Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

FIPA-based Reference Architecture for Efficient Discovery and Selection of Appropriate Cloud Service using Cloud Ontology

Ghulam Abbas*[1] | Amjad Mehmood[2] | Jaime Lloret3[3] | Muhammad Summair Raza[1] | Muhammad Ibrahim[1]

[1]Department of Computer Science, Virtual University of Pakistan, Rawalpindi, Pakistan

[2] Institute of Computing, Kohat University of Science Technology, Kohat, KP, Pakistan

[3]Integrated Management Coastal Research Institute, Universidad Politecnica de Valencia, Spain, de Valencia, Spain

Correspondence

*Jaime Lloret, Integrated Management Coastal Research Institute, Universidad Politecnica de Valencia, Spain. Email: jlloret@dcom.upv.es

Present Address

Virtual University of Pakistan, Rawalpindi Campus

Abstract

Cloud Computing (CC) is considered the latest emerging computing paradigm and has brought revolutionary changes in computing technology. With the advancement in this field, the number of cloud users and service providers is increasing continuously with more diversified services. Consequently, the selection of appropriate cloud service has become a difficult task for a new cloud customer. In case of inappropriate selection of a cloud services, a cloud customer may face the vendor locked-in issue and data portability interoperability problems. These are the major obstacles in the adoption of cloud services. To avoid these complexities, a cloud customer needs to select an appropriate cloud service at the initial stage of the migration to the cloud. Many researches have been proposed to overcome the issues but problems still exist in intercommunication standards among clouds and vendor locked-in issues. This research proposed an IEEE multi-agent FIPA (Foundation for Intelligent Physical Agent) compliance multi-agent reference architecture for cloud discovery and selection using cloud ontology. The proposed approach will mitigate the prevailing vendor locked-in issue and also alleviate the portability and interoperability problems in Cloud Computing. To evaluate the proposed reference architecture and compare it with the state-of-the-art existing approaches, several experiments have been performed by utilizing the commonly used performance measures. Analysis indicates that the proposed approach enables significant improvements in cloud service discovery and selection in terms of search efficiency, execution and response time.

KEYWORDS:

Cloud Computing, Multi Agent System, FIPA, Cloud Discovery and Selection, Cloud Ontology,

1 | INTRODUCTION

Cloud Computing (CC) is an emerging computing paradigm[1,2,3] provides infrastructure, platform and software 'as a service' on pay-per-use model. It is a large pool of easily useable and accessible type of resources such as software applications and development platforms, hardware, storage, online gaming and network applications, etc. that leads to increase the number of cloud users and service providers gradually. There are more than 260 Cloud Computing service providers on the globe and

[0]**Abbreviations:** ANA, anti-nuclear antibodies; APC, antigen-presenting cells; IRF, interferon regulatory factor

approximately 3.6 billion internet users[4]. With the advancement and growth in the Cloud-based solutions, the selection of an appropriate cloud service has become a complex task for a new cloud customer. The general search engines such as Google, Yahoo, and MSN etc. are not much effective to search similar cloud services but also having different nomenclature[5]. A cloud customer may face the vendor locked-in issue in an inappropriate selection of cloud service due the lack of portability and interoperability standards in the Cloud Computing environment. In such case, migration to any other cloud service provider may not be an easy task for the cloud customers. Thus, a standard solution is the need of the time to address these complex issues. This research proposes a FIPA-based reference architecture for the discovery and selection of appropriate cloud services efficiently and effectively by using the cloud ontology. FIPA is basically an IEEE standard organization of computer society for intercommunication of intelligent agents. FIPA is dedicated to the development of intelligent agents by openly developing specifications for agent-based applications. There are a number of standards for the communication among agents of the system; however, FIPA is the most advance standard of IEEE for agent communication and management[6]. The main aim of FIPA is to set standards for inter-communication among software agents for maximize interoperability within and across agents by a set of specifications. The proposed architecture will help the cloud customers in the discovery and selection of the required cloud services at the initial step of the migration to cloud which will mitigate the prevailing vendor locked-in issue as well as alleviate the portability interoperability problems in cloud environment. This research paper is organized of five sections. The related literature is reviewed in Section 3. Materials and methodology for the proposed solution is discussed in Section **??**. Experimental evaluation, results discussion and case studies are presented in Section **??**. Summary including conclusions and future work is delineated in Section 5.

2 | LITERATURE REVIEW

M. Parhi et al. in [1] proposed a semantic cloud service repository where the cloud service providers publish their services semantically by using cloud ontology ensures the standard vocabulary for specific cloud service specifications unambiguously. On the basis of the proposed framework, some experiments are also performed which shows better response time and accuracy measures. The ontology used in the proposed work is confined only to the IaaS layer which helps only for cloud service providers. A. Mehmood et al. in[7] proposed a multi-agent architecture for secure communication among open cloud architectures which is helpful to solve the issues of interoperability and portability. All agents of the system are registered through a registry service by Registry Agent which help the MA (Multi-agents) to interact with each other and also provides security. J. Kang et al. in[8] proposed agent-based search engine that is capable of searching the cloud services as per requirement of cloud customer over the internet. A cloud ontology concept is used to define the specifications and service reasoning. Three kind of reasoning techniques are used in the paper that are: Numerical reasoning, Similarity reasoning and Compatibility reasoning. The main contribution of the research work is to build a user-friendly environment for searching a required cloud service in unified format by using the cloud ontology. Haan T. et al. in[9] proposed a cloud service discovery system (CSDS) for cloud service search facility over the internet by using the cloud ontology. The architecture builds an agent-based discovery which need the cloud ontology for the process of information retrieval of information about cloud services. This process consumes much time to discover the required cloud service to a cloud user. R. Khan et al. in[10] presented an architecture wherein focuses on the issue of interoperability and portability by using intelligent agents and XMPP (Extendible Messaging and presence protocol). Mechanism for mobility and intelligence among agents also discussed. Chang Y. et al. in[5] proposed a framework for mobile agent-based service discovery integrated with ontology and prototype for the service discovery of cloud. This framework helps the cloud users in the process of discovery and selection of cloud services. However, this framework has lacks of implementation for the service discovery to help dynamic constraints and predilections for a specific cloud user. Nagan L. et al. in[11] proposed a service discovery and selection mechanism for semantic cloud based on OWL-S that can carry out the processes of semantic matching of different services of cloud which are dynamically complex and constraint. This mechanism consists of four functional descriptions: Input, Output, preconditions, and effects (IOPE) and non-functional properties (NFP) describes the services. However, the offer and requests related to the services must be represented through a shared ontology to all the service providers which may lead further complications and constraints in service providing in case of partly in free text and as well as if it appears that some cloud service providers integrate for customized ontology. To address these issues, concentration on the ontology learning with the alignment mechanism is necessary for the cloud environment. Tahamtan A. et al.[12] proposed a united business service and cloud ontology for querying capabilities to establish the mapping between the business function and cloud service. However, this mechanism lacks accuracy in matching the business functions in order to maintain the quality of service (QoS) parameters. Gueferl R et al.[13] proposed a mechanism for cloud service discovery based on cloud ontology with their composition of services for complex requirements. The mechanism used in the proposed system can resolve service requests by the user even complex requirements. However, the service composition is a major issue in Cloud Computing. The mechanism explained in a formal way of service-based method as input parameter and output the functional parameters but the parameter for QoS (Quality of service) is not considered for service selection and composition. Uchibayashi T et al. in[14] presented an effective framework for cloud service discovery and selection in a hybrid Cloud Computing environment. For enhancement of the cloud service discovery and selection, they used a heuristic cluster mechanism, however, the methodology for clustering based on the keyword does not consider semantics which may create difficulty in complex and semantic cloud service discovery for the users. Cretella G et al. in[15] proposed a semantic-based methodology that produces well service resource configuration on description based on semantic technology for supporting cloud service developers in terms of discovering functionalities, application development, APIs (Application program Interfaces) and resource required for the application development. K.M. Sim in[16] presented the idea of the agent paradigm to build software tools and test-beds to manage the clouds. The idea has two perspectives: one perspective of Cloud Computing in which contributes several novel approaches by facilitating cloud service discovery, negotiation and service composition and another perspective is a multi-agent system which demonstrates :(1) the application for automating cloud composition, (2) conversion of complex negotiation to commerce, and (3) building a cloud search engine through Cloudle. The Cloudle consists of a Service Discovery Agent (SDA), a Cloud Ontology and a Web interface. By using the Cloud ontology, the SDA has three reasoning: Similarity Reasoning, Compatibility Reasoning and Numeric Reasoning additional with the price and time slot module. SDA (Service Discovery Agent) determines the level of matching of consumers and providers. The service rating module rated the similarity of the functional and technical specifications of the providers and consumers. The output of the system consists of an ordered list in terms of the matching of similarities of functional, technical, and budgetary constraints requested by the consumers. To automate the cloud service composition, the agent adopts a focused selection contract net protocol (FSCNP) for selecting the service of cloud and use SCTs (capability tables) to record the list of cloud agents and services. Opara-Martin et al.[17] presented a critical analysis of the vendor lock-in and its impact on the Cloud Computing migration. A survey revealed that the Cloud Computing paradigm has greatly impacted the IT migration and business applications due to the vendor locked-in issue. The vendor lock-in issue raided due to the lack of standards of interoperability and portability resultantly it embedded in the complex, technological interdependent and heterogeneity in the cloud systems. To alleviate the issue the authors proposed that (1) create awareness of the dependencies and complexities which exist in the cloud environment; (2) assess technology implementation such as APIs and contract for potential areas of vendor lock-in issue; (3) select the standardized services provided by the vendors and platforms and, (4) by ensuring there is sufficient service of portability. All of the above literatures still face the standardization issue for inter-cloud communication. We have used the FIPA-based intercommunication standard which will help to solve the issue by inter-agent communication standards.

3 | MATERIALS AND METHODOLOGY

The proposed reference architecture contains a Multi-Agent System (MAS). The MAS consists of number of agents for different task(s) in the cloud domain such as similarity reasoning, discovery, selection, registry, ranking etc. Agents used in the proposed MAS are Customer Agent (CA), Discovery Agent (DA), Selection Agent (SA), Ranking Agent (RkA), Registry Agent (RgA), Reasoning Agent (RA)and Provider Agent (PA). The agents used in the proposed architecture are divided into two groups:

- Agents that act on the behalf of the Cloud Consumer i.e. Customer Agent, Discovery Agent, Selection Agent, Ranking Agents and Reasoning Agent.

- Agents act on the behalf of Service Provider i.e. Provider Agent and Registry Agent.

Agents that act on the behalf of Cloud Consumers locate the services, contract services, and receive services. And on the other hand, for agents that act on the behalf of the Cloud Provider manage the access to services and the contract are fulfilled. Each agent in the proposed framework have a dynamic behavior based on the service delivery with the FIPA based compliance specification[18,19].

3.1 | CONSTRUCTION OF CLOUD ONTOLOGY

The cloud ontology (CO) contains the knowledge of every entity used in the cloud domain such as attributes, concepts, axioms and relationship among the objects in the cloud environment. Attributes of CO contains all the functional attributes of cloud service providers such as Application Domain, Formal Agreements, Memory Type, Supported OS etc., and the QoS includes all the non-functional attributes like availability, price, rating, response Time etc. These cloud service attributes are used by Cloud Providers like RackSpace, IBM and Amazon EC2. It also consists of the information about the cloud service attributes, performance, and QoS (Quality of Service) attributes of cloud providers[20,21]. It specifies and describes the detailed explanation of every task of the cloud environment. The efficient and effective discovery of cloud service depends on the functionality of the CO. It supports the agents in the domain without any misinterpretation and intercommunication among agents in the framework.

There are many parameters as data properties used in the cloud ontology that represent the functional and non-functional attributes and parameters which are OS, CPU, ECU, price, RAM, response time, bandwidth, service name, service provider's name, status, rating, disk space, URL, rank, availability. These parameters are considered and matched while the discovery of cloud service by a cloud user and then display the list of most appropriate cloud services on the user interface[22,23,24,25].

3.2 | WORKING OF THE PROPOSED MULTI-AGENT SYSTEM

In this section, a complete working methodology of our proposed multi-agent system and ontology for cloud service discovery and selection is presented. With the help of the schematic diagram (Fig. 1), it is very easy to see the working of the multi-agent system of the reference architecture.

The workflow of the proposed architecture is presented in Fig. 4 which shows the working of proposed system. The step by step working of the proposed reference architecture is presented as follows:

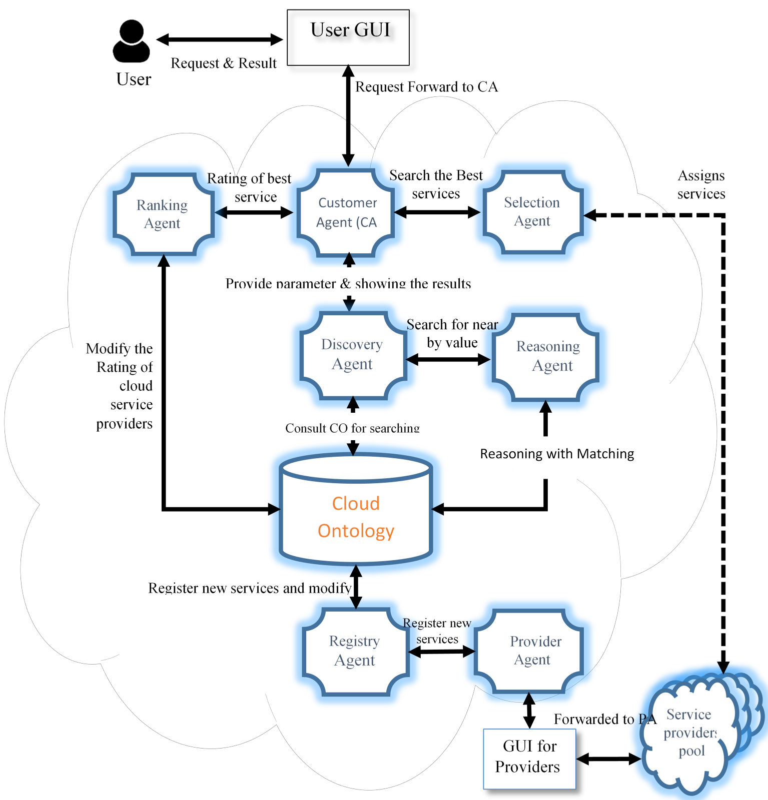

FIGURE 1 Proposed multi-agent reference architecture

Algorithm -1: Matchmaking of cloud resources

```
INPUT: set of requests of cloud resources CRq services.
OUTPUT: set of semantic matched cloud resources S1 a sub
class of S
function Semantic matching (CRq)
S=set of all cloud services
for each S do
   score=0
   CRs=get resources (s)
   foreach r1 ∈ CRs do
      foreach r ∈ CRq do
         score= score +similarity(r,r1)
      end
   end
   if score > 0 then
      S1=S1 U {S}
   end if
end
return S1
......................................
function similarity(r,r1)
   C= common_features (f1 & f2)
   UF1=Unique_features (f1 & f2)
   UF2=unique_fearutres (f1 & f2)
```

$$\text{similarity} = \frac{|C|}{|C| + |UF1| + |UF2|}$$

```
         return similarity
```

FIGURE 2 Algorithm-1

Step 1. On the reception of user request from GUI (Graphical User Interface) which contains number of functional and non-functional parameters in order to search for a required cloud service, is forwarded to Customer Agent (CA) for interpretation of the message. The CA interprets the user request on the basis of primarily analysis on the requirements. There are three ways of agent interaction with the CA i.e. Discovery Agent (DA), Selection Agent (SA) and Ranking Agent (Rk.A). These interactions are mentioned as follows.

Step 2. Once the requirements are formally presented by the user, the Customer Agent forward it to the Discovery Agent (DA). The DA consults with the cloud ontology to find the matching service as requested by the cloud user. The discovery is based on the matching of the services with the cloud user requirements and service specifications. For instance, functional and non-functional requirements of the user such as storage, CPU, memory size, bandwidth, cost. On the successful discovery of appropriate cloud services, the DA send the information to CA for the selection by the SA. Step 3. In case the DA fails to discover, it sends the request to the Reasoning Agent (RA) for semantic matching process consulting with the cloud ontology which give reason about the relations among the cloud services. After the match making process, it returns back the reasoning to the DA. For the similarity matching process between the cloud resources a detail matchmaking, an algorithm has been designed as Algorithm-1. A set of requested resources are provided as an input in CRq from the resources of cloud services CRs. The algorithm Iterates all available resources of cloud service S and calculate the relevancy with the requested resources. Another function is used in the algorithm for the checking of similarity of both the resources.

Algorithm -2: CA Behavior

```
  INPUT: set of string for cloud discovery.
OUTPUT: list of discovered services
Begin
CA received ACL message from GUI and send the
ACLmessage.CFP to DA   // CFP=Call for Proposal
ACL_Message_Reply=Reply from DA
          If ACL_Message_Reply is not empty then
             {
             Call for GUI and print the result list
             CA send QoS parameter to SA by
   ACL.message. cfp
             ACL.Message cfp = new ACL.Message.
   cfp
                   For each SA ∈ SA list do
                             CFP.addReciver[S.A]
                   end for
              }
            ACLmessage send to Rk.A for updation of
          rating of the   particular cloud provider.
          Else if
                   Defuse the request
          end if
End
```

FIGURE 3 Algorithm-2

Step 4. After the completion of the searching process by the Discovery Agent, it returns back the list of matched services to the CA. The Customer Agent forwards the matched services to the Selection Agent for the selection of the most appropriate service from the list provided by the DA. During the Selection process, the Selection Agent (SA) generates a list of all cloud services. Step 5. On the completion of selection process, the CA send ACL message to Ranking Agent (Rk.A) for updating the rating of the respective cloud service provider. The rating of the particular cloud service plays an important role in the calculation of the dynamic rating of a cloud service. The behavior and action of Customer Agent (CA) is described in Algorithm-2.

Step 6. The Provider Agent (PA) interacts with the friendly GUI (Graphical User Interface) that consists of many functional and non-functional attributes. The service provider needs to updates the QoS parameters. The PA also provides a GUI o the Cloud Service Provider for new services, edit and delete the service. Step 7. The Registry Agent (RgA) will validate the results forwarded by the PA (Provider Agent). The Rg.A also validates the format and range of the values in the cloud ontology.

3.3 | PRACTICAL IMPLEMENTATION

JADE (Java Agent Development Framework) is a software framework for agent application and compliance with FIPA specifications for interoperable multi-agent systems. In the proposed architecture, the JADE is used to demonstrate the proposed

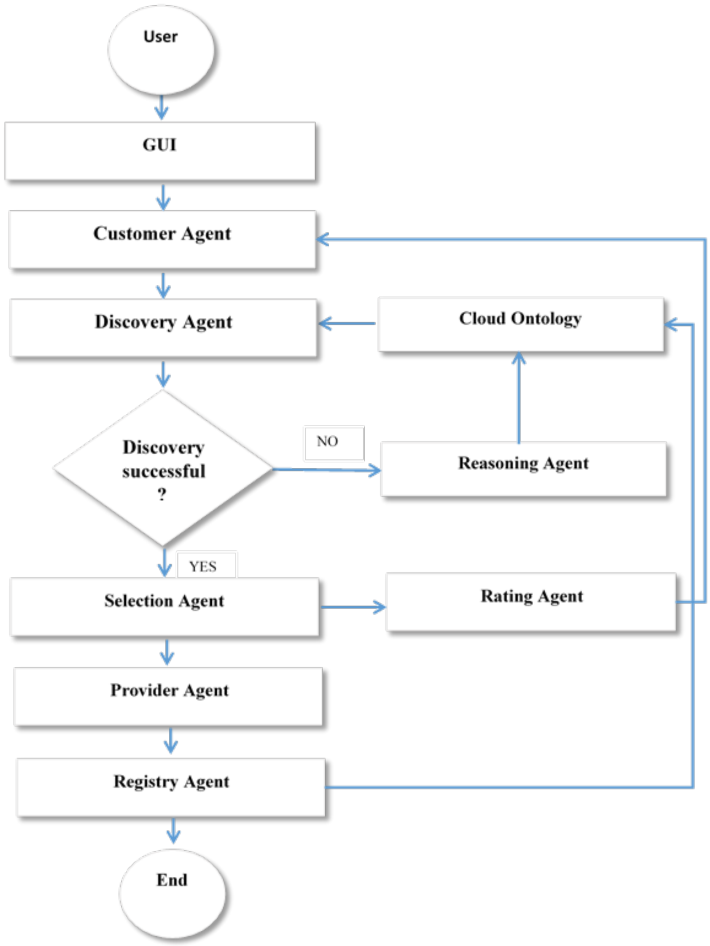

FIGURE 4 Flow diagram of the system

multi-agent framework for discovery and selection of the cloud service. The Jena Semantic web library is used for the creation of cloud ontology which also allows the interconnection of agents and CO. Jena API also includes an RDF API to access the RDF (Resource Description Framework). The JADE comprises of a set of containers which provide services of system level for multi-agent system operation. Each agent container has an RMI (Remote Method Invocation) registry that manages and controls the life cycle of the agents by creating, deleting, suspending and resuming them. The proposed MAS for every agent is constructed as presented in the Fig. 5.

There are seven containers and a main container which consists of registry agent, AMS, DF and RMA. The JADE application provides a complete service to the agents such as message passing, migration to agents and resource management of the agents. The nature of task of each agent depends on the behavior of agent which need to define in the JADE system. Agent's behavior represents logical task and threads for software implementation. The behavior Customer Agent and important agent of the reference architecture are presented in the Algorithm-2.

3.4 | CLOUD SERVICE DISCOVERY

To interact with the framework, a cloud customer needs to communicate with the user-friendly web interface (GUI) where the user will have to register. For cloud discovery, the user needs to choose the functional parameters of the required request as shown in Fig. 6. The functional parameters include the type of service, price range, memory range, bandwidth range, ECU range, CPU speed range, storage range and operating system information. After the submission of the request by the cloud customer,

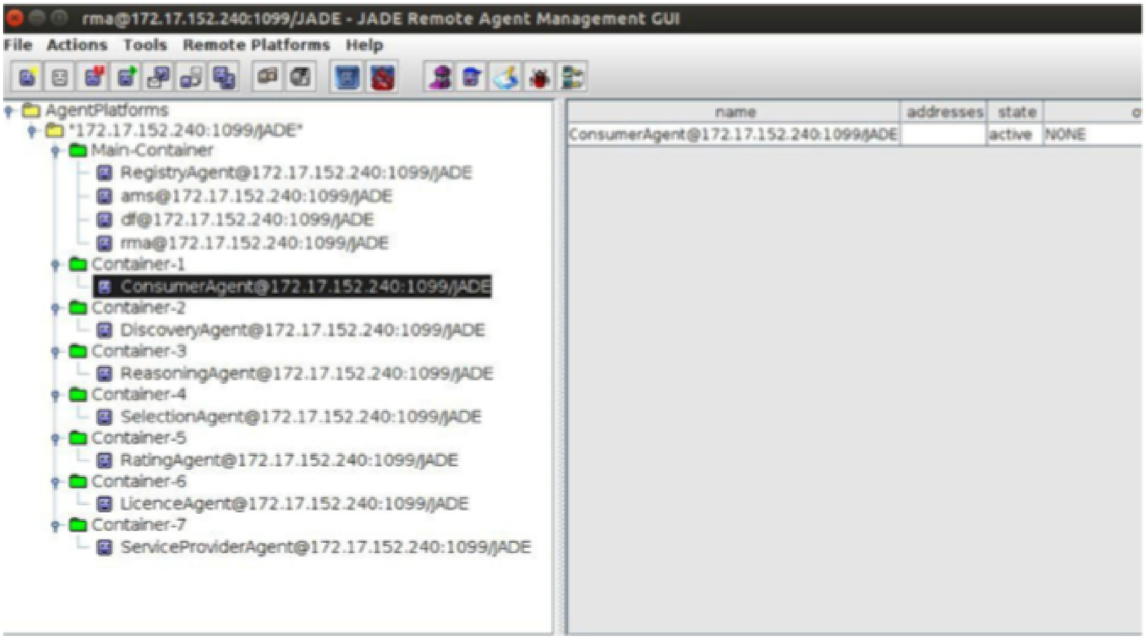

FIGURE 5 JADE application for the proposed MAS architecture

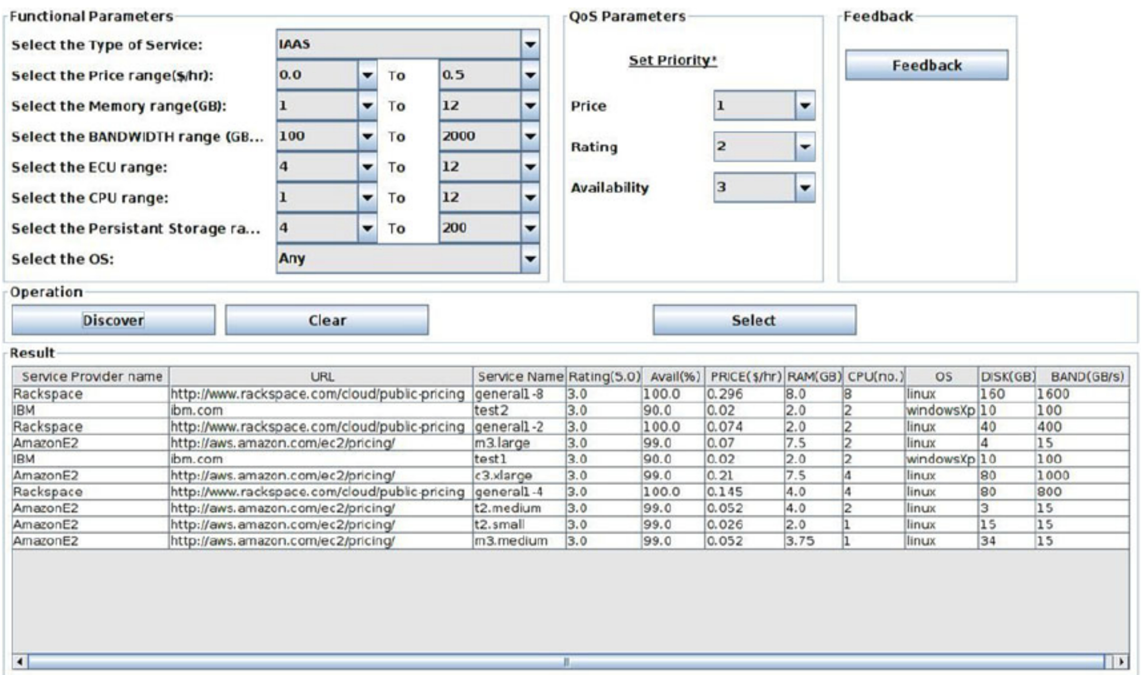

FIGURE 6 User interface for cloud discovery

the system automatically extract the required range available at the bottom of the screen. An ordinary example of the functional parameters and result is shown in Fig. 6.

3.5 | CLOUD SERVICE SELECTION

After the discovery of all possible cloud services range, a cloud user can select the most appropriate service from the list discovered services as shown in Figure. 7. A sample window is presented in the below diagram for the selection of the most appropriate cloud service by cloud user as given in Fig. 7.

4 | EXPERIMENTAL EVALUATION

To evaluate the performance of the proposed reference architecture, a set of experiments have been conducted to compare with the existing state of the art approaches. These experiments are conducted based on values mentioned in the QoS attributes.
Experiment No.1 In this experiment, the proposed approach is evaluated and compared with the contemporary approaches by using the performance measures i.e. Precision, Recall and F-measures. In the context of information retrieval process, the performance measures are evaluated on the basis of retrieved information/documents and the relevant information/documents. Retrieved information is the list of information produced by a web service upon a query and the relevant information is the relevant list of the topics[26].

Precision: It is defined as the fraction of the retrieved documents to that of relevant documents shown in the given formula:

$$Precision = \frac{|\, relevantinformation \cap retrievedinformation \,|}{|\, retrievedinformation \,|} \tag{1}$$

In order to evaluate and compare the performance, 20 iterations are done to search a specific cloud service on both ontology-based and RDB approaches. Table 1 shows the results.

TABLE 1 Experimental values

Approaches	Retrieved Documents	Relevant Documents
Ontology based	20	18
RDB approach	20	12

The value of precision on the basis of the above values is calculated as: Precision of ontology-based approach = 90

Recall: We can calculate the Recall on the above values by the given formula:

$$Recall = \frac{|\, relevantinformation \cap retrievedinformation \,|}{|\, relevantinformation \,|} \tag{2}$$

Recall of ontology-based approach = 95 and Precision of RDB approach = 70

F-Measure: The value of F-measure can be calculated by the following formula

$$F = \frac{2 \times (precision \times recall)}{(precision + recall)} \tag{3}$$

F-measure of ontology-based approach = 92 F-measure of RDB approach = 65The graphical representation of the above observations is as:

The results in Fig. 8 clearly shows the improved results considering the search efficiency of the proposed ontology-based approach against the RDB approaches. These experimental evaluation and analysis show significant improvements in the performance of proposed system. This indicates that the proposed approach is enable to discover the most relevant and required cloud service to a cloud customer in a timely manner.

Experiment No.2 In this experiment, another important parameter of response time is considered. Response time in computing system is the amount of time it takes to respond to a request of any service. Both the agent based and non-agent-based system is considered for evaluating the response time.

Graphical representation of the comparison of response time of the Table 2 for both approaches is shown in the below Fig. 9.
:

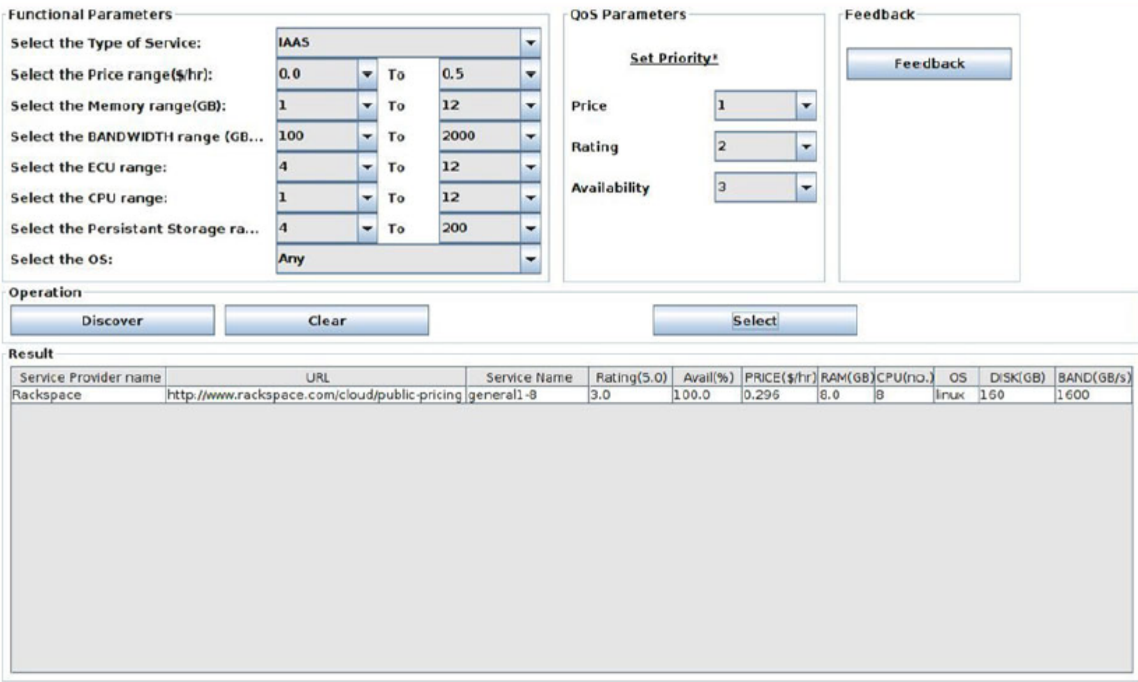

FIGURE 7 Interface for Cloud Service Selection

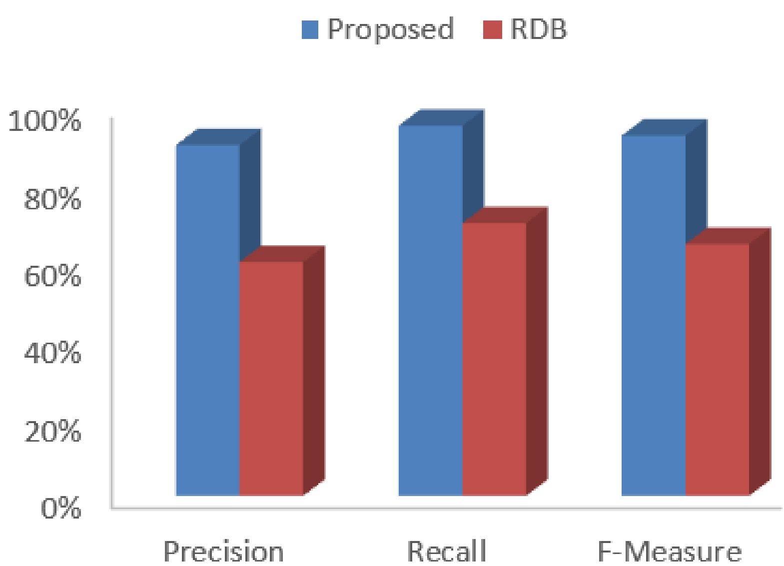

FIGURE 8 Performance measure of search efficiency

Fig. 9 exhibits that the response time of agent-based system is less as compared to non-agent-based approaches. This means that the agent-based system is more efficient than non-agent-based system.

Experiment No.3 In this experiment, another important factor of computing system i.e. Execution Time which is the time to complete a given task or query is measured and evaluated. Execution time to discover an appropriate cloud service is obtained

TABLE 2 Comparison of response time

No of user requests	Response Time in second	
	Agent-based Approach	Non-Agent based Approach
1	0.000542	0.000635
2	0.000985	0.001145
3	0.001375	0.001984
4	0.001955	0.00331
5	0.002821	0.004512

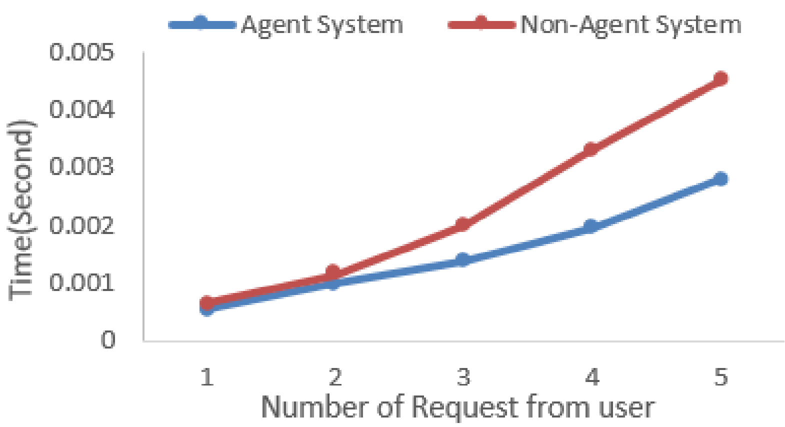

FIGURE 9 Comparison of response time between Agent-based and Non-Agent System

in this experiment. The comparison between ontology-based approached and relational database (RDB) approach by taking 25 iterations is carried out. The obtained results of the comparison are plotted in the Table 3:

The graphical representation of the above comparison is also depicted in the Fig. 10:

Fig. 10 shows that the ontology-based approach is more efficient than traditional approach in terms of execution time.

Case Study: Consider a cloud customer wants to discover and select an appropriate cloud services to shift his/her precious data from traditional computing to cloud computing paradigm. We will see the scenarios of problems, procedure, process and real time experiments of both existing and the proposed approaches by following scenarios:

Scenario-1: Suppose a user subscribe to a service provided by one of the Cloud services providers. In future, the user may not be satisfied with the availed service and want to select the same service from other Cloud vendor and thus migrate the data to the other Cloud. In existing scenario, the user will not be able to directly migrate the application to other Cloud service provider as the Cloud APIs, architecture, and other attributes (for the same service) by different service providers will not match. In this case, the user will face the locked- in issue. To overcome this issue, this work contributed a unified Cloud Ontology that will enable all Cloud service providers to use the same API with same attributes and thus in future it will be easy to migrate from one Cloud service provider to another. The highlighted (i.e., red square) section of the Fig. 11 provides the information about how a user can select a specific service from the set of services offered by different Cloud vendors.

Scenario-2: In the existing Cloud systems, the user's feedbacks are gathered for each of the Cloud service provider independently; however, a holistic approach is needed where rating can be performed to compare the services provided by each of the Cloud vendor. Thus, the proposed system allows the users to provide their feedback about the services provided by each Cloud service providers, and each user can record their feedback and see the comments thus making it appropriate to the select the best service they require. The highlighted (i.e., green square) section of the Fig. 11 provides the information about how a specific service is rated based on the user's feedback.

TABLE 3 Comparison of execution time

No of Iterations	Execution Time in second	
	Ontology based Approach	RDB Approach
I-1	0.000365	0.000372
I-2	0.000325	0.000335
I-3	0.000365	0.000381
I-4	0.000441	0.000492
I-5	0.000526	0.000365
I-6	0.000345	0.000358
I-7	0.000489	0.000495
I-8	0.000467	0.000487
I-9	0.000378	0.000405
I-10	0.000384	0.000421

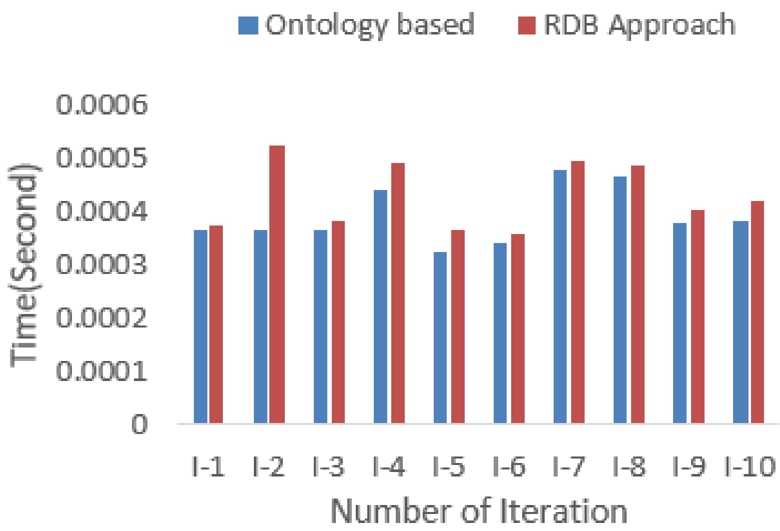

FIGURE 10 Comparison of execution time between Ontology based and RDB Approach

Scenario-3: In the existing approaches, a cloud user searches the required cloud services through general search engines like Google, Yahoo, MSN etc. which gives hundreds of records to select the required cloud service as shown in procedural Fig. 12. The proposed MAS reference architecture is a FIPA compliance system for efficient discovery and selection of appropriate cloud service which discover the most appropriate choice using the cloud ontology as shown in the Fig. 13. The proposed approach is a FIPA compliance system having unified and standard API gives compatible, portable and interoperable cloud services to the users. As a result, the satisfaction level of cloud users becomes high.

Scenario-4: Preform a real-time experiment of both approaches. Consider that the requirement/query of the cloud customer are as follows:

Memory range: 1-12 GB Bandwidth range: 100 to 2000 ECU range: 4 to 12 CPU range: 1 to 12 Storage range: 4 to 2000

A practical example of the existing approach is shown in the screen shot of the Google.com. as highlighted in the Fig. 14, there are 122,000 results on the given query for discovery of cloud service. It might scatter the mind of the cloud service requester that what is the exact solution of the query as every cloud service provider (CSP) have their own API for same requirement.

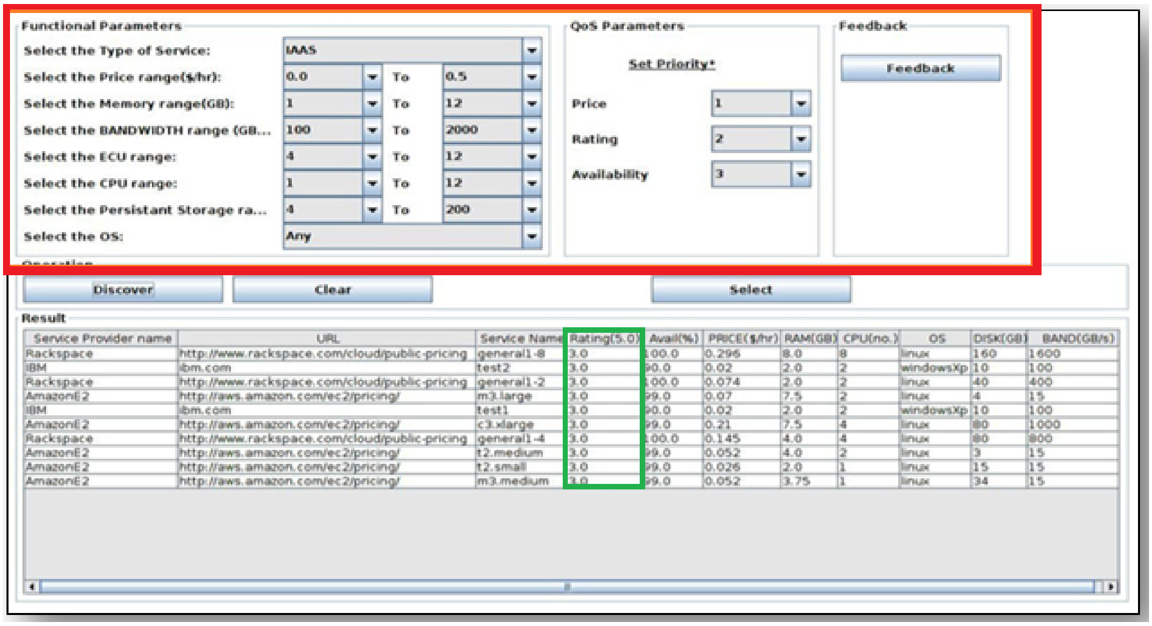

FIGURE 11 Interface for QoS parameter and Rating Cloud Service

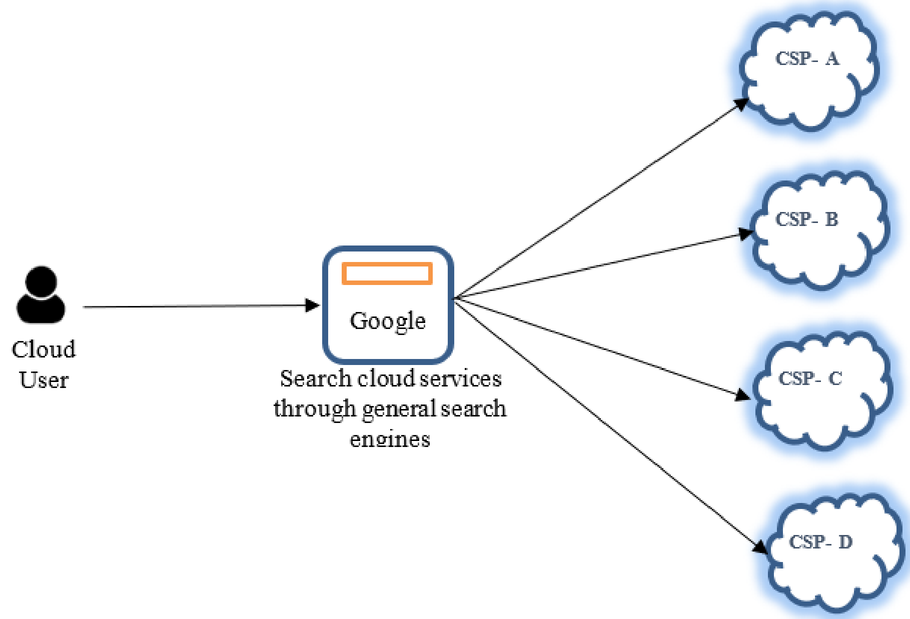

FIGURE 12 Existing Approach to Search a Cloud Service

However, by using the proposed multi-agent system, the same query of the requirements mentioned in the above can be easily get the appropriate cloud services as seen in the above figures (i.e., Fig. 6 and Fig. 7).

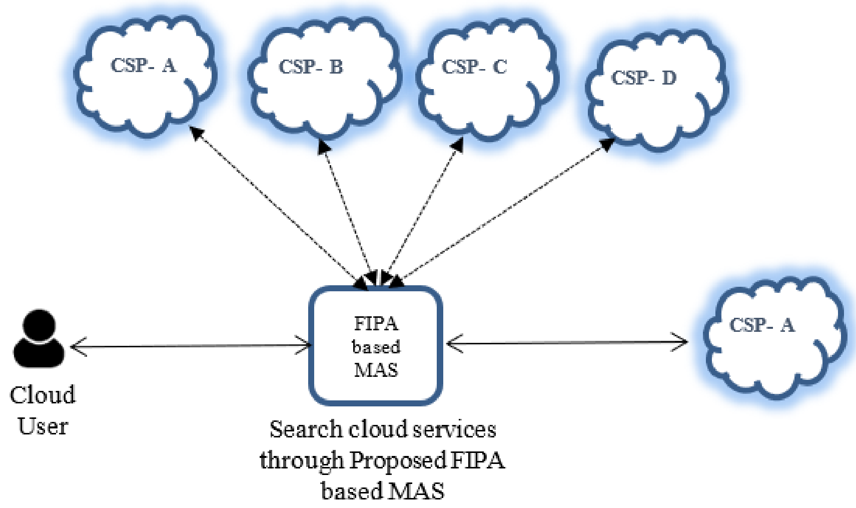

FIGURE 13 Proposed Approach to Discover and Selection of Cloud Services

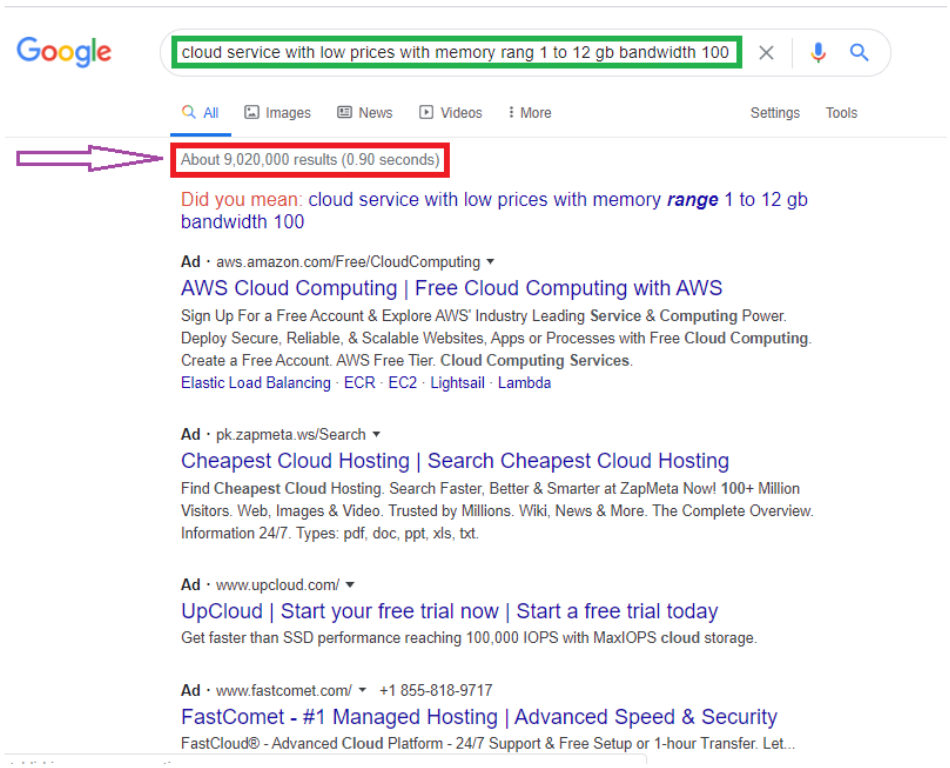

FIGURE 14 Searching of Cloud service in general search engine

5 | CONCLUSION AND FUTURE WORK

A lot of work has been done related to Cloud Computing in various areas related to scheduling, software development, scalability, security, etc. still there are number of challenges like security, privacy, interoperability, portability, reliability and availability problem related to the Cloud computing. In addition, another important challenge is vendor/customer locked-in issue which is the root cause of the interoperability and portability problems. The vendor locked-in issue occurs when a cloud user struck into any unwanted cloud provider or in another word, selection of an inappropriate cloud service by a cloud customer. After migration to the cloud by shifting valuable data to the cloud, the cloud customer can't easily transfer and move to another cloud due to the interoperability and portability problems. This research proposed an efficient multi-agent reference architecture based on IEEE-FIPA compliance architecture and cloud ontology for the discovery and selection of appropriate cloud services. The proposed reference architecture resolved the vendor locked-in issue up to some extend and mitigate the interoperability portability problems in cloud environment. The MAS consists of number of intelligent agents such as Customer Agent, Discovery Agent, Directory Agent, Ranking Agent, Registry Agent, Cloud Service Provider Agents, and Reasoning Agent. Another important component of the reference architecture is the cloud ontology for the knowledge acquisition in the cloud domain. A detail methodology has been presented for the development and implementation of the reference architecture. The JADE supports and compliance the FIPA specification and provide standard inter-communication in multi-agent system by using FIPA-ACL (Agent Communication language). It is used for the development of the intelligent agents and application of interoperability in the multi-agent system. The commonly used performance measures precision, recall and F-measure are used to evaluate the performance of the proposed approach and compare with the existing relational database approaches. After in-depth experimentation, it is found that the reference architecture is much better in terms of search efficiency, response time and execution time than the existing approaches. Thus, the obtained results advocate the effectiveness of the proposed approach for the efficient discovery and selection of an appropriate cloud service. Efforts have been made for improvement in the discovery and selection of cloud service efficiently, however, there is still space to improve in the field of cloud service discovery. In future, plan to extend the approach in other challenging field of Cloud Computing such to discover services between interconnected could, clouds management, security and confidentially by the same approach of FIPA based multi-agent systems.

5.1 | Acknowledgements

References

1. Androcec, D., Vrcek, N., Seva, J. (2012, April). Cloud Computing ontologies: A systematic review. In Proceedings of the third international conference on models and ontology-based design of protocols, architectures and services (pp. 9-14).

2. Ibrahim, M., Iqbal, M. A., Aleem, M., Islam, M. A. (2018). Sim-cumulus: An academic Cloud for the provisioning of network-simulation-as-a-service (nsaas). IEEE Access, 6, 27313-27323.

3. Iqbal, M. A., Aleem, M., Ibrahim, M., Anwar, S., Islam, M. A. (2019). Amazon cloud computing platform EC2 and VANET simulations. International Journal of Ad Hoc and Ubiquitous Computing, 30(3), 127-136.

4. Friedrich S. and Tim K. (2009) : http://www.statista.com/statistics/321215/global-consumer-cloud-computing-users

5. Chang Y. S, Jung T.Y and Chang H.(2012). Integrating intelligent and ontology for service discovery on cloud environment, in preceding of the IEEE international conference on systems, man, and cybernetics (SMC), Seoul Korea, pp3215-3220.

6. Stefan Poslad, Phil Buckle and Rob Hadingham. FIPA-OS agent platform: Open Source for Open Standards,The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents (PAAM 2000) Manchester, UK, April 10-12, 2000

7. A. Mehmood, H. Song and J. Lloret (2014). Multi Agent-based framework for secure and reliable Communication among open clouds. Macrothink Institute, Network Protocoled and Algorithems, Vol 6, No. 4, pp 60-76

8. Kang K M Sim (2010). Cloudle: An Agent-based cloud search engine that consults a cloud ontology. Proc. Intel Conference on Cloud Computing and Virtualization, pp 312-318

9. Haan T. and Sim K. M (2010). An ontology-enhanced cloud services discovery system, in preceding of the international multi-conference of engineers and computer scientists(IMCES), Hong Kong, pp17-19

10. R. Khan and A. Mahmood (2012).Realization of Interoperability and portability among open cloud by using agent's mobility and intelligence., International Journal of Multidisciplinary Science and Engineering (IJMSE), vol. No. 3, No. 7

11. Nagan LD and Kangsabai R (2012).OWl-S based semantic cloud services broker in preceding of the 19th IEEE International Conference on web services (ICWS), HI, USA, IEEE, pp560-57.

12. Tahamtan A, Behesti SA, Anjomshoaa A and Tjoa AM(2012).A cloud repository and discovery framework based on the business and cloud service ontology., in preceding on the IEEE, 8th World congress on service, Honolulu, HI, USA, IEEE pp203-210.

13. Gueferl R et al (2015).Towards a system for cloud services discovery and composition based on ontology. in Nez M, Nguyen N Camacho D , Trawiski B(eds) Computational Collective intelligence, No. 9330 in computer science, Springer Cham pp 34-43

14. Uchibayashi T et al (2015). Toward a cloud ontology clustering mechanism to enhance IaaS service discovery and selection. in Gervasi O et al (edu) computational science and its applications –ICCSA 2015, vol. 9155. Springer, Cham, pp 545-556.

15. Cretella G, Demartino B (2014). A Semantic engine for potring application to the cloud and among clouds., Software practice Exp 45(12),1619-1636.

16. Kwan Mong Sim (2012). Agent–Based Cloud Computing, IEEE Transactions on service computing, Vol. 5, No. 4, October –November 2012

17. Justice Opara-martins, Reza Sahandi and Feng Tian (2016). Critical analysis of vendor lock-in and its impact on Cloud Computing migration: a business perspective, journal of Cloud Computing: Advance, Systems and Application.

18. J Lloret, M Garcia, J Tomas, JJPC Rodrigues, Architecture and protocol for intercloud communication, Information Sciences 258, 434-451. 2014

19. Aliyu, A., Abdullah, A. H., Kaiwartya, O., Cao, Y., Usman, M. J., Kumar, S., ... Raw, R. S. (2018). Cloud Computing in VANETs: architecture, taxonomy, and challenges. IETE Technical Review, 35(5), 523-547.

20. Laghari, A. A., He, H., Khan, A., Kumar, N., Kharel, R. (2018). Quality of experience framework for Cloud Computing (QoC). IEEE Access, 6, 64876-64890.

21. Lin, M., Chen, Z., Liao, H., Xu, Z. (2019). ELECTRE II method to deal with probabilistic linguistic term sets and its application to edge computing. Nonlinear Dynamics, 96(3), 2125-2143.

22. Laghari, A. A., Laghari, R. A., Wagan, A. A., Umrani, A. I. (2019). Effect of Packet Loss and Reorder on Quality of Audio Streaming, EAI Endorsed Transactions on Scalable Information Systems, 24 September 2019, doi: 10.4108/eai.13-7-2018.160390

23. Nayak, S. C., Parida, S., Tripathy, C., Pati, B., Panigrahi, C. R. (2019). Multicriteria decision-making techniques for avoiding similar task scheduling conflict in Cloud Computing. International Journal of Communication Systems, e4126, 16 August 2019. https://doi.org/10.1002/dac.4126

24. [22]. Shobha, R. D., Pounambal, M., Saritha, V. (2019). An efficient algorithm for dynamic task offloading using cloudlets in mobile Cloud Computing. International Journal of Communication Systems, 3890, 28 January 2019, https://doi.org/10.1002/dac.3890

25. Singh, J., Gimekar, A., Venkatesan, S. An efficient lightweight authentication scheme for human-centered industrial Internet of Things. International Journal of Communication Systems, 12 November 2019, https://doi.org/10.1002/dac.4189

26. Irshad, A., Chaudhry, S. A., Shafiq, M., Usman, M., Asif, M., Ghani, A. (2019). A provable and secure mobile user authentication scheme for mobile Cloud Computing services. International Journal of Communication Systems, e3980, 19 June 2019. https://doi.org/10.1002/dac.3980

5.2 | Appendix Section

```
\appendix

\section{Section title of first appendix\label{app1}}
.
.
.
```

How to cite this article: Williams K., B. Hoskins, R. Lee, G. Masato, and T. Woollings (2016), A regime analysis of Atlantic winter jet variability applied to evaluate HadGEM3-GC2, *Q.J.R. Meteorol. Soc.*, *2017;00:1–6*.