Retrieving, Storing, Correlating and Distributing Information for Cloud Management

Spyridon V. Gogouvitis[1], Gregory Katsaros[1], Dimosthenis Kyriazis[1], Athanasios Voulodimos[1], Roman Talyansky[2], and Theodora Varvarigou[1]

[1] Dept. of Electrical and Computer Engineering National Technical University of Athens, Athens, Greece
{spyrosg,gregkats,dimos,thanosv}@mail.ntua.gr, dora@telecom.ntua.gr
[2] SAP Research Israel, Ra'anana, Israel
roman.talyansky@sap.com

Abstract. The emergence of Cloud technologies ultimately affected the service computing ecosystem introducing new roles and relationships as well as new architectural and business models. The increase of the capabilities and potentials of the service providers raised the need for managing the information being available in an efficient way. In this paper we introduce a service management architecture as well as the corresponding information model, which is the placeholder for the information needed to perform management operations. The design of the proposed model was based on the case of a storage service being provided through a cloud infrastructure, but the approach is implemented in a flexible and modular fashion in order to support any service provided through a cloud environment.

Keywords: Cloud management models, information services, storage clouds.

1 Introduction

Cloud computing as a whole is rapidly evolving and becoming one of the most challenging paradigms of Information Technology. Its usefulness for users and enterprises in general is clearly recognized [1], mainly due to the varying business models it can facilitate. Indeed, their reliance on this paradigm is reaching unexpected levels. To this end, more and more cloud providers are contributing towards a quite young, but relatively broad cloud ecosystem. On the other hand, this massive availability of resources and services resulted in an increase in the information generation that the current data models and representations cannot always capture.

Notwithstanding, the rapid evolution of the cloud, along with the new and emerging needs of customers, pose additional challenges with respect to the management of such environments, given the involvement of various entities, such as stakeholders at different levels (i.e. users, providers, brokers, etc) with, in many

K. Vanmechelen, J. Altmann, and O.F. Rana (Eds.): GECON 2012, LNCS 7714, pp. 114–124, 2012.

cases, disparate interests, documents capturing information such as Service Level Agreements (SLAs) and application descriptors, virtualized resources, to name a few. Thus, providers of cloud service models, e.g. Platform as a Service (PaaS) and Infrastructure as a Service (IaaS), deal with huge amounts of information that needs to be collected, managed and evaluated. For this reason, consistent cloud-enabled data models representing the aforementioned multiple entities and their interrelationships are required.

Furthermore, there is a spreading need for efficiently collecting, gathering and storing monitoring information from the underlying cloud infrastructure. This information can range from application-related metrics, such as Web based services response time or HPC jobs completion deadline, to infrastructure-related metrics such as power consumption or resource capacity and utilization. In any case, all this information should be assessed in order to provide Business Level Parameters (BLPs). Thereafter, these pieces of data can be considered in a synergistic way in decision making processes.

In this paper, an information management system and its corresponding data model that considers all the necessary information flows within a cloud environment are presented. Our analysis was based on a cloud storage service but could be extended for any service offered through clouds, since the proposed model is directly extensible by adding structures to describe any kind of cloud service offering.

The remainder of this work is organized as follows: in section 2 we present the major initiatives in the field of information and cloud management, in section 3 we introduce a generic architectural concept that deals with the information management in cloud platforms in a holistic way. In section 4, we elaborate on each of the data models that we propose while in section 5 the mechanisms for retrieving and storing the information are presented. Finally, in section 6 we summarize and discuss our future steps.

2 Related Work

One of the biggest challenges in cloud computing is that there is no single standard or architectural method. The most common and widely adopted cloud model is based on the distinctive layers of Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Each cloud stack raises needs for information management and therefore several initiatives and solutions exist. To this end, there has been a lot of effort spent to the management of resources (IaaS). All the main providers (such as EC2[2], Rackspace[3], Azure [4], NewServers [5]) and technologies (such as OpenNebula [6] OCA, OCCI [7], etc.) offer APIs for the administration and control of the cloud infrastructure. In [8] an application centric management framework is presented that tries to integrate high level requirements throughout the application development process in the cloud. On the other hand, in [9] a hierarchical architecture model of cloud services resources management is presented. The architecture model achieves isolation between users and physical resources, however, from actual conditions, the architecture model is quite complex since a number of preparatory works need to be done, resulting in a certain degree of difficulty. In [10], a multi-cloud management platform that locates between cloud users and cloud sites is proposed. It brings valuable contribution for cloud federation and migration of services but it does not propose an information description scheme. Another interesting initiative is described in [11], where a self-organized cloud management model is presented. This concept is highly aligned with our work but it lacks in details regarding the information schemata as well as certain information flows that reside in the PaaS layer. Finally, an additional IaaS management solution is presented in [12], which uses abstractions for managing resources in a hybrid cloud environment for enterprise users.

In total, we have seen in the literature that most of the cloud management solutions are dealing mainly with the resource allocation or a combination of user requirements towards resource management. There is no holistic approach or modeling of the whole cloud platform lifecycle but individual abstractions of parts of it. In [13] we presented the baseline of the Unified Management model design. To this end, with our work in this paper, we are capturing all the information flows and define models that will manage those datasets in any cloud service situation. In addition, we identify the components responsible for each information flow and its relation with the corresponding model.

3 Management Architecture

The topic of management in cloud environments has several angles to be approached. As presented in the related work, there is the aspect of resource management that mainly lies on the IaaS cloud layer and interacts with the infrastructure middleware or hypervisor. On the other hand, the realization of the SaaS stack demands administration and management in terms of applications and high level requirements. The solution that we present in this section proposes a generic management architecture that is placed on the PaaS cloud layer covering the initial management steps of requirement specification and SLA management, to usage modelling and resource management.

The main focus of this paper is not to implement a management framework, but to identify the basic information flows within a cloud environment and propose a unified model that would be able to capture that information consistently. Therefore, in Figure 1 we present a generic architecture of the management layer.

The introduced generic service management architecture describes the following information flows:

- Initiation and SLA management: throughout this process a user is requesting a service and defines the high-level application requirements and QoS parameters described in an SLA.
- Service instantiation and resource allocation: The second conceptual procedure is the creation of the actual instance for the requested service. The internal analysis will map the high level requirements captured in the signed SLA to low level resource specification that will result in the resource allocation.

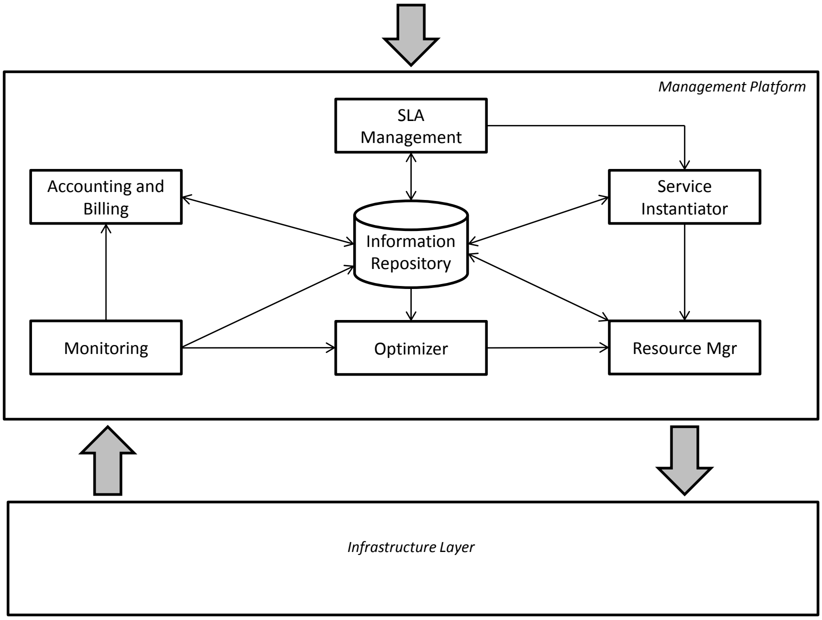

Fig. 1. Generic Management Architecture

– Feedback and optimization: An important feature of every cloud platform is the collection of information from the infrastructure layer and the utilization of that data for Billing & Accounting, SLA violation detection and optimization of the resource allocation with the aim to decrease Total Cost of Ownership (TCO).

4 Unified Management Model

Based on the previous generic architectural approach and the identified information flows, in this section we present several information models that capture the necessary data and create associations between all involved entities. All these models contribute to the overall proposed unified management model. The solution presented is focusing in a storage cloud situation but with minor modifications or additions into some of the defined models could facilitate the information representation of every cloud scenario. We should also note that apart from the basic storage service offered in this cloud, we have incorporated the offering named "storlet" which represents a computation service over a storage entity [14].

4.1 Requirements Model

In order for an application to run effectively over a cloud infrastructure the customer should be able to specify requirements, which will be used by the infrastructure to drive the data access operations of the application. To this end, a Requirements Model capturing the requirements emerging from application attributes modelling and the ones deriving directly from the user needs is necessary. In addition, the model defines structures to describe lower level requirements for the service offerings of the cloud as well as Resource requirements that are used for the resource provisioning.

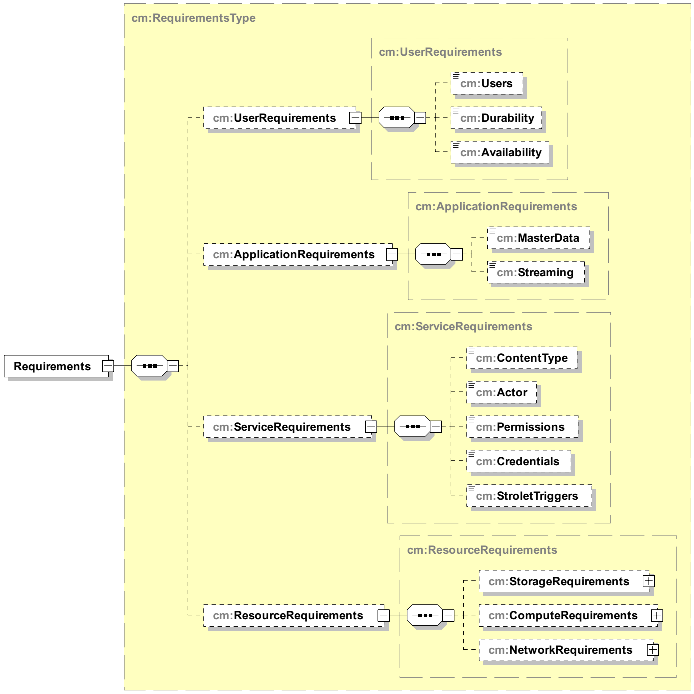

Fig. 2. Requirements Model

As presented in Figure 2 the proposed model covers the following entities:

- User requirements: this entity holds information that will be included in the SLA schema and are provided by the application user. Examples of parameters described are metrics like number of users, durability and availability requirements.
- Application requirements: those requirements are high-level characteristics that will be incorporated in the SLA and will be translated into low level, Storage (or Computational Storage) and Resource requirements. Those parameters will characterize the application to be deployed (Master or Transactional data to be stored, streaming etc.).
- Service requirements: those are the specific requirements that determine a cloud storage service that we have defined in our cloud scenario. This structured and formalized description will include information such as content

type information, data access permissions and other details that may affect the service deployment and operation. The requirements of a Computational Storage service (storlet) are also defined here, including parameters such as CPU speed and memory. This entity will be later incorporated in the Service Model.

– Resource requirements: this entity aims at specifying the resource requirements for the operation of a cloud service (storage as well as storlet). This structure will be utilized during resource provisioning and will keep the desired resources for meeting the constraints described in an SLA.

The transformation of the high level requirements, that are input for the SLA negotiation process, to the low level service and resource requirements is succeeded by the cloud platform throughout the mapping and resource provisioning operation. For example, attributes such as "Master Data" or "Transactional Data" are translated into requirements "low update rates, replicated with eventual consistency" and "high update rates, strong consistency" respectively.

4.2 SLA Model

This model contains the information related to the SLA, that is, the service level agreement between the customer and the service provider. It is separated into two groups:

– the context: which contains non-technical information of the agreement such as participants and dates and
– the terms: which contain the requirements, the conditions and the billing policies.

The requirements that are stated in the SLA utilize the requirement model defined earlier. The customer is able to provide high level requirements (user and application) that are specified for the usage of a specific service offering. Those requirements are being transformed to low level specifications by the platform's services. Moreover, the SLA is enhanced with terms that specify the actual content of the objects to be stored, therefore enabling the support of content-centric access to the cloud. More information can be found in [15].

4.3 Services Model

In this model we capture the information regarding the cloud offerings. The structure keeps general information (ID, name, status details) but also associates the service with a negotiated SLA. In addition, the model incorporates the Service Requirement model and the Resource Model for specifying the respective information. It is also worth noting that computational storage details are also incorporated in this model.

The role of the Service Model is three-fold: (1) to capture all the necessary details of an active service in the cloud, (2) associate each service with a signed

SLA instance and (3) to keep the technical requirements in terms of resources as well as service parameters needed for the instantiation. Through this model the cloud service orchestration is achieved and the component responsible for realizing that information structure is the Service Instantiator in the PaaS layer. Moreover, in case we want to apply our unified management model to another cloud situation, we need just to define an additional Service model according to the requirements of the new service offering.

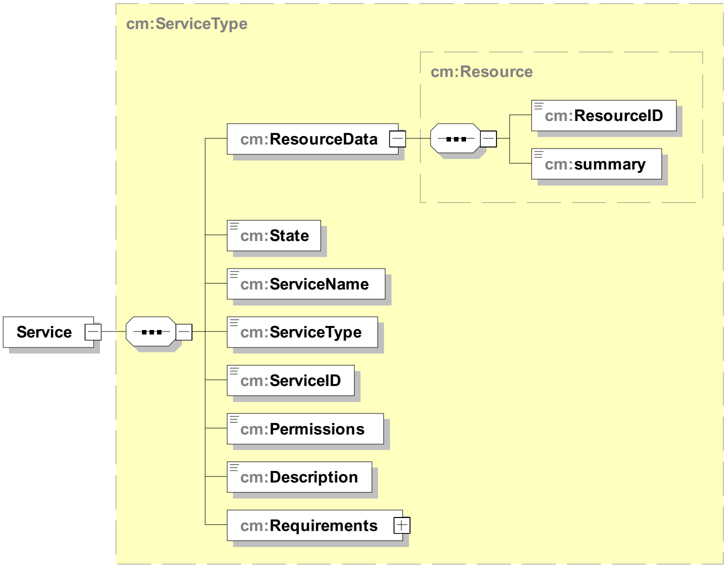

Fig. 3. Services Model

4.4 Resource Model

In every cloud environment the management of the resources is of major importance. Thus, our management model could not be missing this information entity. The model that we have defined has a hierarchical structure in order to facilitate cloud federation and in general a distributed cloud infrastructure management. Therefore, we propose three entities:

– Node: this entity describes the specification of a single node of the resources of a provider. Apart from the static information for each node, we associate each node with the cloud services deployed on this very resource ("service ID list").
– Cluster: the cluster resource model is the parent element of the previous, node entity. It defines characteristics of the whole cluster as well as lists the respective nodes.

– Data Center: hierarchically this structure is the parent element of the cluster entity. Each data center could include one or more clusters. This relationship is defined by the "Clusters" field of the structure. Apart from that, the Data Center model will also specify information such as location, capacity, energy consumption etc.

The existence of a resource model in the PaaS layer of a cloud environment serves the need of management in terms of resource allocation, services deployment and execution and finally optimization. The feedback collected from the Monitoring and processed by the Analyzer will be ingested in the Resource Manager which optimizes the cloud Service Situation (this could be a deployment of a storage service, migration of an active service offering, enabling an elasticity rule etc.) based on the resource model at the given time.

4.5 Usage Model

Usage characteristics (e.g. geographical access distribution or read / write access frequency) encode the knowledge on the typical application behavior. Usage models will be automatically harvested. They capture application access patterns with regard to storage. These characteristics are of major importance since they may be used afterwards for optimizing the mapping of cloud storage resources to application requirements. For example data and storlet placement may be optimized for both the application performance and low operational costs of cloud providers.

In addition to automatic harvesting of usage characteristics, applications may inject into the cloud storage infrastructure a priori knowledge on the application behavior with regard to storage. Self knowledge may be used to further optimize mapping of application requirements to cloud storage resources. Therefore, usage models will also include representations and descriptions to inject application self-knowledge. With regard to multi-tenant applications, the usage characteristics and the information injected by the application as self-knowledge may be encoded in a tenant template that will be used in the subsequent provisioning of data services for new tenants. This enables an optimized management of the new tenant's data and storlets right from the beginning. Figure 4 is an example representation of an application's behavioral characteristics (Usage Model) in the level of cluster that is deployed, cloud that the cluster is hosted and tenant that is authorized to consume that application.

5 Retrieving and Storing Information

The aforementioned models need to populated with real information in order to be used by the various consumers. To this end, within the context of the VISION cloud project, a novel monitoring and aggregation mechanism is being developed that is capable of retrieving the needed information.

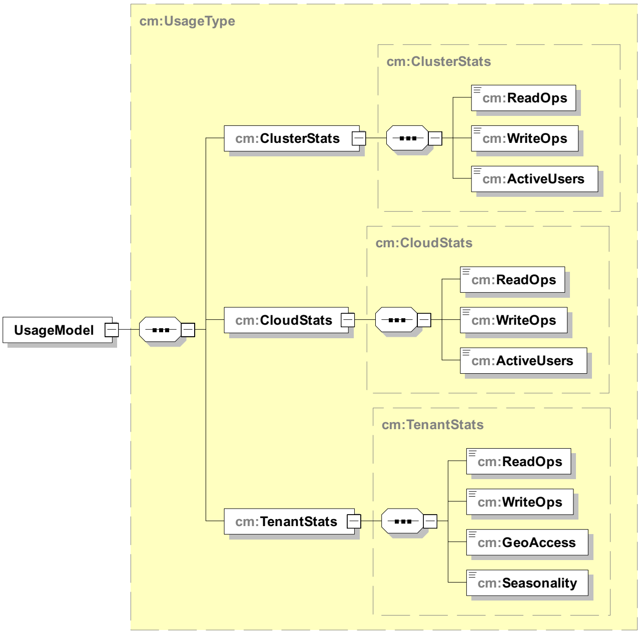

Fig. 4. Usage Model

5.1 Monitoring Architecture

A monitoring instance that runs on every node of the cloud is responsible for collecting local monitoring information and events. Consequently this information is aggregated and propagated to the interested recipients. from the node it resides before processing and storing it, in order for all the information to be available to the decision making modules of the platform. The communication is achieved through the use of 0MQ [16].

Hardware metrics are collected through probes that are configured to pass the relevant information to the local representative of the monitoring subsystem. The modularity of the implementation also allows for other metering systems to be deployed and to be used as sources of information, as long as the specified event format is followed. Services residing on the same host can also utilize the interfaces the monitoring system exposes to propagate events to it using the same format.

5.2 Storing of Information

The information collected need to be stored efficiently and be available to all management modules that require an up to date view of the system. Due to the distributed nature of the cloud, as well as its size, which contains multiple datacenters and clusters, geographically distributed, a NoSQL distributed database is used. Due to the CAP theorem [17] we have chosen to provide availability and eventual consistency. The aforementioned models are transformed into the appropriate structures and the monitoring mechanism is responsible for populating these during run-time. Any module needing to use the information stored is able to query the database and retrieve the needed information.

6 Conclusions and Future Steps

Information management of resources and services in cloud is a challenging task mainly due to the lack of standards and the flexibility of the infrastructure. In this paper we presented a unified management model designed for a storage cloud situation, extendable though for any kind of cloud service offerings. By analyzing the information flows of the cloud service framework we captured the requirements for the management and therefore we defined the core models (requirements, SLAs, services, resources and usage model). In addition, we identified the responsible components for the realization of the models within the service framework of the cloud architecture. The efficient management of the cloud platform is based on the consistency of the models and the ability to associate information of one model to another. In that context, the whole status of the cloud platform can be captured effectively as a single unified model and not separate information structures. That capability allows us to store "snapshots" of the system and in order to migrate it or re-enable it in the future. The future step regarding this work is the investigation of ontologies specifications and the possibility of transforming this unified model into a cloud ontology which could form the management core of a cloud environment. We are also working on validating the model against several cloud scenarios such us cloud federation and hybrid cloud. Finally, we enforce the work on this topic by participating in the Common Cloud Ontologies Working Group [18] where we promote and discuss our design with the research community.

Acknowledgements. This work has been supported by the VISION Cloud project [19] and has been partly funded by the European Commission's ICT activity of the 7th Framework Program under contract number 257019. This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work.

References

1. Buyya, R.: Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility. In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009 (May 2009)
2. Garfinkel, S.L.: An evaluation of amazons grid computing services: Ec2, s3 and sqs. Technical report, Center for (2007)
3. Rackspace Cloud (2009), http://www.rackspacecloud.com

4. Microsoft Azure (2010), http://www.microsoft.com/windowsazure
5. NewServers Inc. (2008), http://www.newservers.com
6. Milojic, D., Llorente, I.M., Montero, R.S.: Opennebula: A cloud management tool. IEEE Internet Computing 15(2), 11–14 (2011)
7. OCCI Working Group (2009), http://occi-wg.org
8. Harmer, T., Wright, P., Cunningham, C., Hawkins, J., Perrott, R.: An application-centric model for cloud management. In: 2010 6th World Congress on Services (SERVICES-1), pp. 439–446 (July 2010)
9. Sun, Y., Xiao, Z., Bao, D., Zhao, J.: An architecture model of management and monitoring on cloud services resources. In: 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 3, V3-207–V3-211 (August 2010)
10. Liu, T., Katsuno, Y., Sun, K., Li, Y., Kushida, T., Chen, Y., Itakura, M.: Multi cloud management for unified cloud services across cloud sites. In: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), pp. 164–169 (September 2011)
11. Liu, Z., Tong, W., Gong, Z., Liu, J., Hu, Y., Guo, S.: Cloud computing model without resource management center. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 442–446 (2011)
12. Yan, S., Lee, B.S., Zhao, G., Ma, D., Mohamed, P.: Infrastructure management of hybrid cloud for enterprise users. In: 2011 5th International DMTF Academic Alliance Workshop on Systems and Virtualization Management (SVM), pp. 1–6 (October 2011)
13. Voulodimos, A., Gogouvitis, S., Mavrogeorgi, N., Talyansky, R., Kyriazis, D., Koutsoutos, S., Alexandrou, V., Kolodner, E., Brand, P., Varvarigou, T.: A unified management model for data intensive storage clouds. In: 2011 First International Symposium on Network Cloud Computing and Applications (NCCA), pp. 69–72 (November 2011)
14. Kolodner, E., Tal, S., Kyriazis, D., Naor, D., Allalouf, M., Bonelli, L., Brand, P., Eckert, A., Elmroth, E., Gogouvitis, S., Harnik, D., Hernandez, F., Jaeger, M., Lakew, E., Lopez, J., Lorenz, M., Messina, A., Shulman-Peleg, A., Talyansky, R., Voulodimos, A., Wolfsthal, Y.: A cloud environment for data-intensive storage services. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), November 29-December 1, pp. 357–366 (2011)
15. Mavrogeorgi, N., Gogouvitis, S., Voulodimos, A., Katsaros, G., Koutsoutos, S., Kiriazis, D., Varvarigou, T., Kolodner, E.K.: Content based slas in cloud computing environments. In: 2012 IEEE Fifth International Conference on Cloud Computing, pp. 977–978 (2012)
16. Zeromq, http://www.zeromq.org/
17. Brewer, E.A.: Towards robust distributed systems (abstract). In: Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing, PODC 2000, p. 7. ACM, New York (2000)
18. Common Cloud Ontologies WG, http://www.holaportal.eu/content/abdo-ymo-neo-ut-os-facilisis-abbas
19. VISION Cloud, http://www.visioncloud.eu/