Standards Based Cloud Clients

Michael Behrens, *R2AD, LLC*
David Moolenaar, *R2AD, LLC*
Mark A. Carlson, *Chair, SNIA Cloud Storage Technical Work Group*
clouds@r2ad.com

Abstract

The computing industry is experiencing a huge shift in the way we use computers at home, at work, and especially in-between as we have hand-held mobile computing devices which also work as telephones. This paper explores how client devices can use cloud oriented software. One area covered is cloud management and the other is cloud based applications or Software as a Service. Recent specifications have been produced in the cloud standards space, namely Open Cloud Computing Infrastructure (OCCI) and Cloud Data Management Interface (CDMI). This paper explores how cloud based clients might take advantage of these.

1. Introduction

Traditional models of client software development included stand-alone applications and client-server applications. Following this is the distributed computing model in which there are n-tiers of servers and or clients. Peer-2-Peer is another which could be considered a variation of client-server computing. However with the concept of a cloud, we now might consider client-cloud based applications as a form of distributed computing. In this case, the client is interacting with an endpoint which represents a large server side potential. The "one to many" cloud connections enable scalability through parallelism.

This paper covers two uses of cloud clients: administration and end-user. A special case is also considered in which the client application itself is executing within the cloud and the client is merely acting as a dumb-terminal.

2. Cloud Access Specifications

A number of attempts are being made to standardize access to the cloud. Efforts are underway by standard development organizations such as the Open Grid Forum (OGF), Storage Networking Industry Association (SNIA), Distributed Management Task Force (DMTF), National Institute of Standards and Technology (NIST), and others. In our case, we have focused on two specifications: Open Cloud Computing Infrastructure (OCCI) [1] and Cloud Data Management Interface (CDMI) [2].

2.1. OCCI

The OCCI specification is developed by the Open Grid Forum (OGF) Standards Development Organization (SDO). OCCI is a boundary protocol/API that acts and fronts as a service front-end to your current internal infrastructure management framework (IMF). The following diagram shows OCCI's place in the communications chain.

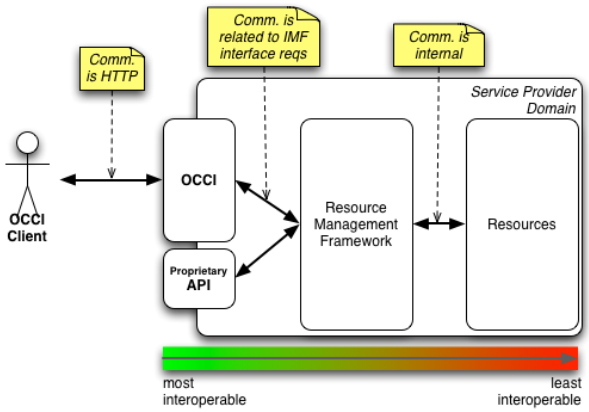

Fig. 1 OCCI as a Boundary API

OCCI consists of a set of specifications (Core and models, HTTP header rendering, Infrastructure models, and XHTML5 rendering). These documents, while written to address a standard interface for cloud computing, are in fact directly applicable to any distributed computing architecture. They represent the "State of the Art" in terms of internet computing today. They support Rich Internet Application Development, Service Oriented computing, and offer a scalable and dynamic approach to creating semantic oriented services.

The OCCI specifications refer to this architecture as a Resource Oriented Architecture (ROA). It defines RESTful [3] interfaces based on the Hypertext Transfer Protocol (HTTP). Each resource (a computer, or storage element, etc) is identified by a URI. One or more "representations" of that resource can be requested using the HTTP "GET" verb. The PUT and POST verbs in HTTP can be used to create and update resources. Associations of resources can be conveyed in the HTTP headers, using the link tags.

With this as the foundation of the specification, IaaS services can be built. Following this pattern, additional layers of the specification can be added to provide support for PaaS and SaaS.

2.2. CDMI

The Cloud Data Management Interface (CDMI) is developed by Storage Network Industry Association (SNIA). It provides a RESTful API set to manage storage in the cloud. CDMI provides cloud data management in support of both public and private storage. In addition to providing management of storage, it also provides a means to create objects within containers, providing the building blocks necessary for any basic application that needs to store data. CDMI is the functional interface that applications may use to create, retrieve, update, and delete data elements from the cloud. It also supports the addition of metadata which can be used to tag data within containers. CDMI uses many different types of metadata, including HTTP metadata, data system metadata, user metadata, and storage system metadata.

From a storage access perspective, CDMI also supports various standard protocols such as CIFS, NFS, iSCSI, and its own standardized RESTful data path.

From a client perspective, basic HTTP GET and PUT requests can be issued and the resultant JavaScript Object Notation (JSON) message bodies can be used within a browser context or parsed by other clients.

Together with client side software, the CDMI specification can be used to provide cloud services such as cloud2cloud (C2C) migration, data backup services, searches, and application storage.

3. Implementing a Standards based Cloud Client

Using both specifications together, our team has developed a basic cloud management client which demonstrations the core functionality of both the OCCI and the CDMI specification. Following a use-case driven development process, we first used OCCI to access a list of cloud compute nodes. Then, using CDMI, we access a set of storage resources.

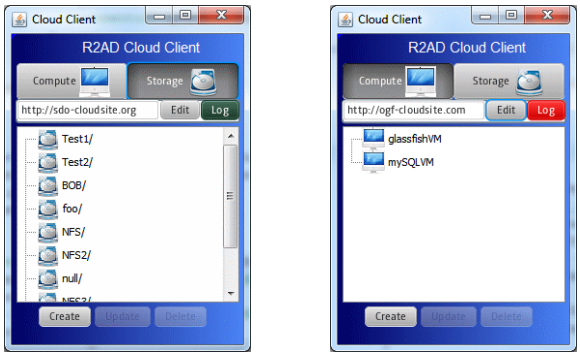

Fig. 2 - Compute Resources and Storage Resources

Using both together, we've created a mashup of the two specifications that allow the creation of compute resources that reference the CDMI storage.

Figure 2 shows the compute tab with the sample virtual machines that are residing on the OCCI controlled servers and also the storage tab listing the available storage containers that the CDMI cloud service exposes. Creating or editing a compute resource also provides the ability to assign storage.

3.1 Cloud Management Client Design

The design of the application developed included a login component, a presentation layer, and parsers for both XML and JSON documents that are received from the cloud based services.

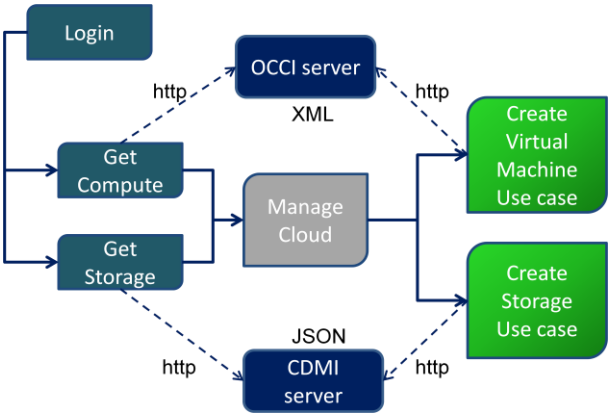

Fig. 3 - Cloud Management Design

A number of lessons were learned during the development and testing of the client, namely:

1) The network is not always present. Developing a cache of recently access data would be useful. A recommendation for cloud storage standards would be to add support for cache synchronization.
2) Remote administration tools were valuable. These included the obvious tools such as SSH access, but also included tools with provided full screen remote desktop access such as Oracle Secure Global Desktop.
3) Deciding to make the code open source before the development cycle would have saved some time later one. Currently, the client code is available online from http://cloud.r2ad.net.

7. References

[1] Open Cloud Computing Infrastructure (OCCI), *http://forge.ogf.org/sf/go/projects.occi-wg/wiki*.

[2] Storage Networking Industry Association, "Cloud Data Management Interface (CDMI)", 2010 [Online]. Available: *http://snia.org/cloud*

[3] Roy Fielding, Representational State Transfer (REST), http://portal.acm.org/citation.cfm?id=93229