International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 07, July 2015
Available at http://internationaljournalofresearch.org

Meta Cloud: Improve the Performance of Winds of Change from Vendor Lock

Mrs. Laxmi Lingutla[1] & Mr. Vakiti Srinivas Reddy[2]

[1] Associate Professor ,Department of Computer Science & Engineering, MLR institute of Technology, Hyderabad, Telangana, India ,Email: laxmi.slv@gmail.com
[2] Department of Computer Science & Engineering, MLR Institute of Technology, Hyderabad, Telangana, India, Email: srinivasreddy71412@gmail.com

ABSTRACT

In view of low expenses and high flexibility, moving to the cloud is in reality convincing. Regardless of the evident favorable circumstances of distributed computing, numerous organizations falter to move into the cloud, basically in view of concerns identified with accessibility of administration, information lock-in, and legitimate vulnerabilities. Lock-in is especially hazardous for the accompanying reasons. Firstly, despite the fact that accessibility of open mists is for the most part high, inevitable blackouts still happen. In the event that this is the situation, organizations bolted into such a cloud are basically at a halt until the cloud is back on the web. Also, open cloud suppliers for the most part don't promise specific administration level assertions, i.e., organizations bolted into a cloud have no ensures that this cloud will keep on providing the obliged Quality of Service (Qos) tomorrow. The terms of administration of most open cloud suppliers permit the supplier to singularly change evaluating of their administration whenever. Subsequently, a business secured a cloud has no mid- or long haul control over their IT costs. Currently migrating to another cloud is not ignorable. Services like EC2 and S3 are incompatible and doesn't follow specific standards but follow their own. For complicating still more many companies not only build on public on Public Clouds but their own Private Clouds will be combined with public cloud, evolving to the hybrid cloud. The concept of Meta Cloud is introduced that includes runtime and design time components. This Meta Cloud minimizes the technical incompatibilities and thus switching to another cloud. This helps the customers to choose appropriate cloud services for particular appropriate use case.

Keywords: Meta Cloud Quality of Service; accessibility

INTRODUCTION

Cloud computing provides lots of services to users and storage benefits to users. Since there is demand for storage in cloud we need to manage the ever-increasing data [1]. So we proposed an Authorized intelligent data compression technique known as de duplication which eliminates the duplicate copies of data in cloud and store only unique files. Since there is lot of issues with quality of service and technical failure occurring the cloud can migrate the data from one cloud to another by providing security.

Cloud service providers provide both parallel computing resources and available storage at low cost [2]. As increasing, amount of data tend to be stored in cloud and provided users who have specified privileges. As the stored volume of data is increasing in the cloud, we need to manage the stored data. To manage the data in cloud computing, the technique

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 07, July 2015
Available at http://internationaljournalofresearch.org

called De duplication is being used which has recently gained more attention.

Data de duplication is a well know intelligent data compression technique used for eliminating the duplicate copies of redundant data in the storage. It is the best technique which can be used for improving the storage utilization and also used to reduce the number of bytes transferred over the network. Instead of keeping the redundant copies of same data it eliminates and keeps only one copy of the physical data and referring the other duplicate data to that copy [6]. Since intelligent data compression technique has lot of benefits, it is sensitive to both outsider and insider attacks. In the traditional encryption system the de duplication process is very difficult in aspects of confidentiality because the users had to encrypt data with their own keys that will provide the different cipher texts for the same identical two copies of data that will make de duplication impossible to be carried out.

Convergent encryption technique has been proposed that will provide confidentiality which makes de duplication feasible. It uses convergent key to encrypt/decrypt the data copy which is obtained after computing the value cryptographic hash of the content. After the key is generated, the users will retain keys and later send only the cipher text of the content to the cloud. The same copy of the data will produce same convergent key that will make de duplication possible. Since to prevent attacks and unauthorized access, the secure proof of ownership concept is provided to authenticate the users. The file can be decrypted only with the convergent key corresponding to Data Owner [7]. Thus the proof of ownership will prevent unauthorized users to access and convergent encryption will allow the de duplication to perform on cipher texts. Each of the file uploaded to the cloud has set of privileges that will allow specifying the access rights and duplicating check to be performed by users. The users who have the particular privileges only will be allowed to carry out the duplication check for the files stored in the cloud [3][4].

However, migration is currently far from trivial. Myriad cloud providers are f loading the market with a confusing body of services, including computer services such as the Amazon Elastic Compute Cloud (EC2) and VMware vs Cloud, or key-value stores, such as the Amazon Simple Storage Service (S3). Some of these services are conceptually comparable to each other, whereas others are vastly different, but they're all, ultimately, technically incompatible and follow no standards but their own. To further complicate the situation, many companies not (only) build on public clouds for their cloud computing needs, but combine public offerings with their own private clouds, leading to so-called hybrid cloud setups [12]. Here, we introduce the concept of a Meta cloud that incorporates design time and runtime components [5]. This Meta cloud would abstract away from existing offerings' technical incompatibilities, thus mitigating vendor lock-in. It helps users find the right set of cloud services for a particular use case and supports an application's initial deployment and runtime migration.

RELATED WORK

Let's consider a Web-based sports portal for an event such as the Olympic Games, which allows users to place bets. An event this large requires an enormously efficient and reliable infrastructure, and the cloud computing paradigm provides the necessary flexibility and

International Journal of Research
International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 07, July 2015
Available at http://internationaljournalofresearch.org

elasticity for such a scenario [8]. Cloud offerings from a multitude of service providers calls for a meta cloud to smoothen the edges of the jagged cloud landscape. This meta cloud could solve the vendor lock-in problems that current public and hybrid cloud users face.

Service providers handle short-term usage spikes without needing respective dedicated resources available continuously. The problem, however, is that once an application has been developed based on one particular provider's cloud services and using its specific API, that application is bound to that provider; deploying it on another cloud would usually require completely redesigning and rewriting it. Such vendor lock-in leads to strong dependence on the cloud service operator [13]. In the sports portal example, in addition to the ability to scale applications up and down by dynamically allocating and releasing resources, we must consider additional aspects, such as resource costs and regional communication bandwidth and latency. Let's assume the sports betting portal application is based on a load balancer that forwards HTTP requests to numerous computing nodes hosting a Web application that lets users submit a bet. Request handlers place but records in a message queue and subsequently store them in a relational database.

Let's further assume a service provider realizes this scenario using only Amazon Web Services (AWS), EC2 to host applications, Simple Queue Service (SQS) as its cloud message queue, and the Relational Database Service (RDS) as a database system [9]. Instead of being bound to one cloud operator, however, the betting application should be hosted in an optimal cloud environment. To leverage a more diverse cloud landscape, support flexibility, and avoid vendor lock-in, the meta cloud must achieve two main goals:

• Find the optimal combination of cloud services for a certain application with regard to QoS for users and price for hosting; and

• Develop a cloud-based application once, and then run it anywhere, including support for runtime migration.

The meta cloud utilizes asset formats to characterize cement emphasizes that the application requires from the cloud. Case in point, an application must have the capacity to indicate that it obliges a given number of figuring assets, Internet access, and database stockpiling. Some present apparatuses and activities — for instance, Amazon's Cloud formation (http:// aws.amazon.com/ cloud shaping/) or the approaching TOSCA detail (www. Desert spring open.org/ councils/ tosca) — are moving in the direction of comparable objectives and can be adjusted to give these obliged peculiarities to the meta cloud [12].

Based on Open Stack, and enhanced with Meta cloud's enterprise-targeted characteristics, our awarded platform fives completely secure, highly available, scalable, and customizable congregated calculate, preservation, and environment of networking to your information center. Dashboard: Open Stack dashboard of Meta cloud is a complete web portal self-service for people of administration and end users to supervise their clouds. By exploring the Open Stack capability API we create it simple to make up and arrange storage, instances, and projects. The dashboard also gives transparency into the utilization and performance of user cloud through Meta cloud's observing services [14]. Finally, we are familiar with how time overwhelming client management is, so our dashboard contains integration with enterprise certification systems so user don't have to reconstruct their user base

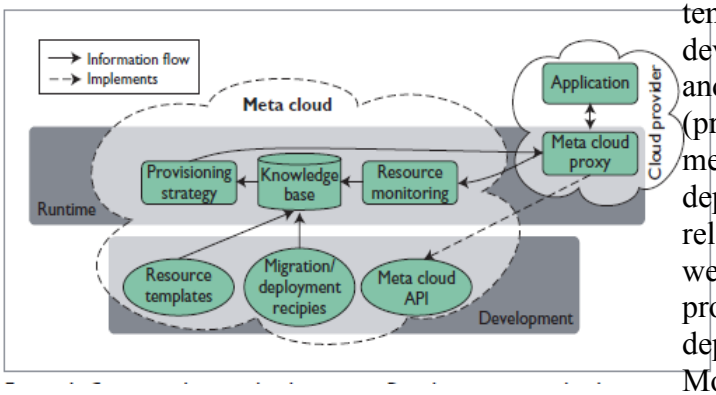
International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 07, July 2015
Available at http://internationaljournalofresearch.org

which is existing already and permits for complete self-service project formation and administration [10].

PROPOSED SYSTEM

To some extent, the Meta cloud can be realized based on a combination of existing tools and concepts, part of which are presented in the previous section. The main components of the Meta cloud, depicted in Fig., are described in the following and their interplay is illustrated using the previously introduced sports betting portal example. The components of the Meta cloud can be distinguished whether they are mainly important for cloud software engineers during development time or whether they perform tasks during runtime.

.



A.Meta Cloud API:

The Meta cloud API provides a unified programming interface to abstract from the differences among provider API implementations. For customers, the use of the Meta cloud API prevents their application from being hard-wired to a specific cloud service offering. The Meta cloud API can be built upon available cloud provider abstraction APIs, like lib cloud, fog, and jclouds, as previously mentioned. While these mostly deal with key-value stores and compute services, in principle all services can be covered that are abstract enough to be offered by more than one provider and whose specific APIs do not differ too much, conceptually.

B. Resource Templates

Developers describe the cloud services necessary to run an application using resource templates. They allow specifying service types with additional properties, and a graph model is used to express the interrelation and functional dependencies between services. The Meta cloud resource templates are created using a simple domain-specific language (DSL), allowing for the concise specification of required resources. The resource definitions is based on a hierarchical composition model, allowing for the creation of configurable and reusable template components, enabling developers and their teams to share and reuse common resource templates in different projects. Using the DSL, developers model their application components and their basic runtime requirements, such as (provider-independently normalized) CPU, memory, and I/O capacities, as well as dependencies and weighted communication relations between these components. The weighted component relations are used by the provisioning strategy to determine the optimal deployment configuration for the application. Moreover, resource templates allow for the definition of constraints based on costs, component proximity, and geographical distribution.

B.Migration/Deployment Recipes:

Deployment recipes are an important ingredient for automation in the Meta cloud infrastructure. The recipes allow for controlled deployment of the application including installation of packages, starting of required services, managing package and application parameters, and establishing links between

related components. Automation tools such as Chef provide an extensive set of functionalities, which are directly integrated into the Meta cloud environment. Migration recipes go one step further and describe how to migrate an application during runtime, e.g., migrating storage functionality from one service provider to another. Recipes only describe initial deployment and migration, the actual process is executed by the provisioning strategy and the Meta cloud proxy using aforementioned automation tools.

D. Meta Cloud Proxy:

The Meta cloud provides proxy objects, which are deployed with the application and run on the provisioned cloud resources. They serve as mediator between the application and the cloud provider. These proxies expose the Meta cloud API to the application, transform application requests into cloud provider specific requests, and forward them to the respective cloud services. The proxy provides means to execute deployment and migration recipes triggered by the Meta cloud's provisioning strategy. Moreover, proxy objects send QoS statistics to the resource monitoring component running within the Meta cloud. The data are gained by intercepting calls of the application to the underlying cloud services and measuring their processing time, or by executing short benchmark programs Applications can also define and monitor custom QoS metrics that are sent to the resource monitoring component via the proxy object to enable advanced, application-specific management strategies. To avoid high load and computational bottlenecks, the communication between proxies and the Meta cloud is kept at a minimum. Proxies do not run inside the Meta cloud, and regular service calls from the application to the proxy are not routed through the Meta cloud, either.

CONCLUSION

The meta cloud can help relieve merchant lock-in and guarantees transparent utilization of distributed computing administrations. The vast majority of the fundamental advances important to understand the meta cloud as of now exist, yet need incorporation. In this way, coordinating these state-of-the-craftsmanship devices guarantees a colossal jump around the meta cloud. Most of the basic technologies necessary to realize the Meta cloud already exist, yet lack integration. Thus, integrating these state-of-the-art tools promises a huge leap toward the Meta cloud. To avoid Meta cloud lock in, the community must drive the ideas and create a truly open Meta cloud with added value for all customers and broad support for different providers and implementation technologies.

REFERENCES

[1]Gartner survey feb 2013: from http://www.slideshare.net/GaldeMerkline/pwc-cloudenabledtelcoopportunitiespdf

[2] Peter Mell, and Tim Grance, Draft NIST Working Definition of Cloud Computing, 2009: from http://csrc.nist.gov/groups/SNS/cloud-computing/

[3] R. Buyya, C. S. Yeo, and S. Venugopal, Market oriented cloud computing: vision, hype, and reality, for delivering IT services as computing utilities, Proc. 10th IEEE International Conference on High Performance Computing and Communications,

Dalian, China, Sept 2008.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, Above the clouds: A berkeley view of

cloud computing, University of California, Berkeley, Tech Rep USB-EECS-2009-28, Feb 2009.

[5] Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R.,Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., and Rabkin, A. (2010). A View of Cloud Computing. Communications of the ACM, 53(4):50– 58.

[6] Chang, F., Viswanathan, R., and Wood, T. L. (2012). Placement in Clouds for Application-Level Latency Requirements. In 5th IEEE International Conference onCloud Computing (CLOUD), pages 327–335.

[7] Huebscher, M. C. and McCann, J. A. (2008). A survey of autonomic computing—degrees, models, and applications. ACM Computing Surveys, 40(3):7:1– 7:28.

[8] Inzinger, C., Satzger, B., Hummer, W., and Dustdar, S. (2012). Specification and deployment of distributed monitoring and adaptation infrastructures. In Workshop on Performance Assessment and Auditing in Service Computing (PAASC) held in conjunction with ICSOC.

[9] Jung, G., Hiltunen, M., Joshi, K., Schlichting, R., and Pu, C. (2010). Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures.In 30th IEEE International Conference on DistributedComputing Systems (ICDCS), pages 62–73.

[10] Kim, H., el Khamra, Y., Jha, S., and Parashar, M. (2010). Exploring application and infrastructure adaptation on hybrid grid-cloud infrastructure. In 19th ACM International Symposium on High Performance Distributed Computing (HPDC), pages 402–412.

[11] Satzger, B., Hummer, W., Inzinger, C., and Dustdar, S.(2013). Winds of Change: From Vendor Lock-In to the Meta Cloud. Internet Computing,17(1).

[12] Zhang, J. and Cheng, B. H. C. (2006). Model-based development of dynamically adaptive software. In 28th International Conference on Software engineering (ICSE), pages 371–380.

[13] M. Armbrust et al., "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, 2010, pp. 50–58.

[14] B.P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," Proc. Int'l Conf. Networked Computing and Advanced Information Management, IEEE CS Press, 2009, pp. 44–51.

[15]. J. Skene, D.D. Lamanna, and W. Emmerich, "Precise Service Level Agreements," Proc. 26th Int'l Conf. Software Eng. (ICSE 04), IEEE CS Press, 2004, pp. 179–188.