



A common API for delivering services over multi-vendor cloud resources

Luís A. Bastião Silva*, Carlos Costa, José Luís Oliveira

University of Aveiro, DETI/IEETA, Portugal

ARTICLE INFO

Article history:

Received 1 March 2012

Received in revised form

24 November 2012

Accepted 16 April 2013

Available online 24 April 2013

Keywords:

Cloud services

Cloud standardization

Cloud storage

Cloud databases

ABSTRACT

The increasing pace of evolution in business computing services leads enterprises to outsource secondary operations that are not part of their core business. The cloud computing market has been growing over the past few years and, consequently, many cloud companies are now offering a rich set of features to their consumers. Unfortunately, those cloud players have created new services with different APIs, which imply that cloud-oriented applications might be instantiated in one single cloud provider. This scenario is not desirable to the IT industry because their applications will become provider-dependent. In this paper we present a platform that allows applications to interoperate with distinct cloud providers' services using a normalized interface. The proposed approach provides a common API that minimizes the present deficit of cloud API standardization and provides secure and redundant services allocation. Moreover, services from different cloud providers can be combined and decorated with additional functionalities like, for instance, redundancy and ciphering on-the-fly.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

The increasing pace of evolution in business computing services leads enterprises to renew their way of operating. Business requirements have also changed and outsourcing has been adopted by many industries, allowing the enterprise to focus more on its core business (Bieberstein, 2006). The Cloud computing market has grown significantly over the past few years and, following the natural progress of business models, there is a great interest in the IT industry in migrating services to this kind of infrastructures (Hajjat et al., 2010; Leavitt, 2009). In order to respond to this demand, many cloud companies increasingly offer new features to their consumers—using a rich set of services on the server side, and/or providing a platform where users can more easily deploy their applications. For instance, Amazon Web Services (Amazon, 2011) has released services such as Simple Storage Service (S3), Amazon SQS, SimpleDB and many others.

However, despite this evolution, most cloud companies have been creating distinct APIs for their services, which implies that the applications developed for the cloud can only be instantiated in one single provider, i.e. they are locked to each vendor. To achieve portability and interoperability, it is clear that stockholders, i.e. the user community and the marketplace, must adopt a common design (Lee, 2010). Some efforts have been made to create standards, in order to grant interoperability among the various cloud providers, e.g. SNIA (Association, 2011b) and CDMI (Association,

2011a). The desired scenario consists of creating specifications and interfaces that could be used by all cloud service providers. However, these standards are still drafts, with reduced practical impact. Furthermore, several efforts have been made to standardize Infrastructure-as-a-Service (IaaS), while the Platform-as-a-Service (PaaS) still does not have any consistent unified API.

To tackle this issue, a notable effort has been made by several communities, such as jclouds (jclouds, 2010), libcloud (libcloud, 2011) and simplecloud (simplecloud, 2011), to provide a unique programmatic API to deal with multiple cloud solutions. Unfortunately, they do support a limited number of providers and the extension to new ones is not simple. Most of them still focus on IaaS, and there is still a gap in the standardization of other services like storage, database and notification services. The transparent combination of multiple cloud resources will allow applications to communicate with any cloud, even with different providers at same time, leading to the Sky computing concept (Keahey et al., 2009). Although this paradigm was initially used just for IaaS, recently several efforts have been made to extend it to PaaS (Petcu et al., 2011; Di Martino et al., 2011). However, Sky computing for PaaS is still not fully developed and only possible architecture has been briefly discussed until now.

Standardization may solve many problems in Cloud computing, but there are also some challenges that must be considered during the process. Several usage scenarios imply dealing with critical information, and applications need to manage access to resources to avoid data tampering. Those privacy concerns are a real problem for some corporations as there is information that they intend to keep safeguarded (Kumbhare et al., 2012). Moreover, Amazon released a new service named AWS Storage Gateway, where the

* Corresponding author. Tel.: +351 916427877.

E-mail address: bastiao@ua.pt (L.A. Bastião Silva).

main idea is to cipher and decipher safeguarded data over the cloud, i.e. the data is secure encrypted/decrypted in house. However, it is an Amazon service and it does not work with other cloud providers.

This paper presents a platform that allows client applications to easily interoperate with distinct cloud providers, combining and decorating services like, for instance, dynamic data storage across multiple and incompatible infrastructures. Throughout the paper, we will describe a set of APIs related to cloud services, focusing on storage, columnar databases and publishing/subscription, i.e. the principal resources consumed by Internet applications. Finally, a case study will be presented and a discussion of the advantages and drawbacks of the solution will be provided. The rest of the manuscript is organized as follows. The next section will give a background of Cloud computing, giving special emphasis to storage, columnar data and publishing/subscribe services. Also, standardization and interoperability will be discussed. Section 3 presents the system architecture and discusses the proposed abstractions. Section 4 presents a case study, i.e. a clinical solution instantiated with this platform. Section 5 presents a discussion of advantages and drawbacks of the proposed solution, as well as a comparative analysis of other solutions. Finally, the main conclusions of the paper will be presented in a summary in Section 6.

2. Background and related work

2.1. Cloud computing services

IT solutions have been mostly supported by on-premise software, i.e. hosted on private enterprise datacenters. With the emergence of cloud computing, off-premise software has increased its applicability and nowadays applications are offered as services through cloud providers. Thus, cloud computing has been adapted to customers' requirements, creating dissimilar development models among cloud providers. Clearly, there are self-services offered by these companies, where resources are available at anytime and anywhere for customers. Moreover, the resources can be distributed over multiple locations, in order to improve reliability. In the following sections, we will focus on the description of three of these services: Storage-as-a-Service, Database-as-a-Service and Notification Service.

2.1.1. Storage-as-a-service

Storage-as-a-Service (SaaS) is the ability to offer remote storage in a local virtualized way, for any operating system and application. Nowadays, Cloud providers are offering storage using the Blobstore concept, which, per se, is not new. In the past, these concepts were used in Database Management Systems (DBMS) in the storage and movement of large data blocks. Blobstores are associative memories, i.e. key-value storage providers, where the blob is unstructured data (value) stored in a container and the lookup is performed through a text key. A container is a namespace or domain for the blobs. A blob is always uploaded to a container. The blobstores have a list of containers where the developer can create, remove or rename them. The container holds content, which can be blobs, folders or virtual path. Also, the blobstore in cloud services has an access control list to authorize people to access the data.

In practice, blobstore service allows customers to store data in a container under the Cloud. For instance, Amazon S3, Microsoft Azure and OpenStack have their own blobstore APIs. These services are considered PaaS because they allow developers to take advantage of remote storage service to support their application data in a transparent way. There are many examples of SaaS usage, for instance, the Dropbox application which stores customers' files

in Amazon S3 or commercial web portals that store great quantities of pictures in cloud blobstores.

2.1.2. Database-as-a-service

Database-as-a-Service (DaaS) is a new paradigm that outsources the burden effort to the cloud provider. Therefore, the database is hosted in a remote datacenter and can be shared between users, in a transparent way. For instance, Amazon AWS, Windows Azure and Rackspace (2011) offer a database as a service where customers pay for what they use. There is a new type of databases, called columnar data, which are organized in a key-value structure. These databases store the information by column, instead of the traditional ones that store the information by row. It has several advantages in computing large amounts of information and cloud providers are now offering these databases. All database operations are supported by these services, for instance, creating tables, loading and accessing data in the tables. Cloud players often supply an API to access the database, and execute operations through a web service API. Furthermore, database maintainers do not need to worry about the server's redundancy, upgrades, backup plan and recovery from disaster. Nonetheless, some enterprises are concerned about ensuring data privacy. In fact, this is one of the weaknesses of DaaS. Despite the provision of Server Level Agreements (SLA) by Cloud Providers, there are legal issues that need a very high level of privacy and confidentiality and these organizations do not have clear data in any case. Also, store procedures and triggers might not be supported in the overwhelming majority of cloud providers supplying DaaS. Finally, performance might deteriorate because applications will access the data in remote datacenters when located in the public cloud provider.

2.1.3. Notification service

Most information systems and computer applications rely heavily on updated and real time information. In addition, many different applications need to exchange information, i.e. the capability to publish a message from an application and immediately deliver it to other applications that subscribe the same channel. Nowadays, many applications provide this functionality supported on a polling strategy, i.e. checking periodically for new messages. They supply these features in a transparent way to the end-user, although there are extra computational resources requirements and in some cases without the expected efficiency. Cloud providers were influenced by this tendency and have also created services to communicate through a message-based model. Notification services refer to the ability to communicate with another remote entity through Cloud Services. There is some variety of communication services, but essentially they use the Publish/Subscribe model. Several companies are working on notification services, i.e. Amazon SQS (2011), PubNub (PubNub, 2011) and Azure Queue (Corporation). All of them have the same concept: a web service that sends notifications to hanging users through an event-driven workflow.

2.2. Interoperability and standardization

Nowadays there is great competition in the industry to provide better and more services over the Cloud. However, the services provided by different players are not typically compatible (Petcu et al., 2011; Shan et al., 2012; Dillon et al., 2010b). Undoubtedly, interoperability and portability is required to allow applications to be ported more easily between different cloud providers.

Recently, several groups have been formed to create standards and common interfaces that could allow interoperability between distinct cloud solutions (Parameswaran and Chaddha, 2009). For example, Storage Network Industry Association (SNIA) (Association, 2011b) has been working on a storage data standard

Download English Version:

<https://daneshyari.com/en/article/10343150>

Download Persian Version:

<https://daneshyari.com/article/10343150>

[Daneshyari.com](https://daneshyari.com)