A multi agent broker framework and decision support for enhanced cloud computing discovery

Mohamed Galal Hafez Rady
College of Computing and Information Technology,
Arab Academy for Science, Technology & Maritime Transport
Alexandria, Egypt
mohamedgalalhafezrady@gmail.com

Mohamed Shaheen Elgamel
College of Computing and Information Technology,
Arab Academy for Science, Technology & Maritime Transport
Alexandria, Egypt
mohamed.shaheen@aast.edu

Osama Mohamed Badawy
College of Computing and Information Technology,
Arab Academy for Science, Technology & Maritime Transport
Alexandria, Egypt
obadawy@aast.edu

Abstract— It is remarkable that the tremendous expansion in the world of cloud computing in recent times, and in light of the Covid 19 pandemic that hit the entire world, the challenge has become more difficult and the intense focus on moving to the world of cloud computing is a necessity not a luxury. Hence, the biggest and most difficult challenge from both sides emerged. Therefore, this study attempts to provide an intelligent search scope to meet the clients' needs in finding the best service provider for their financial capabilities at the right time. It also gives a greater opportunity for service providers to display their services more clearly, by introduce multi-agent framework with decision support approach and recommendation concept, also a proposed cloud service discovery testbed introduced with management tools and an efficient GUI dashboard.

Keywords— Cloud Computing, Multi Agent, Service discovery, Cloud marketplace, Covid 19, Quality of Service (QoS).

I. Introduction

In light of the covid 19 pandemic, many organizations are focusing on launching a cloud computing business and turning the IT department from a cost center to a profit center via helping to monetize Information Technology ability on the cloud and present productive cloud services of their organizations, whether inside or outside them. The use of cloud computing, which is evident through network computing such as the web, and through the means that provide high-speed Internet services that work to increase and spread the various fields under distributed computing [1][2]. There are five service that make up cloud service life cycle (CSLC) such that requirements, discovery, negotiation, composition, and consumption [3][4]. The challenge now has become difficult in-service discovery, on the part of service providers around the world, where it has become necessary to expand the provision of services. On the other hand, the entry of a large segment of customers who recently realized the necessity and importance of cloud computing in light of the covid 19 pandemic. First: On the part of the customer in the search for his requirements of computing in light of the financial capabilities of each institution or company and the time of service provision. Second: On the part of the service providers, where competition has become more intense and reaching the customer is more difficult. The traditional searches in the famous search engines such as Google, Yahoo and Bing have proven that they cannot meet the needs of the customer because there are no standard designations for the requirements [5]. Also, they do not take into account the elements of service quality. Therefore, this study tries to provide an automated and classified framework based on multi agent. An agent is described as entity whose status is demonstrated to include intellectual components such as obligations, capabilities, beliefs and choices [6][7][8]. Agents run without the direct intervention of humans, and feature a few sorts of manage over their moves and inner state [9][10][11] that allow providers to offer their advertisements in formal and organized way also, allow consumers to search and fined their requirements. This framework supported by decision support concept to help consumers and providers with 'Goal Seek' concept. Also, proposed testbed was developed and implemented with management tools that allow based on parameters to enhanced testing procedure with GUI dashboard for monitoring system and performance.

The paper proceeds as follows: Section II discusses literature review and related work. Section III introduce proposed Intelligent Cloud Discovery System (ICDS). Section IV presents practical implementation. Section V introduces proposed Cloud Computing Discovery Service Testbed. Section VI simulation and results and section VII conclusion and future work.

II. Literature review

The main challenge in cloud service discovery segment, is ability to found the cloud services registered in the search engine's database via way of means of matching consumer's technical, functional and budgetary requirements [12].

[13] provided a multi-agent system with ontology, a prototype for cloud discovery service also developed. This study dependent of semantic to enhanced the cloud search but service discovery implementation not introduced to solve dynamic constraints. System accuracy evaluated by calculate recall and precision. [14] developed framework based on ontology that generating a fully automated and dynamic mobile agent allow the system to provide desired information on cloud without user interference. Also cloud service discovery technique not found in this study. multi-agent based distributed semantic service discovery introduced by [15], this study

focuses at semantic concept of three various points of view: measuring the runtime efficiency message overhead of the SmartCluster algorithm for service clustering, measuring the efficiency of the SmartDirect and SmartMap algorithms for service retrieval and measuring the accuracy of SmartDiscover and by OWL-S MX measure the presented system was compared with the commonly used service discovery algorithm. This study lake of multi-agent implementation and the interactions between different agent not introduced. OWL-S- primarily based totally semantic cloud service discovery and selection framework presented by [16], this study introduced effective matching by semantic approach of cloud services described which take into consideration quality of service for cloud requirements by described the service totally into functional such that (Inputs, Outputs, Preconditions and Effects) In tune with Non-Functional Properties (NFPs). However, this study totally dependent on shared ontology over all service providers where it must be used by offers and requests, this approach may cause further complications in the event that service descriptions are introduced partly in free text, also these problems could be appearing if some service providers making a mixing custom ontology. [17] presented an agent-based framework that designed and implemented in many layered to simplify the discovery services in cloud computing area. [18] proposed a model for cloud service that based on agents for helping in discovery, publication and selection, where through consumer interface customers can make a requests and matching their requirements. These two studies miss a clear and specific concept for handling service discovery and search algorithm in cloud computing area, also quality of service constraint that affect in results by overlapping with consumers requirement for example price and time. [19] mentioned approximately the significance of cloud computing service ontology and selection. Also, the cloud ontology database was built, which act as a warehouse for different advertisements that provided from providers registered in cloud service. This approach is likely to be problematic which service providers that taking charge of registering in internal cloud ‘inter-cloud registry’ in cooperating the problem of results in automatic service. [20] Presented the same approaches like previous articles but in different way using the new strong agent property called mobile agent, where this migrate agent possesses many features such as auto adaptability, synchronism and autonomy, also migration from one platform to another to perform assigned tasks is one of its most important features. In this article the author proposed a web services discovery system which consists of three parts. Web Interface Part, Cloud Computing Part and Mobile Agents Part. The Mobile agent part include some agent with different tasks. Interface agent (Ia) is used to transport the customer's request to the A region and display the customer's specified web services via the web interface, Researcher agent (Ra) is used to perform keyword-based words searches. The last agent is Transport agent (Ta) It has three functions that it can perform, the first of which is to deliver the customer's request to Area B, secondly, to transfer the modified web services by search agents to Area B, and finally, when the services are defined in Area B, they transfer all of them to Area A. The main advantage of using the mobile agent is dealing intelligently in terms of independence and automatic adaptation if fault tolerance occurs in either area A or B, where this agent can keep the customer’s request without prejudice to it until the problem of fault tolerance is resolved. Unlike the previous study [13],[15] that used a semantic technique and shared ontology to satisfy a user query, [20] designed an algorithm called comparison and filtering to solve the problem. [21] Try to introduce solutions to improve public cloud services (selection, discovery) infrastructure, specifically for hybrid cloud services with the help and use of ontology and agents. The basic idea organizing this study is to provide a strong and effective search system to find suitable cloud services for users and monitor the different cases of the cloud, especially hybrid ones. The monitoring process is carried out by an agent where this agent stores the information in the cloud services 's ontology, and this agent also allows the possibility of choosing different services for the public cloud, taking into account the structure and contents of the service and also the current case. The dynamic information of the private internal cloud is also collected and stored in the ontology database and by performing matching operations between both the ontology of the public and the internal clouds, it is possible to find search results for the appropriate cloud services for users. By using agents, the network's use of the public cloud is measured at varying and regular intervals, as the system can use the effective information that indicates the current state of the public cloud to serve users in search and selection processes and to provide good results to match users' requirements with the offered services. [22] Aims to build a service-oriented system where this system is based on meeting the requirements of users through support and maintenance, and all this is done through a wireless platform based on a structure that was built using the agent. This system consists of a group of agents that cooperate with each other to perform a set of tasks aimed at choosing the best offers presented that are compatible with the requirements of users while taking into account the quality of service, for example (Wireless User Agent) which is considered part of (Wireless Terminal) also (Agent Based Server) as this agent is the main crossing point in discovering and selecting the required cloud service. The results based on the simulation indicated that the presented system achieved success in choosing the best network resource, as well as fair use of resources. [23] This study provides an advanced concept for a previous study for the same researcher as it relies on cloud ontology and Semantic research methods. Two different concepts were presented in principle, where the first concept is based from the researcher's point of view on the simplest concepts of the cloud, and the other carries a faster and stronger concept. The correspondences between the different cloud offerings are discovered and defined through the means of concept and the logic of qualitative asset similarity. This study also presented two different types of recommendation concepts, where the difference between the two concepts is due to the expectation of feature rate. The first concept carries a maximum, unlike the last concept, which considers the average similarity between each of the cloud offers offered and the different demands of users. This system tries to act as a broker to match the needs of the users with the available offers. According to the results of this study, the latter concept is considered to achieve the best results in achieving the match rate and returns of cloud services for all users. The basic idea in this study is the Semantic not intelligent Agents and no information about the construction of the Agents.

III. PROPOSED INTELLIGENT CLOUD DISCOVERY SYSTEM (ICDS)

The proposed system as shown in Fig. 1 is based mainly on three main agents, Registration Agent, Broker Agent Manager and Decision Support Agent. In the beginning, each of the Providers and the Users register in the system, where each Provider places the advertisements by choosing from a classified list and predefined by the system, it is displayed from the database where each item is specified by a unique number and is not repeated in the same category, and in the case of new item there is a screen dedicated to entering it, according to the classification followed, all of this through an interface for the providers. Likewise, a user interface is available and performs the same task, but in this case, the user selects his/her requirements. Then the search process begins, where the task is transferred to Broker Agent Manager, who in turn assigns the task to one of the brokers, taking into account the load balance. The job of the broker is to match the requirements of the users with one of its providers, If the requirements do not match, meaning the broker fails to fined provider's advertisements that agree with the user's requirements, the broker agent sends the job to the Broker Agent Manager. Where the latter searches the database for another broker agent that has providers that meet the requirements of the user, in case it exists, the task transferred to the new broker to complete the task. In the event that the mission fails, meaning that there are no brokers with providers that meet the users requirements , the task is transferred to the Decisions Support Agent, which in turn enters the analysis phase where it does a what if analysis to nominate the closest broker that has a providers that is almost identical to the user's specification and sends the notification to User to approve or reject the new recommended providers advertisements. At the same time, the broker manager agent addresses the Decisions Support Agent to analyze the reason for the failure and notify the provider of the reasons for the failure so that the provider can modify according to the user's requirements and vis versa. In turn, Decisions Support Agent continues to perform a permanent search within the database where if a new provider subscribes, or if a provider modifies his advertisements to match the user's requirements, he sends a notification to the user and subsequently transfers the task to the new provider to complete the task.

A. Registration Agent (RA)

The registration agent contain two agent (User Agent and Provider Agent) that both agent response to provide a friendly graphical user interface that allow provider to register advertisements by selection from predefined dropdown list holds the cloud computing properties, for clarification, in case of cloud computing a sets of functional and non-functional configuration properties (e.g. CPU number of cores, RAM capacity, Hard Disk (HD) capacity and Operating system (OS)), which this parameters classified and stored in database. Also allow users to select his/her requirements. The search results appear in user interface also if there is any notification or recommendation will appear.

B. Broker Agent Manager (BAM)

Broker Agent Manager consists of three agents (Selection Agent, Recommendation Agent, Notification Agent) When users and providers register, Registration Agent send the task to Broker Agent Manager which generate new Selection Agent (SA) or assign the user/provider to free Selection Agent according to load balancing, which each new Selection Agent has its Queue that accept maximum 10 users and 10 providers this number for testing only but can be more or less according to environment specifications. Broker Agent Manager can generate infinite number of Selection Agent. When the Selection Agent finished its results reported to Broker Agent Manager, Algorithm 1 illustrates the selection process. In case of success the (BAM) send the result to (RA) that appear to user and provider, in case of failed, (BAM) message the Recommendation agent to start searching in database for another Broker Agent That has a provider meet the user requirements. Also report the result to (BAM) in case of success the (BAM) send the result to (RA) that appear to user and provider, in case of failed, BAM message Decision Support Agent (DSA) to start analysis when the last finished the (BAM) order the Notification Agent to propagate the analysis results. It will be discussed and clarified in the next section.

Algorithm 1: Selection Algorithm

Input: *User requirements REQ*
Output: *All potential providers PROV*

```
Initialize {providers} to Empty
For Each REQ in Requesters
  For Each PROV in providers
  IF REQ.CPU_CORE less than or equal to PROV.CPU_CORE
   AND REQ.RAM_CAP less than or equal to PROV. RAM_CAP
   AND REQ.OS_ID equal to PROV. OS_ID
   AND REQ.HD_ CAP less than or equal to PROV. HD_ CAP
   AND REQ.SETUP_DAYS greater than or equal to PROV. SETUP_DAYS
   AND REQ. MAX_PRICE greater than or equal to PROV. MIN_PRICE
  THEN add PROV into {providers}
  END IF
  END FOR
END FOR
```

C. Decision Support Agent (DSA)

Decision support Agent provide a concept of analysis and consists of three agents (What If? Agent, Reason for Fail Agent, Active Search Agent). This proposed agent (DSA) aims to increase the chance of user to find her/his requirements not only that matched or similar to requirements but also why not change his/her mind by helping from (DSA). 'What If? Agent' provide 'Goal Seek' scenario by changing in QoS parameters to increase search results without prejudice to users' requirements. There are two parameters can change (setup days, Maximum acceptable price) there is for testing only. Also, in parallel (DSA) calculate why request failed by 'Reason for fail Agent' and send to user/providers by Notification Agent. In final the request sends to 'Active search Agent', this agent act as a lazy agent that hold the request and when new provider register to system performs the request to it. Also, when any providers change his/her specification. Proposed system agents and tasks are summarized in Table I.

TABLE I. AGENTS WITH THEIR TASKS IN THE PROPOSED SYSTEM.

Agent		Tasks
Registration	***User/Provider Agent***	1-Registration 2-Add/Update and Delete Requirements/Advertisements 3-Display Results 4- Display Notification 5- Send messages to Active Search Agent in case new registration or update in Requirements/Advertisements
Broker Manager	***Broker Agent Manager***	1- Generate Selection Agent 2- Remote and control load balance 3- Assign tasks to (Selection, Recommendation and Notification) Agent 4- Send and receive messages from Decision Support Agent & Registration Agent
	Selection Agent	1- Match user requirements with provider using selection algorithm 2-Reports to Broker Agent Manager
	Recommendation Agent	1- Search in database for another Selection Agents that meet user requirements 2- Ranking results 3- Reports to Broker Agent Manager
	Notification Agent	1- Receive messages from Broker Agent Manager 2-Send Notifications to Users and Providers 3- Control notifications and send status to Broker Agent Manager
Decision Support	***What If? Agent***	1- Make a Goal Seek analysis 2- Send analysis results to Broker Agent Manager
	Reason for Fail Agent	1- calculate why request failed 2- Send and receive message from/to Broker Agent Manager
	Active Search Agent	1- Receive messages from Registration Agent 2- Make an active search 3- Send and receive messages from/to Broker Agent Manager

IV. PRACTICAL IMPLEMENTATION

The proposed system is designed and implemented using JADE (Java Agent Development Framework) based on JAVA and connected to database engine (Microsoft SQL server) on an Operating system (Windows 10), by using hardware having the following specifications: A Core i7 2.70 GHz CPU, 12GB RAM, and 1000 GB hard disk. JADE is a Java-primarily based totally and open-source software program framework for growing multi-agent systems and introduce services, like agent migration, message delivery and resource management, also the power of JADE platform is its support of the FIPA (Foundation for Intelligent Physical Agents) communication [24][25][26]. In this proposed system implementation, there is a set of containers that configure JADE platform, which tool up services for multi-agent operating system level. The agents perform cooperation among them to accomplish the specific tasks Fig. 2. The proposed JADE system consists of eight agents (UserAgent/ProviderAgent, BrokerManagerAgent, SelectionAgent, RecommendationAgent, NotificationAgent, WhatIFAgent, ReasonForFailAgent and ActiveSearchAgent). These agents run on JADE platform. Also, this study presented a testbed framework to test and experiment the system accuracy and comparison.

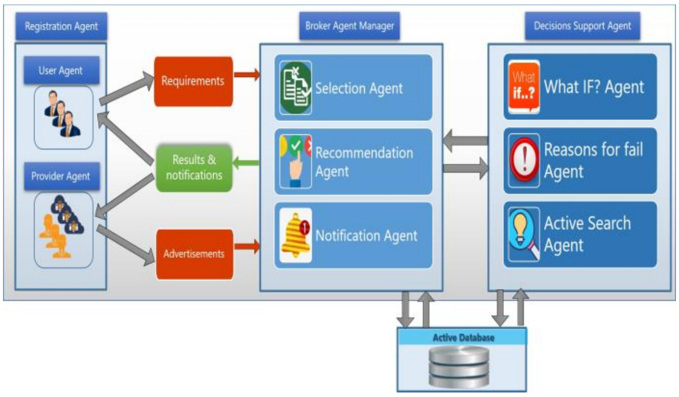


Fig. 1. Proposed Intelligent Cloud Discovery System (ICDS)

V. CLOUD COMPUTING DISCOVERY SERVICE TESTBED

This study proposed a framework testbed that designed and implemented using PHP 7.2.5 (Laravel framework). Laravel is web framework totally free and open source based on PHP language, designed and developed by "Taylor Otwell". Laravel using the MVC (model, view and controller) architectural pattern for developing and follow Symfony [27]. Laravel have many features such that, one of the powerful ways to connect and access database also modular packaging system with a devoted dependency supervisor. This testbed contains two tabs (Dashboard and Management).

A. Management:

that hold four functions to generate a new system for testing as shown in Fig. 3.

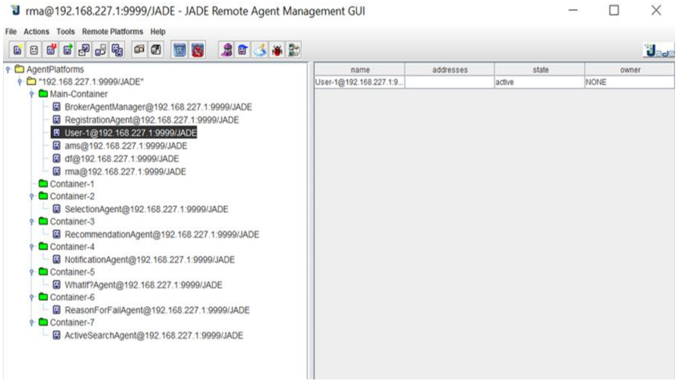
- *System initialization*, this function takes two parameters (number of Providers and number of Requesters), when calling this function delete all previous data and reset all ID's finally generate new registered Users and Providers according to the parameters number.
- *Generate Advertisements,* when execute this function the system automatically and randomly creates a collection of Advertisements and assigned to each provider, that created from previous step also the numbers of advertisements can be parameters
- *Generate Requests,* likewise Generate Advertisement but in this case for requesters.
- *Run System,* invoke and run ICDS all procedures at one time.

B. Dashboard:

Is a user interface-based information management that visually measuring, analyzing and monitoring the system by displays key performance indicators (KPI) to facilitate the analysis process as shown in Fig. 4. This information presented in perfect graphs that can help in testing and developments and also can be customized according to different logic to be used by researchers in this field.

VI. SIMULATION AND RESULTS

Using the proposed testbed, requests and advertisements are randomly generated, and they contain the information of the typical resources (CPU core, RAM capacity, HDD capacity, Operating System OS, maximum setup days and maximum acceptable price for users or minimum acceptable price for providers), all CPU, RAM, HDD and OS selected randomly from predefined and classified database, setup days selected from rang (1..10) and price range (10..100) for testing only.


rma@192.168.227.1:9999/JADE - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help

```
AgentPlatforms
  "192.168.227.1:9999/JADE"
    Main-Container
      BrokerAgentManager@192.168.227.1:9999/JADE
      RegistrationAgent@192.168.227.1:9999/JADE
      User-1@192.168.227.1:9999/JADE
      ams@192.168.227.1:9999/JADE
      df@192.168.227.1:9999/JADE
      rma@192.168.227.1:9999/JADE
    Container-1
    Container-2
      SelectionAgent@192.168.227.1:9999/JADE
    Container-3
      RecommendationAgent@192.168.227.1:9999/JADE
    Container-4
      NotificationAgent@192.168.227.1:9999/JADE
    Container-5
      WhatIf?Agent@192.168.227.1:9999/JADE
    Container-6
      ReasonForFailAgent@192.168.227.1:9999/JADE
    Container-7
      ActiveSearchAgent@192.168.227.1:9999/JADE
```

name	addresses	state	owner
User-1@192.168.227.1:9...		active	NONE

Fig. 2. Proposed multi-agent-based framework using JADE

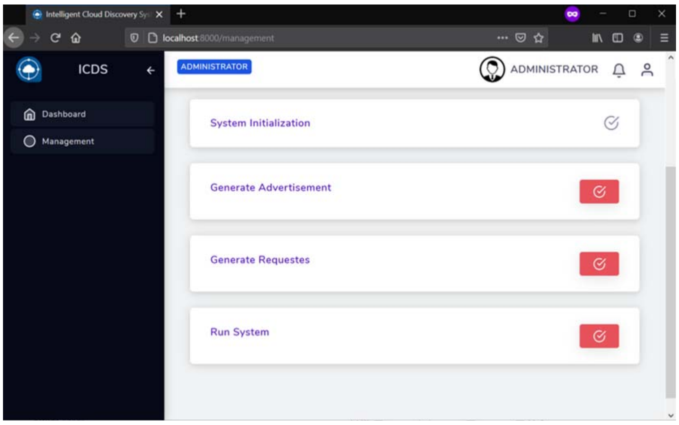


Fig. 3. Cloud Computing Discovery Service Management in proposed Testbed

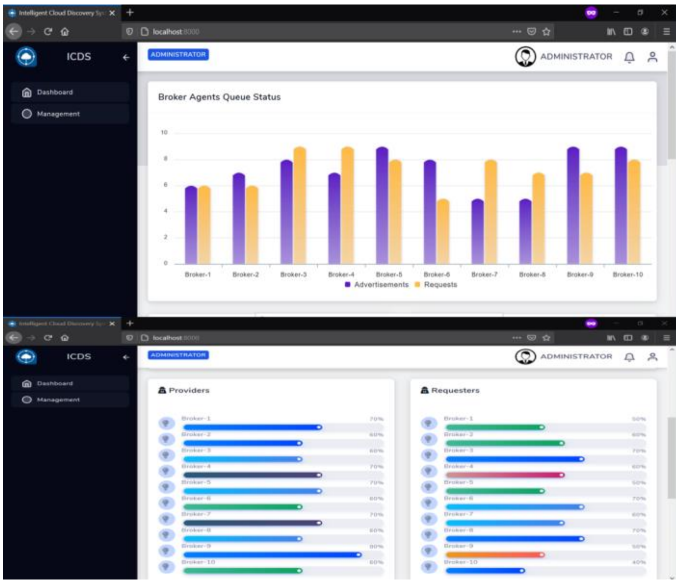


Fig. 4. Cloud Computing Discovery Service Dashboard in proposed Testbed

The experiments in this study aim to determine and discover the efficiency of the system by calculating both the average connection time and the success rate as the evaluation parameter. Connection time also known as Execution Time and is taken as one of the significant factors in information retrieval systems, which can be expressed as the amount of time from the time the user submits the request until the contact with the required provider [28][29], and the average connection time can be formulated with the following equation [23].

$$T_{avg} = \frac{\sum_{i=1}^{N}(TC_i - TS_i)}{N} \qquad (1)$$

Where TS_i is the time the user places the request, TC_i is the time the user contact with the required provider and N is the total number of all connections. As for the success rate, it is the calculation of the number of requests that have actually been executed (R_c) from the total number of requests submitted by users (R_n), equation can show as follow [23].

$$Success\ Rate_i = \frac{R_c}{R_n} \qquad (2)$$

The experiment examined the effect of different numbers of broker agents each time different combinations of provider agents and user agents were used. The experiment was carried out in two different approaches. Without involving the Recommendation and Decision Support (W/O REC & D.S.) against the same system while including the Recommendation and Decision Support (W/ REC & D.S) of investigation of the difference between them. The experimental results were collected by entering 400 user agents in addition to 400 provider agents. The numbers of broker agents changed as follows (1, 10, 30, 50). Each time, the average connection time and success rate are measured. The results are illustrated as shown in Table II.

TABLE II. COMPARISON OF AVERAGE CONNECTION TIME AND SUCCESS RATE.

No. of Broker Agent	**Average Connection Time (Second)**		**Success Rate**	
	W/O REC & D.S.	***W/ REC & D.S.***	***W/O REC & D.S.***	***W/ REC & D.S.***
1	502	750	0.42	0.88
10	102	204	0.33	0.84
30	50	100	0.27	0.8
50	30	90	0.21	0.77

From the results indicators, as shown in Fig. 5, for measuring the average connection time by comparing the number of broker agents with the average connection time, it is noticed that when using a greater number of broker agents, this leads to obtaining less connection time than its counterpart when reducing the number of broker agents. This can be attributed to the increase in the work load of broker agents if their number is small, or rather the increase in the queue capacity of each broker agent, as it is required to deal with a larger number of both the user agent and the provider, and this increases the time used. This is in contrast to the conclusion from Fig. 6, which monitors the success rate, as the success rate increases as the numbers used by the broker agents decrease. Secondly, by comparing the two different approaches used for the experiment, it is noticeable that using the approach with recommendation and decision support (W/ REC & D.S) achieved high success rates compared to other approach (W/O REC & D.S.) without recommendation and decision support, and this is due to the recommendation and decision support function that gives greater opportunities to the user agents in finding suitable provider agents with their needs, but of course, the function of recommendation and decision support will require more time, and this is what the results showed when comparing the two systems in terms of average connection time.

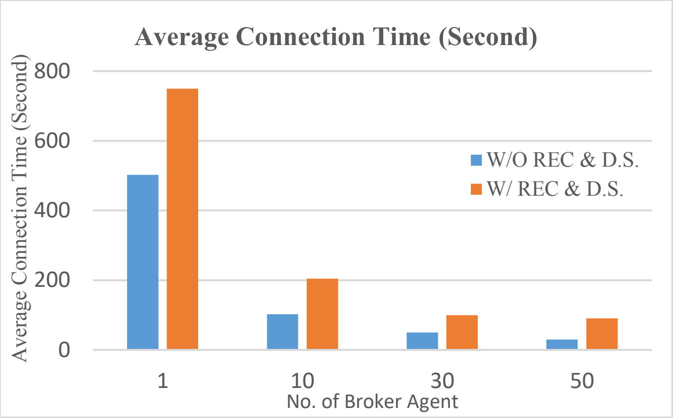


Fig. 5. Comparisone of Average Connection Time between two approches *(W/O REC & D.S. and W/ REC & D.S.)*

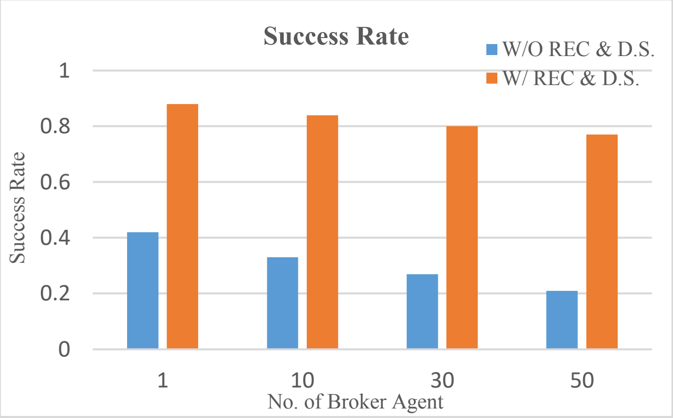


Fig. 6. Comparisone of Success Rate between two approches *(W/O REC & D.S. and W/ REC & D.S.)*

VII. Conclusion and Future Work

In this paper, a multi-agent-based framework has been proposed that uses an intelligent multi agents and decision support concept for cloud computing service discovery to solve the complexity of search and selection in a cloud environment by designed a classified cloud computing requirement stored in database to solve and minimize the gap between both service providers and consumers taking in its account quality of service. Additionally, this paper implemented a GUI and generic testbed that contains a management section that allow developers, researchers and cloud services marketplace to test their logic and performance, also, the dashboard and monitoring system introduced. The proposed intelligent cloud service discovery framework has been tested and compared in terms of two different approaches. Without involving the Recommendation and Decision Support against the same system while including the Recommendation and Decision Support using proposed testbed. From the results of experiments, the approach when using the Recommendation and Decision Support mechanism showed better performance and achieved high rates in performance measures such that efficiency, scalability and success rates compared to the other approach that does not use recommendation and decision support. As a future work, semantic concept and cloud services ontology will improve cloud computing discovery and selection. Also improving the efficiency of the method of recommendation and decision support to increase the efficiency of the system and reduce the connection time.

References

[1] Dinesh R, Ashwin R, "Multi-Agent based Cloud Services," in International Conference on E-Governance & Cloud Computing Services(EGov '12), 2012.

[2] "NIST," 22 December 2015. [Online]. Available: www.nist.gov/itl/cloud.

[3] K.P. Joshi, T. Finin, and Y. Yesha, "Integrated Lifecycle of IT Services in a Cloud Environment," in Proc. Third Int'l Conf. Virtual Computing Initiative (ICVCI '09), 2009.

[4] D. Talia, " Clouds Meet Agents: Toward Intelligent Cloud Services," Internet Computing, IEEE, vol. 16, no. 2, 2012.

[5] H. Samer, "Generic-distributed framework for cloud services marketplace based on unified ontology," Journal of Advanced Research 8 (2017) 569–576.

[6] Y. Shoham, " Agent-oriented programming," Technical Report STAN-CS-1335-90, Computer Science Department, Stanford University, Stanford, CA 94305, 1990.

[7] Y. Shoham, "Agent-oriented Programming," Artificial Intelligence, vol. 60, no. 1, pp. 51-92, March 1993.

[8] A. Kantamneni, L. Browna, G. Parker and W. Weaver, "Survey of multi-agent systems for microgrid control," Engineering Applications of Artificial Intelligence, vol. 45, pp. 192-203, 2015.

[9] M. Wooldridge. "Agent-based computing." Interoperable Communication Networks. Vol l, Number 1, pp. 71-97. January 1998

[10] Gilbert, D.; Aparicio, M.; Atkinson, B.; Brady, S.; Ciccarino, J.; Grosof, B.; O'Connor, P.; Osisek, D.; Pritko, S.; Spagna, R.; and Wilson, L. 1995. IBM Intelligent Agent Strategy, IBM Corporation.

[11] Sycara, K.P. (1998) Multiagent Systems, Artificial Intelligence, summer 1998, pp. 79-92.

[12] K. M. Sim, "Agent-Based Cloud computing," IEEE Transactions on services computing, vol. 5, no. 4, October-December 2012.

[13] Y.-S. Chang, T.-Y. Juang, C.-H. Chang and J.-S. Yen, "Integrating Intelligent Agent and Ontology for Services Discovery on Cloud Environment," in IEEE International Conference on Systems, Man, and Cybernetics, 2012.

[14] Y.S. Chang, C.T. Yang and Y.C. Luo, "An ontology based agent generation for information retrieval on cloud environment," J Univers Comput Sci 17(8):1135–1160, 2011

[15] S. Dasgupta, A. Aroor, F. Shen and Y. Lee, "SMARTSPACE: Multiagent Based Distributed Platform for Semantic Service Discovery," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, vol. 44, no. 7, 2014.

[16] L. D. Ngan and R. Kanagasabai, "OWL-S Based Semantic Cloud Service Broker," 2012 IEEE 19th International Conference on Web Services, Honolulu, HI, 2012, pp. 560-567, doi: 10.1109/ICWS.2012.103.

[17] Maheswari JU, Karpagam GR. Ontology based comprehensive architecture for service discovery in emergency cloud. Int J Eng Technol 2014;6(1):242–51.

[18] Reshma VK, Saravana BB. Cloud service publication and discovery using ontology. Int J Scient Eng Res 2012;3(5):563–7

[19] P. M. Ricordel and Y. Demazeau, "From analysis to deployment: a Multi-Agent Platform Survey," Engineering Societies in the Agents World. Springer, pp. 93-105, 2000.

[20] S. Hamza, B. Aïcha-Nabila, K. Okba and A. Youssef, "A Cloud computing approach based on mobile agents for web services discovery," in Innovative Computing Technology (INTECH), 2012 Second International Conference, 2012.

[21] T. Uchibayashi and B. Apduhan, "A Framework of an Agent-based Support System for IaaS Service Discovery," in 13th International Conference on Computational Science and Its Applications, 2013.

[22] Y. Zhou and Y. Yang, "An Agent-based Scheme for Supporting Service and Resource Management in Wireless Cloud," in Ninth International Conference on Grid and Cloud Computing, 2010.

[23] J. Kang and K. M. Sim, "Ontology-enhanced agent-based cloud service discovery," Int. J. of Cloud Computing, vol. 5, no. 1/2, pp. 144 - 171, 2016.

[24] R. H. Bordini, L. Braubach and M. Dastani, "A Survey of Programming Languages and Platforms for Multi-Agent Systems," Informatica, vol. 30, pp. 33-44, 2006.

[25] S. Prakash and A. Singh, "A SURVEY OF IMPLEMENTATION OF DISTRIBUTED MULTI AGENT SYSTEM USING THE JADE TECHNOLOGY," International Journal of Advanced Technology & Engineering Research (IJATER), pp. 71-75, 2014.

[26] K. Kravari and N. Bassiliades, "A Survey of Agent Platforms," Journal of Artificial Societies and Social Simulation, vol. 18, no. 11, 2015

[27] "The real-time community site Voten goes open-source". Laravel News. June 16, 2017. Retrieved April 30, 2020.

[28] H. Samer, "Generic-distributed framework for cloud services marketplace based on unified ontology," Journal of Advanced Research 8 (2017) 569–576.

[29] M. Parhi, B.K. Pattanayak and M.R. Patra, "Amulti-agent-based framework for cloud service discovery", In: Service Oriented Computing and Applications, 12:137–154, 2018.