International Journal of Science Engineering and Advance Technology

Dynamic Data Security Assurance In Cloud Computing

Sk Davood #1, M Ganesh Babu #2
#1Student of M.Tech (CSE) and Department of Computer Science Engineering,
#2 Asst.Prof, Department of Computer Science and Engineering, Sir C R Reddy College of Engineering, Eluru, W.G.DIST

Abstract:
Dynamic data security assurance very difficult task in cloud worlds. In general we are using meta clouds for secure communication with secrecy of data management on clouds. But clouds are having heavy difficulties with on ensured secure data managements. In the clouds we did not maintain priviligation on clouds. In resent clouds all the people view all users information. In the clouds we are using tpa for auditability of data. But these tpas also eat the data of data owners without privileges. And also cloud servers are worked like very flexible manner these type of servers also malicious the data user and owners of information. So we are introducing Dynamic data security assurance on clouds. These are very accountable and flexible. Clouds maintained with respective 27% secrecy maintained and 42% flexible on meta clouds .

Key words- meta cloud, gusts, data lock in.

I.Introduction

At the level of all these problems, we can identify a need for businesses remotly monitor the cloud they're using and be able to rapidly changes on auditability. However, migration is currently far from trivial. Myriad cloud providers are flooding the market with a clouds. So many years onwards so many persons are worked with clouds. But the cloud computing is including un support for runtime migration .So we are need the migration of clouds and providing better serviceability. several approaches try to tackle at least parts of the problem.

Cloud Computing usecase

Let's view a Web-based sports issue for an moment such as the distributed games, which allows users to place bets. An event this large requires an enormously efficient and reliable infrastructure, and the cloud computing technology provides the flexibility and elasticity for such an application. It starts 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. 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 bet 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. Instead of being bound to one cloud operator, however, the betting application should be hosted in an optimal cloud environment.

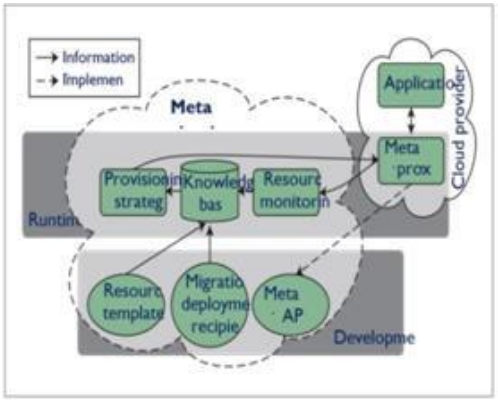
To leverage a more diverse cloud landscape, support flexibility, and avoid vendor lockin, the meta cloud must achieve two main goals:

•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, then run it anywhere, including support for runtime migration.

Lately, the meta cloud idea has received some attention, and several approaches try to tackle at least parts of the problem.

Cloud Api

The meta cloud API provides a unified programming interface to abstract from the differences among provider API implementations. For customers, using this API specific cloud service offering.



The meta cloud API can build on available cloud provider abstraction APIs, as previously mentioned. Although these deal mostly with keyvalue stores and compute services, in principle, all services can be

covered that are abstract enough for more than one provider to offer and whose specific APIs don't differ too much, conceptually.resource templates in different projects. Using the DSL, developers model their application components and their basic runtime requirements, such as (providerindependently normalized) CPU, memory, and I/O capacities, as well as dependencies and weighted communication relations between these components. The provisioning strategy uses the weighted component relations to determine the application's optimal

Resource Monitoring:

On an application's request, the resource monitoring component receives data collected by meta cloud proxies about the resources they're using. The component filters and processes these data and then stores them on the knowledge base for further processing. This helps generate comprehensive QoS information about cloud customer has an account with and providers that offer possibilities for creating (sub)accounts on the fly. Several information sources contribute to the knowledge base: meta cloud proxies regularly send data about application behavior and cloud service QoS.

Cloud-centric migration makes the meta cloud infrastructure responsible for most migration aspects, leading to issues with application specific intricacies, whereas in application-centric migration, the meta cloud only triggers the migration process, leaving its execution mostly to the application. We argue that the meta cloud should control the migration process but offer many interception points for applications to influence the process at all stages

Conclusion:

The meta cloud can help mitigate vendor lock-in and promises transparent use of cloud computing services. Most of the basic technologies necessary to realize the meta cloud already exist, yet lack integration. Thus, integrating these stateofthe-art tools promises a huge leap toward the meta cloud. To avoid meta cloud lockin, 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] S. Abdelwahab, B. Hamdaoui, M. Guizani, and A. Rayes. Enabling smart cloud services through remote sensing: An internet of everything enabler. Internet of Things Journal, IEEE, 1(3):276–288, June 2014.

[2] M. T. Beck, A. Fischer, H. de Meer, J. F. Botero, and X. Hesselbach. A distributed, parallel, and generic virtual network embedding framework. In Communications (ICC), 2013 IEEE International Conference on, pages 3471–3475. IEEE, 2013.

[3] A. Belbekkouche, M. Hasan, A. Karmouch, et al. Resource discovery and allocation in network virtualization. Communications Surveys & Tutorials, IEEE, 14(4):1114–1128, 2012.

[4] K. Birman. The promise, and limitations, of gossip protocols. ACM SIGOPS Operating Systems Review, 41(5):8–13, 2007.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. Information Theory, IEEE Transactions on, 52(6):2508–2530, 2006.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. 2011.

[7] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang. Virtual network embedding through topology-aware node ranking. ACM SIGCOMM Computer Communication Review, 41(2):38–47, 2011.

[8] R. Duan and S. Pettie. Linear-time approximation for maximum weight matching. Journal of the ACM (JACM), 61(1):1, 2014.

[9] S. Ferretti. Gossiping for resource discovering: An analysis based on complex network theory. Future Generation Computer Systems, 29(6):1631–1644, 2013.

[10] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach. Virtual network embedding: A survey. Communications Surveys & Tutorials, IEEE, 15(4):1888–1906, 2013.

[11] F. Garin and L. Schenato. A survey on distributed estimation and control applications using linear consensus algorithms. In Networked Control Systems, pages 75– 107. Springer, 2010.

[12] I. Houidi, W. Louati, and D. Zeghlache. A distributed virtual network mapping algorithm. In Communications, 2008. ICC'08. IEEE International Conference on, pages 5634–5640. IEEE, 2008.