Migrating Web Applications to Clouds with Microservice Architectures

Jyhjong Lin[1], Lendy Chaoyu Lin[2], Shiche Huang[3]

Department of Information Management, Ming Chuan University, Taoyuan, Taiwan.

[1]jlin@mail.mcu.edu.tw, [2]lendy.lin@gmail.com, [3]iambeckhuang@gmail.com

Abst+ract

For the recent advances of Cloud Computing technologies, cloud applications have been popularity for their rich set of features. Therefore, their practical use for promoting business values can be expected. As such, cloud applications are recognized as a trend for the next generation of existing Web applications, and hence how to migrate these Web applications to the clouds becomes a desired field in the literature. For this need, we present a migration method that employs the well-known microservice architectures to support an effective migration of Web applications into the clouds.

Key words: Cloud computing, migration method, Web application, microservice architecture

Introduction

For the recent advances of Cloud Computing technologies, their use on applications has been most widely addressed due to the rich set of features in such cloud applications. These applications can be deployed on the clouds that make users use them in a low cost-, threshold-, and risk-way. Therefore, their practical use on business can be expected as a trend for the next generation of business applications.

In terms of the architecture for existing Web applications, client-server patterns were most used in the past decades; almost all existing Web applications were constructed using these styles. However, as stated above, cloud applications have been recognized as a trend for the next generation of business applications; how to migrate these Web applications to the clouds becomes a desired field. For this need, some discussions about the migration work have been presented in [1-5]. In general, these statements have addressed some important issues about the migration process. However, some shorts can still be found: (1) few considerations take into accounts of the architecture of both Web applications and clouds; and (2) few sentences are stated about the cloud requirements for the distributed styles of the migrated cloud applications.

Therefore, to address these limits, we present herein a method for directing the migration process. The method (1) starts from the identification of the architecture and profile of the Web application, and then the discussion of the requirements for clouds based on the well-known microservice architectures [6-8], (2) through the identification and selection of the clouds whose service models satisfy the cloud requirements, and (3) finally ends at the deployment of the application into selected clouds. For illustration, the method is applied to the migration of a Bed & Breakfast Booking System (B&BBS) to its cloud version.

The migration method

Step 1 – Baseline (Web-based) Identification

The method starts from the identification of the architecture and profile (i.e., the baseline) of the Web application. As an example, Figure 1 shows the architecture of a B&BBS where

1. It is a 4-layer structure where **Customers** and **B&Bs** access the system through four components: ***Access Interface***, ***Preference Agent***, ***Booking Agent*** and ***B&B Agent***.
2. ***Access Interface*** helps (1) **Customers** issue query or booking requests to and receive their preferred B&B information or booking confirmation from ***B&B Agent*** (via ***Booking Agent***); and (2) **B&Bs** update their information through ***B&B Agent***.
3. ***Preference Agent*** collects query or booking requests ever from **Customers** for analyzing their preferences to help ***B&B Agent*** catch their preferred B&B information.
4. ***Booking Agent*** forwards query or booking requests from **Customers** to ***B&B Agent*** that returns back the preferred B&B information or booking confirmation based on these requests.
5. ***B&B Agent*** catches the preferred B&B information for **Customers** based on their preferences from ***Preference Agent*** and then returns it to **Customers** (via ***Booking Agent***) for their booking decisions. It also helps to confirm the booking requests from **Customers** and accomplish the information updates from **B&Bs**.

With the application architecture, the next work is to capture its profile to size the application. In general, there are two kinds of profile data: (1) use data about its executions (e.g., CPU, memory, storage, I/O, and network uses); and (2) action data about its users (e.g., # of active users, request rates, transaction rates, and request/transaction latencies).

Step 2 – Target (Cloud-based Microservices) Identification

With the baseline architecture and profile, the next step is to identify the cloud requirements for satisfying its targeted cloud- deployed versions. For identifying the possible cloud-based architecture on which the components in the baseline may be deployed, we employ herein the well-known microservice architectures as discussed below.

1. As introduced in [8], the microservice architectural style is an approach to model and develop an application as a suite of small services with each running in its own processes and communicating among each other with light

mechanisms. In addition, it specifically organizes these services around

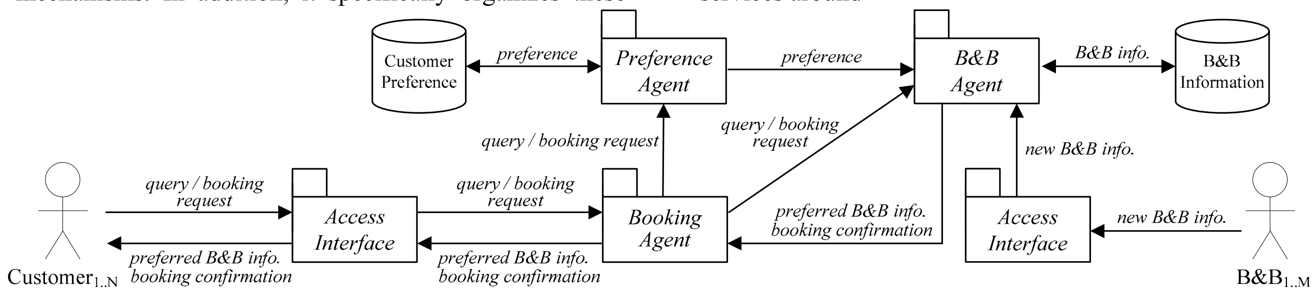

Figure 1: the architecture for Bed & Breakfast Booking Systems

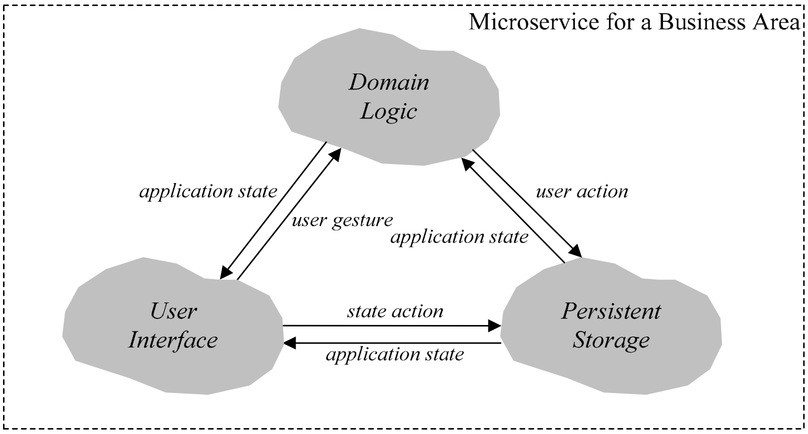

Figure 2: the cloud-based microservice architecture

business capabilities such that a broad-stack structure is taken for a specific business area, including for instance the user interface, the domain logic, the persistent storage, and any external collaboration for achieving the business. Therefore, as shown in Figure 2, a microservice has the architectural features as below.

2. It is organized around a business capability and hence structured with broad-stack modules for achieving the business such as the user interface, the domain logic, the persistent storage, and any internal/external collaboration.
3. It is encapsulated with a firm module boundary and hence can be developed, deployed and operated in an independent way.
4. It decentralizes transaction and data storage decisions and hence completes its own transactions and manages its own databases.
5. It is usually collaborating with other services to accomplish desired business capabilities (i.e., application functions). The collaborations may follow specific design patterns as presented in [8.
6. It needs extensive supports of infrastructure automation techniques and hence is suitable on the well infrastructure-supported cloud environments (i.e., IaaS/PaaS/IaaSDaaS models) for its development, deployment, and operation.

Thereafter, with the (cloud-based) microservice architecture, the identified requirements may include

1. For architectural components, they may be deployed on respective clouds for achieving a microservice. For instances in our B&BBS example, three microservices may be identified: (1) query requests for B&B information; (2) booking requests for B&B confirmation; and (3) B&B information updates. In particular, for these microservices,
   (1) ***Access Interface*** is imposed for helping **Customers/ B&Bs** get accesses with B&BBS and thus considerably deployed on a cloud for user interface.
   (2) ***Preference Agent*** is used for deriving the preferences of **Customers**; in addition, ***Booking Agent*** and ***B&B Agent*** are imposed for processing requests/updates for **Customers**/**B&Bs**; they are thus considerably deployed on a cloud for domain logic.
   (3) ***Customer Preference*** and ***B&B Information*** are used for storing the preferences of **Customers** and the B&B information from **B&Bs**; they are thus considerably deployed on a cloud for persistent storage.
2. For execution profiles, their QoSs in deployed clouds are required to support non-functional purposes such as customized user interfaces, access modes, performance, reliability, security, and scalability.

Therefore, as shown in Figure 3, the architectural components in our B&BBS example are deployed on respective clouds for achieving each of the three identified microservices. After then, for its non-functional purposes, such QoSs in these deployed clouds are required as customized user interfaces and B&B information/confirmation accesses, and reliability of the query/ booking/update services.

Step 3 – Candidate Identification

After identifying the requirements for clouds, the method continues to identify the candidate clouds whose service

models

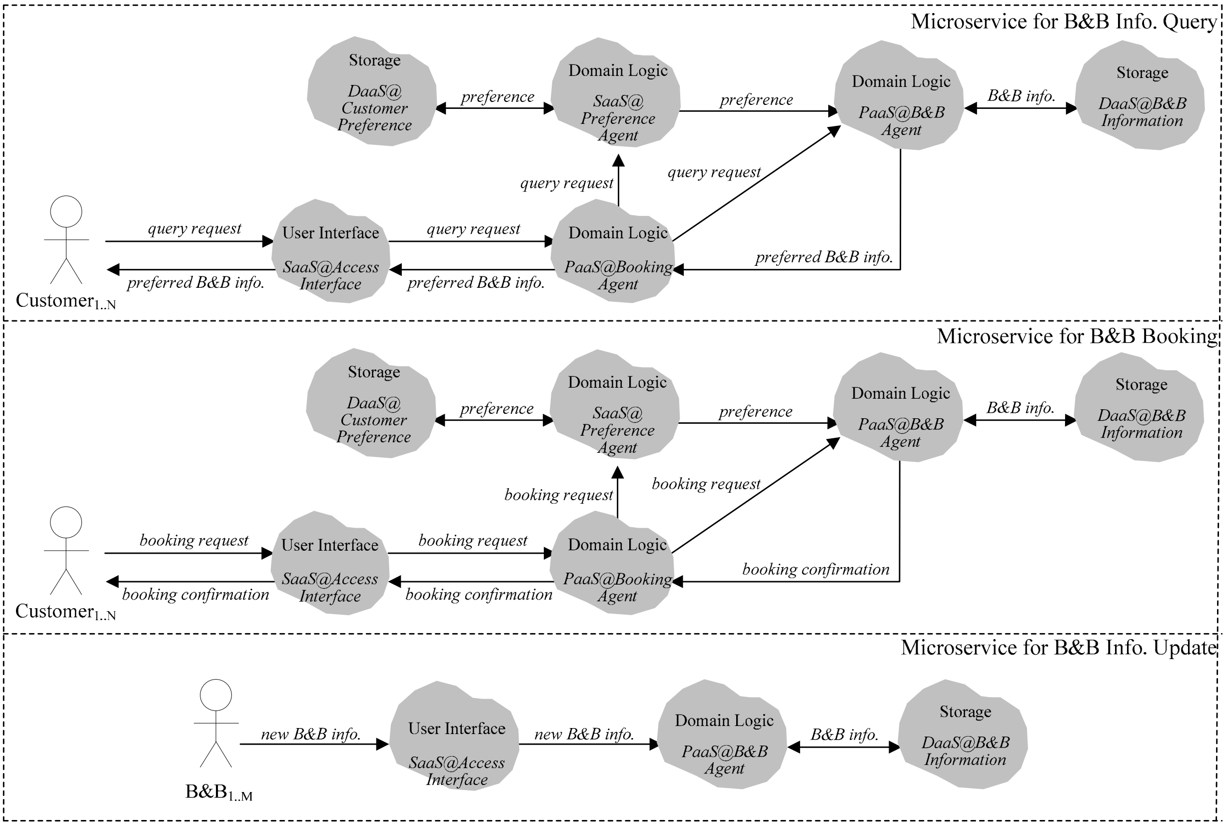

Figure 3: the microservice-structured architecture for Bed & Breakfast Booking Systems

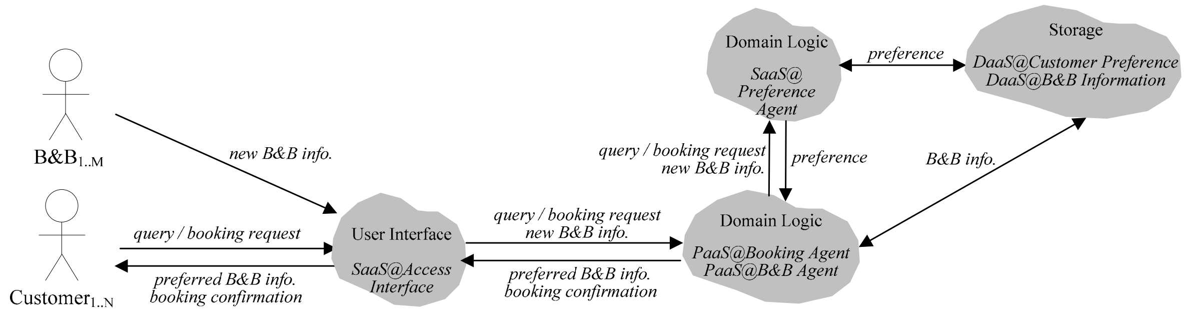

Figure 4: the cloud-based architecture of Bed & Breakfast Booking Systems

Figure 5: the IM B&B Booking Service (IM-B&BBS) for ***Access Interface***

– SaaS or PaaS or IaaS or DaaS satisfy these requirements. For this, therefore, it is good to consider all available environments that provide either of the following service models:

1. In SaaS model, the cloud provides services by which the application services can be replaced where specific QoS features are required for ensuring their replacement, e.g., SLAs, service replacement, and data/access portability.
2. In PaaS model, the cloud provides platform services on which the application can be deployed under specific QoS features such as SLAs, application deployment, service compatibility, and data/access portability.
3. In IaaS model, the cloud provides infrastructure services like servers, storages, and networks where the application and its residual platforms can use under such QoS features as SLAs, application deployment, service compatibility, and data/ access portability.
4. In DaaS model, the cloud provides storage services in which the application data can be stored under specific QoS features such a SLAs, data deployment/accessibility, and storage portability.

As a result, some clouds may be identified to satisfy the cloud requirements and then become the candidates for the migration. For example, Figure 3 shows the possible candidate clouds for the B&BBS where as an instance for ***Access Interface***, three SaaS clouds are identified as the candidates since their user- interface services may be provided by some available (existing) SaaS services to help respectively **Customers** and **B&Bs** access the B&BBS.

After then, with the required clouds for achieving the three microservices, their integration is considered such that services for the same context of functional purposes can be provided in a specific cloud. Therefore, as shown in Figure 4, the architectural components of the B&BBS are respectively deployed on four clouds where as an instance for ***Access Interface***, a SaaS cloud is required that provides the same context of user-interface services to help both of **Customers** and **B&Bs** access the B&BBS.

Step 4 – Cloud Selection

With the candidate clouds identified, the next step is to select from them the clouds to be migrated. In general, the selection can be achieved by the QoS features identified above for satisfying the cloud requirements. For example, based on the above QoS features, a candidate whose service models have the best credits may be selected as the targeted cloud. Continue the above instance for ***Access Interface***, a prototype SaaS cloud has been constructed by the Department of Information Management at Ming Chuan University, named IM B&B Booking Service (IM-B&BBS) as shown in Figure 5, and can be selected due to its well-developed downloadable APP for easy accesses with the B&BBS.

Conclusions

In this paper, we present a method for directing the migration of Web applications to selected clouds. In particular, it employs the well-known microservice architectures to support an effective migration of the Web applications into the clouds. For illustration, the method is applied to the migration of a B&BBS to its cloud version that takes advantages of cloud computing and its service models to help customers receive customized services in a low cost-, threshold-, and risk-way.

As our future work, we will continue to explore the real migrations of the B&BBS to the clouds where the most popular clouds [9] such as Google GAE and Amazon EC2 will be used as the deployed platforms. In fact, with the phased steps to gradually identify the application/cloud architectures and features and then conduct the deployment on the most suitable clouds, the quality of the migrated applications can be expected. Finally, for the most critical issues in the migration such as identifying available clouds and then selecting desired ones from them, we will also study explicit formal approaches such as semantic ontologies that support the identification and selection from available clouds in a systematic and managed manner.

Acknowledgements

The work in this paper is supported by the Ministry of Science and Technology in Taiwan under the Grant MOST 104-2410-H-130 -025.

References

[1] M. Chauhan and M. Babar, 2012, "Towards Process Support for Migrating Applications to Cloud Computing," Proc. of 2012 International Conference on Cloud and Service Computing, 80-87.

[2] W. Abderrahim and Z. Choukair, Sep. 2014, "A Framework Architecture Based Model for Cloud Computing Adaptive Migration," Proc. of 2014 Global Information Infrastructure and Networking Symposium (GIIS), 1-6.

[3] A. Gunka, et al., 2013, "Moving An Application to The Cloud: An Evolutionary Approach," Proc. of 2013 International Workshop on Multi-Cloud Applications and Federated Clouds, 35-42.

[4] S. Zardari and R. Bahsoon, 2011, "Cloud Adoption: A Goal-Oriented Requirements Engineering Approach," Proc. of 2nd International Workshop on Software Engineering for Cloud Computing, ACM, 29-35.

[5] R. Zota and A. Fratila, 2013, "Toward The Selection of An Enterprise Architecture Model for A Cloud Environment," Proc. of 11th RoEduNet International Conference, IEEE Computer Society, 149-154.

[6] W. Hasselbring, 2016, "Microservices for Scalability: Keynote Talk Abstract," Proc. of 7th International Conference on Performance Engineering, ACM/SPEC, 133-134.

[7] G. Toffetti, et al., 2015, "An Architecture for Self-Managing Microservices," Proc. of 1st International Workshop on Automated Incident Management in Cloud, 19-24.

[8] N. Viennot, et al., 2015, "Synapse: A Microservices Architecture for Heterogeneous Database Web Applications", Proc. of 10th European Conference on Computer Systems, Article No. 21.

[9] P. Fisher, 2011, *Cloud Computing: Assessing Azure, Amazon EC2, Google App Engine and Hadoop for IT Decision Making and Developer Career Growth*, Springer-Verlag.