International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

Smart Cloud Services Marketplace based on NLU Technologies

Samer Hasan [1], Prof. Kuda Nageswara Rao[2]
[1,2] *Department of Computer Science & System Engineering, College of Engineering (A), Andhra University, India.*

Abstract: ***Cloud computing makes our life easy by delivering computing resources as utility like telephony, water and gas. In cloud computing users should pay only for what they consumed. Nowadays, cloud service providers deliver a huge number of cloud services with almost the same features which makes the cloud services discovery and selection process as a big challenge for the end consumers. Using existing search engines results in a lot of unrelated outcome which increases the cloud service discovery and selection process time and effort. In this paper, we present an enhanced cloud services marketplace framework to facility the cloud services trading between providers and consumers and to make could services more visible for all consumers. Proposed framework receives users' requests as a voice commands or flat-text then translates them based on Natural Language Understanding technologies. In additional, we enhanced the matching algorithm by adding different weights for attributes based on consumer preferences. Experiments showed an enhancement in the overall user experience and better matching for user request.***

Keywords: ***Cloud computing, cloud services, service matching, cloud marketplace, cloud architecture.***

I. INTRODUCTION

Cloud computing [1] is a paradigm that enables access to a shared computing resources where consumer should pay only for what he consumed. There are three different levels of cloud services. Infrastructure as a Service (IaaS) where users hire an infrastructure resources from another party , Platform as a Services (PaaS) where users develop applications on a haired platform and Software as a Service where users use an applications developed and operated by another party based on pay as you use model. Nowadays, there are big number of cloud service providers who provide a huge number of similar cloud services which makes the process of finding and selecting the appropriate cloud service a tedious task for the users. Using general search engines like Google increases the time and effort of finding the cloud services. In this paper, we present an enhanced architecture for cloud services marketplace based on the work proposed on [2]. Enhanced architecture allows users to enter the cloud service requests as a voice commands or as plain texts. In addition, enhanced architecture improved the matching algorithm by applying different weights for user requested attributes based on user preferences. . Experiments showed an enhancement in the overall user experience and better matching for user requests. The rest of this paper is organised as following: section 2 surveys the related work, section 3 presents the enhancement architecture, section four displays the results of the experiments and section 5 is a conclusion and future work.

II. RELATED WORK

Researchers proposed different approaches for receiving consumers requests in cloud services discovery and selection systems. Zhang et al. [3] presented Cloud Recommender System based on Owl ontology and Sql matching. Cloud Recommender architecture consists of three layers: Repository, Application Logic Layer and User Interface. User Interface enables cloud consumer to select the basic configuration parameters related to compute services based on predefined parameters. Sim [4] developed cloud service discovery system based on Multi-Agents and search engine. The proposed system consists of a service discovery agent that consults cloud ontology and multiple cloud crawlers for building its database of services. User can submit the request by selecting the values from lists and checkboxes. Han and Sim [5] built a cloud discovery system based-on agent and ontology. User enters queries for cloud services through the user interface and the system return the best matching service with recommendations. Kang and Sim [6] presented a Cloud computing search engine. Cloud consumers enter requirements through a web interface. Desirable parameters are recommended by user profiling agent based-on user history log. Dastjerdi et al. [7] proposed ontology-based architecture for appliances development on cloud service providers. Web portal provides graphical interface to capture the clients requirements and requirements ontology captures requested functional and non-functional properties. Saravana et al. [8] presented service recommendation system based-on semantic technologies. User Interface receives the user request then parses and refines it to generate SPARQL quires. Nagireddi and Mishra [9] proposed a generic search engine for cloud service discovery based-on

ontology. Search engine provide clients with interface to enter queries and receive the results based-on service characteristics and AHP mechanism. Afify et al. [10] developed a unified ontology that serves as semantic based repository to facilitate SaaS publication, discovery and selection processes. To overcome the limitation of SPARQL language, WordNet ontology is used to semantically expand the service description and user request. In [11] User agent provides a user interface to the cloud consumers.

III. ARCHITECTURE

This work is an extension for work presented in [2]. We enhanced the user interface to accept the user requests as a flat-text or voice commands based on NLU technologies. On the other hand, we enhanced the matching algorithm by adding different weights for each cloud service attribute based on user preferences. As shown in fig. 1 enhanced cloud services marketplace added four new components to User Interface Sub-System as following:

A. Voice Receiver is responsible for receiving the user commands as a voice.
B. Speech to Text is responsible for converting voice commands into text.
C. Text receiver is responsible for receiving the user commands as a flat-text.
D. NLU is responsible for extracting the commands meaning based on CFG rules.

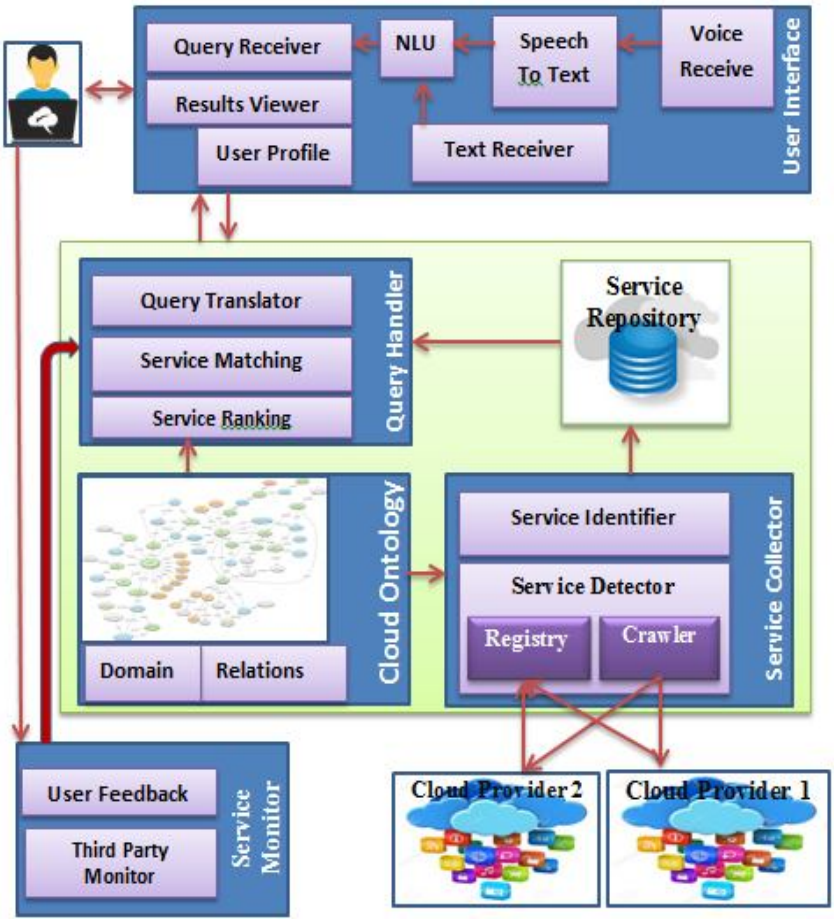

Fig. 1 Enhanced cloud services marketplace

E. Natural Language Understanding

Natural Language Understanding (NLU) is a field of artificial intelligence that attempts to understand the complete meaning of the text by adopting different technologies like Context- Free Grammars (CFG) [12]. CFG[13] generates patterns from strings based on recursive rewriting productions. CFG is consisting of a set of grammar rules as following:

1) Start symbols S
2) set (N) of non-terminal symbols
3) set (T) of terminal symbols
4) set of rules (P)

There are two approaches to draw a different derivation tree: Top-down Approach and Bottom-up Approach. Fig. 2 shows parsed tree for the command "please find virtual desktop price 10 USD per month storage 10 gb location India" based on CFG rules.

F. Enhanced Matching Algorithm

As an enhancement for the work proposed in [2], we improved the matching algorithm by adding different weights for each cloud service attribute in user request. This improvement helps cloud consumers to find the appropriate cloud services by focusing on specific attributes more than others. Enhanced matching algorithms for cloud services marketplace is improved as following.

$$ms = \prod_{i=1}^{v} DSim\left(cs^{ai}, ur^{ai}\right) * \frac{\sum_{j=1}^{u} w_j * PDSim\left(cs^{aj}, ur^{aj}\right)}{\sum_{j=1}^{u} w_j}$$

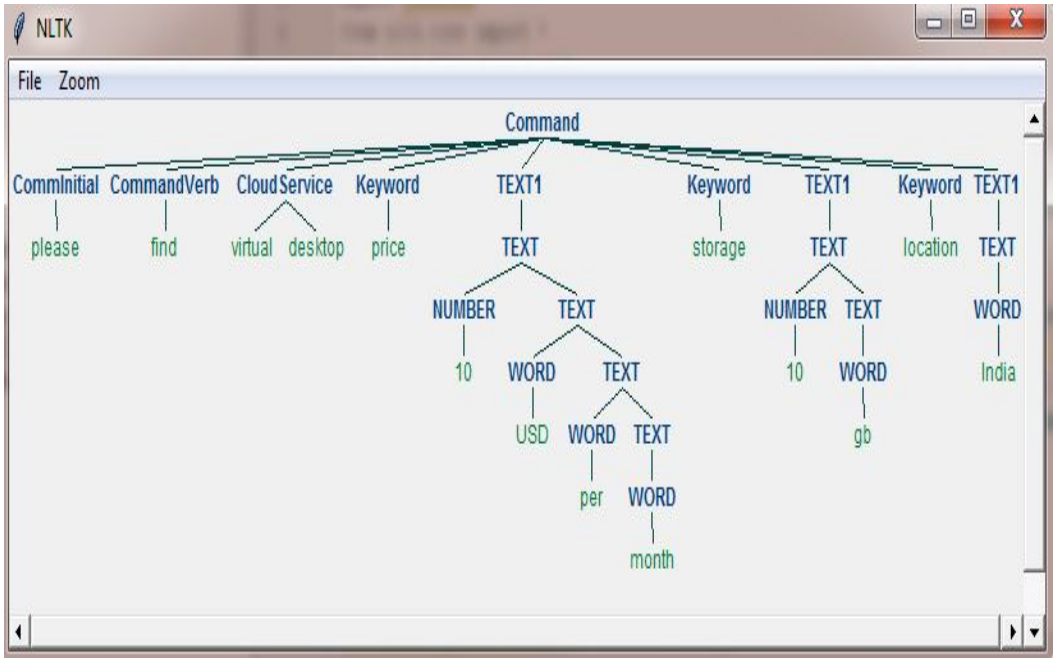

Fig. 2 NLTK parsed tree for user command

IV. RESULTS AND DISCUSSION

CFG is implemented in Python using NLTK [14] library. Dataset is collected from cloud provider's websites and stored in repository. Results are compared with work proposed in [2] refers it as MP for the user request (VCPU=4, Price=30 USD/month, Storage=75GB, Ram=10GB, Availability=99%). Two different set of weights are used for the same user request attributes EMP1 = (3, 5, 2, 4, 1), EMP2 = (3, 1, 4, 2, 5) respectively. Two parameters are used for comparison: Number of matched services and average matching score as following.

A. Number of Matched Services

Number of matched services determine the time that consumer should spend to find the best matched service. As shown on Fig. 3 , Enhanced cloud services marketplace showed higher number of matched cloud services for user request with weights 2 (EMP2) in compare with unweight query MP . On the other hand, Enhanced cloud services marketplace showed lower number of matched cloud services for user request with weights 1 (EMP1) in compare with unweight query MP. As a result, changing the weights for user requested attributes will directly affect the number of the matched services and user can get best matched queries by using the appropriate weight for each attribute.

B. Average Score

Average score represents the mean of score of all matched services at specific threshold. It affects the number of cloud services that will be displayed as results of the user queries. User can change the requested attribute's weights to change the matching score and

the number of matched services. As shown in Fig. 4, all requests showed almost the same values for all thresholds. EMP1 showed Zero value for threshold = 0.9 because the number of matched services for this threshold is zero.

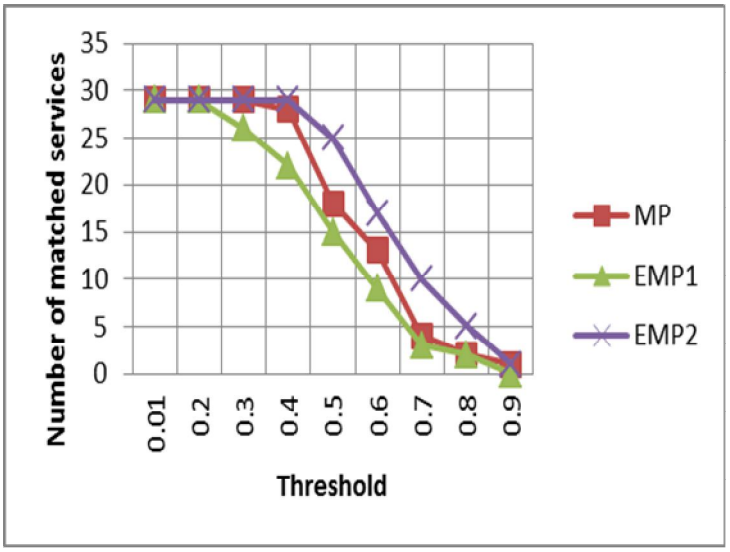

Fig. 3 number of matched services per threshold.

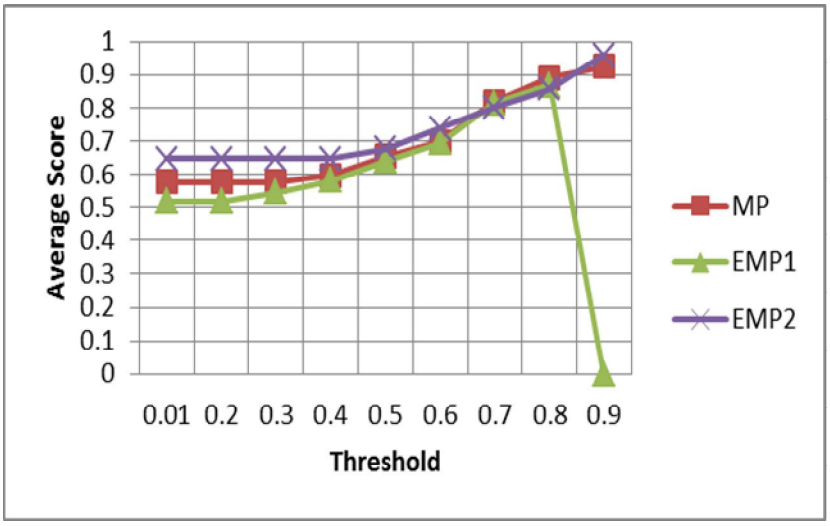

Fig. 4 Average score per threshold

V. CONCLUSION

Cloud computing changes our life and the way that we use computing resources. The current process for finding the appropriate cloud services is hard and time consuming. This work, presented an enhanced architecture for cloud services marketplace based on NLU technologies. In proposed architecture, user can enter requests by voice commands or pain texts. On the other hand, this work improved the matching algorithm by adding different weights for requested attributes based on user preferences. Experiments showed an improvement in the overall user experience with cloud services marketplace with better matching results. As a future work, we plan to use Artificial Intelligence and Machine Learning technologies to extract the requested attributes from user requests.

REFERENCES

[1] SP 800-145, The NIST Definition of Cloud Computing (September 2011), is available at: http://dx.doi.org/10.6028/NIST.SP.800-145.

[2] S Hasan, V.V.Kumari , "Generic-distributed framework for cloud services marketplace basedon unified ontology - Journal of Advanced Research, Volume 8, Issue 6, November 2017, Pages 569-576. Elsevier. https://

[3] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, M. Menzel, and S Nepal. (2012). An Ontology based System for Cloud Infrastructure Services Discovery. In 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). Pittsburgh, Pennsylvania, USA.

[4] K. M. Sim, Agent-Based Cloud Computing, IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 4,b(2012).
[5] T. Han, K.M. Sim, An ontology-enhanced cloud service discovery system, in Proceedings of the International MultiConference of Engineers and Computer Scientists, 2010, pp. 17-19
[6] J. Kang, K. M. Sim, Ontology and Search Engine for Cloud Computing System, International Conference on System Science and Engineering, 2011, pp. 276-281.
[7] A. V. Dastjerdi, S. G. H. Tabatabaei, and R. Buyya, An effective architecture for automated appliance management system applying ontology-based cloud discovery, 2010 10th 1EEEACM International Conference on Cluster Cloud and Grid Computing, 2010, pp.104-112
[8] S. Balaji, K. N. K. b. , R. Kumar, Fuzzy service conceptual ontology system for cloud service recommendation, Computers and Electrical Engineering 000 (2016), 2016, pp. 1–12
[9] V. Nagireddi and S . Mishra , "A naive approach for cloud service discovery mechanism using ontology " , 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), I EEE, (2013).
[10] Y. Afify, I. Moawad, N. Badr, M. Tolba, A semantic-based software-as-a-service (saas) discovery and selection system,
[11] J. Kang, K. M. Sim, Towards Agents and Ontology for Cloud Service Discovery, International Conference on Cyber- Enabled Distributed Computing and Knowledge Discovery, 2011
[12] American Association for Artificial Intelligence Brief History of AI https://aitopics.org/misc/brief-histor
[13] Hopcroft, John E.; Ullman, Jeffrey D. (1979), Introduction to Automata Theory, Languages, and Computation, Addison-
[14] Wesley. Chapter 4: Context-Free Grammars, pp. 77–106; Chapter 6: Properties of Context-Free Languages, pp. 125–NLTK, The Natural Language Toolkit http://www.nltk.org/.