The Journal of Systems and Software 86 (2013) 2294–2308

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A service-oriented framework for developing cross cloud migratable software

Joaquín Guillén[a,*], Javier Miranda[b], Juan Manuel Murillo[b], Carlos Canal[c]

[a] GloIn, Calle Azorín 2, Cáceres, Spain
[b] Department of Information Technology and Telematic Systems Engineering, University of Extremadura, Cáceres, Spain
[c] Department of Computer Science, University of Málaga, Spain

ARTICLE INFO

Article history:
Received 1 March 2012
Received in revised form
17 September 2012
Accepted 6 December 2012
Available online 3 January 2013

Keywords:
Cloud framework
Adaptation
Cross-cloud applications

ABSTRACT

Whilst cloud computing has burst into the current scene as a technology that allows companies to access high computing rates at limited costs, cloud vendors have rushed to provide tools that allow developers to build software for their cloud platforms. The software developed with these tools is often tightly coupled to their services and restrictions. Consequently vendor lock in becomes a common problem which multiple cloud users have to tackle in order to exploit the full potential of cloud computing. A scenario where component-based applications are developed for being deployed across several clouds, and each component can independently be deployed in one cloud or another, remains fictitious due to the complexity and the cost of their development. This paper presents a cloud development framework for developing cloud agnostic applications that may be deployed indifferently across multiple cloud platforms. Information about cloud deployment and cloud integration is separated from the source code and managed by the framework. Interoperability between interdependent components deployed in different clouds is achieved by automatically generating services and service clients. This allows software developers to segment their applications into different modules that can easily be deployed and redistributed across heterogeneous cloud platforms.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Cloud computing has experienced an admirable growth throughout the past years due to the high acceptance rate it has had among companies with technological backgrounds (Leavitt, 2009). The combination of a powerful technology with a competitive business model, which in this case results from combining cloud computing with utility computing, has been presented as extremely beneficial for companies with limited amounts of resources. Such benefits have been widely documented by many researchers (Armbrust et al., 2009; Zhang et al., 2010), emphasizing the lack of an initial investment, the lower operation costs and the apparently infinite computing resources offered by the technology. These benefits, in conjunction with the great amount of cloud providers that exist, each with their own strengths and weaknesses, allow companies to choose which cloud platform they want to develop their software for (Li et al., 2010).

However, besides the outstanding benefits provided by the technology, applications are often subject to numerous limitations that keep them from exploiting the potential of cloud. Most cloud vendors establish a series of restrictions for their cloud platforms, and certify that they are satisfied by providing tools that must be used throughout an application's development process. The software developed with these tools will be coupled to the libraries and services provided by each cloud provider, hence preventing these applications from being deployed in other clouds. This problem, which is commonly known as vendor lock-in (Chow et al., 2009), presents one of the main drawbacks of cloud computing: applications developed for specific cloud platforms cannot easily be migrated to other cloud platforms and users become vulnerable to any changes made by their providers (Armbrust et al., 2010). Additionally the lack of standardization among cloud platforms prevents users from creating software that can be deployed across multiple clouds (Petcu et al., 2011), where different components of a single system could reside in different platforms.

Thus, the current scenario does not allow cloud users to take full advantage of the utility computing model since it only considers choosing a cloud platform at early stages of projects, where development has not begun. Ideally, users should be able to migrate their applications towards different clouds if they are not satisfied with the services and costs of their providers. Moreover, considering that different modules or components of an application may be subject to different requirements or technological restraints, users may find it difficult to choose a cloud platform that complies with all the application's requirements and may thereby have to consider deploying their applications across multiple clouds (Tsai et al., 2010). Cross cloud development is an arduous task which

* Corresponding author. Tel.: +34 685556021.
E-mail addresses: jguillen@gloin.es (J. Guillén), jmircar@unex.es (J. Miranda), juanmamu@unex.es (J.M. Murillo), canal@lcc.uma.es (C. Canal).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.12.033

involves using the tools provided by each different cloud provider, as well as developing the services and clients for integrating the components deployed in separate clouds. Consequently, migrating a component from one cloud to another would require redeveloping its dependent components, either if they are hosted or not in other clouds.

Considering that standardization is currently not an option for most significant cloud computing providers (Dillon et al., 2010), the aforementioned limitations suggest that new tools are required for developing applications that can be deployed across multiple clouds as well as migrated from one cloud to another in order to take advantage of the business model associated to cloud computing. These tools should contribute to lower the barriers that users continue to encounter in their leap towards cloud computing, and thereby impulse cloud as the IT infrastructure of the future. Software should be developed in the same manner independently of which platform it will be hosted on, thereby not having to consider matters related to integration between the software and the cloud architecture as well as the integration between software hosted by different clouds.

This paper presents a framework for developing applications that are decoupled from the architecture, services, and libraries provided by cloud vendors, such that they can be deployed across one or several clouds. The framework allows developers to choose which components are deployed in each of the supported cloud platforms. Deployment metadata is separated from the source code and managed by the framework. The metadata represents a deployment plan that describes how dependent modules hosted in different clouds communicate with each other and the services provided by their cloud, this way relieving developers from implementing cloud integration modules and interoperability services. The application source code, in conjunction with the deployment plan, is interpreted by the framework, which generates cloud compliant software artefacts that are deployed in each cloud platform.

The development of cross-cloud migratable applications is currently a topic of great interest which is being tackled from many perspectives by different authors. The main contributions of the framework, which differentiate it from alternate solutions, are summarized in the following points:

- *Feature models*, commonly used in Product Line Engineering, are used for modelling the variability of the cloud platforms supported by the framework.
- *Software adaptation* techniques are applied; adapters are automatically generated to allow the software artefacts to be integrated with their corresponding clouds.
- Applications can be developed as if they were going to be deployed in an in-house environment, being completely agnostic to any cloud related information.

Such contributions have be integrated in the framework in order to improves the current state of the art in cloud application development. Their adoption allows applications to be designed without having to consider the intricacies of each cloud platform and without being coupled to any cloud or intermediate platform.

The framework described in this work has been conceived for developing software for cloud platforms that allow hosting and deploying applications. The prevailing categorization of cloud computing models considers the following variants (TechTarget, 2010):

- *Infrastructure as a Service* (IaaS): physical or virtual machines are provided as a service, upon which users can install an operating system and any required software.
- Platform as a Service (PaaS): in this model a computing platform which will usually include an operating system, a programming language an execution environment, a database and a web server, is provided as a service.
- Software as a Service (SaaS): in this model services an applications are provided as a service. The underlying infrastructure is not accessible to the users.

Hence, the framework is intended to be applied upon applications targeted towards IaaS and PaaS clouds, as only these two models can host externally developed software.

Both the framework and its associated tools have been developed in conjunction with Gloin S.L., a Spanish start-up company that works on providing technology for enhancing the software development processes of its customers. The framework has been applied to a real industrial case in one of Gloin's projects; the results of this experience are reviewed in the final sections of this paper.

The remainder of this paper is structured as follows. Section 2 contains a description of the background and the motivations that have led us to our work. Section 3 describes our proposal; we discuss the principles on which our work is based and the architecture of the framework. Section 4 contains the technical design of the framework; in this section technical details are provided about the technology in which the framework has been constructed. Section 5 explains the development processes that must be followed by developers in order to use the framework. Section 6 contains a case study where the framework is used to implement and deploy an application across more than one cloud. Section 7 analyses the results obtained from the case study; it presents its strengths and weaknesses in order to plan future milestones. Section 8 summarizes the work carried out by other authors that is related to ours. Finally, Section 9 contains the conclusions extracted from our work.

2. Background and motivation

Cloud computing is a technology that provides numerous benefits to both its providers and its users, however the interests of each of these parties are of a very different nature. Whilst providers have seen in this technology a means of expanding their business models whilst recouping the cost of their underutilized server farms, users have seen the opportunity to access powerful computing resources flexibly and at an accessible price.

Becoming a cloud provider requires companies to have extremely large-scale data centres and highly qualified staff for their maintenance. The initial investment required for setting up such infrastructure is inaccessible for most companies; however companies such as Microsoft, Google and Amazon, among many others, have already invested in large server farms over the years. For these companies cloud computing has emerged as a great opportunity for leveraging the investment that they have made (Armbrust et al., 2009). Furthermore, cloud computing has been commercialized with a competitive business model which allows providers to target both small and large companies, thereby making the technology accessible to a wider range of users. The underlying interest of companies to return the investment they have made by gaining as many customers as possible and guaranteeing that these customers consume as many of their services as possible, has frustrated most efforts for standardization and cloud platforms integration (Dillon et al., 2010).

On the other hand, cloud users in the form of companies with technological interests, have seen that cloud can provide them with access to an unlimited number of computing resources that are accessible from numerous devices, such as smartphones, tablets and PCs (Anderson and Rainie, 2010). This allows cloud users to manage their services and applications in a much more flexible manner, being able to scale up and down the number of computing resources that they manage, as well as the cost of such resources.

Download English Version:

https://daneshyari.com/en/article/10343148

Download Persian Version:

https://daneshyari.com/article/10343148

Daneshyari.com