Cloud Service Reservation using PTN mechanism in Ontology enhanced Agent-based System

Remya Gopinadh [#1], K. Saravanan [*2]

[#]Final Year ME, Dept. of CSE, Regional Centre of Anna University, Tirunelveli, India
[*]Assistant Professor, Dept. of CSE, Regional Centre of Anna University, Tirunelveli, India

***Abstract*— The relationship between the cloud provider and the cloud consumer must be described with a Service Level Agreement (SLA). To establish SLA for utilizing a cloud service, there are two issues which are the following: 1) Determining when to use the service (time slot); 2) Determining the price of the service. This paper presents an agent-based cloud service discovery approach that consults ontology to retrieve information about cloud services and implements Price and Timeslot Negotiation (PTN) Mechanism for Cloud Service Reservation. The important contributions of this work include: developing an agent-based search engine for cloud service discovery and effective adaptation of agent-based Price and Timeslot Negotiation mechanisms for cloud service negotiation. Cloud Service Reasoning Agent (CSRA) that enables the Cloud Service Discovery System (CSDA) is used to reason about the relations of cloud services, rate the search results and design and construction of cloud ontology. Cloud agents in PTNs are designed to concurrently make multiple proposals in a negotiation round that generate aggregated utility, differing only in terms of individual price and time-slot utilities.**

***Keywords*- Cloud computing, Cloud service discovery, Search engines, Cloud ontology, Service negotiation.**

I. INTRODUCTION

Cloud computing is an architecture for providing computing resources as a service. Cloud computing entrusts services with a consumer's data, software and computation over a network. The consumer of the cloud can obtain the services through the network. In other words, users are using or buying computing services from others. Cloud can provide Anything as a Service (AaaS). In general, cloud provides application, computation power, storage, bandwidth, database etc. As the resource pool is very large, users can scale the application in the cloud to any level. The cloud makes it possible for you to access your information from anywhere on demand basis.

Cloud computing is a collection of parallel distributed, and web–accessible that should be dynamically composed and virtualized based on consumer requirements. Cloud participants, namely, service providers and consumers, are self–interested, autonomous parties that should interact and coordinate among themselves to make an effective and efficient use of cloud resources. The distributed nature and inherent dynamism of cloud systems as well as the self–interested autonomy of cloud participants emphasized the need for agent–based solutions. An agent is a computer system that is capable of autonomous (independent) actions, that is, deciding for itself and figuring out what needs to be done to satisfy its design objectives [17]. To successfully interact, agents require the ability to cooperate, coordinate, and negotiate with each other.

Cloud ontology contains a set of cloud concepts and interrelationships among these concepts. We propose a similar ontology based semantic model that captures the features and capabilities available from a cloud provider's infrastructure. These capabilities are logically grouped together and exposed as standardized units of provisioning and configuration to be consumed by another cloud provider. For determining the similarity between two concepts with cloud ontology, there are three kinds of reasoning methods, 1) Similarity reasoning which determines the similarity between two concepts by counting common reachable nodes, 2) Equivalent reasoning which determines the similarity between two sibling concepts based on those label values, and 3) Numerical reasoning which calculates the similarity between two numeric concepts based on those label values. With empirical results of these three reasoning methods with cloud ontology, this paper shows that the search engine is able to provide an efficient search mechanism to find appropriate cloud services.

There are no specialized search engines for the consumers who want to find the cloud services. In an agent-based cloud service discovery approach, a search engine that consults cloud ontology for reasoning about the relations of cloud services and retrieves the relevant service information. The cloud consumer thus reserves the required service.

Even though the reservation manager can identify a common time slot that is acceptable to both consumer and provider agents, it did not provide a utility function for characterizing agents' preferences for different time slots. A tradeoff algorithm has been designed to enhance both the negotiation speed and the aggregated utility of price and time slot in a multi-issue negotiation. At the end of the negotiation process, provider and consumer commit to an agreement. This agreement in the SOA is referred to as a SLA.

The objective of this work is as follows: To design an agent-based cloud system that implements cloud ontology for cloud service discovery and PTN mechanism for cloud service reservation. Rest of the paper organized as follows: Section 2 contains other work related to cloud ontology, cloud discovery system and PTN mechanism. Section 3 gives an overview about the system architecture. Section 4 explains the cloud search engine and its components for cloud service discovery. Section 5 explains the PTN mechanism. Section 6 explains the implementation of a PTN mechanism in cloud for service reservation. Section 7 contains the conclusion of the paper and future work.

II. RELATED WORK

Since this work explores the issue of designing an agent based system for the cloud service discovery and service reservation, areas related to this work include the following: 1) Cloud service discovery; 2) Cloud ontology; and 3) Cloud service reservation and negotiation.

Ontology can provide a controlled vocabulary of concepts, each with semantics which are an explicitly defined and machine understandable. It also provides a shared understanding of a domain of interest to support communication among computers and human by defining shared and common domain theories. In the field of information retrieval, ontology which consists of a set of concepts and relationship between concepts is used for dealing with user queries. This paper provides ontology for infrastructure services. The relevant concepts related with the infrastructure services are classified and the relationship between the concepts is defined. The search engine includes technical specifications for operating system, CPU, clock speed, memory etc.

There are several tools available for developing the ontology. In this work ontology is developed in Protégé 4 [19]. In this paper cloud ontology defines the domain model of IaaS layer. This ontology facilitates the description of infrastructure services and the discovery of these services based on their functionality. In future work , will extend the ontology to cover both PaaS and SaaS layers.

In ontology based Cloud Framework[3] Kaushik and Chana demonstrates that by using ontology based architecture, cloud can be easily accessed and updated using semantic web queries. This framework provides solution for the increasing complexity of cloud by showing that Web Ontology Language (OWL) is efficient for cloud service discovery. Kang and Sim[4] introduces cloud portal with various service categories and cloud service search engine for cloud computing system. The similarity among cloud services are determined using similarity reasoning methods [4][5][6]. Youseff, Butrico and Da Silva[7] presents a summary of assimilation of cloud computing, with a classification of its components, and their relationships as well as their dependency on some of the prior concepts from other fields in computing. The proposed cloud computing ontology has been depicted as five layers, such as cloud application layer, cloud software environment layer, cloud software infrastructure layer, software kernel and hardware and firmware. The infrastructure layer is further divided into computational resources, data storage and communication.

Cloud service reservation using PTN mechanism [2] is implemented in an agent-based cloud. The negotiation mechanism facilitates PTNs between cloud agents and tradeoff which deals between price and time-slot utilities. Another novelty of this work is formulating a novel time-slot utility function that characterizes preferences for different time slots. An automated negotiation engine [16] that identifies mutually acceptable terms and also demonstrates how the negotiation engine enables users to control the quality of service levels they require. A bilateral protocol[12] for SLA negotiation using the Alternate Offers mechanism is able to respond to an offer by modifying some of its terms to generate a counter offers.

III. SYSTEM OVERVIEW

The architecture diagram for Cloud Service Discovery and Reservation (Fig 1) provides an interface for cloud search, cloud service registration and cloud service reservation, negotiation session between provider agents and consumer agents and includes a registry of available infrastructure services and negotiation results.

A. Cloud Service Registration

The cloud service providers register their services into the cloud registry. The service description includes the following specifications.

- Category of service,
- Technical specifications (OS, CPU clock, memory, disk capacity and etc.) and
- Cost specification (maximum acceptable price and timeslot range).

These service specification, known as provider service proposal is registered into the registry. The consumers can search for the matching services or they can directly select a service from the registry and make proposals against the provider proposal.

B. Cloud Search Engine

In our Cloud Search Engine, consumers can specify the type of cloud services. Furthermore, consumers can specify three kinds of requirements:

- Functional requirements (category of service),
- Technical requirements (OS, CPU clock, memory, disk capacity and etc.) and
- Cost requirements (maximum acceptable price and timeslot range) as input parameters.

Once consumers send those input parameters to the cloud service search engine, CSDA consults the cloud ontology and it returns the list of cloud services ordered by aggregated similarity (service utility). The cloud ontology provides meta information which describes data semantics. It contains a set of cloud concepts and their individuals as well as the relationships between individuals.

C. Cloud Service Reservation

Among the retrieved best services from the search engine, the consumer can reserve required service. Consumers can

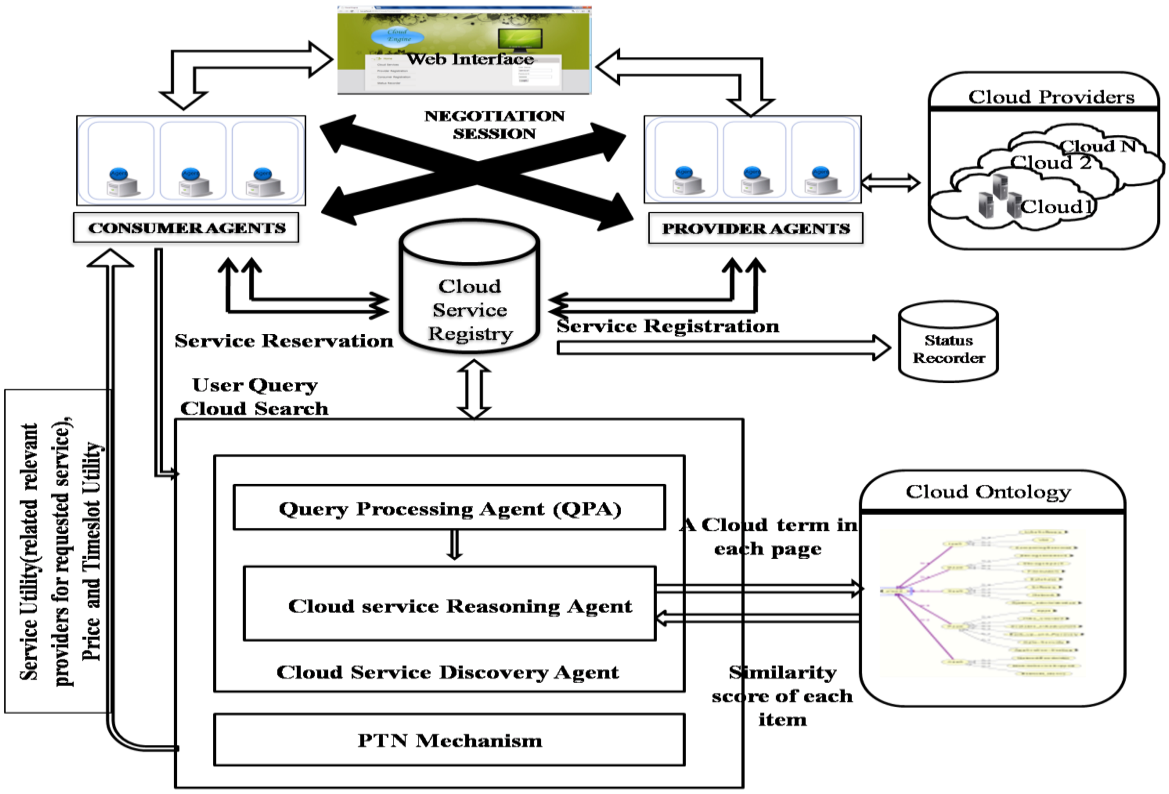

Fig. 1 Cloud Service Discovery and Reservation

reserve more than one service providers for the same service. For better negotiation result, the consumers can even make multiple proposals that differ in terms of price and timeslot.

D. Negotiation Session

During negotiation the cloud provider agents send confirmation to the cloud consumers if the required service is available for the requested timeslot. If the consumer receives more than one confirmation for the same service from different provider agents, the consumer selects the services with highest PTN utility.

E. Cloud Registry

The cloud registry contains the registered service providers and their services. It also keeps the information about the price utility, timeslot utility and average utility of each service proposals made by both consumers and providers. This information is used for the calculation of PTN utility that determines the exact reservation for cloud services.

F. Status Recorder

The agent based cloud system includes a status recorder. The cloud status recorder records all negotiation and reservation results of all provider and consumer agents in the agent based system.

The following algorithm is used for cloud service discovery and negotiation.

Algorithm:

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]

Service Utility=$\sum_{k=0}^{N} term(k) \times Weight(k)$

Where Weight (k) = 1/N is uniformly distributed.

PTN utilty(P)=1/2(Consumer Utility+Provider Utility)

endFor

6. Rating web-pages used by the Service Utility.

7. Select the web-page/s which has the highest Service Utility as the best Cloud service and reserve for services.

8. At each negotiation round t, do the following:

I. From each service provider consumer may receives confirmation.

II. If the consumer receives more than one confirmation, then consumer agent accepts the service that generates highest PTN utility and rejects all other reservation.

IV. CLOUD SEARCH ENGINE

Cloud Search Engine consists of a Cloud Service Discovery Agent (CSDA) and a Web Interface.

A. Web Interface:

The web interface provides an interface for Cloud Search, Cloud Service Registration and Cloud Service Reservation. The cloud providers register their services with attribute specification and users who want to use a cloud service send queries with attribute specification to the system through the web interface. After the Cloud Search, the Cloud Service Discovery System retrieves the related relevant providers for requested service.

B. Cloud Service Discovery Agent

CSDA consist of a Query Processor Agent and Cloud Service Reasoning Agent.

1) Query Processor Agent: The customers can submit queries into the web interface, then the matching attributes from services has been filtered. Consumers can specify the attributes in the web interface and these attributes are needed for reasoning or similarity search. The consumer has to communicate with the query processing agent and make use of the required services and resources.

2) Cloud Service Reasoning Agent: The Cloud Service Reasoning Agent (CSRA) deals with the Cloud Ontology for performing Service Reasoning and Rating.

a) *Service Reasoning:* The reasoning agent (RA) consults a cloud ontology for performing Service Reasoning. All information supplied by a user is used to determine similarity between two services.

There are three methods to determine similarity:

- *Similarity reasoning:* Similarity reasoning is the similarity between two concepts by counting common reachable nodes. The similarity of concepts represents the degree of commonality between concepts. It can be determined as in (1)

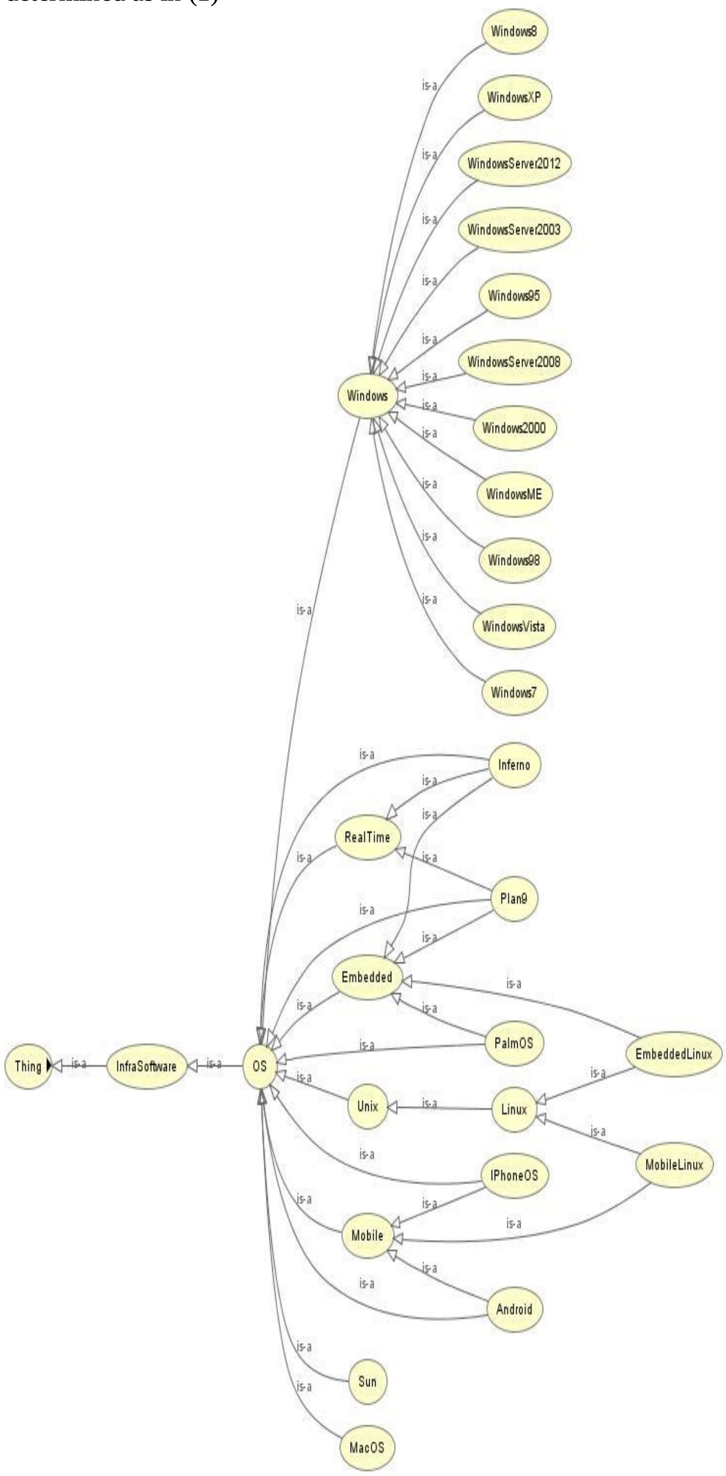

Fig. 2 Cloud Ontology in terms of OS

$$Sim(x,y) = \mu\left|\frac{\alpha(x)\cap\alpha(y)}{\alpha(x)}\right| + (1-\mu)\left|\frac{\alpha(x)\cap\alpha(y)}{\alpha(y)}\right| \quad (1)$$

Where x and y are two concepts, μ €[0,1] determines the degree of influence of generalizations (generation is the opposite of specialization), which depends upon a hierarchical ontology. μ is set to 0.5 so that both measures are given equal consideration. Sim (x,y)=0 means that x is totally not similar to y and Sim (x,y)=1 means that x is fully similar to y. But with similarity reasoning, there is no way to compare between two sibling nodes. Hence, we designed equivalent reasoning.

- *Equivalent reasoning:* Equivalent reasoning is the reasoning for determining similarity between two sibling concepts based on those label values. i.e. Determining the compatibility between two different versions of software. Since two sibling nodes representing different versions of software, will have high degree of similarity, but differ only in terms of chronological ordering. It can be determined as in (2)

$$Eqv\ (x,y) = Sim\ (x,y) + \frac{0.8^{|c1-c2|}}{10} \qquad (2)$$

Where c1is the label value of concept x and c2 is the label value of concept y and Sim(x,y) is the similarity method which is used for Similarity reasoning

- *Numerical reasoning:* Numerical reasoning is used for determining the similarity between two numeric concepts based on those label values. The similarity between two numeric values in same domain can be calculated as following formula.

$$Sim\ (x,y,c) = 1 - \left| \frac{a-b}{Max_c - Min_c} \right| \qquad (3)$$

Where a is first numeric value and b is second numeric value and c is concept name. Max_c and Min_c is the Maximum and minimum values of each domain respectively.

b) *Rating:* An aggregated utility (Service Utility) is used to determine the rating. A service vendor which has the highest utility would be selected as the best services for the user.

V. PRICE AND TIMESLOT NEGOTIATION

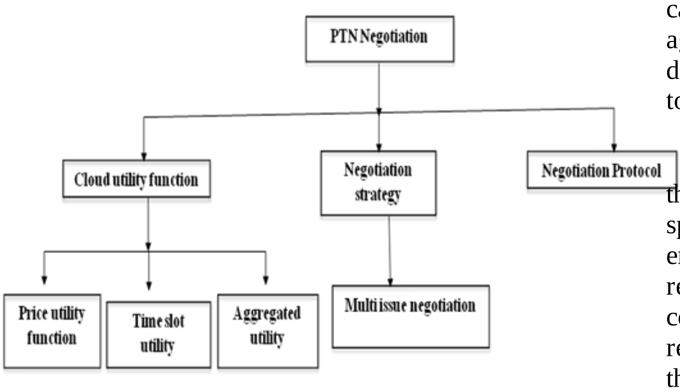

Fig. 3 PTN mechanism

Price and Timeslot Negotiation mechanism provides a negotiation between consumer agent and the provider agent. The CSDA consults the Cloud Ontology and retrieves the best services that fit for the cloud requirements. The PTN mechanism (Fig 3) offer cloud consumers to make multiple proposals that offer only in terms of price and time. For this the web interface includes specification for cost and time information. During cloud service discovery the PTN mechanism is executed and along with the service rating, the users can view the Price and Timeslot Utility for different services. The cloud consumer reserves particular services with different providers and the negotiation is based on the price and timeslot utility.

Cloud utility function represents an agent's level of satisfaction for a negotiation. Since each cloud participant has different preferences for different prices and time slots, a price utility function, a time-slot utility function, and an aggregated utility function are used to model the preference ordering of each proposal and each negotiation outcome. The utility model includes Price utility function, Timeslot utility function and Aggregated utility function [3].

The negotiation strategy considers bilateral negotiations between a consumer and a provider, where both agents are sensitive to time and adopt a time-dependent concession-making strategy for PTNs. Since both agents negotiate on both price and time slot, generating a counterproposal can be making either a concession or a trade-off between price and time slot. Negotiation protocol offers agents to make counteroffers to their opponents in alternate rounds. Both agents generate counterproposals and evaluate their opponent's proposal until either an agreement is made or one of the agent's deadlines is reached. If one of the agents' deadline expires before they reach an agreement, the negotiation fails.

VI. IMPLEMENTATION

In a cloud market, there are many consumers and service providers. Cloud service providers and consumers participate in the cloud market of the through the Registry. All agents participating in the cloud market are registered in the cloud registry. All consumer agents connected to the cloud registry can then recognize and communicate with each provider agent. Provider agents and consumer agents generate service descriptions (Fig 4) and specify their preferences with regard to service name, price and time slot.

The consumer agents can manually select the providers or they can perform a cloud service discovery through service specification. For cloud service discovery a cloud search engine can be used. The cloud search engine retrieves all the registered services and the service descriptions. If the consumer want to search for a service that best suit their requirements, the consumer send service description query to the cloud search engine. The query processor filters the consumer's query and retrieves the matching services from the cloud registry. The cloud service reasoning agent consults the

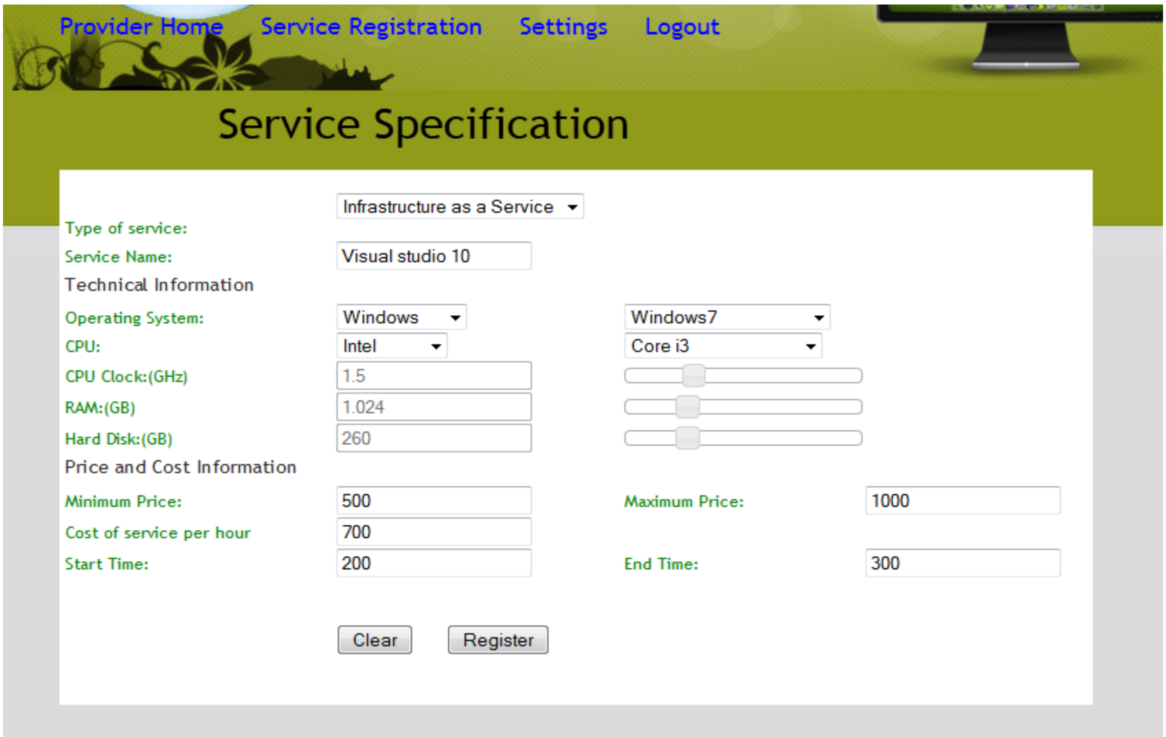

Fig. 4 Service Specifications

cloud ontology and three types of reasoning is performed. After reasoning the rating of the service is calculated. The best matching services will have highest rating. Also PTN mechanism calculates the price utility, timeslot utility and average utility for each service. Finally the cloud service discovery agent retrieves the matching services in the increasing order of rating and average service utility (Fig 5).

The consumer agent reserves the cloud service to the provider agents. The provider agents may get more service requests from different consumers. The PTN mechanism searches for mutually acceptable agreement for leasing a service. In PTN mechanism, negotiation agents calculates the price and timeslot utility of consumer and providers. Using

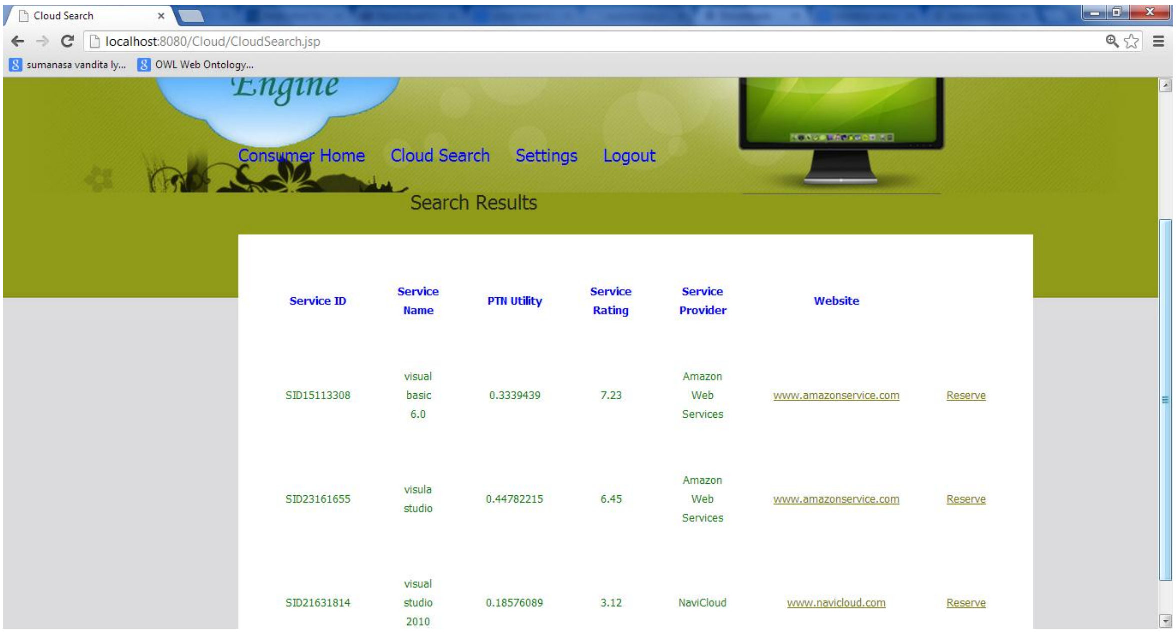

Fig. 5 Service Discovery

these values and the weight assigned for price and timeslot, aggregated utility is calculated for consumer and provider. The total average utility is used to choose the best service scheme based on price and timeslot utilities. Thus the negotiation result contains a set of reserved services that best suits the consumer requirements.

The consumer agent may get multiple confirmations for the same service from different provider agents; in that case consumer accepts the service that generates highest PTN utility and rejects all other reservation. Thus PTN mechanism provides the best utility function that is used to characterize an agent's level of satisfaction in terms of price and timeslot.

Performance Analysis:

The agent-based system with cloud ontology have a better filtering and reasoning functionalities. Therefore cloud search engine retrieves cloud services that are more likely to match with the consumer requirements. Fig. 6 shows agent-based cloud engine with cloud ontology and without cloud ontology. The performance measure is the discovery success rate which is determined by the ratio of number of successes over the number of attempts. The number of cloud service provider is varied from each attempts. The results shows that cloud engine with Cloud ontology shows better performance than the results of using cloud engine without cloud ontology. Also the result shows that with cloud ontology service utility of the retrieved services id greater than 0.5. So based on the observation a consumer can find the appropriate services using cloud engine with cloud ontology.

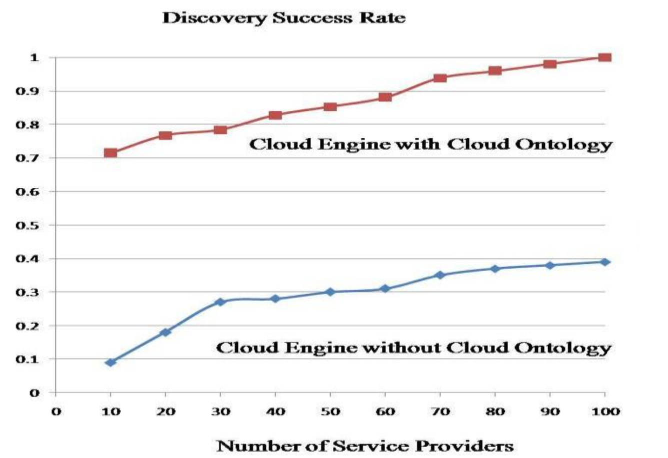

Fig. 6 Discovery Success Rate

To evaluate the performance of the *PTN* mechanism, agent based-system used the following as the performance measures: Average total Utility U^{avg}_{Total} of the negotiating pair and Negotiation Speed.

U^{avg}_{Total} is the level of satisfaction in terms of price and time slot with the service to be provided.

$$U^{avg}_{Total}(P,T) = (U^{P}_{Total}(P,T) + U^{C}_{Total}(P,T))/2 \quad (4)$$

TABLE 1 INPUT DATA SOURCES CLOUD RESERVATION

Input Data	Possible Values
Cloud Loading	CL=N_{res}/N_{tot} $(0 \le CL \le 1)$
No. of provider agents	Integer
No. of consumer agents	Integer

$$U^{P}_{Total}(P,T) = w^{P}_{P}.U^{P}_{P}(P) + w^{P}_{T}.U^{P}_{T}(T) \quad (5)$$

$$U^{C}_{Total}(P,T) = w^{C}_{P}.U^{C}_{P}(P) + w^{C}_{T}.U^{C}_{T}(T) \quad (6)$$

$U^{P}_{Total}(P,T)$ is the Aggregated utility of Provider.

$U^{C}_{Total}(P,T)$ is the Aggregated Utility of Consumer.

w^{P}_{P} and w^{C}_{P} are the weights of price assigned by the provider and consumer respectively.

w^{P}_{T} and w^{C}_{T} are the weights of timeslot assigned by the provider and consumer respectively.

The cloud loading $(0 \le CL \le 1)$ in Table 1 represents and simulates different levels of utilization of cloud services. *CL* is the ratio of the following: 1) *N*res, which is the number of time slots already reserved, and 2) *N*tot, which is the total number of time slots in the reservation queues of all service providers.

The average total utility of the negotiating pair and negotiation were recorded for different cloud loads and varied the number of proposals. Fig 7 and Fig 8 shows that as the number of concurrent proposals(R1, R2 and R3) increased, higher utility and faster negotiation speed were achieved. But the amount of improvement in terms of total utility and negotiation speed decreased with increasing the number of concurrent proposals. Therefore, it suggests that, when adopting the PTN mechanism, an agent can achieve a high total utility and a fast negotiation speed without having to generate an excessively large number of concurrent proposals.

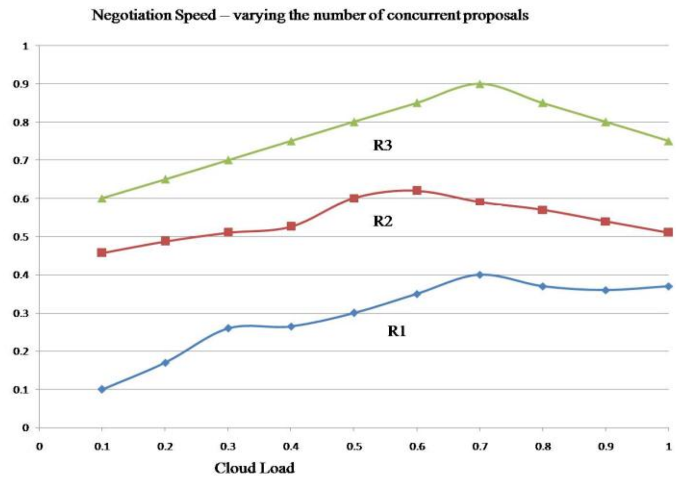

Fig. 7 Negotiation Speed Vs. Cloud Load

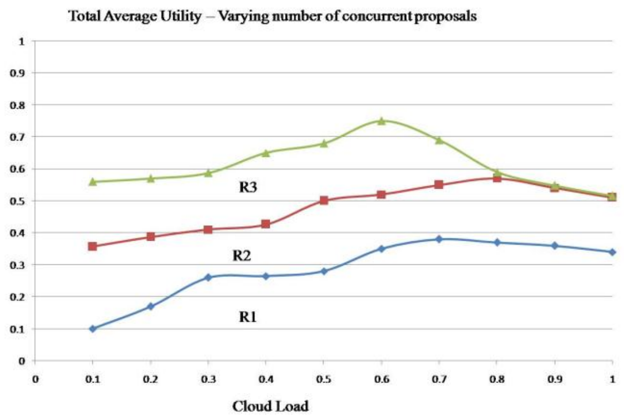

Fig. 8 Total average Utility Vs. Cloud Load

VII. CONCLUSION

This paper presented an agent-based cloud system for cloud service reservation. It is designed for the cloud service providers who want to sell their services to a maximum price and the consumers who want to find a cloud service that best suit their price and timeslot. The cloud engine with cloud ontology shows better performance for cloud service discovery. Cloud engine is more successful to find cloud services that meet requirements of consumer. Since PTN mechanism is implemented in an agent-based cloud, PTN agents can concurrently make multiple proposals that generate the aggregated utility differ only in terms of individual price and timeslot utilities. The performance analysis shows that agents adopting PTN mechanism can reach an agreement, successfully acquire the desired cloud services and achieve higher utilities. Thus a successful agent based cloud system is designed and implemented for cloud service discovery and reservation.

The future work includes a low level specification for the user preferences and enhancing the proposed trade-off algorithm by adaptively controlling the number of concurrent proposals in a burst mode proposal to reduce the computational complexity. We can include negotiation issues other than price and timeslot.

REFERENCES

[1] Seokho Son and Kwang Mong Sim, *Senior Member, IEEE* , "A Price-and-Time-Slot- Negotiation Mechanism for Cloud Service Reservations" IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 42, No. 3, June 2012

[2] Abhishek Kaushik and Inderveer Chana, "Ontology Based Cloud Framework" Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications - ACCTHPCA, June 2012

[3] Jaeyong Kang and Kwang Mong Sim, "Cloud Portal With A Cloud Service Search Engine", 2011 International Conference on Information and Intelligent Computing IPCSIT vol.18

[4] Taekgyeong Han and Kwang Mong Sim, "An Ontology-Enhanced Cloud Service Discovery System", Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol 1

[5] Jaeyong Kang and Kwang Mong Sim, "Cloudle : An Agent-Based Cloud Search Engine That Consults A Cloud Ontology", CCV Conference 2010, May 17–18, 2010, Singapore.

[6] Lamia Youseff and Maria Butrico, Dilma Da Silva, "Toward A Unified Ontology Of Cloud Computing", Proc. Grid Computing Environments Workshop (GCE '08), pp. 1-10, 2008.

[7] K. M. Sim and B. Shi, "Concurrent negotiation and coordination for Grid resource coallocation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 753–766, May 2010.

[8] K. M. Sim, "Grid resource negotiation: Survey and new directions," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 3, pp. 245–257, May 2010.

[9] I. Brandic, D. Music, and S. Dustdar, "Service mediation and negotiation bootstrapping as first achievements towards self-adaptable Grid and Cloud services," in *Proc. GMAC Workshop Conjunction With 6th Int. Conf. Autonomic Comput. Commun.*, Barcelona, Spain, Jun. 2009, pp.

[10] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[11] S. Venugopal, X. Chu, and R. Buyya, "A negotiation mechanism for advance resource reservation using the alternate offers protocol," in *Proc. 16th IWQoS*, Twente, The Netherlands, Jun. 2008, pp. 40–49.

[12] M. A. Netto, K. Bubendorfer, and R. Buyya, "SLA-based advance reservations with flexible and adaptive time QoS parameters," in *Proc. 5th Int. Conf. Service-Oriented Comput.*, Vienna, Austria, Sep. 2007, pp. 119–131.

[13] K. M. Sim, "G-commerce, market-driven G-negotiation agents and Grid resource management," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1381–1394, Dec. 2006.

[14] H. Gimpel, H. Ludwig, A. Dan, and R. Kearney, "PANDA: Specifying policies for automated negotiations of service contracts," in *Proc. ICSOC*, vol. 2910, *LNCS*, New York, 2003, pp. 287–302.

[15] R. Lawley, M. Luck, K. Decker, T. R. Payne, and L. Moreau, "Automated negotiation between publishers and consumers of grid notifications," *Parallel Process. Lett.*, vol. 13, no. 4, pp. 537–548, Dec. 2003.

[16] M. Wooldridge, "An Introduction to Multiagent Systems", second ed. John Wiley & Sons, 2009.

[17] T. Andreasen, H. Bulskov, and R. Kanppe, "From Ontology over Similarity to Query Evaluation," Proc. Second Int'l Conf. Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems (ODBASE), Nov. 2003.

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

[19] *http://www.mysqltutorial.org*

[20] http://help.eclipse.org