Challenges for Provenance in Cloud Computing

Imad M. Abbadi and John Lyle
Department of Computer Science, University of Oxford
Email: firstname . lastname @cs.ox.ac.uk

Abstract

Many applications which require provenance are now moving to cloud infrastructures. However, it is not widely realised that clouds have their *own* need for provenance due to their dynamic nature and the burden this places on their administrators. We analyse the structure of cloud computing to identify the unique challenges facing provenance collection and the scenarios in which additional provenance data could be useful.

1 Introduction

Cloud computing is an increasingly popular approach for the processing of large data sets and computationally expensive programs. This includes scenarios that have clear requirements for maintaining the *provenance* of data, including eScience [5] and healthcare [15], where assurance in the quality and repeatability of results is essential. In addition, clouds have their own application for provenance: the identification of the origins of faults and security violations. However, cloud systems are structured in a fundamentally different way from other distributed systems, such as grids, and therefore present new problems for the collection of provenance data.

In this paper we describe the challenges in collecting provenance for cloud computing and identify where additional work is needed. We believe that there is an opportunity for creating new, practical provenance systems for clouds to support scientific computing as well as satisfying current requirements for better debugging, auditing, forensics, billing and security. These systems must reflect the unique nature of clouds and should take advantage of existing research from grid computing.

1.1 Background: provenance in the cloud

Cloud computing has many competing definitions. The following will be used in this paper:

> 'Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.'[10]

There are three commonly-discussed types of clouds. Infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and software-as-a-service (SaaS). IaaS refers to the provision of virtualized hardware on which the client can run their own operating system and software stack. In PaaS, the operating system and environment are provided and maintained for the client, who then runs their own applications. In SaaS the cloud provider runs and organises the entire software system and provides a specific service.

Provenance, on the other hand, is better defined. It generally refers to information that 'helps determine the derivation history of a data product, starting from its original sources' [14]. This information is clearly valuable in data-intensive computing scenarios, such as scientific computing [12], to provide assurance in quality of results [8] and ensure the repeatability of experiments.

We observe that the problem (if not the concept) of provenance should also be familiar to anyone involved in debugging IT systems. System administrators must identify where an error originated, what caused it and the effects it had. This is particularly true of security violations, and provenance records are closely related to data forensics. These tasks are usually supported through *logging* and *auditing*. This is particularly difficult in complex systems with multiple layers of interacting software and hardware such as a cloud. Clouds are dynamic and heterogeneous by definition [4], and involve several components provided by different vendors which must interoperate. Tracing the origins of faults on cloud infrastructures involves the collection of evidence and data from

diverse sources with difficult to determine causes and effects. Cloud computing therefore is a good example of a situation where the introduction of better provenance data could provide immediate benefits for system administrators as well as users.

As an aside, we note that *logs* and *provenance data* are distinctly different. Logs provide a sequential history of pre-defined actions usually relating to a particular application. Provenance data refers to the history of the *origins* of a particular data object, with perhaps greater requirements for assurance and semantics. Provenance goes beyond an individual application and may refer to many pieces of equipment as well as people. Throughout this paper we refer to logs as being a source of provenance, primarily because in cloud systems, when present, they are used in combination for a similar purpose.

1.2 Related work

The need for additional provenance information in cloud computing storage has been well established by Muniswamy-Reddy et al. [11, 12]. The authors have discussed the requirements for adding data provenance to cloud storage systems and have analysed several alternative implementations. This is in contrast to our work, which considers the entire cloud infrastructure and identifies that existing (but inadequate) tools are already being used for provenance.

The use of provenance for fault tolerance has also been proposed before for grid computing [7, 16]. One aim is to avoid common modes of failure when attempting to use multiple composite web services. This work provides useful motivation for the collection of provenance data, but the move to cloud computing requires a new analysis of current problems in the collection of provenance data.

There are many promising tools which could be adapted for use in cloud environments. Muniswamy-Reddy et al. [11, 12] have already evaluated the use of PASS – the Provenance-Aware Storage System – for cloud provenance. Reilly and Naughton [13] have proposed extending the Condor batch execution system to capture data on execution environments, machine identities, log files, file permissions and more. While there are significant new challenges on a cloud infrastructure, the Provenance-Aware Condor system certainly collects the right kind of provenance data.

2 Cloud dynamics

In this section we present a taxonomy of cloud infrastructures, detailed discussion of which can be found in previous work [1], and then describe its dynamic nature and the challenges this presents for provenance collection.

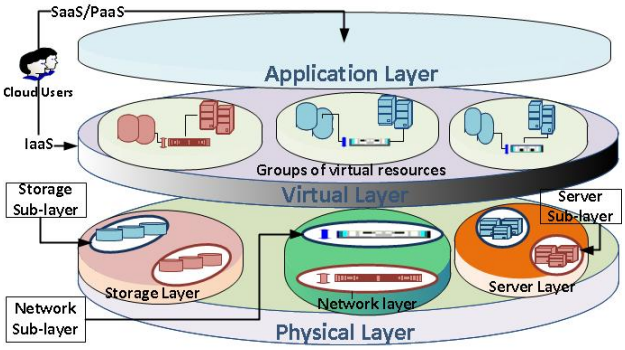

Figure 1: Cloud Taxonomy: 3-D View (source [1])

Cloud infrastructure can be represented as a 3-D cylinder, which can be sliced horizontally and/or vertically (see Figure 1) into layers. A layer represents cloud resources that share common characteristics. The layering concept helps in understanding the relations and interactions between cloud resources. We use the nature of the resource (i.e. physical, virtual, or application) as the key characteristic for horizontal slicing of the cloud. For vertical slicing, on the other hand, we use the function of the resource (i.e. server, network, or storage) as the key characteristic for vertical slicing.

As illustrated in Figure 1, vertical slicing of the physical layer results in three layers: storage layer, server layer, and network layer. Each layer is organized into sub-layers, each of which provides specific properties to serve the needs of the wide range of cloud user requirements. Server, network and storage sub-layers are organized into multiple *collaborating sub-layers*. Sub-layers within each collaborating sub-layer and their resources are carefully selected, interconnected, and even physically positioned to support the overall collaborating sub-layer properties.

Virtual resources are then created and grouped at the virtual layer based on user application requirements. Multiple related *groups* join a *collaborating group* based on user requirements and the nature of the application (e.g. dependency amongst application resources). Sub-layers and groups are associated with properties and policies, which are important for managing the infrastructure. Sub-layers' properties and policies are infrastructure related, while group properties and policies are related to user's application requirements. Each group is hosted at a collaborating sub-layer with physical properties that best match user properties.

Cloud resources communicate in a well organised way, either horizontally and/or vertically, which is defined as follows (for further details see [4]):

Horizontal communication. This is where cloud resources communicate as peers within a layer, sub-

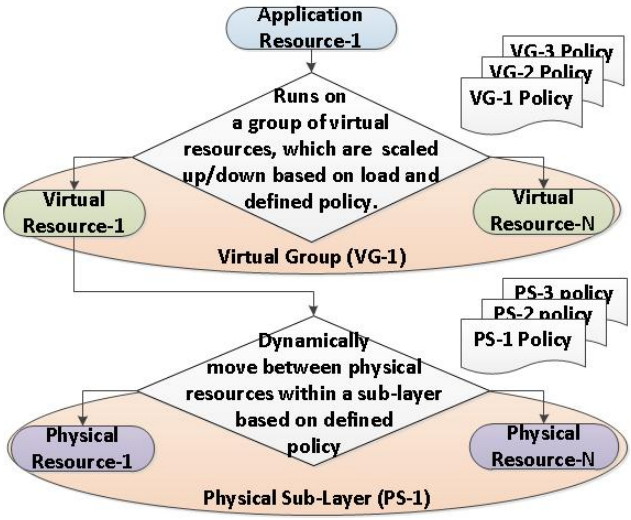

Figure 2: Dynamics of the Cloud

layer, or group. There are many examples of horizontal communication, such as replicating files between peers or virtual machines synchronising shared memory.

Vertical communication. This is where cloud resources communicate with other cloud resources in the same layer or another layer following a process workflow in either an up-down or down-up direction. This would typically work as follows. First, an upper layer's resource runs a process which generates sub-processes that must be run at lower layers. Then the lower layer processes the sub-processes and then sends the outcome to the upper layer. These steps represent an up-down communication channel. Each layer in turn sends their response back in the opposite direction, which represents the down-up communication channel.

Cloud resources are dynamic and hierarchical (see Figure 2). By dynamic we mean the following: (a.) a specific virtual resource can be hosted at many different physical resources at different times according to a policy; (b) similarly a specific application resource can run on multiple virtual resources that are increased or decreased based on load and a predefined policy controlling such behaviour (i.e. elasticity property [1]); and (c.) from (a.) and (b.) we can conclude that a specific application can be hosted under different physical servers.

This is not to say that a cloud's virtual or application resources can run anywhere. The opposite is true: as we discussed earlier cloud resources are well controlled and managed following policies controlling the limits of movement; i.e. an application resource can move within a specific group of virtual resource (e.g. horizontal scalability [1, 2]), and similarly a virtual resource can move within a physical resource's sub-layer boundary [1, 4] .

The dynamic nature of clouds has advantages such as resource consolidation, resilience, scalability and high availability [1, 2]. However, this results in new security, logging and auditing challenges. These need to be solved for a cloud-hosted system requiring provenance collection, and the solutions may in turn improve the other uses for log and audit data. In the next section we identify and categorise cloud logging, auditing and historical data and then derive the challenges.

3 Cloud logging and auditing

Logging, auditing and historical data are of tremendous importance in a cloud. This is especially the case as a cloud is expected to support Internet-scale critical applications. Logging, auditing and historical data have different usage, e.g. pro-active service delivery (incidents monitoring and security monitoring), error investigation, billing, and forensic investigation. Almost all of a cloud's resources generate this data in some way. The importance of such data and its usage is based on the following resource types.

Physical resources generate information related to physical resource status, security and incident reporting. The generated data helps in the direction of finding the cause of incidents and for security monitoring. Cloud providers and forensic investigation teams are the main parties interested in this data.

Virtual resources generate information related to virtual resource status, security and incident reporting. They also generate usage data, which are used for billing customers using IaaS clouds. Cloud providers, forensic investigators, auditors and IaaS cloud customer are the main parties interested in this data.

Application resources generate data related to application resource status, security and incident reporting. They also generate usage data that are used for billing customers using PaaS and SaaS clouds. Cloud providers, forensic investigators, auditors, and cloud customers are the main parties interested in this data.

4 Challenges for cloud provenance

At present, the only way that provenance is provided on a cloud is through linking together log and audit data, collected from multiple resources, to provide the complete history of an event or result. This is no substitute for a

purpose-built provenance system, like those designed for grid systems [14], but is the approach used in practice by clouds for many of the same purposes.

As discussed, cloud systems are composed of dynamically interlinked resources. This means that building a logical sequence of events to investigate an incident for any one application requires data from many sources. These include the application itself, all logs for possible virtual resources that the application could have used, and logs of all physical resources that virtual resources could have used. Administrators must then combine this data correctly by identifying all time intervals when an application used a specific virtual resource, all possible time intervals when these virtual resources used physical resources and then all relevant log files from all related resources. Collecting and combining data from these resources is not easy or practical considering the potential scale of cloud systems.

Therefore, we propose that all layers, sub-layers and groups of a cloud system should incorporate a mechanism to support the collection of linkable data providing the *provenance* of events related to a specific activity.

We now discuss the importance of provenance in a cloud using two simple example scenarios illustrated in Figure 3. We assume that a cloud provider has six physical servers PS_1 to PS_6, and two collaborating sub-layers L_1 and L_2. L_1 is allocated physical servers PS_1 to PS_3, and L_2 is allocated physical servers PS_4 to PS_6. We also assume that the cloud provider hosts an application App. The cloud provider creates group VD_1 in the virtual layer to run App. VD_1 is initially allocated a one virtual resource, VR_1, to host App. VD_1 is associated with a policy allowing it to scale its resources when there is an increase in demand using resources from physical sublayer L_1.

Our first example demonstrates how a simple increase in load, and the corresponding reaction from the cloud, can result in a loss of provenance data.

1. Assume the load on App has dramatically increased.
2. VD_1 responds by instantiating a new virtual resource VR_2 replicating VR_1 inside VD_1.
3. Now both VR_1 and VR_2 process App, which are hosted using L_1. Assume that VR_1 is hosted by PS_1 and VR_2 is hosted by PS_2.
4. PS_2 has hardware problems, which results in incorrect results being generated by App.
5. Load returns to normal and so VD_1 downscales by removing VR_2.
6. Cloud customers discover the problem and call the cloud provider.

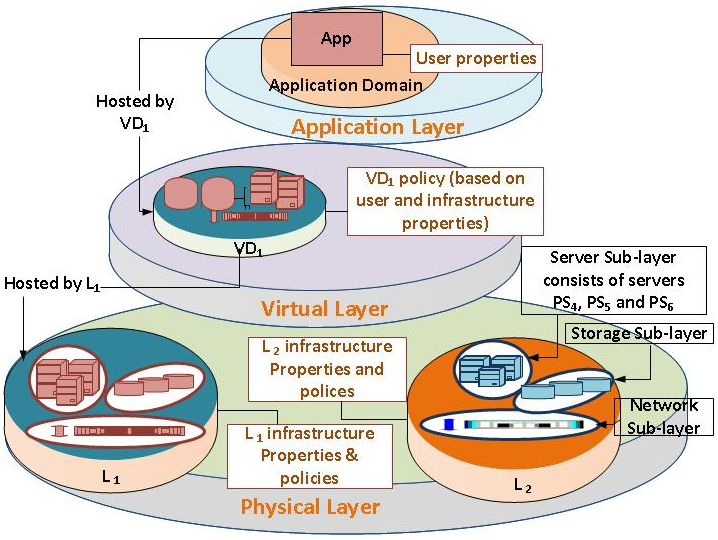

Figure 3: Provenance Scenario

If the cloud provider only examines the logs of files generated by VR_1 and PS_1, then they will not find the root case of the problem or how to rectify it.

Our second scenario focus on forensic provenance in the cloud, as follow.

1. A system administrator reads the policy for VD_1 and understands that App can only be hosted using L_1 resources.
2. The administrator updates the VD_1 policy to force VD_1 to use L_2 resources
3. The administrator then connects to L_2 physical resources and finds out that VD_1 resources are running on PS_4, meaning that App is hosted there. The system administrator connects to PS_4 and indirectly extracts important information from App. PS_4 logs this activity.
4. The administrator restores the original policy, which forces VD_1 resources to switch back to L_1.

If the cloud provider only examines log files generated by L_1 resources, then they will not discover who performed the attack or, even worse, they might never discover that an attack has happened in the first place. This is one of the main challenges that shows the importance of provenance considering the complex cloud infrastructure and enormous distributed resources.

5 Discussion

We have aleady mentioned some use cases for provenance in cloud systems, such as billing, forensic and incident investigation. Current mechanisms provided for this

purpose are associated with many shortcomings. Examples of such shortcomings include the following.

- The methods followed by clouds to support provenance queries are basic and, in many cases, such methods are developed on an ad-hoc basis by cloud system administrators using customized scripts to address a specific event. If the event criteria changes then it is unlikely that the original script would work without modifications; it needs to be manually customized to adapt to new changes.

- Current mechanisms are object specific; i.e. they do not automate the process of managing different log and audit files and linking dependent log and audit records together. For example, investigating an incident would be likely to require going through different log files, each of which may contain a great deal of detailed information. Only experts in the domain can identify the relationship between log files following a manual process supported by simple scripts. This process is error prone, time consuming and expensive. Such a manual process increases the mean time to discover an incident and the mean time to diagnose it. This in turn affects cloud service delivery (or operational trust), as discussed in [3].

- The mechanisms are deployed and fully controlled by cloud providers; i.e. cloud users do not have control over such mechanisms, and neither they can access logging and auditing records. Users have no option but to trust cloud providers. As a result, users cannot be certain that (for example) billing statements accurately reflect real use of resources, cannot be certain that their resources were up and running all the time, and cannot be certain about security breaches or malfunctions affecting their outsourced resources.

- Most importantly, current logging and auditing records are not reasonably protected, which in turn affects the creditability of provenance in the cloud. For example, any authorized system administrator can easily and without being detected update logging records related to physical resources, virtual resources, and even application resources.

There are several different types of cloud deployments: private, public, and community models [10]. There are major differences between such models, including the number of users, supported services, adopted business models, and relationship between cloud providers and their users. Such differences affect the cloud providers' urgency in implementing provenance systems. Abbadi [1] discusses and compares different cloud models in terms of the services they support. He identified that public cloud models support limited services in comparison with the ones provided by private and community cloud models. This is related to many factors including technical limitations and the number of cloud users of each model. However, in the future we can expect many more usage scenarios, including critical infrastructure, which will cause public clouds to adopt all services currently provided by community and private cloud models. Many challenges still need to be addressed before this adoption can happen [2, 3, 4]. One of these is providing automated self-managed services [1, 2], which are software services that can automatically and with minimal human intervention manage cloud environment availability, resilience, adaptability, reliability and scalability to consider security and privacy by design. One of the key challenges in building self-managed services is providing provenance mechanisms to support the management of complex cloud infrastructures. Private and community cloud providers have limited customers, which makes it reasonable to rely on human resources (supported by simple scripts) to provide basic provenance in the cloud. However, moving to public cloud that is expected to have significantly more users requires cloud-specific provenance mechanisms. For example, adaptability service on failure requires provenance mechanisms that would carefully provide the records leading to understand the cause of incidents (e.g. DoS attack, application error, or physical incident). Fortunately, large cloud infrastructures have enormous processing and data storage facilities, making the implementation of such facilities much more reasonable.

Another important point which indirectly depends on cloud provenance is trust establishment. Trust establishment in cloud computing requires collaborative efforts from industry and academia. As discussed by Abbadi [3], establishing trust in cloud systems requires two mutually dependent elements: (a.) support infrastructures with trustworthy mechanisms and tools to help cloud providers automate the process of managing, maintaining, and securing their systems (this includes but is not limited to self-managed services as discussed earlier); and (b.) developing methods to help cloud users and providers establish trust in the operation of the infrastructure by continually assessing its operational status. An important component in both elements is establishing trustworthy cloud provenance mechanisms [9]. By trustworthy we mean that both cloud users and cloud providers can attest to provenance mechanisms ensuring that they have performed their job as expected. One way to implement this is for cloud providers, who are interested in providing trustworthy provenance mechanisms, to support automated management services with incident

provenance describing, for example, the root cause of an incident. Cloud users, on the other hand, would be interested in this to assure them that the information provided by the cloud reflects the real operation of the cloud. For example, it would enable cloud users to ensure that the billing statements reflect real usage of resources. Another example would be attesting the integrity and enforcement of cloud resource policies, which would prevent the second scenario outlined in the previous section.

6 Conclusion and future work

We have discussed the need for cloud provenance, both for the applications a cloud infrastructure will support and for service provider requirements such as debugging, auditing and forensics. We have identified several key factors that make provenance more complicated in cloud systems and shown two examples of why it would be immediately useful in comparison to existing logging approaches.

One of the key objectives of TClouds project[1] is to establish cloud trust models. Trust models have enormous advantages to both cloud users and providers [3]. As discussed earlier, establishing trustworthy cloud computing provenance is fundamental requirement to establish such trust models. We intend to investigate in further detail the requirements for cloud computing provenance, particularly in scenarios under investigation in the TClouds project such as healthcare and public lighting. We would like to attempt both a top-down approach – specifying how a new provenance architecture could be created for clouds of varying complexity – and a bottom-up approach where existing logging systems are combined in order to satisfy requirements without disrupting existing systems. Furthermore, existing grid workflow provenance systems may be applicable for recording cloud vertical communication.

Acknowledgements

This research has been supported by the TCloud project, which is funded by the EU's Seventh Framework Program ([FP7/2007-2013]) under grant agreement number ICT-257243.

References

[1] ABBADI, I. M. Clouds' Infrastructure Taxonomy, Properties, and Management Services. In *CloudComp '11: in proceeding of the International workshop on Cloud Computing: Architecture, Algorithms and Applications (to appear)* (July 2011), LNCS, Springer-Verlag.

[2] ABBADI, I. M. Middleware Services at Cloud Virtual Layer. In *DSOC 2011: Proceedings of the 2nd International Workshop on Dependable Service-Oriented and Cloud computing (to appear)* (Aug. 2011), IEEE Computer Society.

[3] ABBADI, I. M. Operational trust in clouds' environment. In *MoCS 2011: Proceedings of the Workshop on Management of Cloud Systems (to appear)* (June 2011), IEEE Computer Society.

[4] ABBADI, I. M. Toward Trustworthy Clouds' Internet Scale Critical Infrastructure. In *ISPEC '11: Proceedings of the 7th Information Security Practice and Experience Conference (to appear)* (June 2011), LNCS, Springer-Verlag.

[5] BOSE, R., AND FREW, J. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv. 37*, 1 (2005), 1–28.

[6] CHENEY, J., Ed. *First Workshop on the Theory and Practice of Provenance, February 23, 2009, San Francisco, CA, USA, Proceedings* (2009), USENIX.

[7] CRAWL, D., AND ALTINTAS, I. A Provenance-Based Fault Tolerance Mechanism for Scientific Workflows. In *Provenance and Annotation of Data and Processes*, vol. 5272 of *LNCS*. Springer, 2008, pp. 152–159.

[8] GIL, Y., DEELMAN, E., ELLISMAN, M., FAHRINGER, T., FOX, G., GANNON, D., GOBLE, C., LIVNY, M., MOREAU, L., AND MYERS, J. Examining the Challenges of Scientific Workflows. *IEEE Computer 40*, 12 (Dec. 2007), 26–34.

[9] LYLE, J., AND MARTIN, A. Trusted computing and provenance: Better together. In *TaPP '10: Proceedings of the 2nd Workshop on the Theory and Practice of Provenance* (2010), Usenix.

[10] MELL, P., AND GRANCE, T. The NIST Definition of Cloud Computing, 2009. http://csrc.nist.gov/groups/SNS/cloud-computing/clouddefv15.doc.

[11] MUNISWAMY-REDDY, K.-K., MACKO, P., AND SELTZER, M. Provenance for the cloud. In *FAST '10: Proceedings of the 8th USENIX conference on File and storage technologies* (2010), USENIX, pp. 15–14.

[12] MUNISWAMY-REDDY, K.-K., MACKO, P., AND SELTZER, M. I. Making a cloud provenance-aware. In Cheney [6].

[13] REILLY, C. F., AND NAUGHTON, J. F. Transparently Gathering Provenance with Provenance Aware Condor. In Cheney [6].

[14] SIMMHAN, Y. L., PLALE, B., AND GANNON, D. A survey of data provenance in e-science. *SIGMOD Rec. 34*, 3 (2005), 31–36.

[15] VÁZQUEZ-SALCEDA, J., ALVAREZ, S., KIFOR, T., VARGA, L. Z., MILES, S., MOREAU, L., AND WILLMOTT, S. *In R. Annicchiarico, U. Cortés, C. Urdiales (eds.) Agent Technology and E-Health*. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkhäuser Verlag AG, Switzerland, Dec. 2007, ch. EU PROVENANCE Project: An Open Provenance Architecture for Distributed Applications.

[16] XU, J. Provenance-Aware Fault Tolerance for Grid Computing. `http://spiderman-2.laas.fr/IFIPWG/Workshops&Meetings/48/RR/03-Xu.pdf`, 2005.

[1] http://www.tclouds-project.eu