ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

A Survey on SNIA Cloud Data Management Interface for Cloud Storage

Rashmi. G

Assistant Professor, Department of Computer Science & Engineering, KLEDr.MSSCET, Belgaum, India

Abstract: The Storage Networking Industry Association (SNIA) Cloud Data Management Interface (CDMI) is an ISO/IEC standard that provides simplicity and data storage interoperability to customers across a wide range of cloud solutions.This standard is applicable to private, public and hybrid clouds and it provides end users with the ability to control the destiny of their data and ensure hassle-free data access, data protection and data migration form one cloud service to another.This paper presents overview of CDMI which allows applications to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use it to manage containers and the data placed in them. It allows metadata to be set on containers and their contained data elements. This standard is used by administrative and management applications to manage containers, accounts, security access and monitoring/billing information.

Keywords: Cloud Storage, Cloud Computing, CDMI, SNIA, cloud Storage Interface

I. INTRODUCTION

Today everyone with a computer spends a lot of time in collecting data and then trying to find a way to store it. For some people, finding enough storage space to store all the data they've acquired is a real challenge. Some people use larger hard drives. Others prefer external storage devices like thumb drives or compact discs. Some people might delete entire folders containing huge old information in order to make space for new information. But some are choosing to rely on a growing trend: **cloud storage**[1].Cloud storage really refers to saving data to an off-site storage system maintained by a third party. Instead of storing information to personal computer's hard drive or other local storage device, it issavedon to a remote database. The Internet provides the connection between computer and the database. On the surface, cloud storage has several advantages over traditional data storage. For example, if data is stored on a cloud storage system, it can be accessed from any location that has Internet access. There is no need to carry around a physical storage device or use the same computer to save and retrieve your information.

CDMI international standard is intended for application developers who are implementing or using cloud storage. It tells how to access cloud storage and to manage the data stored in it, supports data exchange between cloud systems, exporting data via other protocols such as iSCSI [4] and NFS[5]. It defines RESTful [2] HTTP [3] operations for assessing the following components.

- Capabilities
- Containers
- Objects
- Domains, Users and Groups
- Access Control
- Metadata
- Queries
- Queues
- Compliance
- Logging
- Billing
- Serialization
- Foreign Protocols

1.1 Capabilities

This refers to providing access to a set of configuration parameters. These are either boolean values that represent whether a system supports or do not support things such as queues, protocols, numeric values expressing system limits, such as how much metadata may be placed on an object and so on.

1.2 Containers

These are nothing but enclosures of data as objects and allow CDMI clientsto access objects by either name or object id (OID).

1.3 Objects

These are similar to files with an increased capacity for metadata. As with containers, they may be accessed by either name or OID. When accessed by name, clients use URLs that contain the full pathname of objects to create, read, update and delete them. When accessed by OID, the URL specifies an OID string in the **cdmi-objectid** containers. They may be of any size or type and have user-supplied metadata attached to them.

1.4 Domains, Users and Groups

CDMI supports the concept of a *domain* [13],similar in concept to a domain in the Windows Active Directory model. Users and groups created in a domain share a common administrative database and are known to each other on a "first name" basis.

1.5 Access Control

It is a mechanism by which various types of access to objects and containers are authorized and permitted or denied. CDMIuses the well-known mechanism of an Access Control Lists (ACLs) [6] which is a list of permissions-granting or permissions-denying entries called Access Control Entries (ACEs).

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

1.6 Metadata

Whilemanaging large amounts of data with differing requirements, metadata is a convenient way to express those requirements. CDMI allows Objects and containers to have storage system metadata,data system metadata and arbitrary user specified metadata and the metadata maintained by an ordinary file system.

1.7 Queries

CDMI supports arbitrary queries against CDMI containers.

1.8 Queues

CDMI supports the concept of FIFO (first-in, first-out) queues. These are useful for job scheduling, order processing and other tasks in which lists of things must be processed in order.

1.9 Compliance

CDMI supports retention intervals and retention holds. A retention interval consists of a start time and a retention period. During this time interval, objects are preserved as immutable and may not be deleted. A retention hold is usually placed on an object and they may not be changed nor deleted until all holds placed on them are removed.

1.10 Logging

CDMI allows its clients to login securely and see events locally once the server logs them.

1.11Billing

Summary information suitable for billing clients for on-demand services can be obtained by authorized users from systems that support it.

1.12 Serialization

Serialization of objects and containers allows export of all data and metadata on a system and importation of that data into another cloud system.

1.13 Foreign Protocols

CDMI supports NFS and CIFS [7] protocols and alsoiSCSI LUNs [8].

II. LITERATURE SURVEY

Today businesses, governments, non-profit organizations and individual consumers are all facing growing challenges in storing, managing, protecting and mining thedata being generated in an increasingly digital world. And also, storage vendors often use proprietary APIs [9]. This means that application vendors would have to support a plenty of APIs from a number of different vendors, leading to a lack of commitment from application vendors to support more innovative, object-based storage architectures. There are few APIs such as Amazon's API (as with any cloud vendor's API) are moving target for sure, but the main issue is that these APIs are under the change control of a single vendor. Doesn't matter how "open" the API is (in terms of copyright license) because the vendor can change it to disadvantage a competitor. Each cloud vendor releases their own "open" API – similar but slightly different (enough to get around copyright), almost always RESTful and pretty much they all do the same thing. So, situation has come where in with rapid proliferation of many different interfaces all pretty much the same. But that doesn't help the poor clients. They have to code to N different interfaces to work with N different clouds. And since they are rapidly evolving, they have to keep up with all these API changes over time. To solve the above mentioned problems, SNIA thought of introducing a standard interface CDMI which addresses the accessibility, security, portability and cost issues associated with the relentlessly growing pools of data and supports ingest and retrieve data into and out of a large-scale repository by enabling applications to easily manage this repository and where the data sits. The CDMI also addresses API issues as it is under the change control of a standards body and accommodates requirements from all the cloud storage players in it's standardization process.

More importantly, it was developed under the SNIA IP policy to help prevent any of the specification author companies from gaming the specification with their Intellectual Property. Thus cloud vendors can pick up the CDMI specification and implement it with confidence. They don't need to come up with their own API. CDMI also has a standard way to extend the specification for vendor specific functions that still allows for core compatibility with other vendors. The CDMI specification is focused on simplifying all aspects of cloud storage for subscribers (users), service providers, developers and IT hardware and software vendors. When discussing cloud storage and standards, it is important to distinguish the various resources that are being offered as services. These resources are exposed to clients as functional interfaces (i.e., data paths) and are managed by management interfaces (i.e., control paths). This international standard provides various types of interfaces that are part of offerings today and shows how they are related and defines a model for the interfaces that may be mapped to the various offerings.

In April 2009, more than 140 companies formed the SNIA Cloud Storage Technical Working Group (TWG), and a Google News Group with more than 280 participants was launched. In June 2009, the TWG published its first working documents in the form of Cloud Storage Use Cases and a Reference Model. Based on these, the SNIA Cloud Storage TWG worked on a proposed Cloud Data Management Interface (CDMI) standard throughout the summer. In September 2009, the TWG released draft version 0.8 of the CDMI specification for public review and comment at SNIA's Storage Developer Conference. SNIA Cloud Storage TWG member companies, including Bycast, Cisco, NetApp, Olocity, QLogic, Sun and Xyratex were significant contributors to the draft release CDMI. The 1.0.2 specification is now a cloud storage standard released for public review which helps in defining roles and responsibilities for data ownership, archiving, discovery and retrieval.

An important part of cloud model is the concept of fetching one or more resources from a pool of resources on demand. A relatively recent innovation that has made this possible is virtualization [12]. Thus, cloud storage is simply the delivery of virtualized storage on demand.The formal term that is used for this is Data storage as a Service (DaaS).

2.1 Data Storage as a Service(DaaS)

It simply means providing data storage to users on demand regardless of geographical location.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 7, July 2014

An important part of DaaS offering is the support of legacy clients. Existing data storage interface makes use of standard protocols such as iSCSI for block and CIFS/NFS or WebDAV [14] for file network storage and HTTP for object storage, as shown in Fig. 2.1.

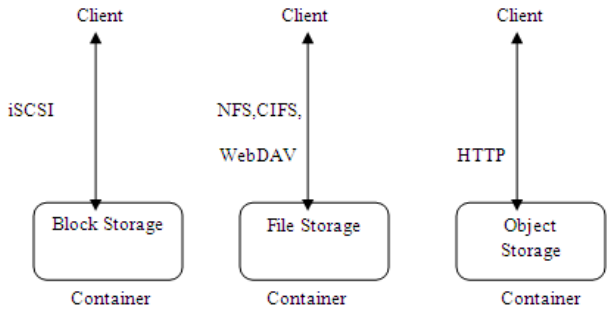

Fig. 2.1 - Existing Data Storage

In Block storage, a block, sometimes called a physical record, is a sequence of bytes or bits, usually containing some whole number of records, having a maximum length, a *block size*. Data thus structured are said to be *blocked*. The process of putting data into blocks is called *blocking*, while *deblocking* is the process of extracting data from blocks. Blocked data is normally stored in a data buffer and read or written a whole block at a time. Blocking reduces the overhead and speeds up the handling of the data-stream. For some devices such as magnetic tape and CKD disk devices blocking reduces the amount of external storage required for the data. Blocking is almost universally employed when storing data to 9-track magnetic tape, to rotating media such as floppy disks, hard disks, optical discs and to flash memory.

Traditional filesystems and access methods were not designed to store hundreds of millions or billions of files in a single namespace. But,this lead to admins storing data in multiple file systems, multiple shares, and complex directory structures – not because the data should be logically organized in that way, but simply because of limitations in file systemsuch as cconverting the type of a file system,migrating to a different file system, long file paths and long file names.This issue became more pressing when data sits in multiple locations, maybe even across on-premise and off-premise, cloud-based storage.

Object storage technology provides the ability to store data in objects, along with metadata that describes the object. Data can be searched based on metadata tags (like a filename – or even better an account number and document type) – as well as manage data based on policies that leverage that metadata. However, this often means that interfacing with storage system through APIs must be considered, as opposed to NFS and CIFS so that applications need to support whatever API storage vendor offers.

When it comes to creating and managing large, distributed content repositories it quickly becomes clear that NFS and CIFS are not ideally suited for this use case. This is where CDMI shines, especially with an object-based storage architecture behind it that was built to support multi-petabyte environments with billions of data sets across hundreds of sites and accommodates retention policies that can reach to "forever".

Managing data and managing containers can have common interfaces. Therefore, the use of metadata is extended from applying to individual data elements to applying to containers of data as well. Thus, any data placed into a container inherits the data system metadata of the container into which it was placed. When creating a new container within an existing container, the new container would similarly inherit the metadata settings of its parent's data system metadata. After a data element is created, the data system metadata may be overridden at the container or individual data element level, as desired.

Even if the provided interface does not support setting metadata on individual data elements, metadata may still be applied to the containers. In such a case, the interface does not provide a mechanism to override metadata that an individual data element inherits from its parent container.

In case of block storage [16], a Logical Unit Number is the granularity of allocation. For file storage [17], a file system is the unit of granularity. In either case, the actual storage space may be thin provisioned. Managing this storage is typically done out of band for these standard data storage interfaces, either through an API or browser-based user interface. This out-of-band interface may be used to invoke other data services such as snapshot [10].

As shown in Figure 2.1, object storage[18] provides a container which 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. Each data object is created, retrieved, updated, and deleted as a separate resource.

Thus, proposed CDMI standard uses object storage to manage large amounts of data with differing requirements.

2.2 Data Management for Cloud Storage

By supporting metadata in a cloud storage interface and prescribing how the storage system and data system metadata is interpreted to meet the requirements of the data, the simplicity required by the cloud storage model may be maintained while still addressing the requirements of enterprise applications and their data. Metadata are of following types.

2.2.1 User metadata[11] is retained by the cloud and may be used to find the data objects and containers by performing a query for specific metadata values. The schema for this metadata may be determined by each application, domain, or user.

2.2.2 Storage system metadata [11] is produced/interpreted by the cloud and basic storage functions (e.g., modification and access statistics, access control).

2.2.3 Data system metadata [11] is interpreted by the cloud as data requirements that control the operation of underlying data services for that data. It refers to an aggregation of data objects in a container or to individual data objects.

2.3 Data and Container Management

Managing data and managing containers can have common interfaces. Therefore, the use of metadata is extended from applying to individual data elements to

applying to containers of data as well. Thus, any data placed into a container inherits the data system metadata of the container into which it was placed. When creating a new container within an existing container, the new container would similarly inherit the metadata settings of its parent's data system metadata. After a data element is created, the data system metadata may be overridden at the container or individual data element level, as desired.

Even if the provided interface does not support setting metadata on individual data elements, metadata may still be applied to the containers. In such a case, the interface does not provide a mechanism to override metadata that an individual data element inherits from its parent container.

III. REFERENCE MODEL

The CDMI model as explained in [11] is used to create, retrieve, update, and delete objects in a cloud.

IV. BENEFITS

CDMI standard benefits following groups.

4.1 Cloud Users: CDMI provides cloud storage users with a simple, common interface to help them discover the appropriate cloud storage service providers for their specific requirements.

4.2 Cloud storage service providers: CDMI provides a common interface for cloud storage service providers to advertise their capabilities and help subscribers discover them and also provides unique, non-standard extensions for service providers.

4.3 Cloud storage service developers: CDMI provides multi-vendor, industry-standard development interface for application developers who want to store data in the cloud and also ensures a broad infrastructure of compatible service providers for application developers.

4.4 Cloud storage service brokers: Brokers are nothing but middle man's who offer services to subscribers such as enterprises or government entities which have complex cloud storage requirements. For example, brokers could offer "cloud insurance" via CDMI by combining a primary and secondary set of cloud storage providers to the broker's customers (subscribers). If the primary cloud storage service provider fails to provide services, then the secondary cloud storage service can take over according to the SLAs [15].

V. OBSERVATION

The CDMI assures cloud consumers that it has been consistently implemented for multi-solution data storage interoperability across a wide range of cloud storage solutions.

VI. CONCLUSION

The CDMI standard supports interoperability and portability between service providers. Based on a RESTful HTTP protocol, it provides both a data path and control path for cloud storage and standardizes a common interoperable data exchange format for securely moving data and its associated data requirements from cloud to cloud. The standard is applicable to public, private and hybrid clouds and benefits both vendors and end-users since vendors gain quicker time to market, and reduced costs, while end users get greater interoperability, reduced training, and lower total cost of ownership.

REFERENCES

[1] Mahima Joshi, Yudhveer Singh Moudgil et al, "SECURE CLOUD STORAGE", International Journal of Computer Science & Communication Networks, Vol 1(2), 171-175, Oct-Nov 2011, ISSN:2249-5789.
[2] http://en.wikipedia.org/wiki/Representational_state_transfer
[3] http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
[4] http://en.wikipedia.org/wiki/ISCSI
[5] http://en.wikipedia.org/wiki/Network_File_System_(protocol)
[6] http://en.wikipedia.org/wiki/Access_control_list
[7] http://technet.microsoft.com/en-us/library/cc939973.aspx
[8] http://en.wikipedia.org/wiki/LUN
[9] http://en.wikipedia.org/wiki/Proprietary_software
[10] http://en.wikipedia.org/wiki/Snapshot_(computer_storage)
[11] http://cdmi.sniacloud.com
[12] Evidence and Cloud Computing: The Virtual Machine Introspection Approach Rainer Poisel1y, Erich Malzer2, and Simon Tjoa11St. Poelten University of Applied SciencesSt. Poelten, Austria.
[13] http://en.wikipedia.org/wiki/Domain_name
[14] http://en.wikipedia.org/wiki/WebDAV
[15] Pankesh Patel, Ajith Ranabahu, AmitSheth Service Level Agreement in Cloud Computing1 KnoesisCenter, Wright State University, USA.
[16] http://en.wikipedia.org/wiki/Block_storage
[17] http://en.wikipedia.org/wiki/File_systems
[18] http://en.wikipedia.org/wiki/Object_storage