Composition of web applications in clouds environment

Zuowen Wang*

School of Computer, Wuhan Polytechnic, Wuhan, 430074, China

Received 1 March 2014, www.cmnt.lv

Abstract

In recent years, for the advances of Cloud Computing technologies, cloud applications have been popularity for their rich set of features. The advantages of cloud applications include that users can utilize them in a low cost, threshold, and risk way; these applications can be quickly deployed on the clouds without duplication of work such that developers can focus on enhancing their QoS to improve core competitiveness. Therefore, their practical use on business with promising values can be expected. As such, cloud applications are recognized as a trend for the next generation of business applications, and hence how to migrate these on-premise applications to the clouds becomes a desired field in the literature. For this need, we present an ontology-based method for the composition process that specifically addresses the cloud features and the composition of on-premise applications into the clouds. In particular, for enabling the selection of desired clouds, the method imposes semantic ontologies on the specifications of the candidate clouds from which the desired ones can be effectively selected. For illustration, the method is applied to the composition of a CSS application to its cloud version.

Keywords: cloud computing, cloud application, semantic ontology, service composition

1 Introduction

Web service has already become an important paradigm for developing web applications. Growing number of web services raise the issue of efficiently locating the desired web services. Many approaches have been proposed with respect to the way in which services are described. Semantic communities provide ontology languages for web services such as OWL-s and WSMO. Services are organized in ontology. To locate desired services depends on semantic match of services. For the advances of Cloud Computing technologies in recent years, cloud applications have been popularity for their rich set of features. The advantages of cloud applications include that users can utilize them in a low cost-, threshold-, and risk-way; these applications can be quickly deployed on the clouds without duplication of work such that developers can focus on enhancing their QoS to improve core competitiveness. Therefore, their practical use on business with promising values can be expected. As such, cloud applications are recognized as a trend for the next generation of business applications.

In terms of the architecture for on-premise applications (e.g., web information systems), client-server patterns were most commonly used in the past decades; almost all kinds of existing on-premise applications were constructed using this style of architectures. However, as stated above, cloud applications have been recognized in recent years as applications to the clouds for taking advantage of cloud applications becomes a desired field in the literature. For this need, some discussions about the composition of on-premise applications into the clouds have been presented in [1-7]. In general, these statements have clarified some important issues about the composition and then proposed respective tips for addressing such issues. Thus, there are already many ideas about the cloud characteristics for composition; for instance, how on-premise applications are smoothly migrated to either of the three service models – SaaS, PaaS, and IaaS – supported in various cloud environments.

For the composition process, nonetheless, any methods that take into considerations of the architecture and characteristics of both on-premise applications and clouds to provide guidance on their composition are still missing. In addition, for the most critical issues in the composition process: the identification of candidate clouds and then the selection of desired ones from these candidates, few discussions can be found for addressing them by explicit formal approaches. Such methods and any formal approaches for addressing these critical issues, in our opinion, should not be negligible since a well-guided process and suitable formal issue approaches are most important for directing the composition of the many on-premise applications in a systematic and managed manner.

In this paper, we therefore present a method for directing the composition process. The method starts at the identification of the architecture and profile of the on-premise application to be migrated, through the discussion of the requirements for clouds and also the identification of the candidate clouds from which desired ones are selected with service models – SaaS or PaaS or IaaS – satisfying the cloud requirements, and finally ends

* ***Corresponding author*** e-mail: zuowenwangwh@163.com

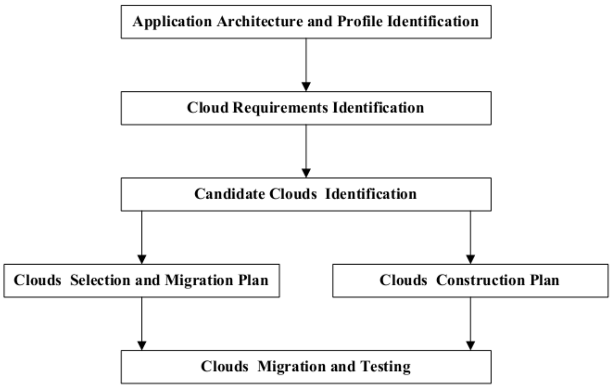

FIGURE 1 The Composition Method

at the deployment of the application on these selected clouds where a deployment and test plan is specified for conducting the deployment and tests. In particular, for enabling the selection of desired clouds, the method imposes semantic ontologies on the specifications of the candidate clouds such that desired clouds can be effectively selected by matching the desired cloud requirements with the provided services of these candidate clouds. Further, for addressing the exceptional situation that no candidate clouds can be found for smooth compositions, some discussions are also presented for conducting how on-premise applications may achieve their cloud-based versions via proper virtual mechanisms where either of the three service models – SaaS, PaaS, and IaaS – can be virtually supported in these mechanisms. For illustration, the method is applied to the composition of a Customer Support System (CSS) [8, 9] application to its cloud version that particularly emphasizes on both of collecting customer information (i.e., knowledge about/from customers) for enterprises and reversely delivering services information from enterprises to benefit customers.

This paper is organized as follows. Section 2 presents the composition method that encompasses three processes: an application-description process, a cloud-identification process, and an application- deployment process. The construction of semantic ontologies for enabling the selection of desired clouds is then presented in Section 3. For illustration, the method is applied in Section 4 to the composition of a CSS application to its cloud version. Finally, Section 5 has the conclusions and future work.

2 The composition method

As shown in Figure 1, the composition method has the following six steps:

1. Application Architecture and Profile Identification that determines (1) the architecture of the on-premise application where imposed components and their dependencies to support functional/non-functional purposes are specifically addressed; and (2) the profile of the on-premise application that includes the usage data about its executions (e.g., CPU, memory, storage, I/O, and network usage data) as well as the action data about its users (e.g., the number of active users, request rates, transaction rates, and request/transaction latencies).

2. Cloud Requirements Identification that clarifies the cloud requirements for the on-premise application based on its architecture and profile, including (1) for its components, the requirements for their deployment on the prospective services in clouds such as virtual machines, data storages, and a/synchronous message channels; and (2) for its usage and user actions, the requirements for their QoS in clouds such as customized user interfaces and access modes, performance, reliability, security, and scalability.

3. Candidate Clouds Identification that identifies the candidate clouds whose services (i.e., service models – SaaS or PaaS or IaaS – provided in clouds) satisfy the cloud requirements identified above. It is noticed that for the components of the on-premise application that have deployment requirements on clouds, there may necessarily have various clouds that collaboratively provide services to satisfy these requirements. Also, in particular, semantic ontologies are imposed on the specifications of these candidate clouds such that the services of these clouds are specified in an integrative and consistent manner.

4. Clouds Selection and Composition Plan that determines from the identified candidates which clouds will be selected for the composition of the on-premise application. As in above, since these candidate clouds are specified in integrative ontologies, the selection of desired ones from them can be effectively achieved by matching the desired cloud requirements with their provided services. After then, the plan about the activities and relevant artifacts involved in the composition into selected clouds will be specified. In general, the activities include (1) deploying the application components on the services in clouds; (2) deploying the interaction mechanisms among application components on the inter/intra-cloud interaction solutions over/in clouds; and (3) refactoring/restructuring deployed components for satisfying the usage and user actions requirements such as customized user interfaces and access modes, performance, reliability, security, and scalability.

5. Clouds Construction Plan that identifies and schedules the alternatives for the situation that no candidate clouds can be found for smooth compositions as stated above. In such a situation, the on-premise application may be considered to achieve its cloud-based version via virtual mechanisms where either of the three service models – SaaS, PaaS, and IaaS – can be virtually supported in these mechanisms.

6. Clouds Composition and Testing that realizes the composition of the on-premise application into selected or constructed clouds in accordance with the composition or construction plan identified above. As in usual, testing of the composition proceeds in accordance with the activities involved in the composition process. where the first step addresses an application-description process, the

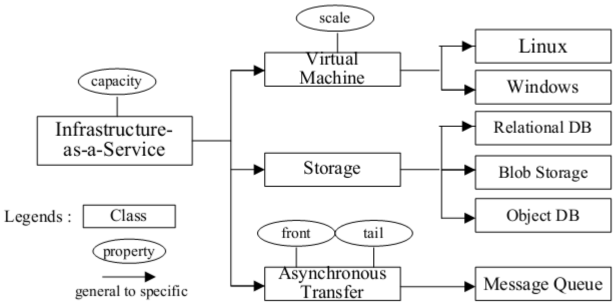

FIGURE 2 The hierarchy of classes for Infrastructure- as -a -Service

middle two encompass a cloud- identification process, and the last three cover an application-deployment process.

2.1 THE COMPOSITION METHOD

In this step, two descriptions about the on-premise application are addressed: (1) the architecture of the on-premise application; and (2) the profile of the on-premise application that includes the usage data about its executions as well as the action data about its users. For the architecture of the on-premise application, it specifically addresses desired architectural components with imposed behaviours and collaborations to support functional/non-functional purposes. As an example, the architecture of a Customer Support System (CSS) [8, 9] that emphasizes via the participant community [10-12] on both of collecting customer knowledge [13-16] for enterprises as well as reversely delivering services information to benefit customers: It is a 4-layer architecture of collaborative components where Customers interact with Enterprises via three intermediaries: Community, Customer Knowledge Agent, and Task Service Provider. It emphasizes on Community to help Customers share information about their desired tasks (e.g., buy or rent services from Enterprises). It emphasizes on collecting customer knowledge by Customer Knowledge Agent to help Enterprises catch their needs (e.g., provide services satisfying their desired tasks). It focuses on delivering services information by Task Service Provider from Enterprises to help Customers make recognition and comparisons.

With the architecture of the on-premise application, it is then good time to capture the profile of the on-premise application that can help size the application before it is migrated to the clouds. In general, the application data should be collected for at least 10 to 14 days to allow figuring out any variances in daily or weekly usage patterns. There are two kinds of data about the application: (1) the usage data about its executions (e.g., CPU, memory, storage, I/O, and network usage data); and (2) the action data about its users (e.g., the number of active users, request rates, transaction rates, and request/transaction latencies). With such application data, it is feasible to have an initial estimate of the cloud resources for the application to be migrated.

2.2 CLOUD REQUIREMENTS IDENTIFICATION

The second step is to identify cloud requirements for the on-premise application based on its architecture and profile, including (1) for its components, the requirements for their deployments on the prospective services in clouds such as virtual machines, data storages, and a/synchronous message channels; and (2) for its usage and user actions, the requirements for their QoS in clouds such as customized user interfaces and access modes, performance, reliability, security, and scalability. For the Customer Support System (CSS) as an example, its five components may require respective deployments on various cloud environments to support the functional/non-functional purposes via deployed clouds. Further, for its purposes of collecting customer knowledge for enterprises and delivering services information to benefit customers, it may require such QoS from these deployed clouds as customized user interfaces and access modes, performance, reliability, security, and scalability.

2.3 CANDIDATE CLOUDS IDENTIFICATION

The third step is to identify the candidate clouds whose services (i.e., service models – SaaS or PaaS or IaaS – provided in clouds) satisfy the cloud requirements identified above. For this, therefore, it is good to consider all of the cloud environments available on-line whose service models may satisfy the cloud requirements identified above. The following describes the possible service models in clouds:

1. Software-as-a-Service (Saas): In this model, the cloud provides application services, which may replace those provided by the on-premise application. With such SaaS services, many QoS features need to be evaluated for determining their replacement with the on-premise application as below.

(1) Their Service-Level-Agreements (SLAs) for availability, scalability, security, and performance; note that the evaluation for specific SLAs such as availability, scalability, and performance can be achieved by assessing the profile of the on-premise application.

(2) The compatibility of the application services with those offered by the SaaS.

(3) The portability of the application data into the SaaS for being accessed by the SaaS services.

(4) The portability of the access control by the application users into the SaaS for the access control by the SaaS users.

(5) The portability of the application interoperability with other services into the SaaS for the interoperable operations by SaaS services.

2. Platform-as-a-Service (Paas): In this model, the cloud provides platform services on which the on-premise application based on certain platforms such as JEE and MS.NET may be deployed under some QoS features as below.

(1) Their SLAs for availability, scalability, security, performance, and configuration (e.g., platform versions, APIs); note that as in SaaS, the evaluation for such SLAs as availability, scalability, and performance can be achieved by assessing the usage data about the application executions and the action data about the application users.

(2) The deployment of the application components and their interaction mechanisms on the PaaS for supporting the functional/non-functional purposes.

(3) The portability of the application services into the PaaS for the access by the deployed application users.

(4) The portability of the application data into the PaaS for being accessed by the deployed application.

(5) The portability of the access control on platforms (e.g., virtual servers) by the application users into the PaaS for the access control on clouds (e.g., virtual machines) by the deployed application users.

(6) The portability of the application interoperability with other services into the PaaS for the interoperable operations by the deployed application.

(7) The portability of the application management into the PaaS for monitoring and managing the deployed application.

3. Infrastructure-as-a-Service (Iaas): In this model, the cloud provides infrastructure services such servers, storages, and networks that the on-premise application and its deployed platforms may use under some QoS features as below.

(1) Their SLAs for availability, scalability, security, performance, and configuration; note that as in PaaS, the evaluation for such SLAs as availability, scalability, and performance can be achieved by assessing the profile of the on-premise application.

(2) The portability of the application services into the IaaS for the access by the deployed application users.

(3) The portability of the application data into the IaaS for being stored in the IaaS storages.

(4) The portability of the access control on infrastructure services (e.g., physical servers) by the application users into the IaaS for the access control on clouds (e.g., physical machines) by the deployed application users.

(5) The portability of the application interoperability with other services into the IaaS for the interoperable operations by the deployed application.

As a result, after considering all possible cloud environments that provide either of the above three service models, it is expected to identify some of them whose service models may satisfy the cloud requirements and then become the candidates to be selected for the realization of the composition. Since, as mentioned above, for enabling the selection of desired clouds from these candidates, semantic ontologies are imposed herein for the specifications of these candidate clouds such that the services of these clouds can be specified in an integrative and consistent manner. For the CSS as an example, Figure 2 shows the possible candidate clouds with service models that may satisfy the cloud requirements for the CSS.

2.4 CLOUDS SELECTION

The fourth step is to determine from the candidates identified above which clouds are selected for the composition of the on-premise application. As mentioned above, since these candidate clouds are specified in integrative ontologies, the selection of desired ones from them can be effectively determined by matching the desired cloud requirements with their provided services: (1) cloud requirements can be easily specified in terms of the structures and vocabularies of these ontologies, and (2) the matching of these requirements with the provided services of these candidate clouds can be easily undertaken. After determining the selection of targeted clouds, the plan about the activities and relevant artifacts involved in the composition into these selected clouds will then be specified. In general, the activities include (1) deploying the application components on the services in respective clouds; (2) deploying the interaction mechanisms among application components on the inter/intra-cloud interaction solutions over/in respective clouds; and (3) refactoring/restructuring any deployed components for satisfying the usage and user actions requirements such as customized user interfaces and access modes, performance, reliability, security, and scalability.

2.5 CLOUDS CONSTRUCTION

The fifth step is to identify and schedule the alternatives for the situation that no candidate clouds can be found at step 3 for smooth compositions. In such a situation, some alternatives may be considered; for instance, the on-premise application may be tailored to achieve a cloud-based version via specific virtual mechanisms [17-27] where either of the three service models – SaaS, PaaS, and IaaS – can be virtually provided through the support of these mechanisms.

The last step is to realize the composition of the on-premise application into selected or constructed clouds in accordance with the composition or construction plan identified above. As in usual, testing of the composition proceeds in accordance with the activities involved in the composition process.

3 Conclusion

In this paper, we present an ontology-based method for directing the composition of on-premise applications into selected clouds. The method takes into considerations of the architecture and characteristics of both on-premise applications and clouds to provide guidance on their composition. It therefore starts at the identification of the architecture and profile of the on-premise application to be migrated, through the discussion of the requirements for clouds, the identification of the provided services of the available clouds, and the selection of the candidate clouds whose service models – SaaS or PaaS or IaaS – satisfy the cloud requirements, and finally ends at the deployment of the application into selected clouds where a deployment and test plan is specified for conducting the deployment and tests. In particular, for enabling the selection of desired clouds, the method imposes semantic ontologies on the specifications of the candidate clouds from which the desired ones can be effectively selected. For illustration, the method is applied to the composition of a CSS application to its cloud version that particularly emphasizes on both of collecting customer knowledge for enterprises and reversely delivering services information from enterprises to benefit customers. Since cloud applications have been recognized in recent years as a trend for the next generation of business applications, how to migrate the many existing on-premise applications to the clouds for taking advantage of cloud applications has thus become a desired field in the literature. However, current discussions about this need mainly focus on some important issues about the composition and then present respective tips for addressing such issues. For the composition process, any methods that take into considerations of the architecture and characteristics of both on-premise applications and clouds to provide guidance on their composition are still missing. Further, for the most critical issues in the composition process: the identification of candidate clouds and then the selection of desired ones from these candidates, few discussions can be found for addressing them by explicit formal approaches. Such methods and any formal approaches for addressing these critical issues, in our opinion, should not be negligible since a well-guided process and suitable formal issue approaches are most important for directing the composition of the many on-premise applications in a systematic and managed manner. The method and corresponding semantic ontologies presented herein provides an effort on this need.

As our future work, we will continue to explore the real composition of existing on-premise applications to the clouds where specific PaaS and IaaS offerings such as Google GAE and Amazon EC2 will be selected as the deployed or used platforms. As one may conceive, while migrating these applications, experiences about the composition can be collected correspondingly for validating the usefulness and effectiveness of the method. In fact, with our systematic and managed steps for gradually identifying and specifying application/cloud features and then conducting the deployment of the applications on the clouds, quality of these migrated applications on clouds can be expected.

References

[1] Banerjee U 2010 *Five Examples of Composition to Cloud* May 2010 http://setandbma.wordpress.com/ 2010/05/28/5-examples-of-composition-to-cloud/

[2] Blaisdell R 2011 *How to Plan Your Composition to the Cloud* Nov. 2011 http://www.rickscloud.com/how-to-plan-your-composition-to-the-cloud/

[3] Carr N, 2011 *Best Practices for Achieving Composition to a Cloud Model* March 2011 http://blogs.dlt.com/practices-achieving-composition-cloud-model/

[4] Cisco Systems 2010 *Planning the Composition of Enterprise Applications to the Cloud* http://www.cisco.com/en/US/services/ps2961/ps10364/ps10370/ps11104/Composition_of_Enterprise_Apps_to_Cloud_White_Paper.pdf

[5] Huey G, Wegner W 2010 *Tips for Migrating Your Applications to the Cloud* August 2010 http://msdn.microsoft.com/ en-us/magazine/ff872379.aspx

[6] Mallya S 2010 *Migrate Your Application to Cloud: Practical Top 10 Checklist* April 2010, http://www.prudentcloud.com/cloud-computing-technology/composition-to-cloud-top-10-checklist-24042010/

[7] Momentum S I 2013 *Cloud Composition* http://cloudcomposition.com/

[8] Lin J 2009 An Object-Oriented Development Method for Consumer Support Systems *International Journal of Software Engineering and Knowledge Engineering* **19**(7) 933-60

[9] Orman L Consumer Support Systems *Communications of the ACM* **50**(4) 49-54

[10] Armstrong A, Hagel J 1996 *The Real Value of Online Communities* Harvard Business Review 134-41

[11] Hoadley C, Kilner P Using technology to transform communities of practice into knowledge-building communities *ACM SIGGROUP Bulletin* **25**(1) 31-40

[12] Krieger B, Müller P Making internet communities work: reflections on an unusual business model *ACM SIGMIS Database* **34**(2) 50-9

[13] Bueren A, et al. 2004 Customer Knowledge Management – Improving Performance of Customer Relationship Management with Knowledge Management *Proc. of 37th International Conference on System Sciences, IEEE Computer Society* 1-10

[14] Davenport T, Klahr P 1998 Managing Customer Knowledge *California Management Review* **40**(3) 195-208

[15] Garcia-Murillo M, Annabi H 2002 Customer Knowledge Management *Journal of the Operational Research Society* **53** 875-84

[16] Mobasher B, Cooley R, Srivastava J 2002 Five Styles of Customer Knowledge Management, and How Smart Companies Use Them To Create Value *European Management Journal* **20**(5) 459-69

[17] Sunrun SunrunVas http://ventem.com.tw/DM11.aspx

[18] Bruijn J, Fensel D, Keller U, Lara R 2005 Using The Web Service Modeling Ontology to Enable Semantic E-Business *CACM* **48**(12) 43-7

[19] Zhang Y, Huang H, Yang D, Zhang H, Chao H, Huang Y 2009 Bring QoS to P2P-based Sematic Service Discovery for the Universal Network *Journal of Ubiquitous Computing* **13**(7) 471-7

[20] Noy N, McGuinness D 2001 *Ontology Development 101: A Guide to Creating Your First Ontology* Stanford Knowledge System Lab Technical Report KSL-01-05, March 2001

[21] Martin D, et al. 2004 *OWL Web Ontology Language for Services (OWL-S)* W3C Member Submission, Nov. 2004: www.w3.org/Submission/OWL-S/

[22] METEOR-S: Semantic Web Services and Processes: http://lsdis.cs.uga.edu/projects/meteor-s/.

[23] Gong H, et al. 2008 Research on the Building and Reasoning of Travel Ontology *Proc. of International Symposium on Intelligent Information Technology Application Workshops, IEEE Computer Society* 94-7.

[24] *Google App Engine (GAE)* https://appengine.google.com/

[25] *Windows Azure* http://www.windowsazure.com/zh-tw/

[26] *Google Compute Engine (GCE)* https://cloud.google.com/products/compute-engine

[27] *Amazon EC2* http://aws.amazon.com/ec2/

Authors

Zuowen Wang, born in October, 1966, Hubei Province, China

Current position, grades: Associate Professor
University studies: Computer science and technology
Scientific interest: Computer software, Software engineering
Publications: 20 papers published in the international or national journals
Experience: He is an associate professor in School of Computer, Wuhan Polytechnic, China. His research interests include computer software and software engineering.