An Ontology-enhanced Cloud Service Discovery System

Taekgyeong Han and Kwang Mong Sim*

Abstract—**This paper presents a Cloud service discovery system (CSDS) that aims to support the Cloud users in finding a Cloud service over the Internet. The CSDS interacts with a Cloud ontology to determine the similarities between and among services. The significance of this project is that it is the first attempt in building an agent-based discovery system that consults an ontology when retrieving information about Cloud services. One of the main contributions of this work is building a Cloud Service Reasoning Agent (CSRA) that enables the CSDS to 1) reason about the relations of Cloud services and 2) rate the search results. Another contribution of this work is designing and constructing a Cloud ontology consisting of a taxonomy of concepts of Cloud services that enables the CSRA to determine the relations of Cloud services using three service reasoning methods: 1) Similarity reasoning, 2) Equivalent reasoning, and 3) Numerical reasoning. Whereas empirical results show that using the Cloud ontology, the CSDS is more successful in finding Cloud services that are closer to users' requirements, the proof-of-concept example demonstrates the major functionalities of the CSDS.**

Index Terms—**Cloud computing, Cloud ontology, Software agent, Web information retrieval**

I. INTRODUCTION

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 [1]. Consumers of Cloud computing will not compute their own computer, but move their programs and data to the Clouds consisting of computation and storage utilities provided by third parties. Cloud computing providers publish Cloud services over the Internet, and consumers normally access these services provided by Cloud application layer through web-portals [2]. To date, however, there is no discovery mechanism for searching different kinds of Clouds. Cloud consumers generally have to search for appropriate Cloud services manually [3]. 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 (e.g., Google, MSN, etc) 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. The significance of this work is that to the best of the authors' knowledge it is the earliest effort in constructing a Cloud service discovery system (CSDS) to assist users in searching for Cloud services more efficiently. However, it is noted here that this work is not designed to compete with or to replace existing generic search engines. Rather, the CSDS in this work employs existing search engines as its initial searching mechanism for gathering information about the Websites of Cloud services. Then, by consulting a Cloud ontology, the CSDS attempts to recognize an appropriate Cloud service among a list of several services. When a consumer submit requests to find Cloud services with their specific requirements, the CSDS returns the best service and recommends other services for the user.

The objective of this project are 1) to develop a CSDS (section Ⅱ), 2) to design and construct a Cloud ontology (section Ⅳ), and 3) a Cloud service reasoning agent (section Ⅲ) for reasoning about the relations among Cloud concepts by consulting the Cloud ontology.

Manuscript received January 12, 2010. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-220-D00092).

Taekgyeong Han is with the Gwangju Institute of Science and Technology, South Korea (e-mail: pilotfh@gist.ac.kr).

*Kwang Mong Sim is with the Gwangju Institute of Science and Technology, South Korea (phone: 82-62-715-2235; fax: 82-62-715-2204; e-mail: prof_sim_2002@yahoo.com) (Corresponding Author).

II. A CLOUD SERVICE DISCOVERY SYSTEM

This section illustrates the prototype of 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). In Fig. 1, there are two components, 1) a *CSDS* helps to find the best Cloud service in behalf of users and 2) a *Cloud ontology* which consists of taxonomy of concepts of different Cloud services to consult with the CSRA. In addition, there is a user interface that allows the user to enter queries contains a service name and requirements considered by their preferences.

Query Processing Agent (QPA): The QPA locates information sources by executing conventional search engines. Although the selection of a search engine is arbitrary, the default search engine is *Google (Search API)*. If the number of searched results is fewer than that specified by a user,

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

generate new alternate queries to have more results [4].

Filtering Agent: It is to relieve users of time consuming and laborious tasks of surfing many websites during an information retrieval process [4]. The relevance of web-pages is determined by adopting three heuristics (1) detecting evidence phrases (EP), (2) counting the frequencies of EP, and (3) considering the nearness among keywords [5].

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. Details of functionalities of the CSRA are given in section Ⅲ.

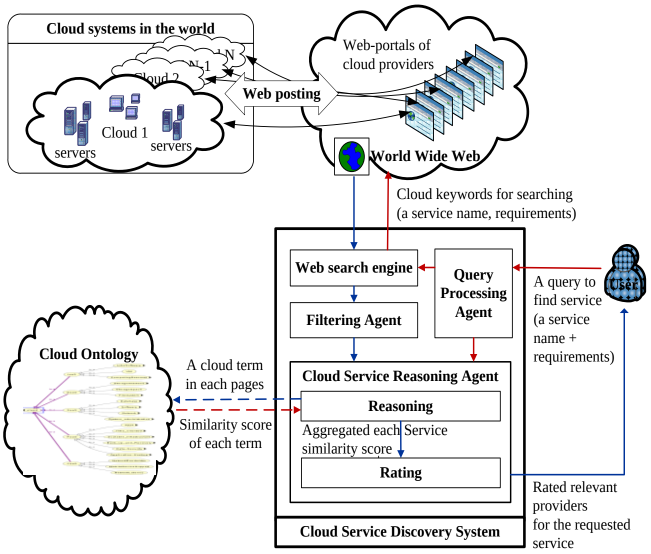


Fig. 1. Cloud Service Discovery System

III. A CLOUD SERVICE REASONING AGENT

A Cloud service reasoning agent (CSRA) carries out two functions: 1) Reasoning and 2) Rating.

Reasoning: A CSRA consults a Cloud ontology for performing service reasoning. All information supplied by a user is used to determine the similarity between two services. There are three methods to determine similarity, 1) Similarity reasoning, 2) Equivalent reasoning, and 3) Numerical reasoning. Because of space limitation, Only similarity reasoning is presented in this paper in section IV. Others will be presented in a future paper.

Rating: An aggregated similarity (i.e., Service Utility) is used to determine the rating as shown in algorithm 1. A web-page which has the highest service utility would be selected as the best service for the user. Other recommended services would be selected as well.

Algorithm 1:
For all filtered results *{Ft(1), Ft(2), Ft(3), …, Ft(N)}*

1. Calculate similarity *q(1)* in user queries *{q (1), q (2), …, q (N)}* with term *t(1)* in the *Ft(N) {t(1), t(2), …, t(N)}*.
   Step 1) Similarity reasoning
   Step 2) Equivalent reasoning
   Step 3) Numerical reasoning
2. If two concepts have the same similarity from the Step 1) Similarity reasoning because they are sibling nodes, then
   do Step 2) Equivalent reasoning.
3. If two concepts are numerical values, then
   Step 3) Numerical reasoning.
4. Otherwise, do Step 1) Similarity reasoning.
5. *From 2, 3, 4, Aggregate Sim(s)* over all terms in the web-page *{t(1), t(2), …, t(N)}.*

   [Aggregation method]

$$ServiceUtility = \sum_{k=0}^{N} term(k) \times weight(k)$$

where *Weight(k) = 1/N* is uniformly distributed.
endFor

6. Rating web-pages used by the *ServiceUtility*.
7. Select a web-page which has the highest *ServiceUtility* as the best Cloud service and other recommendation services as well.

IV. CLOUD ONTOLOGY

Ontology can provide Meta information which describes data semantics [6]. It provides a shared understanding of a domain of interest to support communication among human and computer agents [7]. Ontology contains a set of concepts and relationship between concepts, and can be applied into information retrieval to deal with user queries [8].

In Cloud computing, Clouds are generally divided into three different levels (*IaaS*, *PaaS, and SaaS* [9]) see Fig. 2.

Infrastructure as a Service *(IaaS)* [9] provisions hardware, software, and equipments to deliver software application environments with a resource usage-based pricing model.

Platform as a Service *(PaaS)* [9] offers a high-level integrated environment to build, test, and deploy custom applications. Generally, developers will need to accept some restrictions on the type of software they can write in exchange for built-in application scalability.

Software as a Service *(SaaS)* [9] delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model.

The Cloud ontology in this work represents the relations among Cloud services to facilitate the CSRA in reasoning about the relations between and among Cloud service concepts. It consists of 424 concepts constructed for the service reasoning. There include concepts of Cloud services which are currently being used and many services that may released in the near future. There are three kinds of reasoning methods, 1) Similarity reasoning, 2) Equivalent reasoning, and 3) Numerical reasoning.

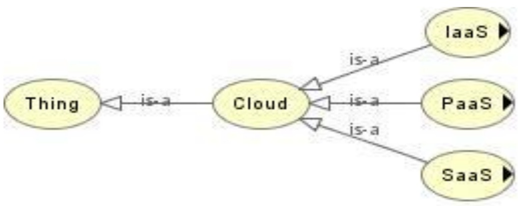


Fig. 2. Cloud three different levels

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

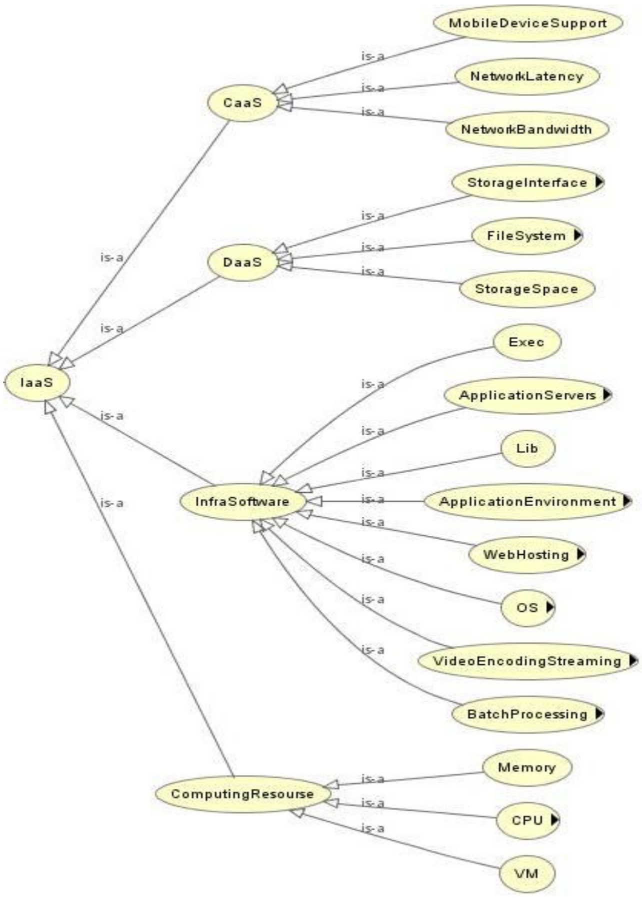


Fig. 3. Cloud Ontology : IaaS

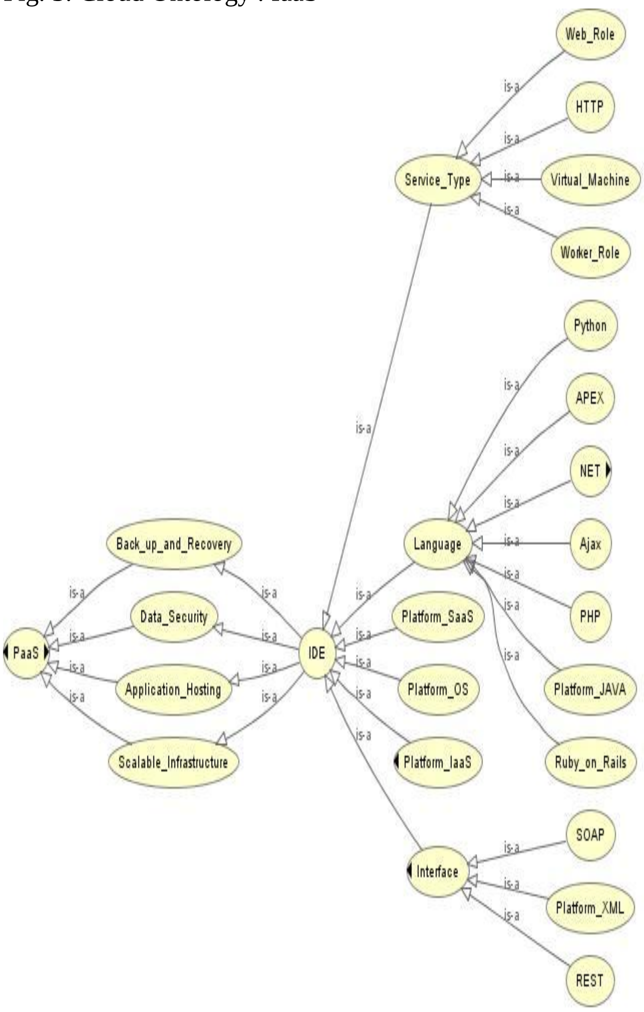


Fig. 4. Cloud Ontology : PaaS

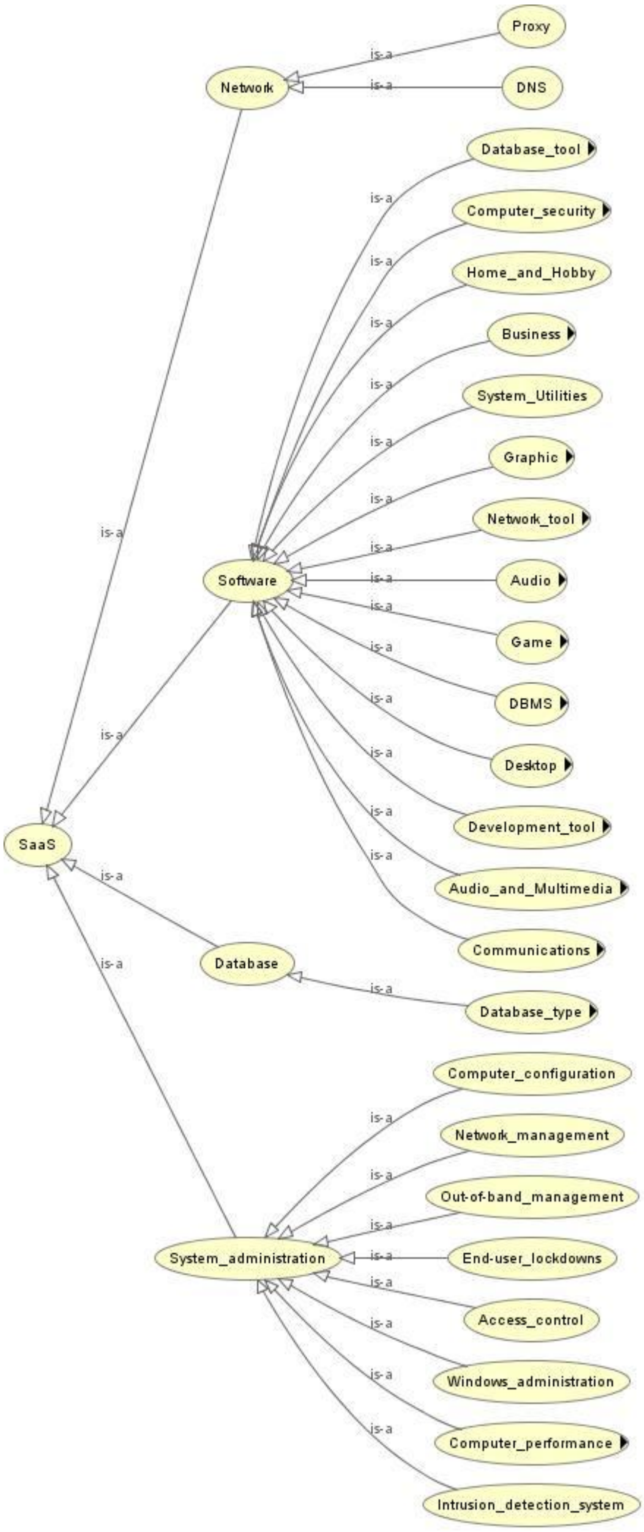


Fig. 5. Cloud Ontology : SaaS

1) **Similarity reasoning**

Similarity reasoning is to calculate similarity between two concepts by counting common reachable nodes. The similarity of concepts represents the degree of commonality between concepts. We compute semantic similarity based on the method in [10] as follows:

$$sim(x,y) = \rho \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1-\rho)\frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|} \qquad (1)$$

where $\rho \in [0,1]$ determines the degree of influence of generalizations.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$\alpha(x)$ is the set of nodes (upwards) reachable from x, we have $\alpha(x) \cap \alpha(y)$ as the reachable nodes shared by x, y, which is an indication of the commonality between concepts x and y [11].

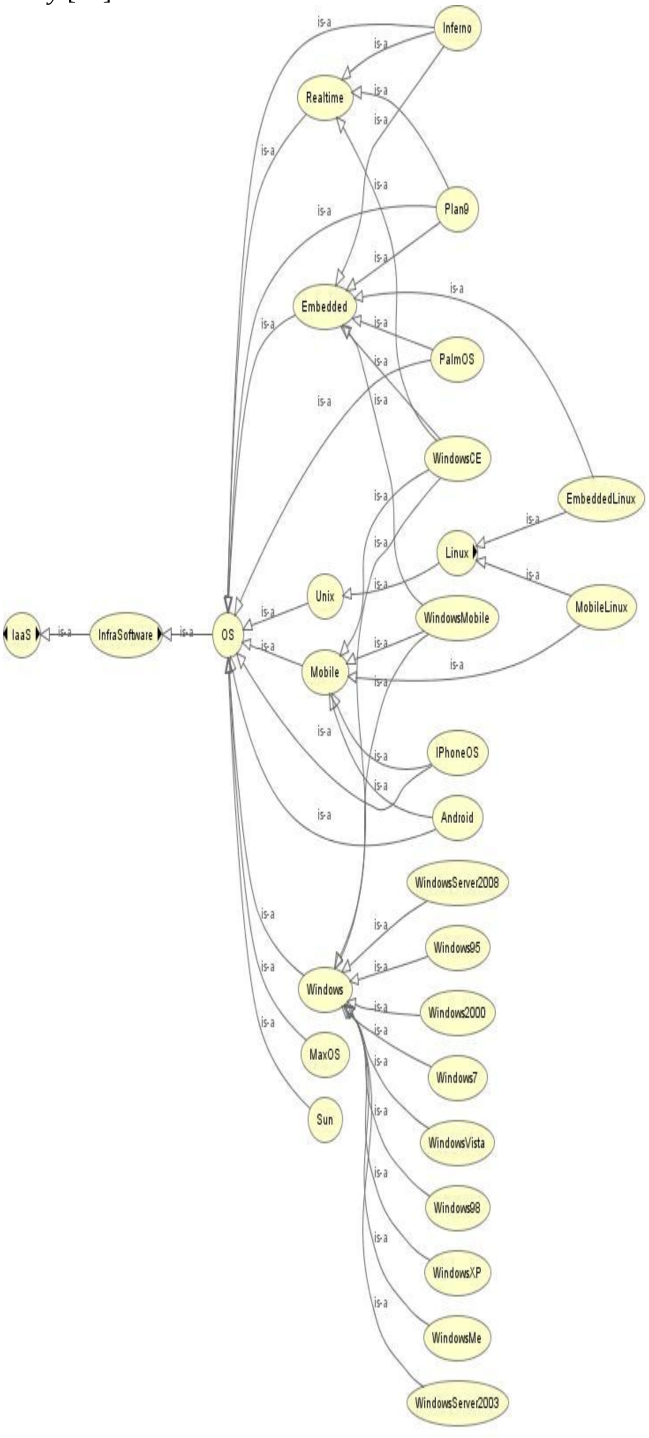


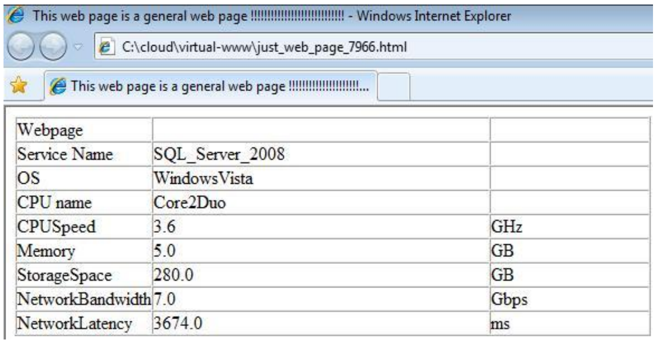
Fig. 6. Relation in terms of OS : IaaS

For example, IaaS - InfraSoftware – OS. In terms of Eq. (1), the concepts UNIX and Windows have 4 reachable nodes (upwards) from themselves, namely,
$\alpha(Unix) = 4$, $\alpha(Windows) = 4$, $\alpha(Linux) = 5$,
$\alpha(Unix) \cap \alpha(Windows) = 3$, $\alpha(Unix) \cap \alpha(Linux) = 4$,
Then, the similarity of $\alpha(Unix) \cap \alpha(Linux) = 4$ is greater than $\alpha(Unix) \cap \alpha(Windows) = 3$.

V. EXPERIMENTATION AND EVALUATION

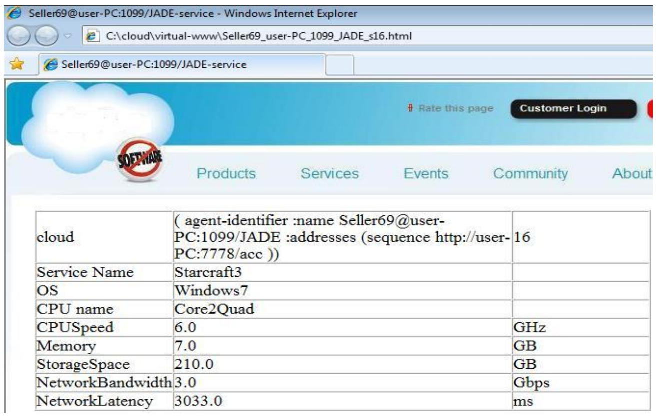
The performance measures are ⅰ) Service Utility (Fig. 9) and ⅱ) Success Rate (Fig. 10), with three comparison schemes, searching a Cloud service 1) without the CSDS, 2) the CSDS without the Cloud ontology, and 3) the CSDS with the Cloud ontology. In case 1), Web-pages are searched with an exact service name and selected a web-page randomly from the searched results. If it is a web-page about a Cloud service, then its service utility is determined. If not, the service utility is assigned as zero which means that the discovery has failed. In case 2), Web-pages which do not include a Cloud term are filtered out from the searched results, and a web-page is selected randomly from the filtered results, and the service utility is calculated. In case 3), Web-pages are rated by the aggregated service utility which is a result of the service reasoning.

For evaluation purpose, we assumed that the WWW replaced by the virtual-www for ease of testing. There are already 10,000 web-pages (not for Cloud services) (Fig. 7) in a directory called the virtual-www and around 30 web-pages (Fig. 8) are automatically generated by each provider when the CSDS is deployed. Depending on the number of providers generated, a total number of web-pages (i.e., Cloud services) would be decided between 11,200 and 15,700 in the virtual-www. The CSDS requires more information consisting of a service name, OS, CPU name and range of values such as CPU clock, RAM, HDD, Network Bandwidth, and Network Latency (see Table Ⅰ).


This web page is a general web page !!!!!!!!!!!!!!!!!!!!!!!! - Windows Internet Explorer
C:\cloud\virtual-www\just_web_page_7966.html

Webpage		
Service Name	SQL_Server_2008	
OS	WindowsVista	
CPU name	Core2Duo	
CPUSpeed	3.6	GHz
Memory	5.0	GB
StorageSpace	280.0	GB
NetworkBandwidth	7.0	Gbps
NetworkLatency	3674.0	ms

Fig. 7. Example of a general web page


Seller69@user-PC:1099/JADE-service - Windows Internet Explorer
C:\cloud\virtual-www\Seller69_user-PC_1099_JADE_s16.html

Customer Login
Products Services Events Community About

cloud	(agent-identifier :name Seller69@user-PC:1099/JADE :addresses (sequence http://user-PC:7778/acc))	16
Service Name	Starcraft3	
OS	Windows7	
CPU name	Core2Quad	
CPUSpeed	6.0	GHz
Memory	7.0	GB
StorageSpace	210.0	GB
NetworkBandwidth	3.0	Gbps
NetworkLatency	3033.0	ms

Fig. 8. Example of a Cloud service page

Service Utility: In Fig. 9, the result of the CSDS with the Cloud ontology shows higher performance than without the CSDS and the CSDS without the Cloud ontology in terms of the service utility. This is because the CSDS have a filtering and a reasoning functionalities which means that web-pages of the Cloud service have higher chance to be selected and is more likely to closer to users' requirements.

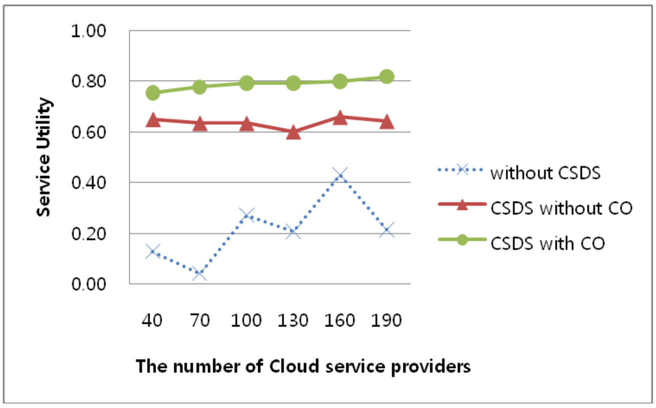


Fig. 9. Service Utility

Success Rate: In Fig. 10, success rate is calculated by the number of successes / the number of attempts. It is assumed that a discovery will fail if the service utility is less than 0.5. Using the CSDS with the Cloud ontology, experimental results show that the service utility of retrieved web-pages is well over 0.5. The results demonstrated that using the CSDS with the Cloud ontology, users are more successful in discovery Cloud services.

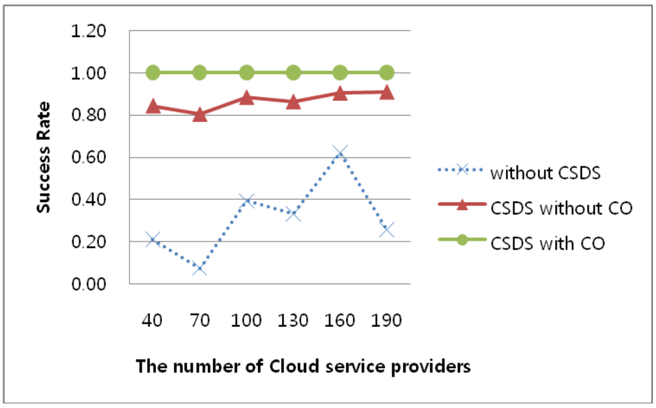


Fig. 10. Success Rate

Table Ⅰ. Experiment settings for simulations

Experiment variables	Value (range)
The number of providers	40, 70, 100, 130, 160, 190
The number of Cloud services provided by each provider	25~35 (web-pages)
The number of Cloud service web-pages in the virtual-www	1200, 2100, 3000, 3900, 4800, 5700 (web-pages)
The number of web-pages in the virtual-www (not for Cloud service)	10,000 web-pages
Total number of web-pages in the virtual-www	11200, 12100, 13000, 13900, 14800, 15700
The number of Cloud services	Around 100 service names
CPU clock	0.1~6.0 GHz
RAM size	0.256~36.0 GB
HDD size	0.1~1000 GB
Network Bandwidth	0.1~10 Gbps
Network Latency	1~5000 ms

VI. PROOF-OF-CONCEPT EXAMPLE

An example is in this section given to demonstrate functionalities of the Cloud service discovery system (CSDS).

Step 0: Initially, when the CSDS is deployed, a number of Cloud providers are generated while each provider is posting around 25 of their services. A total of around 13,000 web-pages existed in the virtual-www including general web-pages.

Step 1: The screen in Fig. 11 shows the user input query which contains a service name (e.g., "Visual_Studio_2010") and requirements (e.g., OS = "Windows7", CPU name = "Core2Quad", CPU clock = "4.6", RAM = "9.0", HDD = "500.0", Network Bandwidth = "5.92", Network Latency = "1667.0"), and Mobile device support = "No").

Step 2: The CSDS automatically search with an exact service name, "Visual_Studio_2010" from the virtual-www and filter web-pages out which do not include the 'Cloud' term. The result is shown in Fig. 12.

Step 3: The CSDS consults the Cloud ontology for service reasoning. Then similarity of each term is aggregated as the service utility shown in Fig. 13.

Step 4: The CSDS takes the highest utility, "0.8275" as the best service among 53 web-pages and rate ordering shown in Fig. 13.

Step 5: The CSDS returns the result of the service discovery and the best service (e.g., provided by "Seller71") (see comparison table Ⅱ) as shown in Fig. 14. Additionally, results for the 3 cases, 1) without the CSDS, 2) the CSDS without the Cloud ontology, and 3) the CSDS with the Cloud ontology are printed into the user interface screen. Other recommended services are also included in turn shown in Fig. 15.

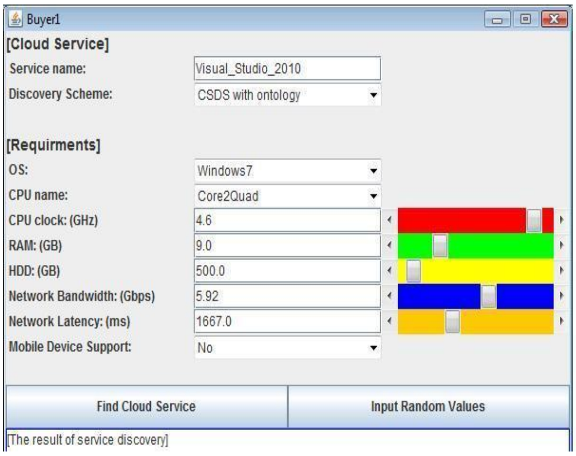


Fig. 11. CSDS User Interface

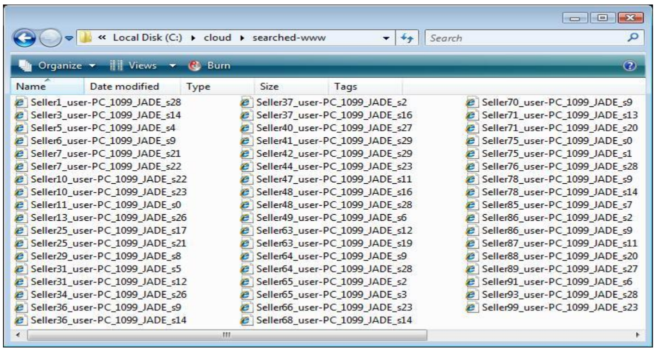


Fig. 12. After searching and filtering from the virtual-www

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

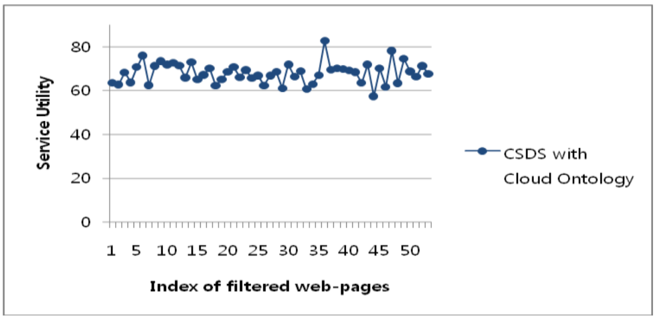


Fig. 13. Difference of Similarity among all filtered web-pages

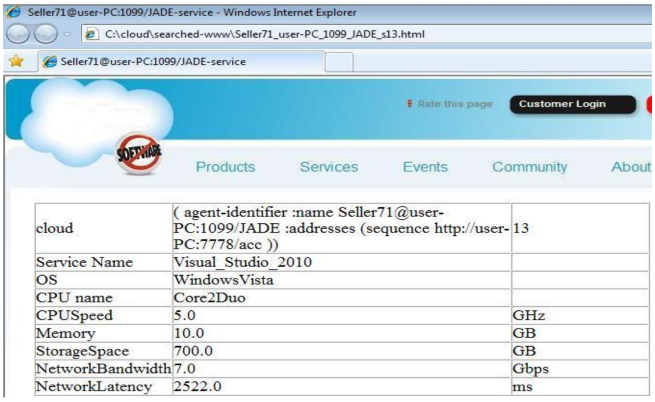


Fig. 14. The best service to be founded

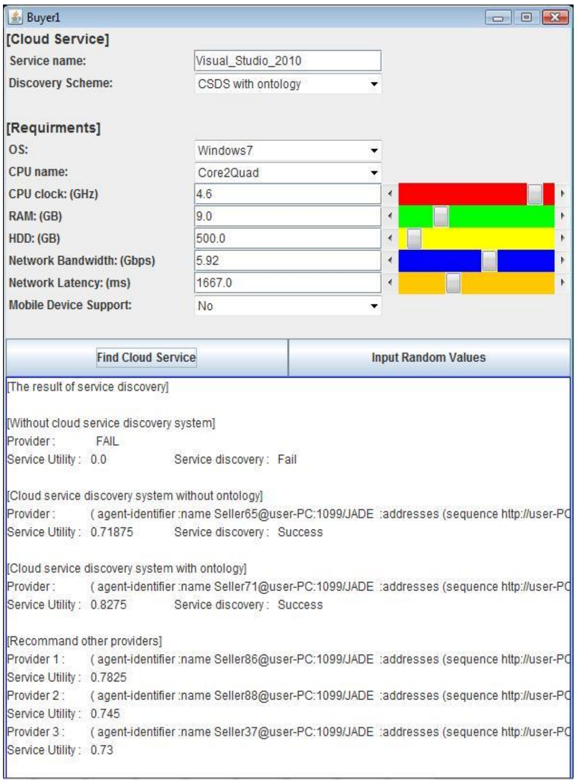


Fig. 15. Results of Cloud Service Discovery

Table Ⅱ. Comparison User Query with Discovery Result

Information	User Query	Discovery Result
Service name	VisualStudio_2010	VisualStudio_2010
OS	Windows7	WindowsVista
CPU name	Core2Quad	Core2Duo
CPU clock	4.6	5.0
RAM	9.0	10.0
HDD	500.0	700.0
Network (Band.)	5.92	7.0
NetworkLatency	1667.0	2522.0

VII. CONCLUSION AND FUTURE WORK

This paper has presented a Cloud service discovery system. It is specially designed for users who want to find a Cloud service over the internet. A Cloud ontology is also introduced for enhancing performance of the CSDS. The contributions of this work include: 1) building of the Cloud service discovery system and 2) constructing the Cloud ontology. It is the first attempt in building an agent-based discovery system that consults an ontology when retrieving information about Cloud services. In present, there are few big Cloud service providers and no various services. When the Cloud computing is more commonly and widely used in the near future, it can be helpful for Cloud users who want to find a Cloud service under their specific preference.

From the empirical results in Section Ⅴ, the CSDS with the Cloud ontology achieved better performance than the CSDS without the Cloud ontology. By consulting a Cloud ontology to reason 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.

Since this is an on-going work, the Cloud service discovery system is currently being enhanced future works include: 1) making more depth of the Cloud ontology so that it can make more difference between two services in terms of service utility and 2) completing functionalities of query processing, filtering and rating in [4], [5] which have been partially implemented.

REFERENCES

[1] Ken Birman, Gregory Chockler, Robbert van Renesse, "TOWARD A CLOUD COMPUTING RESEARCH AGENDA", Cloud Computing Defined.
[2] Lamia Youseff, Maria Butrico, and Dilma Da Silva, "Toward a Unified Ontology of Cloud Computing".
[3] 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.
[4] Kwang Mong Sim and Pui Tak Wong, "Toward Agency and Ontology for Web-Based Information Retrieval", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 34, No. 3, 2004, pp.1-13.
[5] Kwang Mong Sim, "Toward an ontology-enhanced information filtering agent" in ACM SIGMOD Rec., vol. 33, Mar. 2004, pp. 95-100
[6] Dieter Fensel, Ontologies: A silver bullet for knowledge management and electronic commerce, Springer, 2003.
[7] Heiner Stuckenschmidt, Ontology-based information sharing in weekly structured environments, Ph.D. thesis, AI Department, Vrije University Amsterdam, 2002.
[8] FIPA 2001, Foundation for intelligent physical agents. FIPA Ontology Service Specification, http://www.fipa.org/specs/fipa00086/XC00086D.html.
[9] Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu, "Cloud Computing and Grid Computing 360-Degree Compared".
[10] Troels Andreasen, Henrik Bulskov, Rasmus Kanppe, "From Ontology over Similarity to Query Evaluation".
[11] Liangxiu Han, Dave Berry, "Semantic-supported and agent-based decentralized grid resource discovery", Future Generation Computer Systems 24 (2008) 806-812.