State of art ontological infrastructure for Cloud Computing

Dr Farheen Siddiqui[1]
Dept. of computer science Jamia Hamdard, New Delhi

Abstract: *During the past few years, cloud computing has become a key IT buzzword. Although the definition of cloud computing is still "cloudy", the trade press and bloggers label many vendors as cloud computing vendors, and report on their services and issues. Cloud computing is in its infancy in terms of market adoption. However, it is a key IT megatrend that will take root. Though the cloud computing has great features such as on demand self service, resource pooling and rapid elasticity there is a shortage in cloud computing because it hasn't provided a unique semantic ground as that of Semantic Web . So, in order to provide an environment for automatic searching of services, resources ontology is used. These are some of the areas where Ontology is used in Cloud Computing. Ontology based registries are used for dynamic discovery of cloud computing resource across various cloud computing platforms, It can be used to provide intelligent customization framework for Saas and easing the design of security system by proving role based access control using ontology.*

Keywords: *Ontology, cloud computing, SaaS*

I. INTODUCTION

Perhaps the simplest working definition of cloud computing is "being able to access files, data, programs and 3rd party services from a Web browser via the Internet that are hosted by a 3rd party provider"[9] and "paying only for the computing resources and services used". Often cloud computing is used synonymously, inaccurately in my view, with such terms as utility computing (or on-demand computing), software as a service (SaaS), and grid computing. Of these, utility computing and SaaS are merely two of several forms of service cloud computing can provide[5]. Grid computing is simply one type of underlying technologies for implementing cloud computing. The term "cloud" in cloud computing is used synomously with "data center"[6]. Today the computing field is able to envision transitioning into the cloud computing era because of the breath-taking advances in computing and information technologies during the past three decades. The advances include the buildup of the Internet backbone, the widespread adoption of broadband access to the Internet, the powerful network of servers and storage in data centers, the advances in high performance and scalable software infrastructure for the data centers and the Web, etc. The architecture of a cloud includes several key modules [Wikipedia]: user interaction interface, system resource management module with a services catalog, and resource provisioning module. The system resource management module manages a massive network of servers running in parallel. Often it also uses virtualization techniques to dynamically allocate and deallocate computing resources. Intelligent Ontology based registries are used for dynamic discovery of cloud computing resource across various cloud computing platforms, It can be used to provide intelligent customization framework for Saas and easing the design of security system by proving role based access control using ontology.

II. ONTOLOGY BASED INTER CLOUD DIRECTORIES AND EXCHANGES

Intercloud Directories and Exchanges are used to provide effective connectivity and collaboration among disparate cloud providers.This mechanism has cloud catalog which uses ontology to automate an environment whereby software agents discover and consume services. This helps to reduce the n2 complexity resulting in one-many and many-many models.

A. Inter Cloud Topology

One cloud instance must be able to dialog with another cloud instance , which for a particular inter-operability scenario is ready, willing, and able to accept an interoperability transaction , exchanging whatever usage related information which might have been needed as a pre-cursor to the transaction. XMPP[2] is used as an inter cloud protocol for transport and using Semantic Web techniques such as Resource Description Framework (RDF) as a way to specify resources.

Inter Cloud Root : This is analogous to DNS system containing services such as Naming Authority, Trust Authority , Directory Services and other root capabilities.[1]
Inter Cloud Gateway : This provides the mechanism of supporting the entire profile of internet protocols and standards.
Inter Cloud Exchange: This facilitates and mediates the initial negotiating process among clouds.
Resource Catalogs : This catalog provides abstract view of computing resources across disparate cloud platforms.

Intercloud Root instances[1] will host the root XMPP [2] servers containing all the presence information for Intercloud Root instances , Intercloud Exchange instances and Internet visible Intercloud capable Cloud instances. Intercloud Exchanges will host second-tier XMPP servers. Individual clouds will in turn use the resource catalog to match the resources. To accomplish this, one uses Ontology for normalizing meaning across terminology or properties.

B. Ontology Based Cloud Computing Resource Catalog

Since many cloud providers advertise their resource capabilities in the cloud computing resource catalog, management requires careful planning to achieve business objectives and avoid errors. In order to achieve this, semantic web technologies are used in service registries such as UDDI. The taxonomy called tmodel was used in UDDI which served as a proxy for technical solution that lived outside the registry. Taxonomy indicates only class/subclass relationship whereas the ontology based resource catalog describes a domain completely. Ontology helps us to have very accurate description of services by their ability to define properties for the class. Thus this catalog helps us to capture the computing resources across all clouds in terms of " Capabilities", "Policies " and "Structural Relationships".

In order to ensure that the requirements provided by the intercloud enabled cloud provider matches with the infrastructure capabilities in an automated schematic fashion there is a need for declarative semantic model which captures the requirements and constraints of computing resources. So, the ontology based semantic model uses RDF/ OWL to capture the features and capabilities in cloud providers infrastructure. These capabilities are logically grouped together and exposed as standardized units for provisioning and configuration to be consumed by another cloud provider. These are then associated with other policies and constraints for ensuring compliance to access the computer resources. The following represents the ontology structure of such high level semantic model.

C. Sample Code

RDF is used to define classes.Classes are allowed to inherit properties of parent class.Each user defined class is implicitly a sub-class of OWL:Thing. The hierachy of user defined classes in this ontology scheme is Resource Capability ->Cloud Domain Capability->Cloud Capability->Tier Capability->Capability Bundle. The following demonstrates a working language called N-triple which is used to describe ontology semantic model.This is much more user friendly and can be easily converted to RDF using a code converter tool.Below is the sample of N-triple code snippet.

Step 1: In our ontology example, “CloudDomain” is an instance of class “CloudDomainCapability”. It consists of two resources “Cloud1”, “Cloud2”

```
<http://cloud/domain>
<http://ww/csp/resOntology#hasCapability>
<http://cloud/domain/#cloud.l>

<http://cloud/domain>
<http://ww/csp/resOntology#hasCapability>
<http://cloud/domain/#cloud.2>

<http://cloud/domain>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#CloudomainCapabilit
y>.

<http://cloud/domain>
<http://www.w3.org/2000/01/rdf-schema#label>
"Cloud Computing Domain"
<http://www.w3.org/2011/XMLSchema#String>.
```

D. SPARQL

SPARQL (SPARQL Protocol and RDF query language) is very powerful SQL like language for querying and making semantic information machine process-able. The Structure and example of a SPARQL Query is illustrated below:

```
Structure:

PREFIX: Prefix definition (optional)
SELECT : Result form
FROM: Data sources (optional)
WHERE : Graph pattern (-path expression)
•  FILTER
•  OPTIONAL
Example

PREFIX geo : http://www.geography.org/schema.rdf#
SELECT ?X ?Y
FROM http://www.geography.org
WHERE { ?X geo:hasCapital ?Y.
   ?Y geo:areaCode ?Z }
ORDER BY ?X
```

III. ONTOLOGY BASED INTELLIGENT CUSTOMIZATION FRAMEWORK FOR SAAS

The forms of service that cloud computing provides today may be broken down into managed services, SaaS, Web services, utility computing, and platform as a service (PaaS) [7]. The ideas behind these forms of service are not new. The fact that the users can tap into these services from Web browsers via the Internet makes them "cloud" services. A managed service is aimed at delivering an application to an enterprise, rather than to end customers directly. This form of service has been available for a decade. There are many types of managed services provided via the Internet, including virus-scanning services for email, spam-filtering services (Google/Postini, etc.), security services (SecureWorks, IBM, Verizon, etc.), desktop management services (CenterBeam, Everdream, etc.) [8]. There are numerous SaaS vendors (formerly known as application service providers). They run a single application in a data center, and deliver the functionality via the Internet to the users. Enterprise SaaS vendors include salesforce.com (for sales force applications), Oracle/Siebel (CRM applications), Workday (for ERP applications), Citrix (meeting applications), and numerous others. SaaS desktop applications for end users, include Google Apps, Zoho Office, Microsoft WindowsLive, etc. Google Apps include Gmail, Google Talk, Google Calendar, Google Docs (documents, spreadsheets, presentations, collaboration), etc. Microsoft Windows Live includes Hotmail, Messenger, and Photo Gallery. Internet portal sites, Internet search engines, and Internet social networking sites are essentially SaaS vendors for end customers. Web services are similar to SaaS. Web service providers offer APIs that application developers can use in developing applications. Examples of Web services APIs include Google Maps, ADP payroll processing, the U.S. Postal Service, Bloomberg, credit-card processing services, etc. [10]. Many players have recently started to offer computing resources, that is, virtual servers and storage as utility computing service. These include such heavyweights as Amazon, Sun Microsystems, IBM, and AT&T; and new vendors such as Nirvanix, Hatsize, Joyent, Cloudworks, etc [11, 12]. Amazon offers Amazon Web Services (AWS), which today consists of Simple Storage Service (S3), Elastic Compute Cloud (EC2), Simple Queuing Service (which uses S3), and SimpleDB. Amazon charges 15 cents a month per gigabyte of S3 storage, and 10 to 80 cents per hour for EC2 server. Amazon has 370,000 AWS customers. Amazon offers a 10% credit if S3 availability falls below 99.9% in any month. Many startup SaaS vendors provide their services on AWS . Some vendors offer entire virtual desktops with the look and feel of Microsoft Windows to those who cannot afford real desktops. Examples are Desktop Two, Zimdesk, GOPC, and Sun Microsystems' Secure Global Desktop. These virtual desktops come with open source Web applications that run with Flash or Java . PaaS also is a variation of SaaS. PaaS delivers an application development environment (platform) as a service, typically with computing resources for hosting the applications developed on the platform. Vendors include salesforce.com (Force.com), Coghead, Google (Google App Engine), Yahoo (Pipes), and Dapper.net. Google's application development platform, Google App Engine, lets application developers develop Python-based applications, and hosts the application at no cost with up to 500 MB of storage.

Software as a Service (SaaS) with multi-tenancy architecture (MTA) is a model for software delivery whereas software provider publishes a copy of their software on the Web to support multiple tenants or customers in a cloud environment. MTA allows multiple tenants to share a software service with customization [3]so that each tenant may have its own GUI, data,and user interaction. As a consequence, the SaaS software may appear to each tenant as if it is the sole tenant (e.g., keeping confidential data private), while allowing multiple tenants to use the same software (to achieve economy of scale). The SaaS customization need to meet multiple goals : SaaS providers need to support tenants/ customers with a multitude of options and variations using a single code base, such that it is possible for each tenant to have a unique software configuration. Also they need to ensure that the configuration is simple and easy to satisfy tenants' specific requirements without extra development or operation costs. This customization is not only related to functionality but also to the Quality of Service (QoS). A fully customizable SaaS application has a layered architecture.

The framework has four layers:

- Data Layer
- Service Layer
- Business Process Layer
- User Interface Layer

A. DataLayer

In a domain, multiple data ontologis systems can be defined by different communities. Ontology integration is developed to solve these heterogeneity, which refers to build a larger and complete ontology at a higher level using existing ontology systems. An ontology is represented as a tree. Here, key words with similar meaning are given the value "1". In a knowledge context, conceptual similarity is determined. Data layer customization is guided using ontology information. After searching for domain ontology, the template is found and customized using ontology. The ontology semantic information can be matched to database logic designs. The domain objects can represent a large proportion of meta-data. Multiple database schema can be used in MTA, such as XML, sparse table or views.

B. Service Layer

Services can be either simple or composite. Atomic service is a basic service that does fundamental operations. Composite services is made up of related atomic services to do complex tasks. Atomic service customization can be carried out by understanding the unique parts in a description. Every service represents a capability under some terms and conditions. Complicated task can be achieved by ontology. Automated reasoning machinery is developed to address difficult tasks such as automated web service discovery, semantic translation and automated web service composition.

C. Business Process Layer

In this layer, the services and participants have been organized to achieve more complex business tasks,workflows, which consist by a set of activities and represent business processes. Tenants can search a workflow repository using keywords and retrieve the relevant ones according to their interests. The customization process is based on the business domain knowledge in multi-layered workflow with a series of steps or transformation from template objects. Datapass information through flows, and act as the step transformation conditions.

D. User Interface Layer

A UI ontology can be built to provide the concepts, relationships, reasoning, and searching for UI-related elements. The ontology should include UI classification information includes: data collection, data presentation, monitoring, command-and-control, and hybrid (combination of two or more types). Much easier UI customization is to change and configure the appearance and the UI available to the users, including adding/editing/deleting the icons, colors, fonts, titles in pages, menus and page sections.

Using Ontology for Role Based Access Control

IV. **Define Roles with Semantic Information** : According to the semantic in a specific domain, roles are defined as a combination of official positions and job functions. For example, in an IT company, official positions may include the CEO, CFO, Regional Manager, Project Manager, etc. Functions represent the user's daily duties. And based on this the user is assigned various access control mechanisms to work on the data and resources. One employee can be assigned to one or more roles.[4]

Manage Roles Hierarchy with Ontology: In a domain, multiple possible role hierarchies are defined by ontology systems from different communities. For example, ACM and IEEE are two large communities and each has its own standards and practices, and they are similar but distinct. To solve these heterogeneities, we need to build a larger and complete ontology at a higher level using existing ontologies. Thus ontology is represented as a tree.

Role Hierarchy: An unordered and labeled ontology tree is a tuple OT = (V, E) where V is a finite set of nodes, E is a set of edges connecting the edges where $E \subset V \times V$, representing the relationship between nodes. If $(u, v) \in E$, u is the parent of v, denoted as u=parent (v) and v is the child of u, denoted as v=child (u).

Role Numbers and ensure Scalability: The total number of possible roles is the product of every category dimension. However, the actual number of roles is a subset of these combinations, because some roles will not be needed.

IV. CONCLUSIONS

This paper we provide the details about use of ontology in cloud computing. And also the paper discusses some of the methods for cloud ontologies. This is an effort towards a detailed study about the ontology and how to enhance the use of ontology in the private cloud as well as public cloud. Ontology plays a very important role in the cloud computing technology by consolidating view of computing resources present across disparate clouds , providing the ability to protect sensitive information from unauthorized access , customization of Saas and by improving the overall efficiency. Intelligent Ontology based registries are used for dynamic discovery of cloud computing resource across various cloud computing platforms, It can be used to provide intelligent customization framework for Saas and easing the design of security system by proving role based access control using ontology.

References

[1] Using Semantic Web Ontology for Intercloud Directories and Exchanges by David Bernstein and Deepak Vij
[2] http://xmpp.org/xmpp-protocols/rfcs/
[3] OIC: Ontology-based Intelligent Customization Framework for SaaS by Wei-Tek Tsai , Qihong Shao∗, Wu Li
[4] Role-Based Access-Control Using Reference Ontology in Clouds by Wei-Tek Tsai, Qihong Shao
[5] John Foley: "Private Clouds Take Shape", *InformationWeek*, August 9, 2008.
[6] John Foley: "20 Cloud Computing Startups You Should Know", InformationWeek, September 20, 2008.
[7] Galen Gruman: "What Cloud Computing Really Means", *InfoWorld*, April 7, 2008. (Also in *The New York Times*, with Eric Knorr)
[8] Jaesun Han: "The Future of Cloud Computing and Serverplatform", http://www.web2hub.com/blog/82 (in Korean) February 1, 2008.
[9] Steven Hodson: "What Is Cloud Computing?", http://www.winextra.com/2008/05/02/what-is-cloud-computing, May 2, 2008.
[10] J. Nicholas Hoover and Richard Martin: "Demystifying the Cloud", *I nformationWeek Research & Reports*, pp. 30 -3 7, June 23, 2008.
[11] J. Nicholas Hoover: "Outages Force Cloud Computing Users To Rethink Tactics", *I nformationWeek*, August 16, 2008.
[12] "Amazon's S3 Cloud Service Turns Into A Puff of Smoke", *I nformationWeek NewsFilter*, August, 2008.