal provenance information requirement to reproduce a Big Data workflow execution. Since Big Data workflow execution might take a long time in a very large environment, reproducing the whole execution with identical environment and parameter settings might be too costly or impossible. Simulation based reproducibility or partial reproducibility are often more practical. Users might be only interested in the last a few steps or the whole workflow execution with only partial input data. Smart re-run techniques are often useful to determine the minimal steps to run or minimal provenance to record for each reproduce requirement [20].

E. Provenance based Big Data Workflow Performance Prediction

One capability of mining Big Data provenance is to predict future workflow execution performance. If a system records provenance of all workflows it supports, we could use all provenance together to predict the performance of a future workflow execution. There have been some Big Data application performance analytics models [21, 22]. By combining these models and provenance data, we could have more accurate prediction.

F. Provenance based Big Data Workflow Provisioning and Scheduling

Provenance could also be used to help find the best computing resource allocation requirement and scheduling plan for a new workflow execution. Users may have some objectives for their workflow executions, such as execution deadline, monetary cost or both. By looking into the provenance data on execution times, execution costs and environments used by previous workflow executions, we could use machine learning techniques to model the correlation between execution objectives and resource allocations for each workflow. With allocated resources, we could further utilize provenance to help with scheduling each task of the workflow and determining Big Data parameter configurations.

G. Provenance based Data Quality Analysis and Management

Data lineage from provenance has always been a good approach to measure data quality based on input data and data transformation [32, 33]. Current approaches normally use process-based analysis to tell how one dataset's quality depends on which other datasets. In the Big Data era, data quality issues are more challenging because the huge volume and wide variety of data used in an application. Overall quality assessment of a very large dataset is often not enough. We need finer granularity data quality assessment for Big Data. We argue that provenance based data quality should be analyzed from more dimensions. First, we can use sampling or other techniques to determine the quality of each data subset, especially for scenarios that data is partitioned first to achieve parallel batch processing. Second, we might need to check data quality for each time window. It is more suitable for data streaming applications where data quality could suddenly deteriorate due to hardware or weather reason.

VI. Big Data Provenance Reference Architecture

We propose a reference architecture for Big Data provenance platform based on the challenges and opportunities identified in this paper. As Figure 1 shows, the architecture has four main sub-systems: Big Data access, distributed Big Data platform, provenance, applications utilizing provenance. We will explain them separately in this section. Each of these sub systems interact with respective live data and compute engines to gather required provenance data. Following this reference architecture, system developers will select and decide on components in each sub-system based on the targeted provenance usage scenario and capability.

A. Big Data Access

This sub-system is on how to access the different types of Big Data for provenance and distributed systems. Based on the feature of each specific dataset and experiment requirements, system developers need to find the best mechanism/tool to access it. This requires a good coordination between the data and compute systems. Provenance tracking of the execution is also based on this coordination.

B. Distributed Big Data Platform

This sub-system provides construction and execution support for Big Data applications and storage support for Big Data provenance on top of distributed Big Data platforms. Application developers will choose one Big Data workflow system to build their applications, then run them with a specific Big Data Engine. Some systems like Spark can act as both workflow construction tool and DDP execution engine. A Big Data application can be built by wrapping legacy tools or direct programming using DDP programming models. Proper (distributed) databases or file systems also need to be selected and integrated for provenance storage.

C. Provenance

This sub-system determines which provenance information will be recorded and how to record it. We categorize the provenance into three dimensions: data, lineage, and

environment. Rebuilding the exact state of the experiment across these three dimensions is essential to reproduce any data-driven scientific experiment. Data Provenance captures the state of input, intermediate and output data at the time of the experiment. It will choose the best compression algorithms depending on the input data format. Lineage Provenance stores the computational activity of the experiment, which is captured by the storing the instructions that operated on these input datasets. System Provenance collects information about the exact state of the system configuration, which includes both hardware specifications and system-software specifications (OS, libraries, third party tools, etc.).

D. Applications Utilizing Provenance

As explained in Section V, Big Data provenance brings big opportunities, especially on how we could utilize it. As a special type of Big Data, Big Data provenance can be used for provenance query, experiment reproduction, provenance mining, experiment monitoring, data quality management, experiment fault tolerance and many others. Each capability can be a standalone application or internal component in a larger system.

VII. In-Progress Efforts in Kepler

As analyzed above, our current Kepler provenance system cannot well support the Big Data provenance challenges and opportunities yet, especially on provenance usage and environmental provenance recording. Following the reference architecture in Section VI, we are extending Kepler provenance framework. In this section, we summarize some technical efforts. All of them are still under development and this is our first time to discuss them in a publication.

A. Extensible Big Data Provenance Model

We are extending Kepler Provenance Data Model to include execution environment data. To make the new provenance data model flexible and extensible, its database table for environment information is key-value based. It means the same schema can be used to collect different types of environment information, such as local machine, cloud resources, cluster resources and GPU resources.

B. Adaptive Big Data Provenance Recording

By using Kepler sub-workflow to express the processes in DDP patterns, the same provenance recording system can work with multiple Big Data engines. It can be easily configured to switch provenance recording from one Big Data engine to another. There are cases where users are only interested in coarse-grained provenance, not what happens in each DDP function execution. It can be achieved by configuring provenance not to record the sub-workflows within the DDP actors.

C. Environment Information Collection for Workflow Reproducibility and Performance Prediction

Based on the analysis of existing provenance archives for previous Kepler workflow runs, we identify five types of metrics for workflow reproducibility and performance prediction. These metrics are: 1) Static environment information (CPU, memory, core number, GPU,

storage, etc.); 2) Runtime dynamic environment information (usage of the above static environment info, queue, workload, etc.); 3) Prediction time dynamic environment information (queue, workload, etc.); 4) Code/Tool/Actor profiles (programming model environment requirement, data size, parameter values, etc.); 5) Workflow profiles (actor number and type, structure, execution environment type, parameter values, etc.). The collected metrics can also be used for workflow characterization, workflow classification, and workflow performance variability.

D. Provenance Mining

We are investigating how to learn useful knowledge from Big Data provenance. We are focusing on provenance based workflow performance prediction and resource provisioning. We identify two types of workflow performance analysis approaches, namely, single workflow performance prediction and collective workflow performance prediction. The two approaches are illustrated in Figure 2. The first task predicts the execution performance of a new instance of a workflow based on its own execution history. The second one predicts the execution performance of a workflow based on not only its own history but also on other similar workflows' execution histories. Collective workflow performance prediction uses all available provenance data collectively for the next workflow execution. It is more suitable for workflows that have no or little execution provenance, which is a cold-start problem.

For collective workflow performance prediction, we found that although workflows can have various structures and tasks, there is a lot of commonality at the task level. In Kepler, each task is described using an actor and the same actor is often reused in many workflows. Based each actor's provenance data from executions of the same or different workflows. We can train a machine learning model for each actor and predict its performance in the next execution. By combining actor performance prediction based on workflow structure, we can get a good estimation for a new workflow's execution.

We have employed Spark MLlib [9] to implement a decision tree model on top of Kepler provenance. Spark MLlib is built on top of Spark system which is for large-scale data processing. Using Spark provides an efficient and scalable way to process large-scale provenance data. Spark MLlib is a scalable machine learning library including common learning algorithms and utilities. Decision trees can be used for regression modeling, and thus are applicable to workflow performance prediction. After the training phase, the model can be used to predict new workflow execution based on its parameter values.

VIII. Conclusions and Future Work

In this paper, we presented our analysis of the effects of Big Data on tracking and utilization of data and process provenance in workflow-driven analytical and scientific applications. We presented some of the specific challenges we faced in this while building such applicatons, and some open research questions that arise from these challenges. We hope this analysis will serve as a starting point expansion of this discussion at the workshop and other Big Data venues. We built an initial reference architecture for a Big Data provenance platform that we

[9] http://spark.apache.org/mllib/

can use as a testbed in response to the needs of the four scenarios we outlined. We also presented our in-progress approach to model, record, query and mine Big Data provenance. We would like to conclude by noting that although there are overhead and cost related challenges related to provenance, not collecting it has huge implications on the reliability, veracity and reprodicibility of Big Data analysis efforts. Working on any of the outlined ambitious challenges has a potential to make impact in this important area.

As a part of future work, we will create experimental archives for the workflow community to utilize. We will also work with our colleagues in the IPPD project (see acknowledgements) on a data model, ontology and PROV [1] standard extensions. We will conduct experimental analysis of learning from the Big Data workflow provenance and other related system and tool level information to gain more insight on the execution requirements of workflows and provisioning of resources required for workflow execution. We will conduct a cost analysis of provenance recording and extend our approach for adaptive provenance recording decisions. We will also explore case-driven scenarios for compression of provenance data.

Acknowledgments

This work is partially supported by NSF DBI 1062565 and 1331615, NIH P41 GM103426 for NBCR and R25 GM114821 for BBDTC, and DOE DE-SC0012630 for IPPD.

References

1. W3C Working Group. PROV Model Primer. http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/
2. Altintas, I., Barney, O., Jaeger-Frank, E. Provenance collection support in the kepler scientific workflow system. Proc. 1st International Provenance and Annotation Workshop (IPAW 06); Berlin Heidelberg: Springer; 2006. p. 118-132.
3. Davidson SB, Boulakia SC, Eyal A, Ludäscher B, McPhillips TM, Bowers S, Anand MK, Freire J. Provenance in Scientific Workflow Systems. IEEE Data Engineering Bulletin. 2007; 30(4):44–50.
4. De Oliveira, FT., Murta, L., Werner, C., Mattoso, M. Using provenance to improve workflow design. Proc. 2nd International Provenance and Annotation Workshop (IPAW 08); Berlin Heidelberg: Springer; 2008. p. 136-143.
5. Chirigati, FS., Shasha, D., Freire, J. ReproZip: Using Provenance to Support Computational Reproducibility. Proc. USENIX Conference on Theory and Practice of Provenance (TaPP 13); 2013.
6. Crawl, D., Altintas, I. A provenance-based fault tolerance mechanism for scientific workflows. Proc. 2nd International Provenance and Annotation Workshop (IPAW 08); Berlin Heidelberg: Springer; 2008. p. 152-159.
7. Crawl, D., Wang, J., Altintas, I. Provenance for MapReduce-based Data-Intensive Workflows. Proc. 6th Workshop on Workflows in Support of Large-Scale Science (WORKS11) at Supercomputing 2011 (SC2011) Conference. ACM 2011; p. 21-29.
8. Glavic, B. Big data provenance: challenges and implications for benchmarking. In: Rabl, T.Poess, M.Baru, C., Jacobsen, H., editors. Specifying Big Data Benchmarks. Springer; Berlin Heidelberg: 2014. p. 72-80.
9. Jagadish HV, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, Shahabi C. Big data and its technical challenges. Communications of the ACM. 2014; 57(7):86–94.
10. TechAmerica. Demystifying Big Data - A practical guide to transforming the business of government. TechAmerica Foundation's Federal Big Data Commission. 2012. https://www-304.ibm.com/industries/publicsector/fileserve?contentid=239170

11. Wang, J., Crawl, D., Altintas, I., Li, W. Computing in Science & Engineering. Vol. 16. IEEE; Jul-Aug. 2014 Big Data Applications using Workflows for Data Parallel Computing; p. 11-21.
12. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A. Pig latin: a not-so-foreign language for data processing. Proc. 2008 ACM International Conference on Management of Data (SIGMOD 08); New York, NY, USA: ACM; 2008. p. 1099-1110.doi:http://doi.acm.org/10.1145/1376616.1376726
13. Olston, C., Chiou, G., Chitnis, L., Liu, F., Han, Y., Larsson, M., Neumann, A., Rao, VB., Sankarasubramanian, V., Seth, S., Tian, C., ZiCornell, T., Wang, X. Nova: continuous Pig/Hadoop workflows. Proc. 2011 international conference on Management of data, (SIGMOD 11); New York, NY, USA: ACM; 2011. p. 1081-1090.doi:http://doi.acm.org/10.1145/1989323.1989439
14. Wang, J., Tang, Y., Nguyen, M., Altintas, I. A Scalable Data Science Workflow Approach for Big Data Bayesian Network Learning. Proc. 2014 IEEE/ACM International Symposium on Big Data Computing (BDC 2014); 2014. p. 16-25.
15. Ikeda, R., Park, H., Widom, J. Provenance for generalized map and reduce workflows. Proc. of 5th biennial Conference on Innovative Data Systems Research (CIDR 11); 2011. p. 273-283.
16. Park H, Ikeda R, Widom J. RAMP: A System for Capturing and Tracing Provenance in MapReduce Workflows. Proc of the VLDB Endowment. 2011; 4(12):1351–1354.
17. Akoush, S., Sohan, R., Hopper, A. HadoopProv: Towards Provenance as a First Class Citizen in MapReduce. Proc. USENIX conference on Theory and Practice of Provenance (TaPP 2013);
18. Amsterdamer Y, Davidson SB, Deutch D, Milo T, Stoyanovich J, Tannen V. Putting lipstick on pig: Enabling database-style workflow provenance. Proceedings of the VLDB Endowment. 2011; 5(4): 346–357.
19. The OPM Provenance Model (OPM). http://openprovenance.org/
20. Chen, W., Altintas, I., Wang, J., Li, J. Enhancing Smart Re-run of Kepler Scientific Workflows Based on Near Optimum Provenance Caching in Cloud. Proc. 2014 IEEE World Congress on Services (SERVICES 2014); IEEE; 2014. p. 378-384.
21. Vianna E, Comarela G, Pontes T, Almeida J, Almeida V, Wilkinson K, Kuno H, Dayal U. Analytical performance models for MapReduce workloads. International Journal of Parallel Programming. 2013; 41(4):495–525.
22. Herodotou H. Hadoop performance models. 2011 arXiv preprint arXiv:1106.0940.
23. Wang, J., Crawl, D., Altintas, I., Tzoumas, K., Markl, V. Comparison of distributed data-parallelization patterns for big data analysis: A bioinformatics case study. Proc. the 4th International Workshop on Data Intensive Computing in the Clouds (DataCloud 2013); 2013.
24. Sun S, Chen J, Li W, Altinatas I, Lin A, Peltier S, Stocks K, Allen EE, Ellisman M, Grethe J, Wooley J. Community cyberinfrastructure for advanced microbial ecology research and analysis: the CAMERA resource. Nucleic acids research. 2010:gkq1102.
25. Schadt EE, Linderman MD, Sorenson J, Lee L, Nolan GP. Computational solutions to large-scale data management and analysis. Nature Reviews Genetics. 2010; 11(9):647–657.
26. Schönherr S, Forer L, Weißensteiner H, Kronenberg F, Specht G, Kloss-Brandstätter A. Cloudgene: A graphical execution platform for MapReduce programs on private and public clouds. BMC bioinformatics. 2012; 13(1):200. [PubMed: 22888776]
27. Imran, A., Agrawal, R., Walker, J., Gomes, A. A Layer Based Architecture for Provenance in Big Data. Proc. 2nd IEEE International Conference on Big Data (BigData 2014); 2014. p. 1-7.
28. Olston, C., Sarma, AD. Ibis: A provenance manager for multilayer systems. Proc. of 5th biennial Conference on Innovative Data Systems Research (CIDR 11); 2011. p. 152-159.
29. Cheah, YW., Canon, R., Plale, B., Ramakrishnan, L. Milieu: Lightweight and Configurable Big Data Provenance for Science. Proc. IEEE 2nd International Congress on Big Data (BigData Congress 2013); 2013. p. 46-53.
30. Fuerlinger, K., Wright, NJ., Skinner, D. Effective Performance Measurement at Petascale Using IPM. Proc. 16th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2010); 2010. p. 373-380.
31. Sadiq, S. Handbook of data quality. Springer; 2013.
32. Simmhan YL, Plale B, Gannon D. A survey of data provenance in e-science. ACM Sigmod Record. 2005; 34(3):31–36.

33. Simmhan, YL., Plale, B., Gannon, D. Towards a quality model for effective data selection in collaboratories. Proc. 22nd International Conference on Data Engineering Workshops (ICDEW 06); IEEE; 2006. p. 72-72.

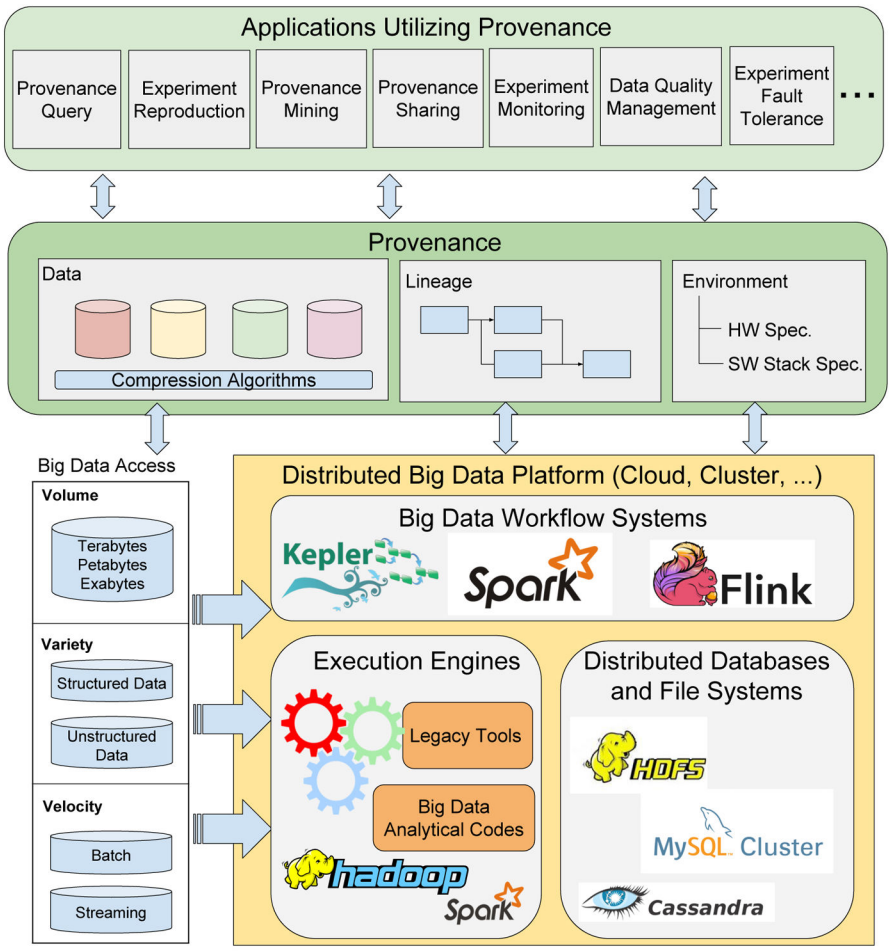

Figure 1.
Proposed Big Data provenance reference architecture

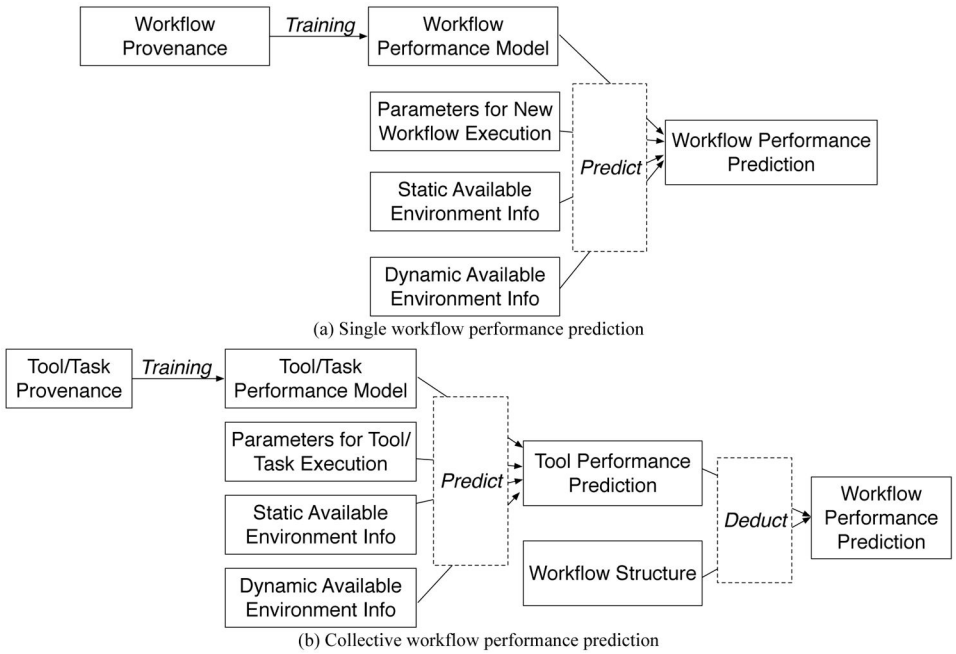

Figure 2.
Workflow performance analysis approaches.

TABLE I

Related Work Comparison on Big Data Provenance

	Provenance Recording	Applicable Big Data Engines	Provenance Usage	Environment Provenance Recording
Kepler	Parallel recording in a MySQL database	Unmodified Hadoop	Parallel query through MySQL Cluster	N/A
RAMP	Parallel recording in files	Extended Hadoop	Backward provenance tracing	N/A
HadoopProv	Parallel recording in files	Modified Hadoop	Parallel query through index files	N/A
Pig Lipstick	Parallel recording in files	Unmodified Pig/Hadoop	Graph operation based query	N/A

TABLE II

Required Big Data Provenance Capability for the Scenarios in Section III

	Provenance Model	Provenance Recording	Provenance Query and Sharing	Reproducibility	Performance Prediction	Provisioning and Scheduling	Data Quality Management
Scenario 1	X	X	X	X			
Scenario 2	X	X	X				X
Scenario 3	X	X				X	
Scenario 4	X	X			X		