Enabling Provenance on Large Scale e-Science Applications

Miguel Branco[1,2] and Luc Moreau[2]

[1] CERN, European Organization for
Nuclear Research, CH-1201 Genève
Miguel.Branco@cern.ch
[2] University of Southampton,
Southampton SO17 1BJ, United Kingdom
L.Moreau@ecs.soton.ac.uk

Abstract. Large-scale e-Science experiments present unprecedented data handling requirements with their multi-petabyte data storages. Complex software applications, such as the ATLAS High Energy Physics experiment at CERN, run throughout Grid computing sites around the world in a distributed environment, with scientists performing concurrent analysis on data and producing new data products shared among the collaboration. In this paper, we introduce a multi-phase infrastructure to achieve data provenance for an e-Science experiment. We propose an infrastructure to integrate provenance onto an existing legacy application with strong emphasis on scalability and explore the relationship between provenance and metadata introducing a model where data provenance is made available as metadata through a separate reasoning phase.

1 Introduction

Large-scale e-Science experiments are underpinned by an iterative e-scientific process by which data are produced, made available to the community, analyzed, reproduced, verified, curated and shared. The success of e-Science experiments, or computationally intensive sciences carried out in highly distributed network environments is often defined by the ability to efficiently analyze data. An aspect to this efficiency is the ability to understand the *origin of data*, which is usually referred to as *provenance*, under analysis: whether it is raw data as received from the detector or the result of possibly very complex computations. Consider the ATLAS detector for the Large Hadron Collider [14] which will deliver hundreds of megabytes of raw data per second when online. This data is expected to be available for studies for up to twenty years. Scientists will be located in their institutes anywhere in the world, analyzing data in isolation or as part of a small group or a larger community, possibly spanning multiple institutions. Their analysis will be undertaken concurrently on data, producing derived data products that may or may not be relevant to others, shared or validated.

In this paper, we present a novel provenance infrastructure suitable for large-scale applications, particularly in how it copes with storage and handling of large data volumes. We present a design with a novel reasoning phase enabling provenance to be

L. Moreau and I. Foster (Eds.): IPAW 2006, LNCS 4145, pp. 55 – 63, 2006.

made available as metadata. In addition, we introduce an approach for ease of integration of provenance onto existing legacy applications. The ATLAS High Energy Physics experiment was used as a scenario for this work.

We start by presenting the constraints for integrating provenance onto large-scale e-Science experiments, then introduce our data provenance definition and present the provenance infrastructure. Finally, we present related work and conclude.

2 Provenance for Large-Scale e-Science Experiments

Provenance is defined, in the Oxford English Dictionary, as *the place of origin of something*, or *the fact of something coming from a particular source*. It is our goal to provide *e-scientists* with the tools to the track origin of data, or *data provenance*, in their experiments.

We start by defining broadly the environment and constraints on which we base our work. Creating a model involves foremost understanding the restrictions of the domain. As such, when modeling a provenance infrastructure, we took into account the requirements and constraints typical of High Energy Physics.

We aimed to ease integration of a provenance infrastructure with the experiment's existing software infrastructure. Indeed, proposing an infrastructure that requires dramatic changes to the software of existing e-Science experiments in order to support *data provenance* might certainly be a useful exercise but would prevent widespread adoption, given the extensive legacy code base. Hence, grappling with the legacy challenge – integration with existing systems – is a core concern from start when defining, modeling and implementing the infrastructure.

Therefore, we have identified a small set of restrictions, which we believe are common to many situations involving integration with a large-scale e-Science experiment. These are: the inability to alter the majority of (deployed) software components; difficulty of layering new middleware across the application; diversity of design practices across existing software, as software components may not follow a single architectural style; usage of complex workflows with interwingled interactions, difficult to identify clearly; the presence of a highly distributed environment for producing and analyzing data as well as for developing software.

Another restriction is the difficulty in understanding the very complex workflows in production. As an example, a single ATLAS workflow involves several gigabytes of software libraries, many remote accesses to databases (detector conditions, calibration, …) and the set of end-user configurable options only within the ATLAS analysis framework are close to a thousand. The execution of the workflow is also a significant computing effort, run on the Grid and partitioned onto multiple tasks executed in parallel throughout computing sites around the world.

With the previous restrictions in mind we now define *provenance*. First we define a documentation record as being the view of an intervenient actor (a client or a service) on part of a process execution. We define *data provenance properties* as the relevant set of descriptive properties of the data products for end users. We also define *reasoning* as the derivation of data provenance properties involved in the execution, by applying a reasoning algorithm to a set of documentation records. Finally, we define

provenance as *the end result of applying context-specific reasoning over a set of records that document the execution of a process, with the goal of deriving a set of properties of the data products involved in the execution*.

An important factor of our definition is that provenance is to be determined in relation to the data and not meant to determine e.g. the provenance of a workflow execution. Our goal is ultimately to allow users to understand the origin of a piece of data by looking at the derived set of provenance properties for the data product under analysis.

3 Creating Provenance-Aware Applications

Provenance-aware applications are defined as applications that implement our provenance infrastructure. Legacy e-Science experiments are defined as complex, large-scale experiments with an existing code base, not designed from start to provide provenance for its end-users.

We start by introducing an important assumption regarding service-orientation architectural style and then present our provenance infrastructure.

3.1 Why Service-Oriented Architectural Style is Useful

We take the view that complex software applications are typically designed using a service-oriented approach [11] or can easily be modeled using service-orientation principles. Integration with large-scale legacy applications is facilitated if coupling the provenance infrastructure has minimal interference with the existing application. The flexibility of the service-oriented model allows for relative ease of integration with legacy systems (e.g. by identifying and wrapping the various legacy components as services). This was seen in our analysis of the ATLAS Experiment where the various components involved with the ATLAS Production System followed diverse architectural styles but were easily modeled as coarse-grained services with simple interactions.

The service-oriented architectural (SOA) style can be defined as having actors (services) which interact with one other by exchanging messages, following a service description. Deciding which components, sub-set of components or composition of components should be a service in SOA is a multi-dimensional problem exceeding the scope of our work. For what concerns our infrastructure, the finer-grained the services the greater is the potential understanding of the data flow, if we consider our infrastructure relies on a protocol (presented below) that captures message exchanges between services. Nonetheless having finer-grained services may lead to potentially recording more information that may be redundant, unimportant and expensive to record, store or analyze.

Alternate approaches to provenance often create new data and workflow modeling languages that enable provenance of data to be fully determined, even mathematically proving the models as complete. These approaches are especially common in the context of relational databases [1,3,4]. While these works provide an important foundation, the legacy challenge – integration into existing systems, particularly given by the requirement to apply new modeling languages – prohibits its applicability in some

contexts. It is precisely these scenarios that we intend to cover by providing a model that impose little restrictions and changes to existing systems.

3.2 A Provenance Infrastructure

To create provenance-aware applications, we have defined a model consisting of four phases: creating documentation, storing, reasoning and querying. The overall model is shown on Figure 1.

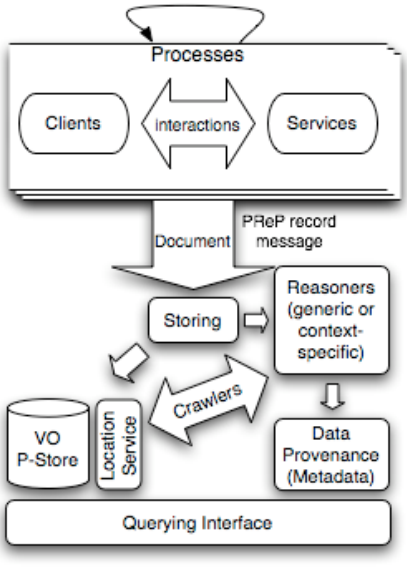

Fig. 1. Logical model for a provenance infrastructure

Documenting a Process. Documenting the execution of a process translates into creating documentation records, as shown on Figure 1. This is done in our infrastructure using PReP [9,10] which also defines a provenance store as the repository for the documentation records. PReP defines a representation of process documentation suitable for service-oriented architectures, introducing a generic protocol for recording provenance. An element of process documentation is called a p-assertion: an assertion that is made by one of the intervening actors and pertains to a process. In PReP, actors (clients and services) involved in a process may provide their view on an interaction. The flow of the process is captured by relating several assertions, along with the corresponding clients and services.

The documentation records conform to a PReP schema. PReP does not limit the scope of the assertions made by the intervenient actors – an actor may assert any information it wishes, as long as it pertains to the process - but in our usage of PReP we assume actors assert a minimal set of assertions (typically the message exchanges and the request-response relationship between messages) to facilitate integration with

legacy applications – that is, not to alter significantly its existing and deployed services, but by only wrapping the external interfaces to support PReP.

Storing Documentation Records for a Process. Once documentation records have been created, they are stored and an index must be created to efficiently locate them. The *storing* phase implements the storage of documentation records.

PReP has introduced the concept of a provenance store (p-store), as the repository for documentation records. It is expected that documentation records may themselves become a significant portion of the total data volume for large-scale experiments such as ATLAS. Therefore in our infrastructure we have extended PReP's p-store concept and defined that documentation records may be replicated onto multiple physical instances. Thus what PReP defines as a provenance store is for us a logically unique provenance store, which may in our model have multiple physical instances. We have defined the existence of a *location* service providing the single entry point to locate documentation records (given a unique identifier) across multiple instances of the same logical repository. The documentation records are write-once-read-many, meaning that contents do not change after recording. This property allows for straightforward replication of data without concerns for concurrency. In addition our design foresees (parts of) a physical instance of a p-store to be serialized on disk, and shipped around computing sites for improved accessibility or archival. Also, we defined the storage layer as providing secure access to documentation records by implementing Grid-enabled authentication and authorization.

Reasoning over Documentation Records. Reasoning consists of analyzing documentation records and extracting a set of data provenance properties for the data elements involved in the workflow execution, as required by a reasoning algorithm. This set of properties constitutes what we refer to in our model as the *data provenance* for a data product. The definition implies that data provenance is overall dependant of each particular reasoning algorithm. Often, reasoning algorithms depend on the context (e.g. for ATLAS, 'luminosity' may be considered part of the data provenance for certain categories of event data). Others are generic even across experiments from different scientific domains (e.g. the software versions used during data processing).

Reasoning is typically done asynchronously in relation to the recording or storage of documentation records: it begins after documentation records have been stored. To reason over documentation records, the records must first be found and (parts of) its contents retrieved. The generic *location* service, shown on Figure 1, allows physical locations for documentation records to be resolved, typically by giving to the location service a globally unique identifier. A reasoning algorithm may require more than a single documentation record as input. Therefore it is usually necessary to find a set of related documentation records, according to some strategy that is dependant of the reasoning being applied. A *crawler* is the component from our infrastructure that can navigate throughout the contents of the provenance repositories (the *p-store* shown on Figure 1), finding physical copies of the required documentation records. The crawler takes into account security policies for accessibility to the documentation records. Also, in experiments with a large set of documentation records, typically only a sub-group of those records can be analyzed at any given time. Restrictions range from time available for reasoning to the accessibility of stored documentation records (e.g., for ATLAS some documentation records may be archived onto tape which leads to a prohibitive

access time for retrieval). When crawlers are actively navigating the documentation records they may detect that certain records are not available, either permanently or temporarily, and feed this information back to the reasoning algorithm.

Querying Newly Defined Data Provenance as Metadata. Ultimately, the goal is to provide the output from the previously described reasoning phase to end users. This is accomplished by defining new metadata attributes, which we referred to as *properties*, associated with the original data products.

Reasoners operate on the documentation records to produce metadata that is kept on a metadata catalog. User queries are then directed to the metadata catalog avoiding the need for end-users to directly query the p-store. This two-phase model is designed to cope with large volumes of data in a scalable manner. As a side effect, the provenance infrastructure becomes a provider of metadata for the application. We call *tagging* of data to the association of data with its data provenance. Tags also serve as indexes to the data products allowing faster and user-friendlier understanding of the data.

Motivation for a Reasoning Phase. One important feature of our infrastructure is that end-users do not directly query over the documentation records contained within the provenance store, but only the metadata catalog. In the next paragraphs, we present a few motivations for this choice.

When analyzing a set of documentation records there is no guarantee that the information contained is consistent, as the documentation records may be incomplete (e.g. the process may have not been fully documented or parts of the p-store may have got lost due to a storage failure). This is particularly relevant for experiments with long lasting workflows, such as the ATLAS experiment, where producing a single dataset (single workflow) may last several months and be computed and stored throughout tens of computing sites in parallel. In some scenarios, incomplete information may still be sufficient to deduce data provenance for a piece of data, e.g. for ATLAS, if the majority of the data products part of a dataset follow a certain software version and the dataset was assigned to be processed at a single computing site, it is safe to assume that all dataset partitions follow the same software version even though the documentation records for each individual partition may not be known. This type of statistical significance is very much dependent on the particular data provenance property being derived (in the example: '*software version*' for a '*dataset*').

Another important motivation comes again from the large volumes of data we expect to handle. For ATLAS, not all documentation records may be accessible at all times (e.g. parts of the provenance store may be archived onto tape) so it is up to the implementers of the reasoner algorithm to take into account a set of context-specific assumptions and decide how the reasoning infrastructure should react.

In addition, end-users should not be given the ability to directly query documentation records, as their queries may cause large sets of the p-store to be read. This is particularly important considering the provenance store may spread many sites with different performance for data access. Finally, in ATLAS, we have identified a large set of provenance queries, which are repeatedly executed by many users (e.g. "what is the set of algorithm versions used to produce dataset X?"), requiring large portions of

the p-store to be read in order to derive the necessary provenance property. Formalizing the reasoning step so that the query to the p-store is performed only once by the infrastructure and in the most optimal access conditions to the documentation records saves many unnecessary queries, improving scalability and manageability of the store.

4 Related Work

Buneman et al [1] have presented models for determining data provenance (or lineage, or pedigree) in the context of relational databases. A technique was also presented on [2] for optimizing archival of data based on recording semantic continuity of elements, introducing two definitions: "where-provenance" - determining where a piece of data came from and "why-provenance". Cui and Wisdom [3] further studied "why-provenance" that can be defined as: "why is a certain piece of data in the database" and "what tuples in a database D contributed to some piece of data d in query Q(D)". All the work described so far has been developed on the context of relational databases, although authors claim is applicable elsewhere. [4] presents techniques to understand provenance in the context of data warehouse systems. Silva et al [6] present a model for "knowledge provenance" including proof-like information on how a question-answering system arrived at the response. Woodruff and Stonebraker [7] argue for fine-grained data lineage and present a method based on an inverse function as the methodology to determine provenance. All these models were built using an idealized, close world of databases. In this world many natural restrictions and simplifications apply and query languages are often created to satisfy the underlying data models (when new data models are not defined). The data warehouse concept cannot apply directly since there are many concurrent, possibly inconsistent, repositories available worldwide. Notions such as "distributed data", "distributed applications" have to be first- class "citizens" of our provenance model. Similarly ad-hoc approaches to provenance, such as electronic logs, can hardly cope with modern day e-Science data demands. Electronic logs are typically updated by manual user input or by simple scripts. In addition, e-logs do not necessarily have any associated reliability in the information they contain since it is mostly updated by humans. Zhao et al. [13] the provenance logs are extended by creating annotations based on concepts drawn from an ontology. This allows records to be linked with each other by means of inference over the associated concepts. COHSE (Conceptual Open Hypermedia Services Environment) integrates the ontology service with an annotation and linking service.

Chimera virtual data system [8] is based on the assumption that explicit representation of computational procedures used to derive data can be defined, enabling on-demand data generation from analyzing the expressive representation of computational procedures.

Szomszor and Moreau [5] present a model for recording and retrieving provenance on large-scale, dynamic and open environments such as those of Grids [12] and Web services where a workflow enactment engine is responsible for interacting with the VO services and is altered to submit information to a provenance repository.

This work is distinguishable from PReP [9,10] in that the former is a provenance recording protocol. We build upon this protocol, detailing the storage handling, and defining an infrastructure to reason over PReP's documentation records systematically

exposing data provenance as metadata, allowing developers to asynchronously build reasoning algorithms and providing end-users with a simplified query interface based on a metadata representation.

Finally, large-scale legacy e-Science experiments require flexible methodologies to handle the processing, publishing, analysis, verification and curation of data and none of these alternate approaches was particularly designed to take into account the naturally distributed environment on which these experiments work.

5 Conclusion

The proposed provenance infrastructure enables provenance-awareness for large-scale e-Science experiments, particularly those handling large volumes of data. The infrastructure allows for provenance recording, storage, reasoning and querying by identifying the multiple stages for provenance integration in an application. It builds upon PReP by specifying how the provenance recording protocol can be integrated with a legacy application, handling large volumes of data in the provenance store and with a separate reasoning phase. Asynchronous reasoning algorithms allow for a flexible and scalable framework for data provenance whose representation relies on a metadata catalog. In terms of future work we intend to continue prototyping the model identified in this paper and evaluate its applicability on a set of scenarios in the context of the ATLAS Experiment.

References

1. P. Buneman, S. Khanna, and W.C. Tan. Data provenance: Some basic issues. In Foundations of Software Technology and Theoretical Computer Science, 2000.
2. P. Buneman, S. Khanna, K.Tajima, and W.C. Tan. Archiving scientific data. In Proc. of the 2002 ACM SIGMOD International Conference on Management of Data, pages 1–12. ACM Press, 2002. ISBN 1-58113-497-5.
3. Y. Cui and J. Widom. Practical lineage tracing in data warehouses. In Proceedings of the 16th International Conference on Data Engineering (ICDE'00), San Diego, California, February 2000.
4. J. Widom Y. Cui. Lineage tracing for general data warehouse transformations. In The VLDB Journal, pages 471–480, 2001.
5. M. Szomszor and L. Moreau. Recording and reasoning over data provenance in web and grid services. In International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE'03), volume 2888 of Lecture Notes in Computer Science, pages 603–620, 2003.
6. R. McCool P. Silva, D. McGuinness. Knowledge provenance infrastructure. IEEE Data Eng. Bul l., 26(4):26–32, 2003.
7. Woodruff and M. Stonebraker. Supporting fine-grained data lineage in a database visualization environment. In ICDE '97: Proceedings of the Thirteenth International Conference on Data Engineering, pages 91–102, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-7807-0.
8. I. Foster, J. Voeckler, M. Wilde, and Y. Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation, 2002.

9. P. Groth, M. Luck, and L. Moreau. Formalising a protocol for recording provenance in grids. In Proc. of the UK OST e-Science second Al l Hands Meeting 2004 (AHM'04), Nottingham, UK, September 2004.
10. P. Groth, M. Luck, and L. Moreau. A protocol for recording provenance in service-oriented grids. In Proceedings of the 8th International Conference on Principles of Distributed Systems (OPODIS'04), Grenoble, France, December 2004.
11. Munindar P. Singh and Michael N. Huhns. Service-Oriented Computing: Semantics, Processes, Agents. John Wiley & Sons, Ltd., 2005.
12. I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. Lecture Notes in Computer Science, 2001.
13. Jun Zhao, Carole Goble, Robert Stevens and Sean Bechhofer. Semantically Linking and Browsing Provenance Logs for e-Science. In International Conference on Semantics of a Networked World Revised Selected papers, Springer LNCS 3226 pp 157-174, Paris, France, June 17 -19, 2004.
14. ATLAS Computing Group, ATLAS Computing Technical Design Report, http://doc.cern.ch/archive/electronic/cern/preprints/lhcc/public/lhcc-2005-022.pdf, June 20, 2005