International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 6 Issue 10||October 2017 || PP. 01-06

Enhanced Percent Distance Similarity Algorithms for Cloud Services Marketplace

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

Abstract: *Cloud computing services become an important part of our life as individual or industry. Cloud computing transferred the computing recourses into utilities like water and electricity based on pay only as you use model. Nowadays, finding the suitable cloud service is a big challenge for the consumer. Using search engine (Google and Yahoo) to find the suitable cloud service is hard and time consuming task because it shows a lot of unrelated results. In this paper, we present architecture for cloud services marketplace where cloud providers and consumers can exchange the cloud services as utilities. In additional, we present Enhanced Percent Distance Similarity (EPDSim) algorithm for numerical matching between cloud services. Proposed algorithm is depending on user requested value only and it is independent of any other attribute value. Experiment results showed that Enhanced Percent Distance Similarity algorithm reduced the execution time by 49% with keeping the same values for all other parameters like: Number of matched service, average score and recall.*

Keywords: *Cloud computing, cloud marketplace, cloud services, service discovery, numerical similarity.*

Date of Submission: 24-08-2017 Date of acceptance: 06-10-2017

I. Introduction

Cloud computing [1] enables on-demand and ubiquitous network access to a pool of shared and configurable computing resources. There three basic delivery models are (SaaS, PaaS and IaaS). In Software as a Service (SaaS), clients use software running on cloud providers' hardware. In Platform as a Service (PaaS), clients deploy software onto providers' hardware. Finally, in Infrastructure as a Service (IaaS), clients deploy and arbitrary applications and have a full access to the system. There are three deployment models for cloud services: Private cloud, Public cloud and Hybrid cloud. Nowadays, finding the suitable cloud service is a big challenge for the cloud consumers. Using search engine (Google and Yahoo) to find the suitable cloud service is hard and time consuming task because it shows a lot of unrelated results. Consumers need to spend a lot of time and effort to filter the search engine results and compare between them to find the best matched cloud service. In 2013 Buyya et al. wrote [2] that "the cloud services discovery and selection process is done by human: a person looks to identify services that match his needs. We imagine in the future that, it will be way to find the cloud services that meets our needs by simply entering our requests in a marketplace that trades cloud services as utilities". In this paper, we present architecture for cloud services marketplace where cloud providers and consumers can exchange the cloud services as utilities. Proposed architecture automates the cloud services matching process and reduces the time and effort to find the appropriate service. In additional, we present Enhanced Percent Distance Similarity (EPDSim) algorithm for numerical matching between cloud services. Proposed algorithm is depending on user requested value and it is independent of any foreign attribute value. Four parameters are used for comparison: Number of matched services, Execution time, Average score and Recall. Experiment results showed that Enhanced Percent Distance Similarity algorithm reduced the execution time by 49% with keeping the same values for all other parameters.

II. Motivation

Desktop-as-a-Service (DaaS) is a virtual desktop handled by third party based on subscription fee model. User can connect and access anywhere at any time with a lot of other benefits. The following steps summarize the current process that consumer should follow to find the appropriate DaaS cloud service:
1- Select search engine: consumer can select one or more from the existing search engines.
2- Select search keywords: This step depends on the consumer skills to find the best keywords.
3- Explore all result websites.
4- Make a list of all available DaaS services and their attributes.
5- Compare and rank the listed cloud services.

6- Select the top matched service.

Table 1 shows the precision of Google search after exploring the first 150 results for each query. The total number of the results is huge but the total number of relevant results is very small. It's very clear that current process for finding the appropriate cloud service is a hard, uncompleted and time-consuming process.

Table 1 Google results for different keywords.

	Total results	Relevant in first 150 results	Revised relevant in first 150 results	Precision
Desktop as a service	1,97,00,000	73	32	0.2
Desktop as a service providers	65,50,000	47	29	0.19
DaaS cloud providers	1,82,000	25	12	0.08

III. Architecture for cloud services marketplace

Fig. 1 shows the proposed architecture for cloud services marketplace. It is divided into four layers and nine components as following:

- User interface layer contains three components: Query Receiver, Results Viewer and User Profile. It's responsible for receiving the user request and displaying the matched results for the users. User profile saves user preferences to show better results in future.
- Service Matching and ranking layer contains two components: Service Matching component is responsible for matching user request with available cloud services. Service Ranking component is responsible for ranking the matched cloud services based on user preferences collected by user profile component.
- Third layer contains services repository that stores all available cloud services.
- Fourth layer contains Registry Component and Crawler Component that collect service descriptions for cloud service providers.

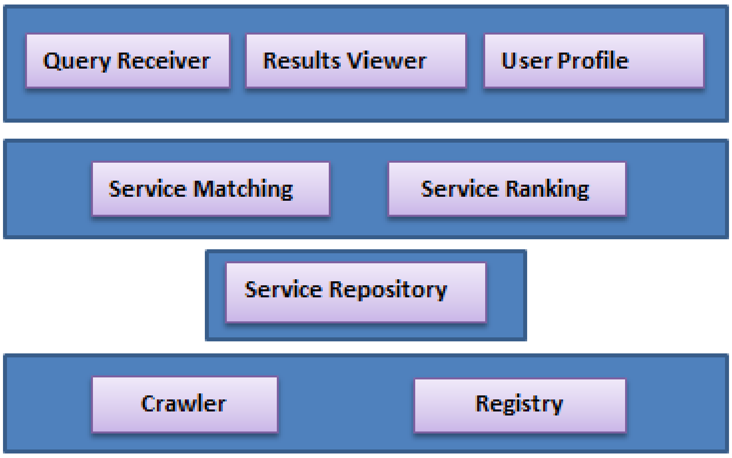


Figure 1 Cloud services marketplace architecture

IV. Related works

Researches done in the field of cloud service discovery and selection don't have a common understanding of the problem. . [3, 4, 5] considered cloud service discovery and selection problem as Multi-Criteria Decision Making (MCDM) and used different algorithms of MCDM . [6, 7, 8, 9] proposed Multi-Agents system based on search engine with three type of matching: similarity, compatibility, and numerical. [10, 11, 12] built and XML solution based on XQuery. In [13, 14] Semantic search engine is presented based-on SPARQL language which is not easy to be used by normal users. In [15] researches presented a cloud service discovery system based on QoS with four matching methods same matching, equivalence matching, containing matching and similarity matching. Unfortunately, all proposed solutions for numerical similarity are depending on external attribute values. Depending on external values increase the search time and reduce the system efficiency. In this work, we present an Enhanced Percent Distance Similarity algorithm that increase system efficiency by not depending on any external value.

V. Numerical Similarity

Semantic similarity determines how much a concept A is similar to concept B. On the other hand [16] numerical similarity measures how much an attribute value of concept A is similar to attribute value in concept B. as an example, if the consumer looking for cloud storage service with 750 GB and Amazon offer a cloud storage solution with 800 GB and Microsoft offer a cloud storage solution with 200 GB, then Amazon solution is numerically more similar to user request than Microsoft solution. Few researches had done in the area of cloud services numerical similarity. Kang and Sim [7] proposed an algorithm to calculate the numerical similarity between cloud services based on max and min value of this attribute in all available cloud services in the system repository. On the other hand, in [16] researchers presented an algorithm to calculate the numerical similarity between cloud services based only on the max value of this attribute in all available cloud services in the system repository. The following paragraphs explain in details the both numerical algorithm and present Enhanced Percent Distance Similarity algorithm which is not depending on any external attribute value.

5.1 Kang and Sim numerical similarity algorithm (KSSim)

As shown in Fig. 2 Kang and Sim numerical similarity (KSSim) algorithm calculates the distance between user request x and min value of this attribute a in all available cloud services in repository then it calculates the distance between user request value y and max value of this attribute a in all available cloud services in repository. Finally, the similarity between user requested attribute value and available candidate cloud service attribute value is calculated based on the following formal.

$$\mathrm{SNSim}(x, y, a) = 1 - \frac{|x-y|}{\max((\max a - x), (x - \min a))} \quad (1)$$

KSSim algorithm needs to calculate the max and min value in each available cloud service attribute to calculate the total similarity with user request that increase the execution time and reduce the system efficiency.

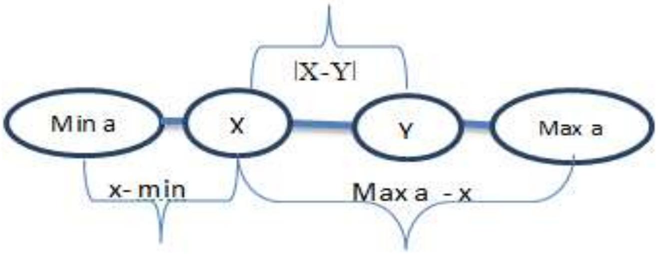


Figure 2 KSSim numerical similarity.

5.2 Max numerical similarity algorithm (MSim)

Researchers in [16] proposed numerical similarity algorithm for cloud services matching based only on the max value of cloud service attribute as following:

$$\mathrm{MSim}(x, y, a) = 1 - \frac{|x-y|}{\max a} \quad (2)$$

MSim needs only to calculate the max value of cloud service attribute to calculate the numerical similarity with user request.

5.3 Enhanced Percent Distance Similarity algorithms (EPDSim)

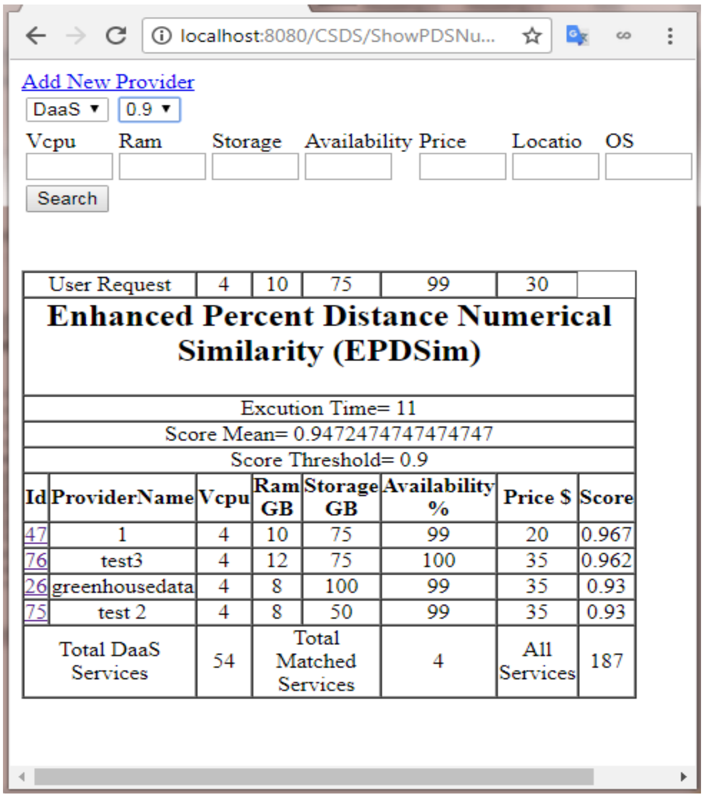
To overcome the limitation of the previous algorithms we present the Enhanced Percent Distance Similarity algorithms (EPDSim) that is independent of any attribute value as following:

$$\mathrm{EPDSim}\ (x, y) = \begin{cases} 1 - \frac{|x-y|}{x}, & y < 2nx \\ 0, & y \geq 2nx \end{cases}, \quad n = 1,2,3 \ldots \quad (3)$$

If $y < 2nx$ then y value is similar to x value and the similarity is $\mathrm{EPDSim}(x, y)$. On the other hand, If $y \geq 2nx$ then the distance between values is too big and similarity between values is zero, so cloud user needs to modify the query to get new results.

VI. Results and discussion

Experiments done on dataset that collected from real cloud services providers and we used laptop core I5 with 6 GB Ram. Fig. 3 shows the results for Enhanced Percent Distance Similarity algorithms (EPDSim) with user request parameters (VCPU=4, Price=30 USD/month, Storage=75GB, Ram=10GB, Availability=99%) , threshold =0.9 and n=2 . Four parameters are used for comparison: Number of matched service, Execution time, Average score and Recall.



User Request		4	10	75	99	30	
Enhanced Percent Distance Numerical Similarity (EPDSim)							
Excution Time= 11							
Score Mean= 0.9472474747474747							
Score Threshold= 0.9							
Id	**ProviderName**	**Vcpu**	**Ram GB**	**Storage GB**	**Availability %**	**Price $**	**Score**
47	1	4	10	75	99	20	0.967
76	test3	4	12	75	100	35	0.962
26	greenhousedata	4	8	100	99	35	0.93
75	test 2	4	8	50	99	35	0.93
Total DaaS Services		54	Total Matched Services		4	All Services	187

Figure 3 cloud service marketplace result with th=0.9

6.1 Number of matched services

As shown in Fig. 4 EPDSim and MSim algorithm showed almost the same value of matched services for all thresholds. On the other hand, MSim showed lower values for the number of matched services for all threshold. Number of matched services determines the number of the results that will display for the user. The bigger number of matched service means more chance for user to find the appropriate cloud service that match the request.

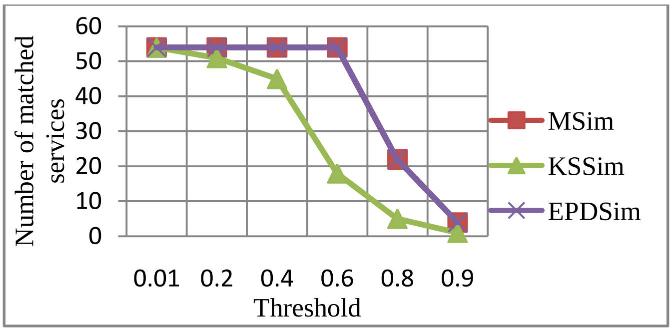


Figure 4 total matched services per threshold

6.2. Execution time

Execution time is the most critical parameter for the success of any marketplace. As shown in Fig. 5 for Enhanced Percent Distance Similarity algorithms (EPDSim) showed the lowest execution time for all threshold values. On the other hand, MSim and KSSim showed higher values for all thresholds. MSim algorithm need to calculate the max value of cloud service attribute to calculate the numerical similarity and KSSim need to calculate the max and min max value of cloud service attribute to calculate the numerical similarity which increase the execution time. EPDSim enhances the user experience with cloud marketplace by reducing the execution time.

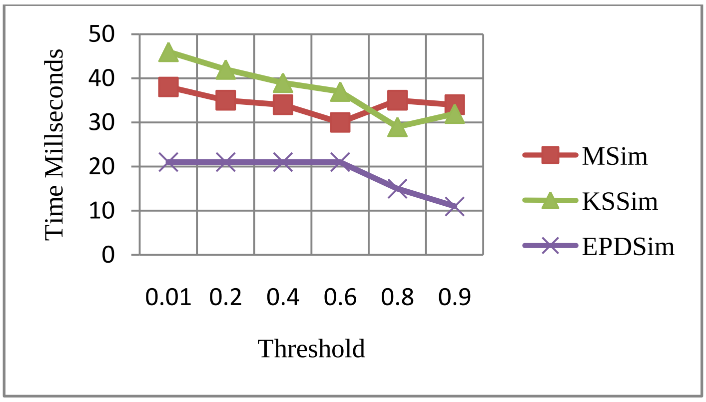


Figure 5 execution time per threshold.

6.3 Average Score

Average score affect the number of matched cloud services for each threshold. As shown in Fig. 6 KSSim showed lower average score for threshold lower than 0.6 while MSim and EPDSim showed higher values. On the other hand, all algorithms showed almost the same average score for the threshold values bigger than 0.6. The higher number for average score will increase the number of matched cloud services per threshold.

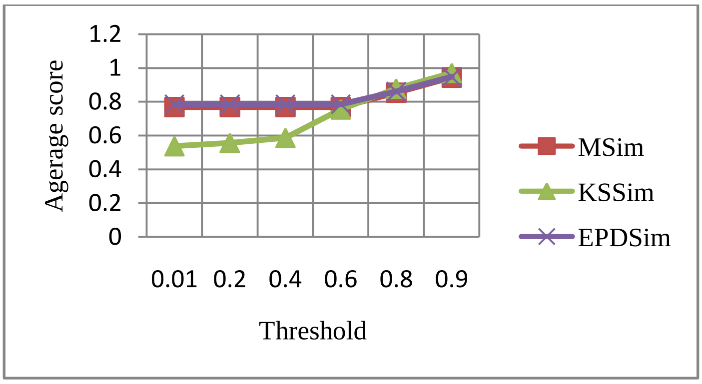


Figure 6 average score per threshold

6.4 Recall

Recall determine the effectiveness and completeness of any system that retrieve the information. It calculates the number of selected services out of the number of all available services [17]. As shown in Fig. 7 EPDSim and MSim showed the higher values for recall for all thresholds while KSSim showed lower values for all threshold because of lower number of matched services.

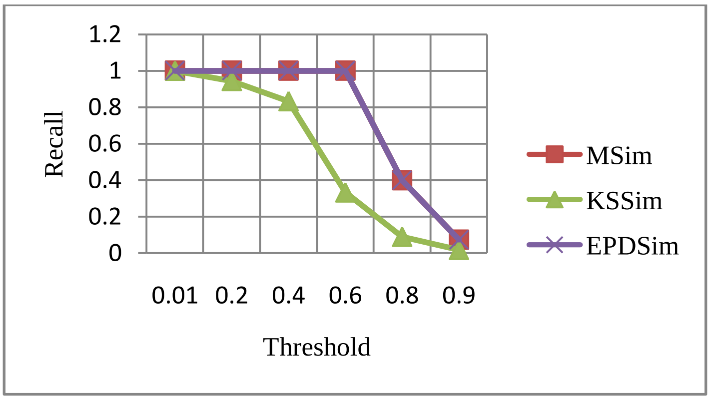


Figure 7 recall per threshold

VII. Conclusion

Cloud computing services become a part of our life as individual or industry. Finding the suitable cloud service by using search engine is a big challenge for the consumers. In this paper, we presented architecture for cloud services marketplace to automate the cloud services matching process. Proposed architecture enhanced the overall user experience and reduced the time and effort of find the appropriate service. In additional, we presented Enhanced Percent Distance Similarity algorithms (EPDSim) for numerical similarity of cloud services. Proposed algorithm enhanced the execution time by 49% and keeping the same values for all other parameters like : Number of matched services, Average score and Recall. As a future work, we plan to improve the user interface to accept the user request as a flat text or voice command.

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]. R. Buyya, C. Vecchiola, and S. Thamarai Selvi, "Mastering Cloud Computing: Foundations And Applications Programming", Morgan Kaufmann Publishers Inc., San Francisco, CA, 2013
[3]. Z. U. Rehman, O. K. Hussain, and F. K. Hussain, Iaas Cloud Selection using MCDM Methods, 2012 IEEE Ninth International Conference on e-Business Engineering, 2012, pp. 246-251.
[4]. M. Godse, S. Mulik, An Approach for Selecting Software-as-a-Service (SaaS) Product, IEEE International Conference on Cloud Computing, 2009, pp. 155-158.
[5]. J. Park , H. Jeong, The QoS-based MCDM system for SaaS ERP applications with social network, Journal of Supercomputing, October 2012, Springer US Publisher.
[6]. K. M. Sim, Agent-Based Cloud Computing, IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 4, 2012.
[7]. J. Kang, K. M. Sim, Towards Agents and Ontology for Cloud Service Discovery, International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2011.
[8]. J. Kang, K. M. Sim, Ontology and Search Engine for Cloud Computing System, International Conference on System Science and Engineering, 2011, pp. 276-281.
[9]. J. Kang and K. M. Sim, Cloudle A Multi-criteria Cloud Service Search Engine, 2010 IEEE Asia- Pacific Services Computing Conference, pp. 339-346, 2010.
[10]. S. Hasan, V. V. Kumari, Decentralized Architecture for Cloud Service Discovery based-on XML Representation, International Journal of Applied Engineering Research, Vol. 11 No.4, 2016.
[11]. H. Ma , K. Schewe , H. Xie, Using XML for cloud specification and XQuery for service discovery, Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, 2010.
[12]. Ruiz-Alvarez, A. and M. Humphrey, An automated approach to cloud storage service selection, in Proceedings of the 2nd International Workshop on Scientific Cloud Computing, 2011, pp. 39-48.
[13]. A. Tahamtan, S. A. Beheshti, A. Anjomshoaa, A. M. Tjoa, A cloud repository and discovery framework based on a unified business and cloud service ontology, Services (SERVICES) 2012 IEEE Eighth World Congress on, 2012, pp. 203-210.
[14]. M. Rodrguez-Garca, R. Valencia-Garca, F.Garca-Snchez, J. Samper-Zapater, I. Gil-Leiva. Semantic annotation and retrieval of services in the cloud, Distributed Computing and Arti_cial Intelligence, volume 217 of Advances in Intelligent Systems and Computing, 2013, pp. 69-77.
[15]. L. Liu et al., Ontology-based service matching in cloud computing, Proc. Fuzzy Systems (FUZZ- IEEE) 2014 IEEE International Conference on,2014, pp. 2544-2550.
[16]. S. Hasan, V. V. Kumari "Numerical Similarity Algorithms for Cloud Service Discovery and Selection System", International Journal of Intelligent Engineering and Systems, Vol.10, No.4, 2017, pp.226-234 DOI: 10.22266/ijies2017.0831.24.
[17]. V. Raghavan , P. Bollmann , and G. Jung, " A critical investigation of recall and precision as measures of retrieval system performance, ACM Transactions on Information Systems,
[18]. Vol.7, No.3, p.205-229, 1989.

International Journal of Engineering Science Invention (IJESI) is UGC approved Journal with Sl. No. 3822, Journal no. 43302.

Samer Hasan "Enhanced Percent Distance Similarity Algorithms for Cloud Services Marketplace" International Journal of Engineering Science Invention(IJESI), vol. 6, no. 9, 2017, pp. 01-06.