Monsoon: Policy-Based Hybrid Cloud Management for Enterprises

Shi Xing Yan, Bu Sung Lee, Guopeng Zhao, Chunqing Chen, Ding Ma, Peer Mohamed

HP Laboratories
HPL-2012-137

Keyword(s):
Cloud Management; hybrid cloud;

Abstract:
Cloud Computing has become more and more prevalent over the past few years, and we have seen the emergence of Infrastructure-as-a-Service (IaaS) which is the most acceptable Cloud Computing service model. However, coupled with the opportunities and benefits brought by IaaS, the adoption of IaaS also faces management complexity in the hybrid cloud environment which enterprise users are mostly building up. A cloud management system, Monsoon proposed in this paper provides enterprise users an interface and portal to manage the cloud infrastructures from multiple public and private cloud service providers. To meet the requirements of the enterprise users, Monsoon has key components such as user management, access control, reporting and analytic tools, corporate account/role management, and policy implementation engine. Corporate Account module supports enterprise users’ subscription and management of multi-level accounts in a hybrid cloud which may consist of multiple public cloud service providers and private cloud. Policy Implementation Engine module allows users to define the geography-based requirements, security level, government regulations and corporate policies and enforces these policies to all the subscription and deployment of user’s cloud infrastructure.
Monsoon: Policy-Based Hybrid Cloud Management for Enterprises

Shi Xing Yan*, Bu Sung Lee*, Guopeng Zhao*, Chunqing Chen*, Ding Ma*, Peer Mohamed*

*HP Labs Singapore
1 Fusionopolis Way
Singapore 138632
{shixing.yan, francis.lee, guopeng.zhao, chunqing.chen, dingm, peer.mohamed}@hp.com

Abstract—Cloud Computing has become more and more prevalent over the past few years, and we have seen the emergence of Infrastructure-as-a-Service (IaaS) which is the most acceptable Cloud Computing service model. However, coupled with the opportunities and benefits brought by IaaS, the adoption of IaaS also faces management complexity in the hybrid cloud environment which enterprise users are mostly building up. A cloud management system, Monsoon proposed in this paper provides enterprise users an interface and portal to manage the cloud infrastructures from multiple public and private cloud service providers. To meet the requirements of the enterprise users, Monsoon has key components such as user management, access control, monitoring and analytic tools, corporate account/role management, and policy implementation engine. Corporate Account module supports enterprise users’ subscription and management of multi-level accounts in a hybrid cloud which may consist of multiple public cloud service providers and private cloud. Policy Implementation Engine module allows users to define the geography-based requirements, security level, government regulations and corporate policies and enforces these policies to all the subscription and deployment of user’s cloud infrastructure.

I. INTRODUCTION

Infrastructure-as-a-Service (IaaS) is one of the service models of cloud computing which offers IT infrastructure including compute, storage and network as a service. The cloud infrastructure is provided to the users in a more scalable and elastic way by IaaS providers through pay-per-use business model. IaaS becomes extremely attractive to IT managers and CIO when it comes to outsourcing IT resources. CIO and IT managers see the possible benefits in terms of reducing capital cost, global accessibility and flexibility on top of other advantages. However, there are also many concerns such as vendor lock-in, security, availability, portability, etc. Many cloud service providers in this emerging market, or more specifically the majority of IaaS providers, are providing similar services with slightly different offerings, but the proprietary APIs to each service are all different which stop the users switching from the service providers. Hence, the standard interfaces become crucial to address these issues to accelerate the adoption of cloud computing. If we look at the exponential growth of Internet after the widely adoption of World Wide Web (WWW) in 1990s, it is an interesting example showing the power and importance of standardization. DMTF Open Virtualization Format (OVF), which is one of the pioneer cloud standards, provides a packaging standard designed to address the portability and deployment of virtual appliances [1]. From OVF’s successful story of being adopted by ANSI [2], we can see that industry users need interoperability and portability in the cloud computing era.

The lack of standards for configuration and management of cloud services is still a major obstacle on the road for the companies who are considering the adoption of IaaS, because the enterprise users may need to manage a hybrid cloud environment which includes both private cloud and public cloud. Since each IaaS service provider has similar provision of cloud computing service, it is possible to design a generic model for each category of service. We have proposed a generic abstraction model of IaaS services to facilitate multisourcing IaaS for increasing availability and security requirements [3]. This generic IaaS model abstracts the commonly used operations from existing key service providers, and it has been designed with full coverage of functionalities to satisfy IaaS users’ requirements. As part of the modeling of IaaS resources, we have defined core elements and attributes as well as extension to cover the additional services provided by different IaaS providers. A common interface has been developed with the implementation of the generic IaaS model in a model-based service integration environment, Web2Exchange [4], and users can use this common interface to access and manage a shared pool of cloud infrastructure from multiple IaaS providers without need to be aware of the different proprietary interfaces for different service providers [5].

Besides demanding the cloud infrastructure provide high availability and security for their business critical system, the enterprise users will also need some more advanced features for their cloud infrastructure management. For example, the concept of Corporate Account is very appealing for enterprise users because they can subscribe to cloud services and allocate users in their organization to different roles. Although Corporate Account has been adopted widely by many Software-as-a-Service (SaaS) providers to support multi-tenancy and organization, the major IaaS providers currently do not have implementation of Corporate Account in the market. In this paper, we will first briefly describe our common interface for the management of hybrid cloud called unified IaaS proxy service in Section II. The IaaS proxy model and its implementation on Web2Exchange are covered in detail.
giving the generic entities of IaaS. In Section III, we will introduce Monsoon which provides a multi-tenant infrastructure management portal for hybrid cloud. In Section IV we will discuss the related work and current issues. Conclusions and future work are described in Section V.

II. UNIFIED IAA S PROXY

A. Generic IaaS Model and Its Evolvement

Since each IaaS service provider has similar provision of cloud computing service, it is possible to design a generic model by abstracting the commonly used operations from existing key service providers. The proposed service models are designed with full coverage of functionalities, which can satisfy IaaS users’ requirements.

As previously proposed in [5], the generic IaaS model inherits from the model-based approach, and contains models for both types and services. The type models provide necessary parameters as well as information about instances of types which are exposed by a set of services during service invocations. Service models cater for basic functionalities which are required to fulfill operational requests.

There are various types in the proposed IaaS model: Resource type, such as compute, storage; Reflection type such as image and snapshot; and other types such as credential and firewall. A number of services are proposed in the service model, including Resource Management Service, Virtual Machine Service, Clone Service and Security Service. The details for both type and service models are described in [5].

For private cloud, enterprise users usually require to manage a cluster of virtual machines. They are also managed at different levels, from the entire cluster to each specific virtual machine. In order to support the enterprise usage on clusters, the generic IaaS model has evolved to include modeling of a cluster of virtual machines. Virtual machine belongs to the Resource type, and is a composition of multiple computes, storages and networks. A collection of virtual machines form a server group, and a cluster consists of several server groups, as shown in Fig. 1. Cluster Service manages the life cycle of a cluster; user can create, start, stop and terminate an entire cluster or a specific server group. In addition, Cluster Service provides services such as monitoring and reporting at various levels.

The detailed implementations of service calls to various IaaS providers are transparent to end users. Users will be only aware of the common interface provided by IaaS proxy in order to communicate with all the supported IaaS providers. With the generic service models, it is possible to support an arbitrary number of service providers. In case of API changes of a specific provider, only the underlying implementation parts need to be modified while the service models need not to be changed.

Figure 1. Modeling of Clusters

B. Implementation in Web2Exchange

Web2Exchange provides a model-based service integration environment to facilitate the service deployment, discovery, and integrations. The models of services on Web2Exchange are described by Managed Object Format (MOF) which is a readable and writable syntax with extension of the DMTF CIM meta-model [4]. In Web2Exchange the service developers can define service models directly in MOF or use source-code annotations to have Web2Exchange to define the service logic.

We implemented our service models for IaaS as a unified IaaS Proxy in Web2Exchange [5]. The IaaS proxy provides a common interface to manage IaaS environments across public clouds and private clouds, so the users only need to interface with our IaaS proxy but not specific APIs from various IaaS providers as shown in Fig. 2. The IaaS proxy can be discovered and made available to other services running on Web2Exchange, and it also provides REST interface to the users and services situated outside of Web2Exchange. For current version, the IaaS proxy supports major public clouds such as Amazon Web Service (AWS) EC2, GoGrid, Rackspace Cloud, HP Cloud and private clouds, such as HP CloudSystem Matrix.

III. INFRASTRUCTURE MANAGEMENT PORTAL

To provide enterprise users a comprehensive management tool for their cloud infrastructure in hybrid cloud environment, we have designed a Monsoon portal which evolved from the IaaS Aggregator portal introduced in [5]. IaaS Aggregator is an early prototype of our infrastructure management tool with basic functionalities for life cycle management. Our Monsoon portal covers both basic cloud infrastructure management tools and advanced features to meet the enterprise users’ requirement in hybrid cloud environment. The overall system architecture of Monsoon is shown in Fig. 2. We can see that the Monsoon portal sits on top of the unified IaaS proxy, and the requests to various IaaS providers across public and private clouds will be interpreted through the RESTful interface of IaaS proxy.

A. Basic Cloud Infrastructure Management Features

The Monsoon portal provides a basic set of features to manage the cloud infrastructure in hybrid cloud. Some of these basic features are listed as follows.
User Management: This module provides basic user and privilege management which are necessary for a multi-tenant management portal. The Monsoon portal requires that all the users need to be registered and authenticated. There is no anonymous user allowed to access the portal. There is a group of portal users with portal Administrator right who are responsible for authorizing the newly registered user and daily maintenance of the Monsoon portal. However, portal Administrator will have no access to users’ confidential data such as API keys to cloud service providers. All the user data are encrypted and stored securely in data layer through secure connection, and only available to the users themselves.
Access Control: This module coordinates with User Management module to provide authentication, authorization, and access control to ensure the portal users only gain access to the authorized resources.

Resource Management: This module provides users a dashboard to manage their cloud infrastructure in hybrid cloud as shown in Fig. 3. All types of virtual resources can be provisioned and managed across public and private clouds. Fig.3 shows that cloud resources from different service providers are currently managed in Monsoon. Each portal user will have their API keys registered with this module to allow Monsoon be authenticated with various cloud service providers.

Monitoring and Reporting: This module enables portal users to monitor his cloud infrastructure in real time. A set of tools will provide the graphs to show statistics of the virtual resource, alert setting, and report generation according to user’s requirements. It can be seen from the screenshot in Fig. 3 that one AWS virtual machine is being monitored in Monsoon, and there are some graphs of its real-time statistics are shown in the bottom of the figure. Besides the real-time and near term statistics for the cloud infrastructure that are monitored and recorded in Monsoon, the historical statistics have also been archived to facilitate the analytics on these data in the future.

B. Advanced Features for Enterprise Users

We have described a few basic modules in the previous section which provide some of the basic cloud infrastructure management in Monsoon. However, in addition to basic management capability enterprise users usually require richer features to fit their needs on security, account/role management, billing, etc. Monsoon provides advanced features to complement the current offerings from cloud service providers. We will choose a few enriched features provided by Monsoon to describe their functionalities and design.

Corporate Account: We have seen that Corporate Account has been widely implemented by SaaS providers and very successful as one of the core business models for cloud computing. Although many private cloud solutions have the capability of provisioning virtual resources in a way identical to Corporate Account, there is no IaaS provider to provide Corporate Account for enterprise users in public cloud. The implementation of Corporate Account in Monsoon will not only complement the lack of single IaaS provider’s offering, but also enable enterprise users to have a centralized Corporate Account for the infrastructure management in hybrid cloud.

The concept of Corporate Account requires the service provider to enable an organizational relationship being mapped into the user/role management, access control and other components in a multi-tenant cloud. This is also different from the current IaaS offering where one virtual resource is designed to be allocated to one user only. A corporate can subscribe to multiple cloud services, and in Monsoon its Corporate Account has a centralized control over all the resources under this corporation. A corporate admin can assign users in their organization to different roles, and different access rights will be assigned to roles. For example, a company assigns employees two roles, developer and tester. Developers are allowed to start and stop a web server, while testers are only allowed to access the web service running on the server. Corporate Account also enables separation of computing resources. For example, the machines of a finance department are not accessible by a developer department. In this way different departments also manage their cloud resources and IT budget for cloud services separately. Moreover, Corporate Account allows the company to have a centralized control, so that spending on cloud services across the entire company can be well managed.

User and Role are two important domain models for the user management in Monsoon. User represents the actual users of the system, and a role is assigned to a user according to the organizational relationship. Therefore a user may have multiple roles and a role can be assigned to multiple users. The role of a user immediately determines the access right of this user, and we only examine his role to decide whether a specific permission can be granted. To support role hierarchy in an organization, a “reports to” relation is modeled, which allows multiple subordinate roles to be supervised by a superior role. Our domain models for user management enable a company to have customizable access control in terms of management hierarchies. A company has multiple Organizational Units (OUs), and an OU allows sub units/departments to implement various levels of the organizational management structure. Multiple resources, resource groups, users, and roles are managed under an OU, but each OU can only be managed by one specific role, i.e., the manager role of the OU. We have adopted and extended the Spring Security framework [6] in Monsoon to achieve granular access control for enterprises to meet their own requirements.

Policy Implementation: Enterprises will not commit to cloud computing, or more specifically IaaS, before IaaS can meet their service level requirements for business critical applications. With Monsoon the enterprise users are able to benefit from multi-sourcing IaaS where the enterprise may deploy their services with more than one IaaS provider and private cloud. Many enterprises do have rigorous requirements for their IT deployment, and these requirements also need to be fulfilled when their IT deployment is in hybrid cloud. For example, the enterprise’s applications/services which require high availability need to be deployed with capability of the geo-redundancy which allows the applications/services to switch from one site to another geographically distant site.

In many countries, some industries, such as banking and healthcare, are also ruled by data protection laws regarding the customer’s personal information, so the enterprise should prevent such data to be stored in servers located in another country. However, data may be stored anywhere in the cloud as many cloud service providers make the physical location of the data transparent to the users. Enterprise may need to employ multiple cloud service providers and utilize private cloud to comply with regulatory requirements on jurisdiction and legal obligation.

Enterprise may have different but interdependent rules that need to be applied in their IT infrastructure as well. Different types of data and applications may require cloud resources to have certain security level, e.g., some applications or services
are not supposed to be put in the same subnet. On the other hand, some applications or services have a lot of data exchange and therefore they should reside in same intranet to optimize the performance. Some enterprises define the rules on load balancing, firewall, etc.

Different service providers may have different properties satisfying enterprise’s policy requirements. In the course of resource provision, automatically implementing the IT policy will alleviate a lot of manual work for a corporation. Policy Implementation Engine (PIE) is proposed to assist enterprise users in defining and implementing policies. The engine contains both a service provider directory and a context-aware policy mapping component. The design of the PIE is shown in Fig.4.

The service provider directory is designed to allow a company which is a corporate account in Monsoon to define the geography-based requirements, security level, government regulations and corporate policies. The policies defined by the company will be implemented to conform to the compliances for the subscription and deployment on IaaS resources by the mapping component. The portal administrator maintains a global repository which stores all the cloud service providers supported in Monsoon. The portal administrator composes all the cloud service providers’ specifications including security level, geographical coverage, price, and other information about each service provider’s cloud offering. These specifications published by the portal administrator become the default value in the global cloud service provider directory. For every Corporate Account there are a group of administrators for this corporation who are responsible to manage the users and resources. If the corporate organization finds that the default service providers’ specification does not fit its needs (e.g., having different interpretation of the security level), the Corporate Admin can customize the specifications (e.g., the security level) and the customized information stored in service provider directory is only visible and applicable to this corporation.

The portal user belonging to one corporation (meaning the user is under a Corporate Account) in the Monsoon can specify his requests for resource provisions which are submitted to PIE. The context-aware policy mapping component searches through the resource pools of service providers, checking against the user’s corporate policies in the service provider directory. We categorize a specification of a policy into hard and soft. A hard specification posts a hard constraint that is necessarily to be satisfied and cannot be relaxed, while a soft specification expresses a preference and can be optimized. For example, government regulations on data protection belong to the hard category, and budget belongs to the soft category. Context-aware policy mapping firstly filters out resources that violate the hard specifications, and then orders the resources according to the soft specifications. A request of the resource which optimizes the soft specifications is generated, and sent to Resource Management module for provisioning. The cloud resource will be ready for use after resource provisioning step.

We have prototyped the PIE based on the Satisfiability Modulo Theories (SMT) and constraint programming techniques [7]. Our prototype can automatically detect violation against compliances of policies and inconsistencies among user requests, and select appropriate providers fulfilling both hard and soft specifications.

Billing and Charging: Monsoon provides the portal to manage cloud infrastructure across public and private clouds including multiple service providers, so the billing and charging from different vendors could be an issue to the users. Enterprise has its own organizational relationship and users under Corporate Account in Monsoon. Its cloud resources may come from its users’ subscriptions with different service providers. Hence, it is crucial for the enterprise users to have a module in Monsoon to aggregate the billing and charging.
information for his Corporate Account. Billing and Charging module provides the ability to accurately track usage of cloud infrastructure and properly bill the charge to the users who could be within an organization under a Corporate Account.

IV. RELATED WORK

Interoperability in a hybrid cloud environment attracts more and more attention recently. There are several working groups engaged in developing cloud standards or management tools to address this issue. From the infrastructure management point of view, some of the initiatives on the cloud standard and cloud projects are related to our project, and we will discuss and compare them with Monsoon.

Initiated by Open Grid Forum, Open Cloud Computing Interface (OCCI) is a RESTful protocol and API for all kinds of management tasks [8]. It focuses on interoperability across clouds with a high degree of extensibility. Sitting on the boundary of a service provider, OCCI acts as a service front-end to provider’s internal management framework. It requires non-trivial effort for providers to adapt this standard by rewriting their existing proprietary APIs. Moreover, the community currently lacks of implementations of OCCI for major cloud providers, such as Amazon and GoGrid. DMTF’s Cloud Management Working Group (CMWG) is chartered to address management interoperability for clouds between service consumers and providers [9]. It aims to deliver a set of specifications including resource models and interaction protocols. It has similar goal with the generic IaaS model in our project. However, it is still in the development phase and lacks of practical implementation examples to support. The Storage Network Industry Association (SNIA) created the Cloud Data Management Interface (CDMI) to enable interoperable cloud storage and data management [10]. Management of cloud storage belongs to the overall cloud computing management. In our research work, the discussion of a specific protocol for managing cloud storage is out of our scope. We focus on the overall infrastructure management for cloud resources, while storage is considered as part of the resource provisioning. OpenNebula is an open-source project to provide a comprehensive solution from setting up a private cloud to cloud infrastructure management across hybrid clouds [11]. Compared to Monsoon, it has some overlaps, such as supporting both public and private clouds. However, OpenNebula uses computing capacity from a public cloud as a means to supplement local infrastructure to meet requirements such as peak demands and high availability. Hence, there is little intention for OpenNebula to focus on interoperability across a wide range of cloud providers, and it only supports EC2 and OCCI for current release. DeltaCloud is an open-source project, aiming to provide one common API for a wide range of service providers [12]. The approach which tries to aggregate multiple proprietary APIs is similar to our common IaaS APIs, but DeltaCloud focuses on the convenience of using one API, instead of complete stack of management tools for enterprise users. Thus, it is merely comparable to the unified IaaS proxy in our project. RightScale also provides management tools to manage the cloud infrastructures over multiple public cloud providers, but it only provides a subset of the basic management features compared to Monsoon [13].

Compared to the related work mentioned above, the unified IaaS Proxy which is one part of Monsoon is similar to some projects in terms of providing a common APIs for IaaS. Some project also provides a web portal or management tools to provide the basic management features, but Monsoon also has many advanced features to complement with cloud service providers’ offering, particularly for enterprise users; e.g., Policy Implementation Engine and Corporate Account. Most of these projects are targeted to manage heterogeneous public clouds only, and Monsoon is designed to manage private, public, and hybrid clouds.

V. CONCLUSION AND FUTURE WORK

Monsoon aims to provide comprehensive cloud infrastructure management in a hybrid cloud environment for enterprise users. A unified IaaS Proxy provides a common interface to manage cloud infrastructures based on an abstraction IaaS model implemented on Web2Exchange. A Monsoon portal is developed to enable users to subscribe, monitor, and manage the full life-cycle of IaaS services from multiple providers across public and private clouds. The unified IaaS Proxy deployed in Monsoon is able to support AWS, GoGrid, RackSpace Cloud, HP CloudSystem Matrix and other AWS-like or Rackspace-like IaaS providers such as OpenStack [14] and Eucalyptus [15]. The advanced modules in Monsoon built for enterprise users, such as Corporate Account, Policy Implementation Engine, greatly help enterprise users to manage their hybrid cloud environment in a more effective manner.

REFERENCES


