A CLOUD COMPUTING AGAINST UNIFIED ONTOLOGY

Nimrat Kaur
M.Tech (C.S.E)
Sai Institute of Engg. & Tech.

Gagandeep Singh
Assist. Prof.
Sai Institute of Engg. & Tech.

Manpreet Kaur
M.Tech (C.S.E)
GIMET, Asr

Abstract: *The emergences of cloud computing deals in various internet applications developers for hosting various applications which users are in need in this day to day life. The idea is cloud ontology for easy Selection, Publication and Discovery in cloud services. Query Processing Agent, Ontological Similarities. In this paper, proposal is an cloud service discovery system with ontological model, in order to solve, for example: A job site, It's not an ordinary jobsite but focusing and specializing in matching the users or clients queries related to jobs and displaying the information even by giving options of updating their information which is needed for this Job Domain. This paper implements the cloud ontology technique to make cloud service discovery system efficient for user query in job site. The aim of this paper is to determine how the CSDS with the Cloud ontology achieved better performance than the CSDS without the Cloud ontology. By consulting a Cloud ontology to reason and to rating about the relations among Cloud services, the CSDS is more successful in locating Cloud services and more likely to discover Cloud services that meet consumers " requirements.*

Keywords: ***Cloud computing, Cloud ontology, Cloud service discovery system (CSDS)***

I. INTRODUCTION

"Cloud computing" is the next natural step in the evolution of on-demand information technology services and products[4]. Cloud computing is Internet (Cloud) based development and use of computer technology (computing) whereby dynamically scalable and often virtualized resources are provided as a service over

the Internet [3]. Cloud computing can be considered a new computing paradigm that allows users to temporary utilize Computing infrastructure over the network, supplied as a service by the cloud-provider at possibly one or more levels of abstraction. The introduction of an organization of the cloud computing knowledge domain, its components and their relations – i.e. an ontology – is necessary to help the research community achieve a better understanding of this novel technology [5].

2. CLOUD ONTOLOGY

Ontology is a conceptual schema about a domain. Ontology provides Meta information which describes data semantics [3]. Ontology contains a set of concepts and relationship between concepts, and can be applied into information retrieval to deal with user queries.

In Cloud computing, Clouds are generally divided into three different levels (*IaaS, PaaS, and SaaS)*

Infrastructure as a Service: Virtual machines (VMs) are the most common form for providing computational resources to cloud users at this layer, where the users get finer-granularity flexibility. This layer allows the users unprecedented flexibility in configuring their settings while protecting the physical infrastructure of the provider's data center.

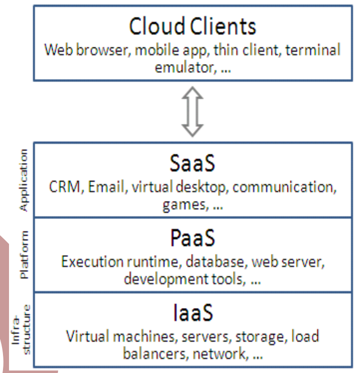


Fig (1): Cloud Ontology

Platform as a service: In the PaaS model, cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server.

Software as a Service: It delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model.

3. PROBLEM STATEMENT

Cloud computing providers publish Cloud services over the Internet, and consumers normally access these services provided by Cloud application layer through web-portals [3]. To date, however, there is no discovery mechanism for searching different kinds of Clouds [7]. Even though there are many existing generic search engines that consumers can use for finding Cloud services, these engines may return URLs containing not relevant web-pages to meet the original service requirements of consumers. Intuitively, visiting all the web-page can be time-consuming job. Whereas generic search engines such as Google, MSN are very effective tools for searching URLs for generic user queries, they are not designed to reason about the relations among the different types of Cloud services and determining which service(s) would be the best or most appropriate service for meeting consumers service requirements. Hence, service discovery mechanisms for reasoning about similarity relations among Cloud services are needed.

4. A CLOUD SERVICE DISCOVERY SYSTEM

A Cloud service discovery system is a specially designed for cloud users who want to find a Cloud service over the internet or to support them in finding for Cloud services more efficiently. By interacting Cloud ontology, the CSDS attempts to recognize an appropriate Cloud service among a list of many services. When a user requests to find Cloud services with their specific requirements, the CSDS returns the best service and also recommends other services for the user.

The goal of this project are 1) to develop a CSDS 2) to design and construct a Cloud ontology and 3) a Cloud service reasoning agent for reasoning about the relations among Cloud concepts by consulting the Cloud ontology[3].

A Cloud service discovery system (CSDS) consisting of a search engine and three different agents, Query Processing Agent, Filtering Agent, and Cloud Service Reasoning Agent (CSRA).

5. SYSTEM ARCHITECTURE

(a) Query Processing Agent: The information kept in the query processing Agent is useful for the users or customers. Customers are able to identify the required services, Resources, and Providers by querying in the query processing Agent. The QPA locates information sources by executing conventional search engines. Depending on the result from executing conventional search engines, a number of relevant information sources are selected based on the desirable number of URLs specified by a user. The default search engine is *Google*. The customers submit the queries, matching attributes from services has been filtered.

(b) Information Filtering Agent: To relieve users of time consuming and laborious tasks of surfing many websites during an information retrieval process [3], IFAs are used to browse multiple websites concurrently. Instead

Of the user visiting one website at a time, several IFAs can be deployed to retrieve the required information from different websites simultaneously [2]. IFA will search for 1) synonymous keywords, 2) more specialized keywords, and 3) more general keywords, by this, the content of the web page can be validated by scanning for exact match of keywords in the query.

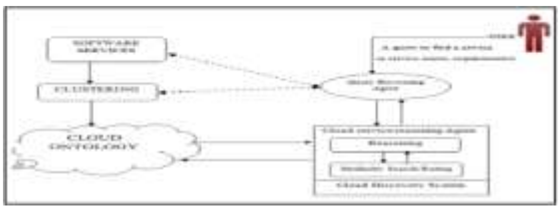


Fig (2): System architecture of CSDS

(c) A Cloud service reasoning agent: It consults with the Cloud ontology to reason about the relations among Cloud services. There are three reasoning methods to determine similarity between and among services. The cloud service reasoning Agent depends on mainly two concepts

1. Similarity Search (Reasoning)
2. Rating.

The cloud service agent helps to perform those functions.

Reasoning: Reasoning is needed to understand a service and its similarity among a list of services. It makes a difference of the service similarity by calculating the similarities between the user input and services [8]. A CSRA consults with cloud ontology to determine the relations among cloud services.

For Example: Similarity reasoning is a method to calculate similarity between two services[1]. One of the reasoning methods considered for this paper is Similarity Reasoning. This helps the easy selection and to locate the materials which the queries submitted by the customers.

We can calculate the similarity between two concepts 'a' and 'b' can be calculated using the following formula[6]:

$$\textbf{sim (a, b)} = \alpha\ \textbf{simcon (a, b)} + \beta\ \textbf{simobj (a, b)} + (1-\alpha-\beta)\ \textbf{simdata (a, b)}$$

α and β are the weights of each clause and $0 \leq shim(a,b) \leq 1$.

This formula includes the following similarities:

1. Concept similarity
2. Object property similarity
3. Data type property similarity

Concept Similarity: Concept similarity computes the similarity among concepts which have the same meanings but are referred to differently.

$$\textbf{simcon (a, b)} = |\textbf{Super (A)} \cap \textbf{Super(B)}| / |\textbf{Super(A)}|$$

Where,

A, B – the most specific concepts of individuals a and b

Super (A) and Super (B) – the sets of all reachable super concepts from the concepts A and B.

Steps to determine the Concept similarity:

Step 1: Consider the two individuals for which the concept similarity is to be calculated, 'a' and 'b'.
Step 2: Count all the reachable super concepts from each of the specific concepts, 'A' and 'B', of the Individuals 'a' and 'b'.
Step 3: Determine the commonly reachable super concepts of 'A' and 'B'.
Step 4: Divide the result of step 3 by the result of step 2 (for individual 'a').
Object Property Similarity: Object properties represent the meanings of concepts and also provide the relationships among concepts in ontology.
simobj (a, b) = _(x,y)_ U Shim (x,y) / |O(a)|
Where,
U= { (x,y) | (a,p,x) _ O(a), (b,p,y) _ O(b)}

O (a) – a set of triples containing the object properties of the individuals a and b

U – The set of object values, with common predicate p of individuals a and b in each triple O(a) and O(b)

Steps to determine object property similarity

Step1: Consider the two individuals for which the concept similarity is to be calculated, 'a' and 'b'.

Step2: Identify O (a) and O (b), which are the sets of triples containing the object properties of individuals 'a' and 'b' respectively.

Step 3: Determine | O (a) |, which is the count of such sets.

Step 4: Identify U as the set of object values with common predicate 'p' of 'a' and 'b' in O (a) and O (b) respectively.

Step 5: Determine sim(x, y) for all object values identified in step 4.

Step 6: Make a summation of all the individual results obtained in step 5.

Step 7: Divide the result of step 6 by the result of step 3 (for individual 'a').

Data type property similarity: These properties represent the context of concepts and they are the attributes of 'concepts' in the ontology.

simdata (a, b) = _(x, y, p)_V Comp (x,y,p) / │D(a) │

Where,

V = {(x,y,p) │(a,p,x) _ D(a), (b,p,y) _ D(b)}

Comp(x,y,p) = 1 – (│x-y│) / Max distance (x,p)

Max distance (x,p) = max **i_I(p)** (│x-i│)

I (p) = {i│(s,p,i) _ Ontology}

D (a) – a set of triples, which contains the datatype properties of individual 'a'

V – A set of datatype values, with common predicate p of individuals a and b

Steps to determine object property similarity

Step1: Consider the two individuals for which the concept similarity is to be calculated, 'a' and 'b'.

Step 2: Identify D (a) and D (b), which are the sets of triples containing the data type properties of individuals 'a'

and 'b' respectively.

Step 3: Determine | O (a) |, which is the count of such sets.

Step 4: Identify V as the set of object values with common predicate 'p' of 'a' and 'b' in D (a) and D (b) respectively.

Step 5: Determine Comp(x, y, p), which is the similarity between the data type values 'x' and 'y' over the Common predicate 'p'.

Step 6: Determine the maximum reachable distance based on 'x' value.

Step 7: Using the result of step 6 and the distance between 'x' and 'y', determine the similarity between the

Data type values of 'a' and 'b'.

Rating: Once the similarity between services is calculated, the proceeding will be to calculate the service utility, based on which the services are ranked. This service helps of the CSRA (Cloud Service Reasoning Agent).This leads to the process the rating of information, according to the users‟ queries and finally it has been published with the rating results. A The customers or users can justify according to the rating results and they can choose the needed information which they got as the reply for their queries which is highly rated or according to the users needed interest and expectation. The rating will be accurate. This helps in the success rate of the rating concept in the cloud[1].

Evaluation of the system

There are two performance measures 1) Service Utility 2)Success Rate

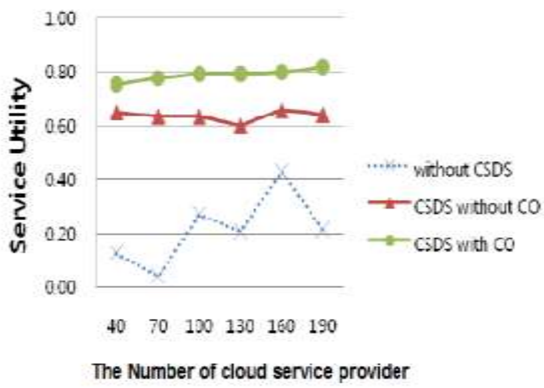
Above two has been compared with Three aspects such as 1) without the CSDS, 2) the CSDS without the Cloud ontology,3) the CSDS with the cloud ontology.

In (1) case, if the selected web-page is about the cloud service, then its service utility can be determined, otherwise service utility can be considered as zero.

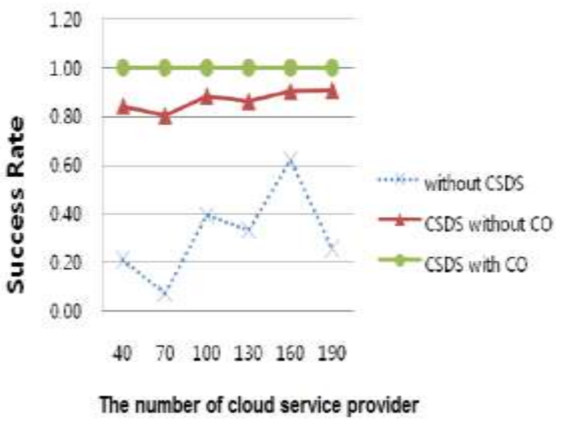
In (2) case, as per condition, the cloud term is not used, if the required web-page is selected through filtered results, the service utility can be determined.

In (3) case, Web-pages are rated by the aggregated service utility which is a result of the CSRA.

Service Utility: In the following Fig (3), it is shown that when CSDS interacts with cloud ontology, it provides the high performance as compare to CSDS without cloud ontology. The reason behind it is that CSDS have filtering and reasoning functionalities which means that web-pages of the Cloud service have higher chance to be selected and is more likely to fulfill the user requirements.



Success Rate: It is assumed that the discovery System fails when the service utility is less than 0.5, but the experimental results shows that Using the CSDS with the Cloud ontology, service utility of retrieved web-pages is well over 0.5. Thus, CSDS with ontological model users are more successful in discovery Cloud services.



CONCLUSION

Cloud service discovery system (CSDS) is a service that helps its users in finding a Cloud service over the Internet.

The contribution of this work is adopting an approach for bolstering web-based information retrieval using a society of agents. It is the first attempt in building an agent-based discovery system that consults ontology when retrieving information about Cloud services. This paper implements the cloud ontology technique to make cloud service discovery system efficient for user query in job site. Its empirical results show that using the Cloud ontology, the CSDS is more successful in finding Cloud services that are closer to users' requirements. The queries asked by the clients are back to clients without any delay.

References:

[1] V.K.Reshma, B, Saravana Belasis "Cloud Service Publication and Discovery Using Ontology."

[2] Kwang Mong Sim and Pui Tak Wong "Toward Agency and Ontology for Web-Based Information Retrieval"

[3] Taekgyeong Han and Kwang Mong Sim* "An Ontology-enhanced Cloud Service Discovery System"

[4] Mladen A. Vouk "Cloud Computing – Issues,Research and Implementations"

[5] "Toward a Unified Ontology of Cloud Computing" Lamia Youseff , Maria Butrico, Dilma Da Silva

[6] S. Bhama, DR. G. R. Karpagam "Realizing the Need for Similarity Based Reasoning Of Cloud Service Discovery"

[7] PHILLIP C-Y SHEU, SHU WANG, QI WANG, KE HAO and RAY PAUL, "Semantic Computing, Cloud Computing, and Semantic Search Engine", 2009 IEEE International Conference on Semantic Computing.

[8] "Intelligent Control and Computer Engineering" Sio-Long Ao,Oscar Castillo,Xu Huang

Nimrat Kaur is pursuing M. Tech in Computer Science & Engineering from Sai Institute of Engineering & Technology. She did her B.Tech degree in Computer Science & Engineering from Baba kuma Engineering College, Amritsar, Punjab, India. Her area of interest is Distributed Database, software Engineering

Er. Gagandeep Singh is working as a Assist. Professor Department of Computer Science & Engineering, Sai Institute of Engineering and Technology, Manawala, Amritsar, Punjab, India. He obtained his B.Tech (Computer Science Engineering) from Sri Sai Institute of Engineering & Technology, Punjab, India, M. Tech (Computer Science & Engineering from ACET, Manawala, Punjab, Presently, India he has 4 years of research and UG & PG teaching experience. He has 7 research papers to his credit in various international journals and conferences. He is member of International Association of Engineers HONG KONG and International Association of Computer Science & Information Technology, SINGAPORE. His areas of research and interest include Web Security, Software Engineering.