Ontology based Cloud Framework

Abhishek Kaushik
Computer Science and Engineering Department
Thapar University
Patiala-147004

Inderveer Chana
Computer Science and Engineering Department
Thapar University
Patiala- 147004

ABSTRACT
The fluctuating demands of software and hardware IT infrastructure have resulted in Cloud Computing to be the fastest growing trend in the Information Technology industry. From a business perspective, organizations adopt cloud computing as they no longer need to buy or maintain expensive and energy-draining equipments. IT administration including licensing issues, software updates and IT security management, all are taken care by the cloud service provider. Removal of this administrative burden allows organizations to concentrate on their core business and be more productive. These characteristics have increased Cloud Computing market share but along with the complexity of cloud has also increased. Now it has becomes more difficult to develop an efficient and highly flexible cloud platform. As web is moving towards Web 2.0 (Semantic Web), it is shifting towards representing things as per their meaning (semantic representation). Cloud Computing is totally based on internet for any possible functioning. It thus becomes mandatory for cloud computing to adopt itself according to the future trends. This paper presents an Ontology based Cloud Framework. The framework demonstrates that by using ontology based architecture cloud can be easily accessed and updated using semantic web queries and the administrative burden of the cloud provide can be reduced considerably.

General Terms
Cloud Computing, Ontology, Standardization, OWL

Keywords
Cloud Computing, Ontology, Semantic Web

1. INTRODUCTION
Cloud Computing is one of the biggest trends in Information Technology. The basic idea behind cloud computing is to pool all the available resources and allocate them to users on the demand, leaving question of “indefinite ownership” out of equation. Cloud computing touches every aspect of Software and Information industry in one or another way. Thousands of clients all over the world shares software and hardware resources without even knowing there source. Different sectors like education, industry, government also uses services like SaaS (Software as Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service) provided by cloud. The “Standardization of Cloud Computing” [1] thus becomes more and more crucial and challenging.

Cloud computing enables a dispersed workforce to work effectively and allows easy collaboration with partners. Many organizations have workers based around the country or the globe. These employees often need access to their systems and a hosted desktop service. Cloud computing technology enables them to access their desktop from any location. This radically improves the organization’s efficiency but simultaneously degrading the cloud flexibility and this degradation in flexibility leads to difficult cloud standardization.

Today’s cloud platform and infrastructure providers are providing much of the same services but require applications to interact with them in completely different ways. So if a user wants to move his applications between two cloud providers he has to re-architect the whole applications according to the new cloud provider. Semantic Web offers a solution to the emerging problem by providing a common cloud ontology using Ontology Web Language (OWL). The Semantic Web is a “Web of data” — of dates and titles and part numbers and chemical properties and any other data one might conceive of. RDF provides the foundation for publishing and linking data [2]. This work presents a flexible cloud architecture based on Owl that can be easily accessed and quires by cloud users.

Rest of paper organized as follows: Section 2 contain other researches related to ontology and cloud, Section 3 is explains the prototype of ontology is used in developing the cloud, Section 4 explains the main classes that are used in cloud ontology and in Section 5 query part is explained then there is “Conclusion” where conclusion of the paper and future work is mentioned last is “References”.

2. RELATED WORK
There are many other impressive works on Cloud Ontology. Lamia Youseff et al. [3] gave a very thorough interpretation of cloud in terms of ontology in their work, covering all the theoretical points but without details of technical implementations. Similarly Yong Beom Ma, Sung Ho Jang, and Jong Sik Lee [4] proposed resource management algorithm in cloud using ontology. In their work they suggest that user’s request for resources is not processed only on the basis of CPU size, operating system or storage but also using agreed SLA’s (service level agreement).


Above research works explain how ontology helps in improving the different aspects of cloud computing, but none of them focuses on the flexibility and simplicity that can be achieved by developing entire cloud platform using ontology. Also the implementation of web semantic languages for user interaction with the cloud is not mentioned in any of these and most of other papers. In this paper we have prototyped a whole cloud platform that can be easily modified according to requirement and in
which queries can be done in a most suitable user friendly manner.

3. CLOUD ONTOLOGY

Ontology is a formal explicit specification of a shared conceptualization of a domain [8]. Ontology plays a backbone for meaning-centered reconfiguration of syntactic structure, which is one aspect of semantic technology [9]. There are several languages are available to develop ontology like Developing Ontology-Grounded Methods and Applications (DOGMA) [10], Frame Logic(F-Logic), LOOM, Web Ontology Language (OWL). In this paper we used OWL [11].

3.1 Ontology Web Language

The OWL (Web Ontology Language) is designed for the purpose of those applications that need to process the content of information instead of just presenting information to the humans. Ontology Web Language is divided in three main types [12] which are:

3.1.1 OWL-Lite

OWL Lite is the easiest version of Owl family. It is easy to learn and write but it restricts the expressiveness very much. This is its main disadvantage.

3.1.2 OWL-DL

OWL DL provides the maximum expressiveness possible while retaining computational completeness. It also permits efficient reasoning support. The disadvantage is that it loses full support to RDF.

3.1.3 OWL-Full

OWL Full include every OWL language primitive. it has so many functionalities that it become difficult to use. Due to the intermediate expressiveness and ease of use we are using OWL-DL for developing the Cloud-Ontology.

3.2 Proposed Cloud Ontology Framework

Figure 1 shows a diagrammatic view of our proposed model. There are two interfaces available one for programmer and another for user to interact with the system.

Different modules of the framework are explained below:

3.2.1 Browser

Web Browser acts as an interface for both User and developer. User can query using Manchester OWL syntax [13] and developer can use SPARQL and Manchester OWL syntax both.

3.2.2 Database

All the information and queries are stored in database. The stored information is used to answer the queries. Changes done by developer simultaneously updates in the database.

3.2.3 OWL Ontology

In OWL Ontology part the ontology rules are developed. These rules define relations between different ontology classes.

3.2.4 Management Services

In management services rules are divided in different categories like billing, data usage, payment. Answer of user queries and to manage ment of entire cloud infrastructure is depends upon these rules.

4. DEVELOPED ONTOLOGY

Cloud Ontology defined in this paper scaled a most basic framework of cloud in terms of ontology. This ontology can be modified further according to requirement by either importing the required ontologies or by custom development. There are several tools available for developing the ontologies like OntoEdit [14], WebODE [15], Protégé [10] etc. In this work, Ontology is developed in Protégé 4.16.

4.1 Ontology Classes

Some of the classes which are created in process of cloud ontology development are discussed below:

4.1.1 Cloud

This class represents the services of cloud. It has three subclasses which are: Iaas (Infrastructure as a Service), Paas (Platform as a Service) and Saas (Software as a service).

4.2 Hypervisor

Hypervisor Class shows different hypervisors that are available on cloud. Hypervisor class has two sub classes: Full Virtualization and Para Virtualization. These sub classes are further sub divided in sub classes like VMware and Xen-hypervisor.
4.3 Operating System
This particular class deals with operating systems choices available on the cloud for the users. It has sub classes like Windows, Ubuntu, Fedora, Linux etc.

4.4 Preference
This is a special class that deals with the SLA [17] (service level agreement) part. Here services are divided in three different categories which are sub classes of preference class and these categories are “High”, “Medium” and “Low”. “High” category services are those services for which user pay highest amount of fee. These services are come with no or very little downtime. Similarly “Medium” category provides same services with little bit high downtime or fewer options with fewer fees. Same hierarchy is followed in “Low” category.

Below given code represents RDF/XML [18] form of the “HIGH” class and its relationship with the other classes.

```xml
<owl:InverseFunctionalProperty>
  <owl:Class rdf:ID="High">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#hasOperatingSystem"/>
        <owl:someValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#Fedora"/>
              <owl:Class rdf:about="#Linux"/>
              <owl:Class rdf:about="#Macintosh"/>
              <owl:Class rdf:about="#Unix"/>
              <owl:Class rdf:about="#Windows"/>
            </owl:unionOf>
            </owl:Class>
        </owl:someValuesFrom>
        </owl:Restriction>
        </owl:equivalentClass>
    </owl:Class>
  </owl:equivalentClass>
</owl:InverseFunctionalProperty>

RDF/XML Format of High Class

4.5 User Account
As the name shows this particular class takes care of different costumer’s personal data. Sub classes like billing, data usage and payment which are user specific are also the part of User Account class.

4.6 Machine
This class defines two sub classes Hardware Machine and Virtual Machine.

4.7 Storage
This class represents the storage type available for users to choose on cloud.

All the above mentioned classes are combined by certain set of ontological rules. These rules help in setting a relationship in between these classes which further helps in executing queries.
5. WEB IMPLEMENTATION

Developed framework of Cloud Ontology is implemented using standalone implementation. “Web Protégé”[19] is the browser which we used to implement a desktop based version of the cloud ontology. MySql is used as the backend database for the Cloud Ontology. Web Protégé is used because of its capability to provide interface for users as well as programmers. Protégé allow users to execute Semantic based queries. Developers can execute SPARQL [20] queries too using Protégé. These queries are developed using Manchester OWL syntax and very easy to write. These queries are similar with the natural language queries.

5.1 User Queries

Two examples of Manchester Owl Syntax Queries are shown below:

5.1.1 Query 1
If a user executes a query like “A cloud Service which has virtual machine that has only Para-virtualization Hypervisor “it will look like in Manchester owl syntax “hasVirtualMachine only paravirtualizationHypervisor” and the answer is shown in equivalent classes tab as “PaasCloud” as shown in Figure 5.

5.1.2 Query 2
Similarly there can be a query like “That preference which allow to use Operating systems Fedora, Linux, Macintosh, Unix, Windows, either one of them or all of them” it look like in Manchester owl syntax as “hasOperatingSystem some ( Fedora or Linux or Macintosh or Unix or Windows)” result corresponding to this query is “High” as shown in Figure 6.

CONCLUSION

This paper presented an ontology based flexible cloud framework that can respond to the semantic queries created using Manchester OWL Syntax. This framework provides solution for the increasing complexity of cloud by showing that how Cloud development in ontology is efficient and fulfills the future requirements. There are several ontologies available for improving the cloud, for example for resource management, service discovery etc. that can be imported into this framework. Also it gives user a better interface to interact with cloud by using Manchester OWL Syntax. In future we will try to further extend this framework and implement it on a live environment.

6. REFERENCES


[2] Semantic Web Available at: http://www.w3.org/standards/semanticweb/


[16] Protégé Available at: http://protege.stanford.edu/


[20] SPARQL Available at: http://www.w3.org/TR/rdf-sparql-query/