OPTIMIS and VISION Cloud: How to Manage Data in Clouds

Spyridon V. Gogouvitis[1], George Kousiouris[1], George Vafiadis[1], Elliot K. Kolodner[2], and Dimosthenis Kyriazis[1]

[1] National Technical University of Athens, Iroon Polytechniou 9, Athens, Greece
[2] IBM Haifa Research Labs, Haifa University, Mt. Carmel, Haifa, Israel
{spyrosg,gkousiou}@mail.ntua.gr, gvaf@iccs.gr, kolodner@il.ibm.com, dimos@mail.ntua.gr

Abstract. In the rapidly evolving Cloud market, the amount of data being generated is growing continuously and as a consequence storage as a service plays an increasingly important role. In this paper, we describe and compare two new approaches, deriving from the EU funded FP7 projects OPTIMIS and VISION Cloud respectively, to filling existing gaps in Cloud storage offerings. We portray the key value-add characteristics of their designs that improve the state of the art for Cloud computing towards providing more advanced features for Cloud-based storage services.

Keywords: Cloud computing, Storage, Data Management.

1 Introduction

An important aspect of the Future Internet is the ever growing amount of data that is generated and stored in the Cloud. Be it user-generated data, such as multimedia content uploaded to social networking environments, or data coming from companies moving to the Cloud to take advantage of the cost savings it has to offer, generated data is growing faster than we can store it.

Storage Cloud solutions, by which storage is virtualized and offered on demand, promise to address the proliferation of data. Key concepts of Cloud Computing, such as the pay-as-you-go model, essentially unlimited scalability and capacity, lower costs and ease of use and management, are also prominent in Storage Clouds.

Nevertheless, issues such as Quality of Service assurances, mobility, interoperability and federation, security and compliance, energy efficiency and others still need to be addressed to enable the provision of data-intensive storage Cloud services. Furthermore, increasing legal requirements with regard to data storage locations is preventing storage providers from fully utilizing their infrastructures according to internal optimization capabilities, e.g., load balancing techniques. In this paper we describe two different approaches to achieving better management of data in Cloud environments, as realized by the OPTIMIS [1] and VISION Cloud [2] EU Projects.

M. Alexander et al. (Eds.): Euro-Par 2011 Workshops, Part I, LNCS 7155, pp. 35–44, 2012.

The remainder of this paper is structured as follows: Section 2 surveys current Cloud Storage offerings while Section 3 presents the physical model that is considered throughout the paper. Section 4 discusses the OPTIMIS approach to data management in Clouds and in Section 5 presents the VISION Cloud project. In Section 6 compares the approaches of the two projects. Finally, Section 6 concludes the paper.

2 Related Work

There are various approaches with regard to Cloud infrastructures that aim at providing storage services. Possibly the most well-known Cloud storage service today is the Amazon Simple Storage Service (S3) [3]. Amazon has not made public details of S3's design, though stating that it is "intentionally built with a minimal feature set". S3 allows writing, reading and deleting an unlimited amount of objects that are stored in buckets. It provides an SLA that guarantees service availability of 99.9%. S3 replicates objects in multiple locations to achieve a durability of 99.999999999%, although the service is not held accountable for any data loss. Amazon S3 buckets in most regions (US West, EU and Asia Pacific) provide read-after-write consistency for PUTS of new objects and eventual consistency for overwrite PUTS and DELETES. Buckets in the US Standard Region provide eventual consistency.

Another commercial solution is Windows Azure [4], which is mainly a PaaS offering that provides four different storage services, namely the Binary Large Object (BLOB) Service for storing text and binary data, the Table service, for structured storage that can be queried, the Queue service for persistent messaging between services and the Windows Azure Drive that allows Windows Azure applications to mount a Page Blob, which is single volume NTFS VHD. All data is stored in 3 replicas. The storage services may be accessed from within a service running in Windows Azure or directly over the Internet using a REST API. The BLOB service offers the following three resources: the storage account, containers, and blobs. Within one's storage account, containers provide a way to organize sets of blobs, which can be either block blobs optimized for streaming or page blobs optimized for random read/write operations and which provide the ability to write to a range of bytes in a blob.

EMC Atmos [5] is another Cloud storage solution with an API similar to Amazon S3. Some of the key differences include objects that are re-writable and user metadata that can be updated, richer geo-dispersion capabilities, two data models, namely a flat object interface and a namespace interface that similar to a file system with folders, and a richer account model that is more suitable for enterprise customers.

Google Storage for Developers [6] is a RESTful online storage web service using Google's infrastructure. It provides strong read-after-write data consistency and eventual consistency on list operations. It uses the notions of buckets, and the user is able to specify the geographic location of each bucket.

Nevertheless, the approaches described in this section do not meet the new challenges of data-intensive services. Starting from the data models, they are basic. Amazon S3, Google Storage, and the Windows Azure Blob Service allow associating user metadata in the form of key value pairs with objects and blobs, but they simply

store the metadata and pass it back. EMC Atmos has a slightly richer model; it allows some of keys (called tags by Atmos) to be listable; this enables retrieving the objects that have a specific tag. The support for federation does not exist or is limited and requires homogeneity. Amazon S3, Google Storage and the Windows Azure Blob Service do not have any support for federation. EMC Atmos allows federating data in an Atmos system in a customer data center with the customer's data in a Cloud, provided it is also implemented with Atmos. No current Cloud storage offering provides computational abilities as an integral part of the Cloud storage system to the best of our knowledge. Access to an object is solely through its name with Amazon S3, Google Storage and the Windows Azure Blob Service. As mentioned above, EMCS Atmos has a slight richer access capability through its listable tags. But no current Cloud storage system has a rich flexible access to storage based on its content and relationships. Finally, the QoS mechanisms and SLAs provided by current offerings are very basic. In our approach, models, requirements and SLA schemas are expressed not only on storage resources and services, but also on the content descriptions for the underlying storage objects, in support of content centric storage.

3 Infrastructure Model

The physical model of the infrastructure is a network of data centers that can span over a large geographic area, connected by a dedicated network. Each data center is composed of one or multiple *storage clusters* containing physical compute, storage and networking resources.

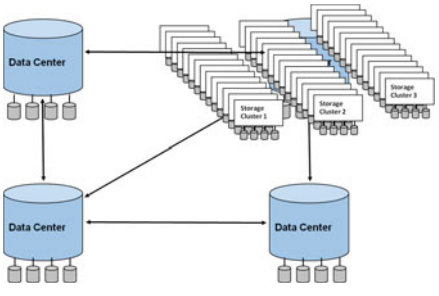

Fig. 1. Physical Model of a Storage Cloud consisting of multiple interconnected Data Centers

A *storage cluster* is composed of storage rich nodes constructed from commodity hardware and connected by commodity interconnect. The idea is to build the storage Cloud from low cost components, ensuring reliability in the software, and building advanced functionality on top of this foundation. For example, given today's hardware, the initial hardware configuration for the nodes could be 4 or 8 way multiprocessors (taking multicore into account) with 12 to 16 GB of RAM. Each node could have 12 to 24 high capacity direct attached disks (e.g., 2TB SATA drives). The architecture, design and implementation should support a system with hundreds of storage clusters, where each storage cluster can have several hundred nodes and the storage clusters are spread out over dozens of data centers.

4 The OPTIMIS Approach

The main goal of the OPTIMIS project is to enable a Cloud toolkit that will be able to accommodate management of a Cloud environment in configurable terms with regard to various factors such as Trust, Risk, Ecology and Cost (TREC). This is done for cases where multiple locations are envisioned to contain data centers, either belonging to one entity or different ones, through various deployment scenarios like federated, hybrid and multi-cloud cases. A more detailed analysis of the project goals and architecture appears in [8].

The OPTIMIS Data Management architecture is portrayed in Fig. 2. It is based on the distributed file system architecture (HDFS [7]) offered by the Hadoop framework. This consists of a central NameNode, which acts like the inode of the system, and a series of DataNodes, which act as the actual storage and processing nodes of the Hadoop cluster. Each file is divided into a number of blocks (with configurable size per file) and distributed over the DataNodes. Different blocks of the same file may belong to different DataNodes. This creates a very flexible framework for managing files, in order to optimize their processing and management actions (e.g. cluster balancing). Suitable RESTful interfaces have been added on top of the HDFS NameNode, in order to offer its functionality as a service. Furthermore, due to the multi-tenant environment, security features have been added in order to encrypt communication between the NameNode and the DataNodes, between the DataNodes themselves but also between the service VMs that utilize the account on the HDFS and the HDFS components. In cases when the storage space is running out, one or more DataNode VMs can be added from an external location. This location may be the same provider, or a different Cloud, through utilizing their available APIs for launching a VM instance of the OPTIMIS version of the HDFS DataNode. As soon as these VMs are launched, they are registered in the distributed file system that runs in the internal IP and can be managed like any other internal VM resource. The HDFS utilizes the storage space of these VMs as part of the file system offered to the services.

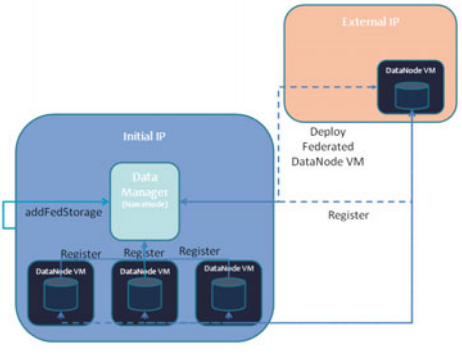

Fig. 2. OPTIMIS Data Manager Architecture

In order to offer data management or storage services as a Cloud Provider, the OPTIMIS solution encapsulates the HDFS and offers account creation for the end users of the OPTIMIS platform. Thus, each user that raises his/her Virtual Machines on the OPTIMIS Cloud is granted access on the HDFS through his/her own personal account. This appears as a directory. Through this form, the service data may be managed through the distributed file system in an efficient way, enabling parallel processing and access from numerous service VMs.

Another major aim of the OPTIMIS data management solution is to be able to capture and apply user requirements with regard to data location constraints. Extensive analysis on the legal requirements has indicated a series of actions that need to be undertaken by the Cloud provider in order to be compliant with the according legislation ([9]). For this reason, rack awareness features of HDFS along with block location information are utilized for monitoring purposes of the data location. An extension is also in progress in order to regulate the block placement policies according to each service's needs. This includes capabilities for processing the SLA in which these needs are expressed and act upon them, making sure that no blocks of data of a specific service are located on servers outside the user's identified geographical region.

Taking under consideration that OPTIMIS scenarios include multiple locations of data centers, that cooperate in distributing the load, what is necessary for a single location is to identify the fittest services that can be transferred to external data centers from a data activity point of view. Due to the fact that in general this load balancing may be performed on the fly and for a specific period of time (to meet for example a peak in demand), the optimal service VMs must be chosen. This is why the OPTIMIS data management module provides interfaces for ranking existing services with regard to their data activity. Through logging mechanisms added internally to the interfaces that are used by the service VM to access the distributed file system, each user action on the data is logged. These extensive logs are temporarily stored for a specific interval. After this interval they are preprocessed as a Map-Reduce task, in order to be transformed in a format suitable for incorporation in the time series prediction framework (Federation Candidate Selector-FCS) but also to reduce the storage space needed for the logs. The FCS component is responsible for processing the concentrated statistical results and create a model that is able to predict the expected user activity with regard to the HDFS (e.g. expected number of bytes read/written) for the forthcoming time period. Thus for example, it can recommend to the Cloud provider to choose a service VM to migrate that has the least activity in the near future. This estimation can also be used for other purposes, such as regulating the replication factor of a service's data on the HDFS. If a peak in read operations is expected for example, by having more replicas of a data block on the distributed data nodes the requestors of these operations will have more sources to choose from and thus balance the load.

Furthermore, in order to have multiple policies with regard to the different TREC factors, the OPTIMIS Data Manager exposes suitable interfaces that enable the Cloud provider to set predefined levels of specific metrics, like eco-efficiency. Each such level corresponds to a set of actions/policies in order to control for example the energy consumption of the infrastructure for which the DM is responsible for. For example, the number of DataNodes can be regulated through these interfaces.

High eco-efficiency may lead to decommissioning of a number of nodes that are used for storage in order to reduce carbon emissions. On the other hand, when performance is needed, extra nodes may be added on the fly in order to improve the system response times.

5 The VISION Cloud Approach

The main aim of VISION Cloud is to provide efficient support for data-intensive applications by taking a content-centric view of storage services. To this end five areas of innovation have been identified and are driving the VISION Cloud platform: i) content is managed through data objects associated with a rich metadata model, ii) avoidance of data lock-in by enabling the migration of data across administrative domains, iii) avoiding costly data transfers by moving computations close to the data through programming agents called *storlets*, iv) enabling efficient retrieval of objects based on their content, properties and the relationships among them and v) providing strong QoS guarantees, security and compliance with international regulations. More details on the aims and architecture can be found in [10] and [11]. A key element to achieve these aims is a data model with rich metadata, which is described in the following paragraphs.

5.1 Data and Account Model

In VISION Cloud the unit of storage is a *data object*. A data object contains data of arbitrary type and size, has a unique identifier that can be used to access it and has *metadata* associated with it. Each data object resides within the context of a single *container*. Containers are the unit of placement, reducing not only the frequency of making global placement decisions, but also the size of the location information that has to be stored globally. *Metadata* can be associated with both data objects as well as containers. We distinguish between two categories of metadata, namely *user* and *system* metadata. While the semantics of the former are transparent to VISION Cloud, the platform provides the facilities needed to create, update and make queries on it. System metadata, on the other hand, has concrete meaning to the Cloud storage system. It either directs the system how to deal with the object (e.g., access control, reliability, performance requirements, etc.), or provides system information about the object (e.g., size, creation time, last access time, etc) to the user.

The account model of VISION Cloud consists of tenants, sub-tenants and users. A *tenant* is an organization that subscribes to the platform's services. A tenant may represent a commercial firm, a governmental organization, or any other organization, including any group of one or more individual persons. It is the entity that negotiates Service Level Agreements (SLAs) with VISION Cloud and is billed accordingly. A tenant may also define *subtenants*. Subtenants can be viewed as different departments of the same organization. A firm consisting of an R&D Department and HR Department could therefore constitute different subtenants of the same tenant. This allows for different service levels to be set according to the requirements of each department, while also providing for isolation where needed. Thus, a commercial firm

is able to "map" its business structure directly on the underlying storage. A *user* is the entity that actually consumes the storage services provided. For auditing, billing and security reasons every operation needs to be associated with a user.

5.2 Data Management

The architecture of VISION Cloud is logically divided into two facets, one concerned with data operations, such as creating and retrieving data objects, and one handling management operations, such as SLA negotiation and placement decisions of containers. The architectural separation between the data and management services is inspired by the unique service model of the storage Cloud. In compute Clouds, the management service is used to provision and manage compute resources, which interact with external entities as defined by the service provider. The storage Cloud is different - once the storage resources are provisioned, they may be used by different, independent service consumers through Cloud storage APIs, with different characteristics and requirements on latency, throughput, availability, reliability, consistency, etc. The data service and the management service are designed to be separate and independent in order to facilitate this differentiation and provide the flexibility required to enable the innovations mentioned earlier.

Most management decisions stem from the SLA under which a container is created. SLAs in VISION Cloud are dynamic, in the sense that a tenant is able to negotiate its terms, cost and penalties with the platform. Through SLAs the tenant is able to define various requirements to the platform, such as:

- Performance requirements, such latency and throughput
- Durability level, by which the probability of data loss is defined
- Availability, by which the probability that the data is available when requested is defined
- Geographic preference, by which a tenant asks that the data is stored within a specific geographic region
- Geographic exclusion, by which a tenant asks that the data is not stored within a geographic region
- Security, such as encryption of data with a specific algorithm
- Data dispersion, which defines the minimum geographic distance that datacenters holding replicas of the data should have.

These SLA requirements are automatically transformed to system requirements. For example a durability of 99.999% can be translated to creating 3 replicas of a given container. The management directives are thereafter added as metadata to containers during their creation process and are used for optimizing their placement.

In order to efficiently make placement decisions a clear and coherent global view of the system is needed. For this reason, VISION Cloud makes use of a monitoring mechanism that is able to collect, aggregate, analyze and distribute monitoring information across the collaborating clusters and data centers. Information from different sources can be collected, such as low-level hardware metrics and high-level software calls. For example every request to read an object can be monitored and logged. The purpose of this is two-fold. On one hand this information is needed for

accounting and billing purposes, as a tenant will be able to bill customers based on the number of requests they execute on the platform. On the other hand this information can be collected for the system itself with the intention to analyze them and derive knowledge that could further enhance the way the system makes management decisions. For example, seeing that a certain container is accessed frequently from a specific geographic location could lead to it being moved closer to that location.

The coupling of management metadata with an advanced monitoring framework also allows for proactive SLA violation detection schemes to be developed. Building on the knowledge that is accumulated though the analysis of monitored parameters, the system is able to proactively detect conditions that could lead to degradation of the QoS delivered and take necessary management actions to avoid a possible SLA violation.

The programming model of VISION Cloud provides storlets, which also use the metadata associated with containers and objects to effectively perform computations on the stored data. Not only can storlets be directed to execute on specific objects based on their metadata, but they are also triggered due to changes in object metadata. The results of these computations can be further used to annotate objects with additional metadata, thereby correlating the knowledge gained with the data.

The rich data model of VISION Cloud enables applications to access data objects based on their content rather than their physical or logical location, though an API that supports queries based on the metadata associated with containers and objects, realizing a content-centric access to storage.

6 Comparison

Both projects recognize the importance of providing storage as a service in the emerging era of the Future Internet but follow different approaches in realizing this vision. In the next paragraphs we summarize the similarities and differences of the two projects regarding the challenges identified by both.

Abstraction Level. A main difference is in the level of implementation. OPTIMIS follows a regular file system approach, while VISION Cloud follows an object-based storage approach with a rich data model.

Data Mobility and Federation. Data mobility and federation is considered by both projects as an important factor. OPTIMIS achieves flexibility based on the storage VM concept that allows for one-step interoperable federation and easier commission and decommission to be used for TREC-based management. VISION Cloud approaches the problem by building upon a high-level abstraction of platforms and devices, enabled by the definition of an object-based data model and its metadata mechanism.

Computations. The coupling of storage with computation is another concern of both projects. OPTIMIS makes use of storage Virtual Machines that can be started on any available computational resource that is part of the Cloud. This approach offers computational power that is in par with traditional Cloud offerings. The combination of the placement of service VMs and storage VMs can lead to improved locality of data. VISION Cloud proposes storlets, a novel programming model for computational

agents, by which computations can be injected into the Cloud and activated by events, such as creation of new data objects or changes in metadata of existing ones. These storlets run on compute resources on the storage servers where their object parameters reside.

Access Semantics to Storage. From a semantics point of view, OPTIMIS does not offer any data semantics but provides distributed storage to the end user in a transparent way. This storage, whether it is in the internal or external provider, is independent of the underlying storage implementation of traditional providers, thus reducing data lock-in. VISION Cloud enables applications to associate rich metadata with data objects, and thereby access the data objects through information about their content, rather than their location or their path in a hierarchical structure.

Placement of Data. Both projects focus on the optimal placement of data but at a different level. OPTIMIS focuses on recommending the suitable service VMs that when federated will have the least overhead with relation to the in-house distributed storage system. VISION Cloud automatically places container replicas based on the resources available in each cluster and data center, geographical constraints, and availability, resiliency and performance goals.

Legal Requirements. Finally, both approaches cover the need for user defined geographical location of data, in order to be in compliance with the legal requirements especially within the European Union area.

	OPTIMIS	**VISION Cloud**
Abstraction Level	Regular File System	Object-based storage
Data Mobility and Federation	Through storage VMs	Object-based data model with integrated metadata
Computations	Service VMs	Storlets
Access to storage	Hierarchical file system	Content-centric access
Optimal placement of data	Analysis of access patterns and legal constraints	Through SLAs and metadata
Legal Requirements	Control of geographical placement of data	Control of geographical placement of data

Fig. 3. Comparison of OPTIMIS and VISION Cloud

7 Conclusion

Providing storage as a service is an important aspect of the emerging Cloud ecosystem. Issues such as ease of management, data mobility and federation, coupling storage with computing power and guaranteeing QoS need to be researched to address the increasing volumes of data that are being produced and need to be processed and stored. In this paper we presented the contrasting approaches that two EU funded projects, OPTIMIS and VISION Cloud, take to data management and discussed their differences and similarities.

Acknowledgments. The research leading to these results is partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257019, in the context of the VISION Cloud Project and grant agreement n° 257115 , in the context of the OPTIMIS project.

References

1. OPTIMIS, http://www.optimis-project.eu/
2. VISION Cloud, http://www.visioncloud.eu/
3. Amazon simple storage service (Amazon S3), http://aws.amazon.com/s3
4. Microsoft Windows Azure Platform, http://www.microsoft.com/azure/default.mspx
5. EMC Atmos storage service, http://www.emccis.com/
6. Google Storage for Developers, http://code.google.com/apis/storage/
7. Hadoop, http://hadoop.apache.org/hdfs/
8. Ferrer, A.J., Hernández, F., Tordsson, J., Elmroth, E., Zsigri, C., Sirvent, R., Guitart, J., Badia, J.R., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, S.K., Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Kipp, A., Wesner, S., Corrales, M., Forgó, N., Sharif, T., Sheridan, C.: OPTIMIS: a Holistic Approach to Cloud Service Provisioning. In: First International Conference on Utility and Cloud Computing (UCC 2010), Chennai, India, December 14-16 (2010)
9. Barnitzke, B., Ziegler, W., Vafiadis, G., Nair, S., Kousiouris, G., Corrales, M., Wäldrich, O., Forgó N., Varvarigou, T.: Legal Restraints and Security Requirements on Personal Data and Their Technical Implementation in Clouds. To Appear in Workshop for E-contracting for Clouds, eChallenges 2011, Florence, Italy (2011)
10. Kolodner, E.K., Naor, D., Tal, S., Koutsoutos, S., Mavrogeorgi, N., Gogouvitis, S., Kyriazis, D., Salant, E.: Data-intensive Storage Services on Clouds: Limitations, Challenges and Enablers. To Appear in eChallenges 2011, Florence, Italy (2011)
11. Kolodner, E.K., Shulman-Peleg, A., Naor, D., Brand, P., Dao, M., Eckert, A., Gogouvitis, S.V., Harnik, D., Jaeger, M.C., Kyriazis, D.P., Lorenz, M., Messina, A., Shribmann, A., Tal, S., Voulodimos, A.S., Wolfsthal, Y.: Data-intensive Storage Services on Clouds: Limitations, Challenges and Enablers. In: Petcu, D., Poletti, J.L.V. (eds.) European Research Activities in Cloud Computing. Expected Date of Publication (to appear, March 2012)