Existing approaches for cross-platform development and deployment of cloud applications

Fotis Gonidis [1], Iraklis Paraskakis [1] and Anthony J. H Simons [2]

[1] South-East European Research Centre (SEERC),
City College – International Faculty of the University of Sheffield,
24 Proxenou Koromila Street, 54622 Thessaloniki, Greece
{fgonidis, iparaskakis}@seerc.org

[2] Department of Computer Science, University of Sheffield,
Regent Court, 211 Portobello Street,
Sheffield S1 4DP, United Kingdom
A.Simons@dcs.shef.ac.uk

Abstract. Cloud computing is a relatively new paradigm that promises to revolutionize the way IT services are provided. However, the issue of cloud application portability could be an impediment for the wide adoption of cloud computing. In this paper, certain existing approaches addressing application portability are presented. Each solution is developed in a different direction. Next we briefly state the scope of our research focus and the future research directions on the field of cross platform development and deployment of cloud applications

Keywords: Cloud Computing, Portability

1 Introduction

The growing interest in cloud application platforms has resulted in a large number of platform offerings being already available on the market. However, different cloud application platform offerings are characterized by considerable heterogeneity. Because of incompatibilities, users that develop applications on a specific platform may encounter significant problems when trying to deploy them in a different environment. This gives rise to the familiar problem of vendor lock-in [1], which has been a challenge long before the advent of cloud computing.

Incompatibilities which may arise while deploying an application across various platforms include the differences in programming languages and databases, specific configuration files and differences in provided services such as the message queue service, file storage service, billing service etc [2]. However, for developers to be able to exploit the full advantages of PaaS [3], they should be able to overcome the conflict points and deploy their cloud applications across multiple platforms, without lock-in to a particular vendor. To achieve this, a new approach to cloud application

development must be adopted. The key concept is for users not to develop applications directly against proprietary platform environments. Rather, developers should use either standard and widely adopted technologies, or abstraction layers which decouple application development from specific target platforms. Ensuring portability across cloud providers would eliminate the vendor lock-in problem and in turn, this would increase consumers' trust towards cloud computing and public cloud services.

This paper examines existing approaches which address the issue of application portability. Rather than providing an exhaustive list of existing approaches we focus on certain approaches which follow a different solution paradigm. Next we briefly state our own research scope and future directions towards addressing the issue of application portability.

2. Existing approaches

Significant work has been done in order to minimize the vendor lock-in effect and allow the development of applications which can be deployed across multiple platforms. In this section we briefly state some prominent approaches that adopt different solution architecture.

1) **Wrapper approach.** The wrapper approach consists of an API library which abstracts the access to the native APIs offered by the specific cloud service providers. jClouds [4] is a major example. It provides abstraction for major cloud storage services such as Amazon S3 and Microsoft Azureblob. It allows the application to perform actions on files located in remote stores which are offered by a cloud provider.

   From a technical point of view, jClouds is a client implementation of the REST API [5] which cloud providers expose to allow access to their storage service. Therefore developers can use a single method offered by the library to access any of the supported storage services.

   Alternatively, developers should implement their own HTTP requests for each storage service or use the client implementations offered by the cloud providers. In both cases a significant development effort is needed when deploying the application across different cloud platforms. jClouds reduces the effort by introducing an abstraction layer between the application and the storage service. Similar solution which offers a wrapper API library, is the Simple Cloud API [6].
2) **Adoption of common standards.** Instead of using a wrapper library to abstract the access to the native cloud service APIs, users can choose cloud platforms that adopt common standards and thus avoid any lock-in. Openshift is such a cloud platform which does not use any proprietary API [7]. Another example of an open source platform contributing to application portability is the European project mOSAIC [8]. mOSAIC adopts standard and widely used technologies. On top of that the platform offers the so called interoperability API [9]. Developers can exploit the interoperability API in order to access resources such as databases, file storage and message queuing service from various cloud providers. However, the range of the supported resources is significantly limited.

While the cloud platforms which adopt common standards minimise the lock-in effect, they usually do not offer a range of services that users could exploit in order to enrich their applications. mOSAIC allows a certain access to cloud resources such as file storage, databases, messaging system , but the developers are required to use a certain programming concept which increases the learning curve of the platform.

3) **Model-driven engineering (MDE) approach.** Another approach towards achieving cloud application portability is to exploit MDE techniques. Using MDE, the application creation process begins with building a platform independent model (PIM). Then, using automated model transformations, the PIM is translated into a platform specific model (PSM) targeting particular cloud platforms.
MobiCloud [10] is an example which leverages the benefits of model-driven engineering (MDE) field. MobiCloud is a domain specific language (DSL) that allows developers to create simple Create, Read, Update, Delete (CRUD) applications using a graphical editor and a scripting language. The platform automatically generates the source code and the required configuration files for uploading the application on Google App Engine and Amazon EC2.
Although MobiCloud enables the development and deployment of cross platform applications, it has a limited application scope. Developers can only create simple CRUD applications and target a limited set of cloud platforms.
Additional examples which leverage MDE techniques are [11] [12], which both propose the definition of a modeling language for describing cloud applications.

3. Future research directions

In the previous section we mentioned certain existing approaches, each one of them addressing the issue of application portability from a different perspective. The first one offers a third party library which abstracts certain storage service providers. The second one constitutes an open source cloud platform where applications can be deployed and executed. The third one leverages model driven engineering techniques and offers an online development environment for creating simple CRUD applications. Each one of the approaches involves certain advantages and disadvantages as mentioned in section 2. In this section we briefly state our own future research directions in the field of cross-platform development and deployment of cloud applications.

Our intention is to create an abstraction framework which can be used by developers to allow access to specific cloud services independent of the cloud provider. The target cloud services can be a message queue service or a platform specific service such as the billing service. Our motivation for choosing this scope is that, to the best of our knowledge, this is a relatively unexplored area and there is no or little related work on the field. In addition to that, the functionality of the framework can be enriched by leveraging existing solutions, such as the jClouds storage service.

An initial high-level overview of the envisioned solution is shown in Figure 1. The abstraction framework will consist of two parts. The first one will contain platform

independent models which will represent the cloud services that the framework will support. For each supported service a set of models will be available. A domain specific language can be designed to leverage the use of the models. The second part will consist of concrete adapters which will be responsible for transforming the abstract models into concrete source code for the target cloud platform such as Google App Engine, Amazon Elastic Beanstalk etc.

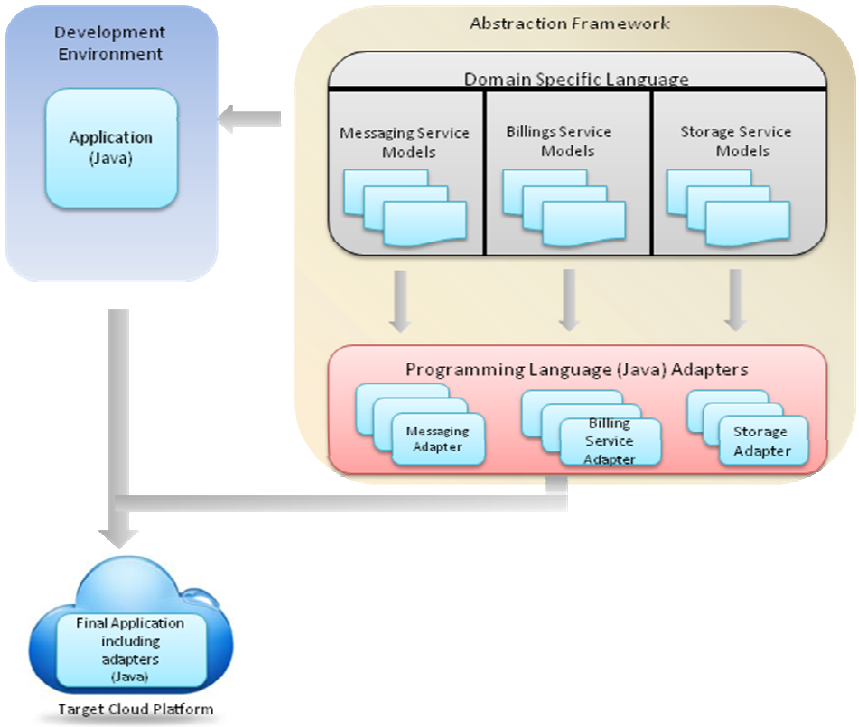

Fig. 1. High level architecture of the abstraction framework

The developer will initiate the development of an application using a popular development environment such as Eclipse and a programming language such as Java. When the application needs to use a cloud service which is supported by the framework, the developer will use the domain specific language in order to configure the service and model the desired functionality. Then, the source code for the specific cloud platform will be generated, using the appropriate adapter. The generated source code will be merged with the rest of the application and deployed on the target platform.

The contribution of the proposed research directions will be threefold. First, it will systematically explore the challenges involved when trying to abstract cloud service providers. Second, it will attempt to propose a formal methodology and best practices for abstracting cloud services. Third, the envisioned abstraction framework will attempt to expand its functionality by leveraging existing solutions. For example, a storage service can be offered by integrating the jClouds framework.

4. Conclusions

In this paper the issue of cross platform development and deployment of cloud applications was introduced. Certain existing approaches, addressing the issue of application portability, were presented. Each approach follows a different solution direction ranging from a wrapper library, to adopting common standards and MDE techniques. Next, our research directions towards addressing application portability were briefly stated. Our intention is to focus on areas that have not been yet extensively explored, such as cloud messaging services or billing services. Additionally, we intend to explore the benefits of introducing a domain specific language for allowing easy access and configuration of the offered services.

References

1. Neal Leavitt, "Is Cloud Computing Really Ready for Prime Time?", *Computer*, vol. 42, no. 1, pp. 15–20, Jan. 2009.
2. Fotis Gonidis, Iraklis Paraskakis, and Dimitrios Kourtesis, "Cloud Application Portability. An Initial View", in *6th Balkan Conference in Informatics*, Thessaloniki, Greece, 2013.
3. F. Gonidis, I. Paraskakis, and D. Kourtesis, "Addressing the Challenge of Application Portability in Cloud Platforms", in *7th South-East European Doctoral Student Conference*, Thessaloniki, 2012, pp. 565–576.
4. "jclouds(May 2013)", [Online], Available: http://jclouds.incubator.apache.org/, [Accessed: 10-May-2013].
5. R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture", in *Proceedings of the 22nd international conference on Software engineering*, New York, NY, USA, 2000, pp. 407–416.
6. "SimpleCloud(May 2013", [Online], Available: http://www.simplecloud.org/, [Accessed: 10-May-2013].
7. "Openshift(Jun 2013)", [Online], Available: https://www.openshift.com/, [accessed: 30-Jun-2013].
8. D. Petcu, C. Craciun, M. Neagul, I. Lazcanotegui, and M. Rak, "Building an interoperability API for Sky computing", in *2011 International Conference on High Performance Computing and Simulation (HPCS)*, 2011, pp. 405–411.
9. Gorka Echevarria, "Description of mOSAIC`s API", D1.3, Aug. 2011.
10. A. Ranabahu and A. Sheth, "Semantics Centric Solutions for Application and Data Portability in Cloud Computing", in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 2010, pp. 234 –241.
11. J. Esparza-Peidro and F.D. Munoz-Esco□, "Towards the Next Generation of Model-Driven Cloud Platforms", TR-ITI-SIDI-2011/001, Valencia, 2011
12. M. Hamdaqa, T. Livogiannis, and L. Tahvildari, "A reference model for developing cloud applications," in Proceedings of the 1st International Conference on Cloud Computing and Services Science, 2011, pp 98-103