September 2009

Cloud Storage for Cloud Computing

Table of Contents

List of Figures

Introduction

The Cloud has become a new vehicle for delivering resources such as computing and storage to customers on demand. Rather than being a new technology in itself, the cloud is a new business model wrapped around new technologies such as server virtualization that take advantage of economies of scale and multi-tenancy to reduce the cost of using information technology resources.

This paper discusses the business drivers in the Cloud delivery mechanism and business model, what the requirements are in this space, and how standard interfaces, coordinated between different organizations can meet the emerging needs for interoperability and portability of data between clouds.

Cloud Computing Overview

Recent interest in Cloud Computing has been driven by new offerings of computing resources that are attractive due to per-use pricing and elastic scalability, providing a significant advantage over the typical acquisition and deployment of equipment that was previously required. The effect has been a shift to outsourcing of not only equipment setup, but also the ongoing IT administration of the resources as well.

From Server Consolidation to Cloud Computing

The needed changes to applications, in order to take advantage of this model, are the same as those required for server consolidation – which had already been taking place for several years prior to the advent of the Cloud. The increased resource utilization and reduction in power and cooling requirements achieved by server consolidation are now being expanded into the cloud.

The role of server virtualization software

The new technology underlying this is the **system virtual machine** that allows multiple instances of an operating system and associated applications to run on single physical machine. Delivering this over the network, on demand, is termed ***Infrastructure as a Service*** (IaaS). The IaaS offerings on the market today allow quick provisioning and deployment of applications and their underlying operating systems onto an infrastructure that expands and contracts as needed to handle the load. Thus the resources that are used can be better matched to the demand on the applications.

How is all this managed?

IaaS offerings typically provide an interface that allows the deployment and management of virtual images onto their infrastructure. The lifecycle of these image instances, the amount of resources allocated to these instances and the storage that they use can all be managed through these interfaces. In many cases, this interface is based on REST (short for REpresentational State Transfer) HTTP operations. Without the overhead of many similar protocols the REST approach allows users to easily access their services. Every resource is uniquely addressed using a Uniform Resource Identifier (URI). Based on a set of operations – create, retrieve, update and delete – resources can be managed. Currently three types of resources are considered: storage, network and compute resources. Those resources can be linked together to form a virtual machine with assigned attributes. For example, it is possible to provision a machine that has 2GB of RAM, one hard disk and one network interface.

Standardizing Cloud Computing Interfaces

Having a programmable interface to the IaaS infrastructure means that you can write client software that uses this interface to manage your use of the Cloud. Many cloud providers have licensed their proprietary APIs freely allowing anyone to implement a similar cloud infrastructure. Despite the accessibility of open APIs, cloud community members have been slow to uniformly adopt any proprietary interface controlled by a single company. The Open Source community has attempted responses, but this has done little to stem the tide of API proliferation. In fact, Open Source projects have increased the tally of interfaces to navigate in a torrent of proprietary APIs.

What is needed instead is a vendor neutral, standard API for cloud computing that all vendors can implement with minimal risk and assured stability. This will allow customers to move their application stacks from one cloud vendor to another, avoiding lock-in and reducing costs.

Introducing OCCI

The Open Grid Forum™ has created a working group to standardize such an interface. The Open Cloud Computing Interface (OCCI) is a free, open, community consensus driven API, targeting cloud infrastructure services. The API shields IT data centers and cloud partners from the disparities existing between the lineup of proprietary and open cloud APIs.

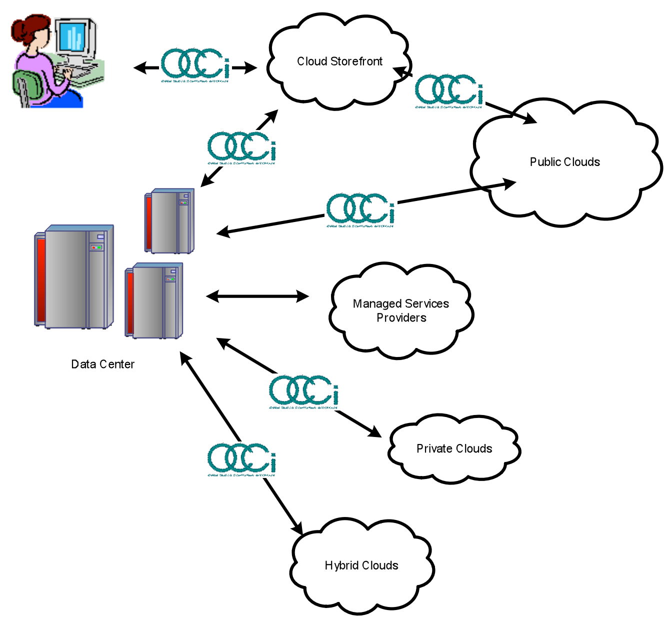


Figure 1: The OCCI API

The OCCI Reference Architecture

The OCCI has adopted a "Resource Oriented Architecture (ROA)" to represent key components comprising cloud infrastructure services. Each resource (identified by a canonical URI) can have multiple representations that may or may not be hypertext (e.g. HTML). The OCCI working group is planning mappings of the API to several formats. Atom/Pub, JSON and Plain Text are planned for the initial release of the standard. A single URI entry point defines an OCCI interface. Interfaces expose "nouns" which have "attributes" and on which "verbs" can be performed.

Figure 1 shows how the components of an OCCI URI aligns to IaaS Resources:

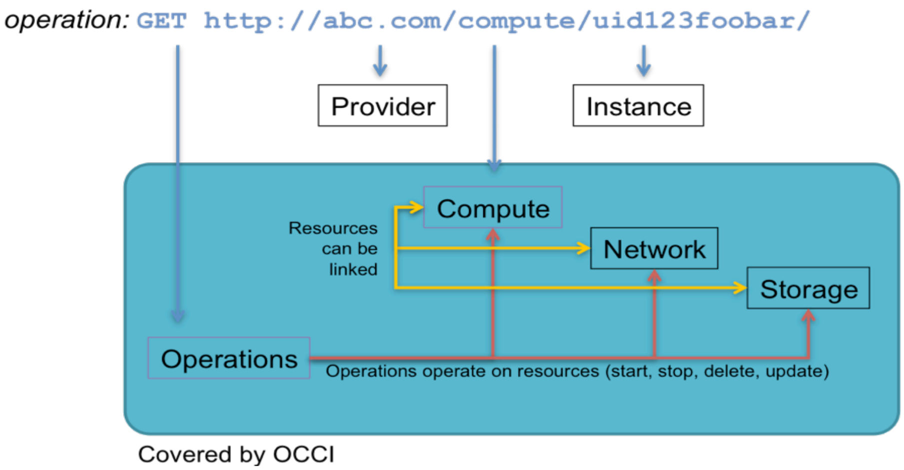


Figure 2: Alignment of OCCI URI to IaaS Resources

Attributes are exposed as key-value pairs and the appropriate verbs as links. The attributes may be described as a URI. Adopting URI support affords the convenience of referencing (linking to) other interfaces including SNIA's Cloud Data Management Interface (CDMI), for example.

The API implements CRUD operations: Create, Retrieve, Update and Delete. Each is mapped to HTTP verbs POST, GET, PUT and DELETE respectively. HEAD and OPTIONS verbs may be used to retrieve metadata and valid operations without the entity body to improve performance. All HTTP functionality can take full advantage of existing internet infrastructure including caches, proxies, gateways and other advanced functionality.

All metadata, including associations between resources is exposed via HTTP headers (e.g. the Link: header). The interface, natively expressed as Atom, executes as close as possible to the underlying Hyper Text Transfer Protocol (HTTP). In one case where the HTTP protocol did not explicitly support Atom collections, an Internet Draft (draft-johnston-http-category-header-00.txt) for a new HTTP header supporting Atom collections, has been submitted by an OCCI working group coordinator to the IETF for standardization.

OCCI provides the capabilities to govern the definition, creation, deployment, operation and retirement of infrastructures services. Using a simplified service lifecycle model, it supports the most common life cycle states offered by cloud providers. In the event providers do not support or report service life cycle states, OCCI does not mandate compliance, defining the life cycle model as only a recommendation. Cloud providers wishing to do so, can comply with the OCCI service life cycle recommendations.

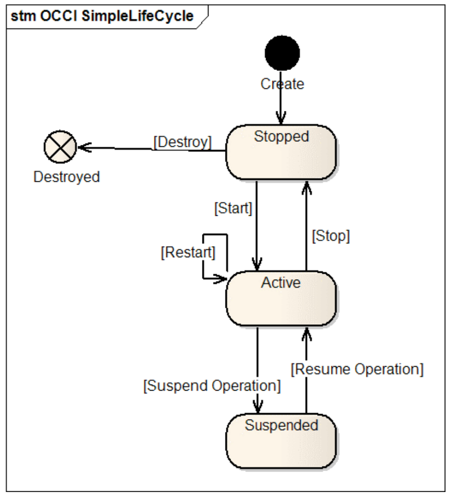


Figure 3: The OCCI Lifecycle Model

With OCCI, cloud computing clients can invoke a new application stack, manage its lifecycle and manage the resources that it uses. The OCCI interface can also be used to assign storage to a virtual machine in order to run the application stack such as that exported by SNIA's CDMI interface. Next we will examine the means of managing that storage and the data in it.

Cloud Storage Overview

Just like Cloud Computing, Cloud Storage has also been increasing in popularity recently due to many of the same reasons as Cloud Computing. Cloud Storage delivers virtualized storage on demand, over a network based on a request for a given quality of service (QoS). There is no need to purchase storage or in some cases even provision it before storing data. You only pay for the amount of storage your data is actually consuming.

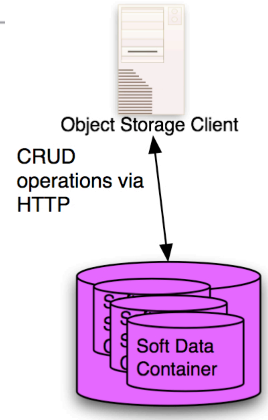
Some of the Use Cases

Cloud storage is used in many different ways. For example: local data (such as on a laptop) can be backed up to cloud storage; a virtual disk can be "synched" to the cloud and distributed to other computers; and the cloud can be used as an archive to retain (under policy) data for regulatory or other purposes.

Web facing applications

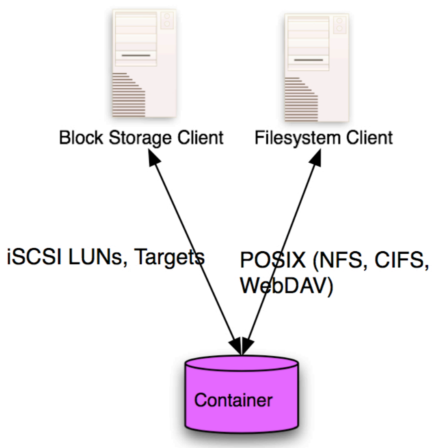
For applications that provide data directly to their clients via the network, cloud storage can be used to store that data and the client can be redirected to a location at the cloud storage provider for the data. Media such as audio and video files are an example of this, and the network requirements for streaming data files can be made to scale in order to meet the demand without affecting the application.

The type of interface used for this is just HTTP. Fetching the file can be done from a browser without having to do any special coding, and the correct application is invoked automatically. But how do you get the file there in the first place and how do you make sure the storage you use is of the right type and QoS? Again many offerings expose an interface for these operations, and it's not surprising that many of these interfaces use REST principals as well. This is typically a data object interface with operations for creating, reading, updating and deleting the individual data objects via HTTP operations.



Storage for Cloud Computing

For cloud computing boot images, cloud storage is almost always offered via traditional block and file interfaces such as iSCSI or NFS. These are then mounted by the virtual machine and attached to a guest for use by cloud computing. Additional drives and filesystems can be similarly provisioned. Of course cloud computing applications can use the data object interface as well, once they are running.



What makes Cloud Storage different?

The difference between the purchase of a dedicated appliance and that of cloud storage is not the functional interface, but merely the fact that the storage is delivered on demand. The customer pays for either what they actually use or in other cases, what they have allocated for use. In the case of block storage, a LUN or virtual volume is the granularity of allocation. For file protocols, a filesystem is the unit of granularity. In either case, the actual storage space can be thin provisioned and billed for based on actual usage. Data services such as compression and deduplication can be used to further reduce the actual space consumed.
The management of this storage is typically done out of band of these standard Data Storage interfaces, either through an API, or more commonly, though an administrative browser based user interface. This interface may be used to invoke other data services as well, such as snapshot and cloning.

Introducing CDMI

The Storage Networking Industry Association™ has created a technical work group to address the need for a cloud storage standard. The new Cloud Data Management Interface (CDMI) is meant to enable interoperable cloud storage and data management. In CDMI, the underlying storage space exposed by the above interfaces is abstracted using the notion of a container. A container is not only a useful abstraction for storage space, but also serves as a grouping of the data stored in it, and a point of control for applying data services in the aggregate.

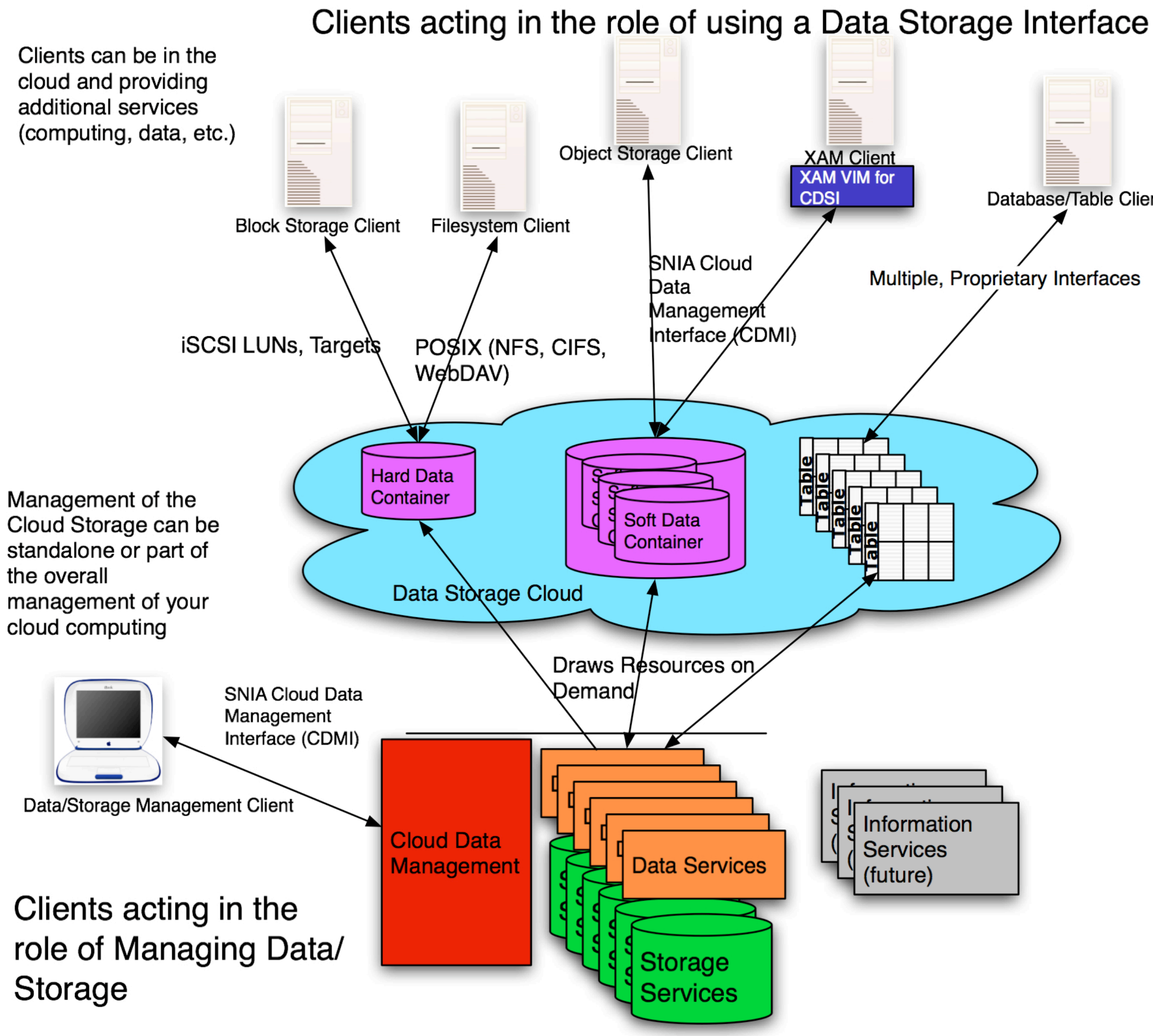


Figure 4: The Cloud Storage Reference Model

CDMI provides not only a data object interface with CRUD semantics; it also can be used to manage containers exported for use by cloud computing infrastructures as shown above in Figure 4.

CDMI for Cloud Computing
With a common cloud computing management infrastructure

Using CDMI and OCCI for a Cloud Computing Infrastructure

CDMI Containers are accessible not only via CDMI as a data path, but other protocols as well. This is especially useful for using CDMI as the storage interface for a cloud computing environment as shown in Figure 5 below:

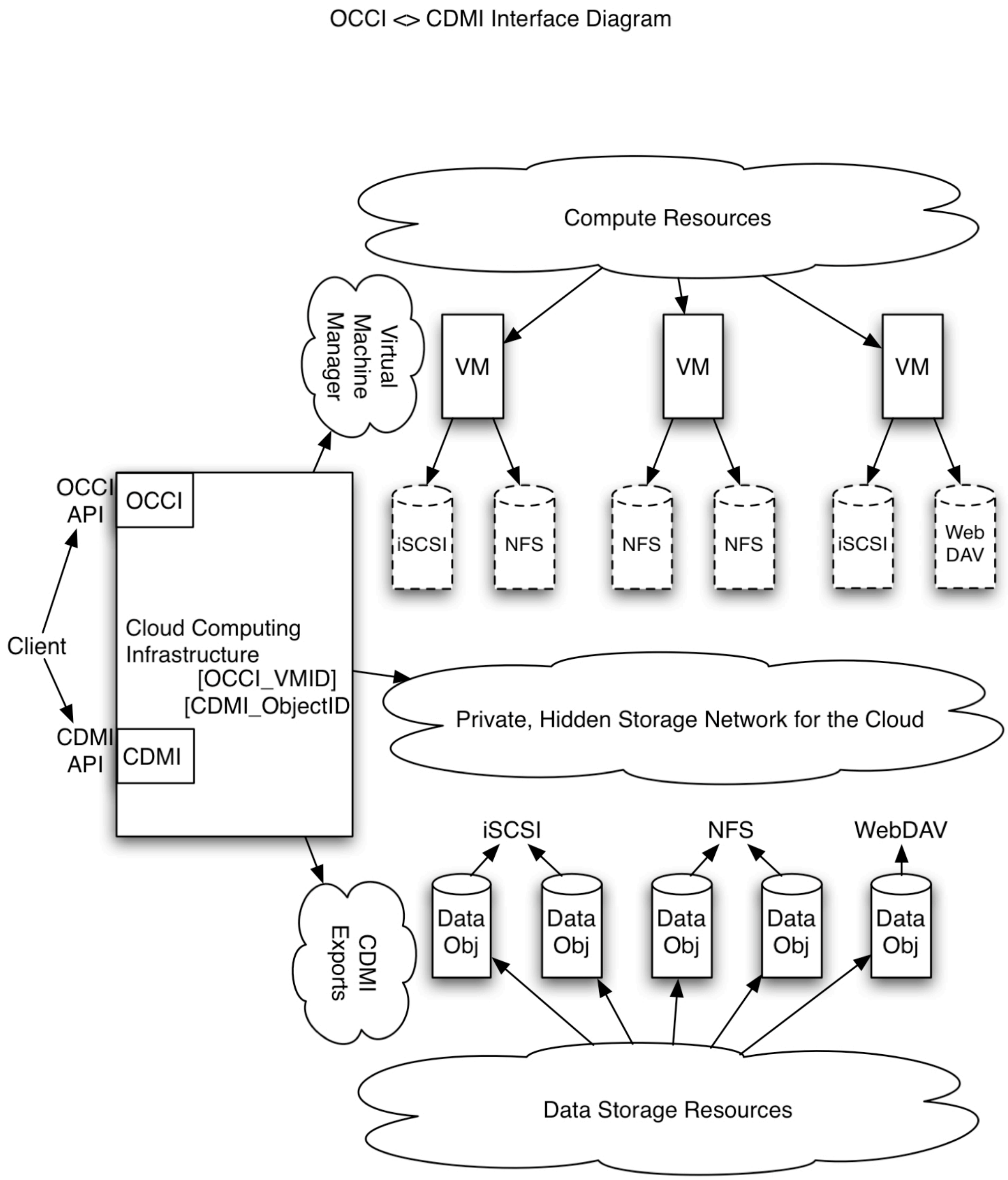


Figure 5: CDMI and OCCI in an integrated cloud computing environment

The exported CDMI containers can be used by the Virtual Machines in the Cloud Computing environment as virtual disks on each guest as shown. With the internal knowledge of the network and the Virtual Machine, the cloud infrastructure management application can attach exported CDMI containers to the Virtual Machines.

How it works

The cloud computing infrastructure management shown above supports both OCCI and CDMI interfaces. To achieve interoperably, CDMI provides a type of export that contains information obtained via the OCCI interface. In addition, OCCI provides a type of storage that corresponds to exported CDMI containers.
OCCI and CDMI can achieve interoperability initiating storage export configurations from either OCCI or CDMI interfaces as starting points. Although the outcome is the same, there are differences between the procedures using CDMI's interface over the OCCI's as a starting point. Below, we present examples of interoperability initiating storage export from both CDMI and OCCI approaches

A client of both interfaces would perform the following operations as an example:

- The Client creates a CDMI Container through the CDMI interface and exports it as an OCCI export type. The CDMI Container ObjectID is returned as a result.
- The Client then creates a Virtual Machine through the OCCI interface and attaches a storage volume of type CDMI using the ObjectID. The OCCI Virtual Machine ID is returned as a result.
- The Client then updates the CDMI Container object export information with the OCCI Virtual Machine ID to allow the Virtual Machine access to the container.
- The Client then starts the Virtual Machine through the OCCI interface.

Standards Coordination

As can be seen above OCCI and CDMI are standards working towards interoperable cloud computing and cloud storage. The standards are being coordinated through an alliance between the OGF and the SNIA as well as through a cross-SDO cloud standards collaboration group at http://cloud-standards.org. OCCI will take advantage of the storage that CDMI has provisioned and configured. Since both interfaces use similar principles and technologies, it is likely that a single client could manage both the computing and storage needs of an application, scaling both to meet the demands placed on them.

About the SNIA

The Storage Networking Industry Association (SNIA) is a not-for-profit global organization, made up of some 400 member companies spanning virtually the entire storage industry. SNIA's mission is to lead the storage industry worldwide in developing and promoting standards, technologies, and educational services to empower organizations in the management of information. To this end, the SNIA is uniquely committed to delivering standards, education, and services that will propel open storage networking solutions into the broader market. For additional information, visit the SNIA web site at www.snia.org.

About Open Grid Forum:

OGF is the premier world-wide community for the development and adoption of best practices and standards for applied distributed computing technologies. OGF's open forum and process enable communities of users,

infrastructure providers, and software developers from around the globe in research, business and government to work together on key issues and promote interoperable solutions.
http://www.ogf.org