Towards Model-Defined Cloud of Clouds

Xiaodong Zhang[1], Mauricio Tsugawa[2], Ying Zhang[1], Hui Song[3], Chun Cao[4], Gang Huang[*1], Jose Fortes[2]

[1]Peking University, China, {xdzh, zhang.ying, hg}@pku.edu.cn
[2]University of Florida, USA, {tsugawa, fortes}@acis.ufl.edu
[3]SINTEF, Norway, hui.song@sintef.no
[4]Nanjing University, China, caochun@nju.edu.cn

Abstract. With the growth in the number of Cloud Service Providers, many enterprises and organizations are now able to use multiple Cloud platforms in order to achieve improved overall Quality of Service (QoS), reliability and cost efficiency. However, due to the diversity in architecture and functionalities among different Cloud platforms, it is difficult to build a system that simultaneously manages multiple Clouds, i.e., a Cloud of Clouds. In this paper, we present a model-defined approach to the development of a Cloud of Clouds, and a real case study to demonstrate its feasibility.

Keywords: model-defined, Cloud of Clouds, runtime model

1 Introduction

With the growth in the number of Cloud Service Providers, many enterprises and organizations can use and combine services from multiple providers. The use of multiple Clouds brings many advantages: cost optimization, Quality of Service (QoS) improvements, high availability, avoiding vendor lock-in, disaster recovery and so on. Due to differences in services and exported interfaces (APIs), current practice is to interact with each Cloud separately: e.g., use CloudStack tools to interact with CloudStack Clouds and use VMware vCloud to interact with VMware Clouds. While this approach does not require additional software development, it has many disadvantages: administrators may need to continually switch among two or more completely different Cloud systems, the global and unified view of all available resources is not provided, and different Clouds do not interact with each other.

In order to efficiently use resources and services from multiple Cloud providers, there is a need for a management system that can interact with different Clouds offering a unified view of the entire system. Such management system is often referred as inter-Cloud, sky computing, or "Cloud of Clouds".

There are many challenges to the development of Cloud of Clouds. This work focuses on three main challenges: (1) each Cloud system has its own way of organizing resources, and exposes different management interfaces, making it difficult to develop an integrated and unified management system; (2) Cloud providers are physically

* Corresponding Author

located on geographically separate sites, and a networking system that can be dynamically reconfigured is needed to enable the interaction among different Clouds; and (3) there are different personalized management requirements consisting of specific scenarios and appropriate management styles.

A runtime model is a causally connected self-representation of the associated systems. We have developed a model-based runtime management tool called SM@RT (Supporting Model AT Run Time [6,7]), which provides the synchronization engine between a runtime model and its corresponding running system.

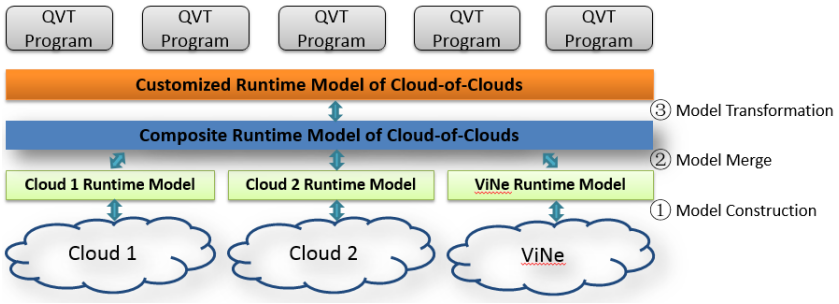

Fig. 1. Overview of Model-Defined Cloud of Clouds

In this paper, we present a model-defined approach to the development of a Cloud-of-Clouds (as shown in Figure 1). First, the run-time models of multiple Cloud platforms, and a virtual networking system (to enable and manage the connectivity among multiple Clouds) are constructed. ViNe[4] has been selected as the network virtualization component due to its dynamic network reconfiguration ability and availability of programmatic interfaces. Second, These run-time models are merged to form a composite model. Finally, the composite model is transformed into a customized model that meets the personalized requirements of an enterprise or an organization. After the three steps, all the participant Clouds can be managed in a unified and personalized manner through manipulations on the customized model. Network connectivity between all virtual machines on these Clouds can be enabled. Management tasks such as VM scheduling and VM placement can be carried out through executing different QVT (Query/View/Transformation) programs on the customized model, without the need to interact with the interfaces of underlying Cloud systems.

2 Related Work

In recent years, the interest in Cloud of Clouds has been increasing both in academia and industry. Some libraries such as Apache jclouds and Apache Libcloud provide abstraction layers facilitating the provisioning and deployment of multiple Cloud systems through a unified interface. They support numerous IaaS providers such as OpenStack, CloudStack, Eucalyptus and Rackspace. Commercial products and research projects such as RightScale and EnStratus also adopt a similar approach. While these products and projects effectively foster their deployment and maintenance, they remain on the code-level, which makes redesign difficult and error-prone. Liu et.al.

[1] propose a Multi-Cloud management platform that locates between Cloud users and Cloud sites and provides unified Cloud services from the SOA perspective. Ferry et.al. [2] propose a model-based framework called CLOUDMF to manage multiple Clouds. Liu and Ferry's work are on a higher level, but both lack the support of inter-Cloud network connection. Houidi et.al. [3] present a Cloud broker framework to enable inter-Cloud links between two different Clouds by using OpenFlow technology, however, it needs the network infrastructure to support OpenFlow, which is currently not widely deployed. Riteau et.al. [5] present an approach to building dynamic computing infrastructures over distributed Clouds and propose an inter-Cloud live migration mechanism called Shrinker. It shows good inter-Cloud network performance, however, it uses Nimbus to achieve the unified management of multiple Clouds, which is at code-level and not flexible to meet personalized requirements.

3 Example

We use an example to illustrate the proposed approach. There are two Cloud systems in different geographical locations: a Cloud managed by PKU InternetwareCloud[2], and a Cloud managed by OpenStack[3].

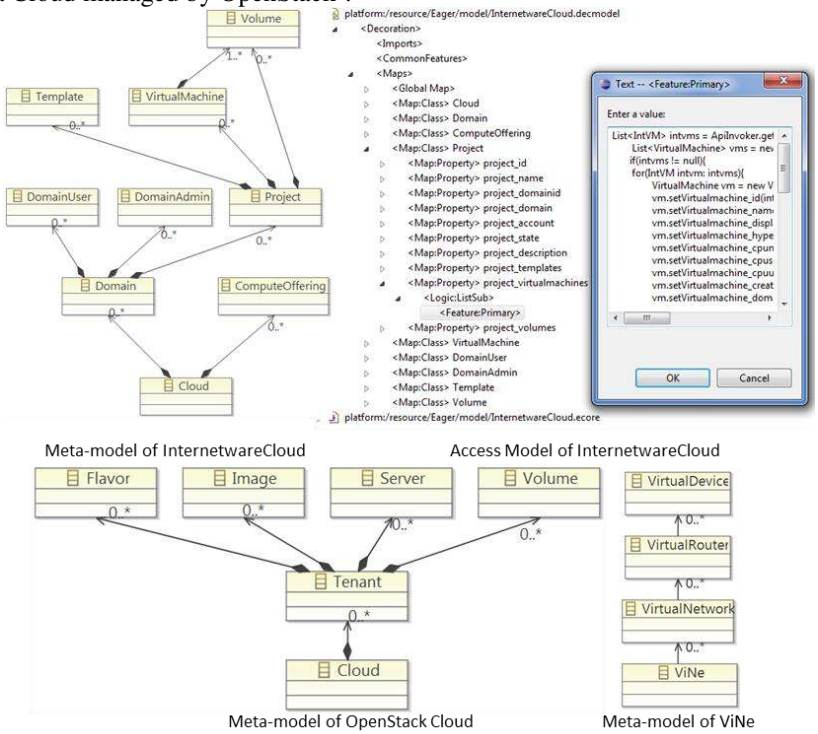

Fig. 2. Meta-model of each system and an access model sample

[2] A IaaS Cloud system. The Cloud website: http://internetwarecloud.acis.ufl.edu
[3] The Cloud website: https://openstack.futuregrid.tacc.utexas.edu/horizon

3.1. Construct the runtime models of both Clouds and ViNe

First, we construct the architecture-based meta-model and the access model, as shown in fig.2 (which illustrates the meta-model of InternetwareCloud). Although the architecture-based meta-model is constructed as an Ecore model, the equivalent UML model without the attributes of each class is shown for a better presentation. In this particular case, we abstract the following elements: Cloud, Domain, ComputeOffering, DomainUser, DomainAdmin, Project, VirtualMachine, Template, and Volume. The access model describes how to access and manipulate these elements. Taking the meta-model and the access model as input, SM@RT tool can automatically generate the runtime model.

3.2. Construct the composite model through model merge

In order to manage both Clouds in a unified way, we merge the runtime models of InternetwareCloud and OpenStack Cloud to construct the composite model. The composite model integrates all the managed elements and their management methods. We also propose a data synchronization mechanism between the Cloud models and the composite model. More details can be found in [8].

3.3. Transform the composite model to the customized model

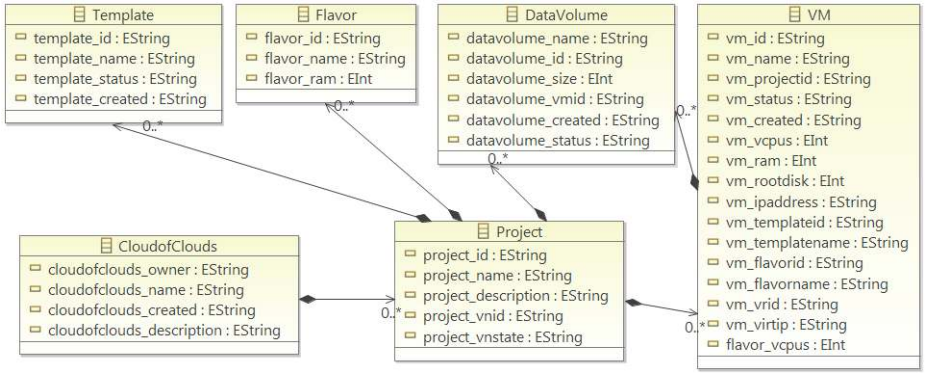

Fig. 3. The managed elements in the customized model

The composite model is an aggregation of the runtime models of each Cloud and ViNe. In order to meet the personalized management requirements of the Cloud of Clouds, we use the model transformation to transform the composite model to the customized model. The transformation is based on a set of mapping rules between the two models. Figure 3 shows the managed elements in the customized model. For example, the element of "Cloud" is transformed to the element of "CloudofClouds". The elements of "Tenant" and "VirtualNetwork" are transformed to the elements of "Project". The elements of "Server", "VirtualRouter" and the elements of "Volume" whose attribute "volume_type" is "root" are transformed to the element of "VM".

3.4. Inter-Cloud network connection

ViNe is a user-level virtual network software developed at the University of Florida. The ViNe network consists of a set of Virtual Networks (VNs). Each VN consists of a set of Virtual Routers (VRs). Any machine deployed with ViNe software can serve as

a VR and VRs in the same VN have network connection with each other. In this case study, every VM serves as a VR. We transform the elements of "VirtualRouter" and the elements of "Server" to the elements of "VM". Every time an element of "VM" is created on the customized model, an element of "VirtualRouter" will be created on the runtime model of ViNe, and the ViNe software will be configured automatically on this VM. By using our approach, the administrators don't have to do the virtual network configuration manually, which can be complex and error-prone.

4 Conclusion and Future Work

In this paper, we presented a model-defined approach to the development of a Cloud of Clouds. Through model construction, model merge and model transformation, multiple Clouds can be managed in a unified and personalized manner. By merging with the runtime model of ViNe, inter-Cloud communication is enabled in the model-defined Cloud of Clouds, which considerably reduces the need for user interventions.

In a runtime model, all the resources are modeled as managed elements and all the management interfaces are abstracted as manipulations on the elements. From this perspective, there's no difference between developing based on system management interfaces with developing based on runtime models in regards to feasibility. In our approach, we also guarantee the synchronization between the customized model, the composite model and the distributed models. Therefore, our approach is feasible.

We plan to leverage the developed base system to implement advanced management tasks such as virtual machine placement in a Cloud of Clouds environment.

ACKNOWLEDGMENT. This work is supported by the National High Technology Research and Development Program of China (863 Program) under Grant No. 2013AA01A208; the National Natural Science Foundation of China under Grant No. 61361120097 and No. 61300002.

References

1. Liu, T., et.al. Multi cloud management for unified cloud services across cloud sites. CCIS 2011. IEEE.
2. Ferry, N., et.al. Managing multi-cloud systems with CloudMF. In Proceedings of the Second Nordic Symposium on Cloud Computing & Internet Technologies (pp. 38-45). ACM.
3. Houidi, I., et.al. Cloud service delivery across multiple cloud platforms. SCC 2011. IEEE.
4. Tsugawa, M., et.al. A virtual network architecture for grid computing. IPDPS 2006. IEEE.
5. P. Riteau. Building dynamic computing infrastructures over distributed clouds. NCCA 2011. IEEE.
6. H. Song, et.al. Generating Synchronization Engines between Running Systems and Their Model-Based Views. Models in Software Engineering, Pages 140-154, 2010.
7. H. Song, et.al. Inferring Meta-Models for Runtime System Data from the Clients of Management APIs. MODELS 2010.
8. X. Zhang, X. Chen, Y. Zhang, Y. Wu, W. Yao, G. Huang, Q. Lin. Runtime Model Based Management of Diverse Cloud Resources. MODELS2013