

Semantic based Inter-Operable Cloud Resource Allocation

R.S.Amshavalli^a, and Usha Kiruthika^b

^aDepartment of Computer Science and Engineering, SRMIST – Ramapuram.

^bDepartment of Computer Science and Engineering, SRM Institute of Science and Technology - Kattankulathur Chennai, India

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Cloud computing is the process of lending the required resources that includes computing power, network access, database storage, applications, and other IT resources in on demand basis through a cloud services platform with pay-as-you-go pricing. Here, users access services according to their requirements, without knowing where the services are hosted or how they're delivered. So Resource Management is tedious process between different clouds. Since different clouds will have different types of cloud virtualization techniques with corresponding middleware's with unique mechanism and protocols, we face lot of interoperability issues. So there is a need to address these interoperability issues between different clouds to achieve the highest availability of cloud resources. This motivates us to conceive a proposal Semantic based Interoperable Cloud Resource Allocation (SICRA) that provides a common mechanism to access the resources of different clouds Proposed work also incorporates the semantic services for identifying best potential resource. Thus an ontology based indexing and retrieving algorithms are designed and provided with ranking which in turn increases the success rate.

Keywords: Intercloud, Interoperability, Semantic search, Resource allocation, OWL, Ontology, Indexing, Retrieving.

1. Introduction

Inter-cloud infrastructures are essential because it is not possible for single cloud provider to attain the fullest satisfaction of their customers. Hence it is mandatory to sort out cloud frameworks that supplement each other, for example, to obtain resources from other competing cloud frameworks. Be that as it may, it is hard to give the potential resources from different cloud suppliers inspite of the fact that the executives arrangements and management policies about different resources are diverse in nature. Having these distinctions, it is difficult to achieve interoperability among them, which in turn makes the resource management a complicated one. So we are in need of common mechanism to address these kind of interoperability issues and thereby increases the availability of cloud resources. In this paper, Semantic based Interoperable Cloud Resource Allocation (SICRA) framework have been designed to manage different clouds.

This paper also works on replacing traditional keyword search with semantic search on cloud resources. Keyword search focus on indentifying the actual term wherever they appear. This framework is viable if the specialist knows precisely what they are searching for. The actual problem arises when we come across the situation, where single word will have multiple meanings, so keyword search most frequently return irrelevant results(false positives), neglecting to disambiguate unstructured content. For example, a keyword search on the term "ACID" might include not only references to chemical acid but also the "ACID" (Atomicity Commit Isolation Durability) properties of DBMS. So, we have incorporated semantic search in the place of keyword search. Unlike keyword search, semantic search takes user's actual intent into consideration so as to get at the contextual meaning of terms

In this directions, various researchers have recognized that ontology based retrieval is one of the best in terms of precision and recall for semantic search engine. However, they did not specify explicitly any scheme/ algorithm for ontology based indexing and retrieval for cloud resources. This motivates us to conceive a proposal that includes (a) suggesting indexing algorithm to effectively maintain ontology, (b) suggesting retrieving algorithm to process user query and produce results with better precision and recall. Thus a framework is proposed to focus on the management of Intercloud components with semantic technology to provide the ranking and thereby improving the accuracy and availability of cloud resources. The following sections are framed as: section two highlights the related works that helps in understanding the underlying problem, and section three explains the methodology to overcome the issues with our proposed architecture. Implementation information has been discussed in section four, section five provides experimental environment. And section six concludes the proposed work and it also points out the further possibilities of extending the work.

2. Related Work

There has been a great deal of research work completed in understanding the concept of interoperability issue in multiple clouds. Some of the related works which helps in understanding the concepts of Intercloud and semantic are listed in this part. The Intercloud Resource sharing has been tried out in a cloud framework with traditional way of search instead of using semantic concepts which shows difference in performances. Paper[9] proposed the blueprint for developing InterCloud, in Hadoop environment. But it has not been implemented using clouds with different middleware. It helps in addressing the differences between the clouds which follows their own mechanism and protocols. Suresh Kumar, Naresh Kumar, Manjeet Singh, Asok De [1] proposed OWL

Based Indexing and Retrieving Algorithm. This work mainly focused on retrieving the suitable resource from the huge knowledge base by applying the algorithm. This work helps in identifying the demerits of keyword search over the semantic search. It also gives the importance of ontology. Yong Boem Ma et al [3] had given the detailed description of ontology. In their work, they insisted up on the importance of using ontology based resource management in cloud environment. This motivates us to incorporate semantic techniques in cloud resource allocation. Jorge Ejarque et al [8] had worked on incorporating semantic technology in resource allocation in service provider. Through their work, they had overcome the issues of traditional keyword search system. They also uses OWL for creating ontologies.[10]

3. Proposed Architecture

The proposed SICRA framework is designed to obtain high interoperability among different cloud setup. It is focused on two major areas such as Semantic Interoperable Cloud Resource Allocation (SICRA) (Figure 1) and Semantic technology (Figure 2).

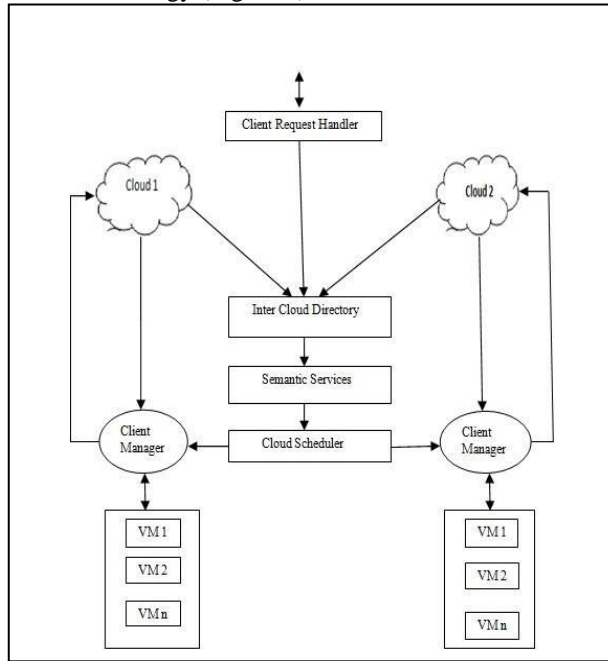


Fig 1: SICRA

A. Request Handler

The Request handler acts as a intermediate component and gets the requests and provides appropriate output to the clients. Once it receives the request from the client, it sends the request to the semantic components. Finally it receives corresponding output from the client manager of the appropriate cloud.

B. Intercloud directory

Inter cloud directory is the collection of resources available in different clouds that are participating in the deal. Each clouds register their available resources with their detailed description, in inter cloud directory. Here it maintains a database for maintain the list of resources with their information. After receiving request, it sends the request to semantic service. It also provides the list of resources to semantic services.

C. Cloud Scheduler

Semantic services contacts cloud scheduler with the output of the semantic search. Only the cloud scheduler will be in touch with all the participating clouds. Once it receives the result of the semantic search, it initiates the appropriate cloud and instruct it to execute the job with that particular virtual machines. One of the main advantage of the proposed work is that, the cloud scheduler gives top five ranked resources. So even if the identified resource is busy, it can suggest with the other option

D. Client Manager

Each cloud provider’s client manager will be in contact with the central device. Client manager will receive the list of top n resources and start executing the task with the appropriate virtual machine. If the said resource is busy with other task, it will check for the other options in the list and go for it. If the next ranked resource is not associated with that cloud provider, it will intimate the same to the cloud scheduler. Then the cloud scheduler will initiate the appropriate cloud provider. After the VMs executed the task, only the client manager will hand it over to the request handler. Thus client manger acts as the intermediate component between the central device and the cloud provider.

E. SEMANTIC SERVICE:

Resource discovery in cloud computing environment is the concept of identifying the best potential cloud resource among the pool of resources offered by multiple cloud service providers (Amazon, VMware, Rackspace etc). In turn these cloud resources are used for virtual resource creation and application execution.

Task execution is done by forwarding the client's application requests to the cloud provider. They in turn creates the required number of virtual machines with the help of available physical resources and maps the virtual resources to client's application requests. It is done based on the motivation that, creating virtual machine and task execution is done in minimal time. The following are the semantic components.

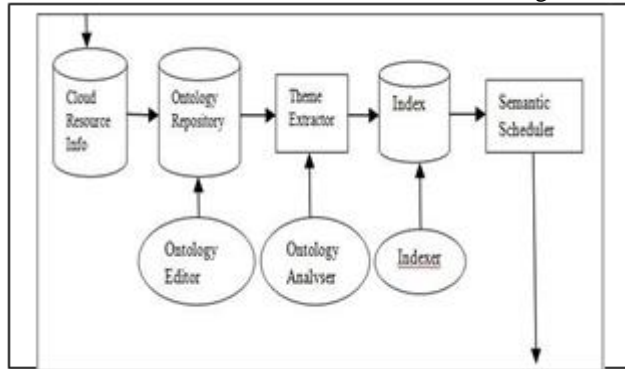


Fig 2: Semantic Service

F. Ontology Repository

Various researchers have recognized that ontology based retrieval is one of the best in terms of precision and recall for semantic search. Ontology helps in extracting the properties of complete knowledge base, including the similarities and differences between different entities. Ontology can be created using ontology editor. Ontology are created for each available resources. It provides the each level of concepts such as the requirements of the application including LOC, Hardware information, Software information, Operating System Details, Network Details, History of the application and etc. in addition with properties like hasLOC, hasMemorycapacity, hasOperatingSystemName, hasRAMSpeed, hasApplicationName and etc. to explain the characteristics of cloud resources.

G. Theme Extractor

Ontology analyzers like jena [18] or pellet [17] can be used to extract themes for each resources. Theme includes, specifications, history of execution, success rate (Number of task executed successfully out of allotted task), down rate (Number of tasks failed only because of the down of resource). Based on the theme, ontology analyzers will perform semantic search and provides the result to the indexer. Indexer will index the results using indexing algorithm.

H. Semantic Scheduler

Semantic scheduler will follow the retrieval algorithm for retrieving the best result from the index. Then the semantic scheduler will forwards the result to the cloud scheduler.

Request handler, receives all the client's application requests and forwards it to Inter Cloud Directory (ICD). All the participating cloud providers register the details of their available resources with ICD. Then ICD forwards the above said details to semantic service. In semantic service. For each registered resource, separate ontologies are created using protégé ontology editor. After analyzing the ontology, reasoners like jena reasoner or pellet reasoner extract the themes for each resource. Based on the details in the theme, resources are indexed using proposed indexing algorithm. Now semantic scheduler decides the appropriate resource and it sends the information to the cloud scheduler. It also sends top five best suited resource list. Then the cloud scheduler initiates the appropriate cloud. Particular cloud provider executes the task using the selected resources and provides the result to user through request handler. If found resource is busy, the cloud scheduler informs the same to semantic scheduler. Then semantic scheduler follows the same step for second ranked resource.

4. Implementation and Working Principle

We have designed the index of retrieval system by keeping in view the structure of thematic document, in which, we have n number of resources and with each resource, certain properties are associated with their values. Therefore, to keep the size of index as small as possible and to avoid redundancy, we have divided our index at three levels named as primary, secondary and tertiary index.

Level 1 or Primary Index:

It contains the information about C_Id and corresponding thematic content of document as shown in Table 2 above. This information is required because the document content is to be responded to end user if the document qualifies for the user query.

C_Id	Thematic Content

Table 1: Primary Index

Level 2 or Secondary Index

It contains the information about C_Id, resource identification number (R_Id) and property name and property value (jointly called PV_ID). The structure of this part along with some typical entries is as shown in Table 3 given below. This information is required to list property name, property value of each R_Id corresponding to C_Id. TABLE 1.

C_Id	R_Id	PV_Id

Table 2: secondary index

Level 3 or Tertiary Index:

It contains the information about property name (P_N), its value and PV_Id, as shown in structure in Table 2 below. This information is required to know how many types of unique property name and property value by our scheme at any moment of time.

R_Id	PV_Id	P_name	P_Value

Table 3: Tertiary index

The responsibility of the indexer is to provide ranking for the intermediate thematic repository. So here, we proposed indexing algorithm to index thematic repository in above designed index. In this algorithm, it has shown that how the above index designed is generated.

```

INDEXING ALGORITHM:

{
1 Repeat step 2 for each cloud i.e. C_Id of thematic
Repository.
2 Repeat step 3 for each R_Id in current C_Id
3 Repeat step 4 for each P_Name and P_Value of current
R_Id.
4 if(P_Name and P_Value exist in Level 3 index
   i.e. Table 2) Then
   Get PV_Id from it else
   (i) Generate unique PV_Id
   (ii) Insert PV_Id, P_Name, P_Value,
   into Table 2.
   5 if (R_Id exist in Table 2) Then
   Get R_Id from it
   else
   (i) Generate unique R_Id
   (ii) insert R_Id into Table 2
   6 if (C_Id, R_Id and PV_Id not exist in Table 1)
   (i) Insert C_Id, R_Id, PV_Id into Table 1.
}
    
```

RETRIEVAL SCHEME:

Retriever provides only the relevant resources to user. It analyses the index and after applying the retrieving algorithm, it retrieves the potential resource. We have couple of parameters to access the accuracy of retrieved resource: Recall and precision. Recall is the number of retrieved relevant resources in out of available relevant resources in index Precision means number of retrieved relevant resource in out of total retrieved resources. And. We proposed three retrieval algorithms as shown below

If user just give resource id (not familiar about its property name and value).If user give property name and its value.

If user give resource id, property name and value also(i.e. completely familiar with resource).

Retrieval Algorithm_1 (R_Id)

```

1 If R_Id exist in table 1 Then
(i) Get all PV_Id corresponding to current R_Id from table 1
(ii) Corresponding to each PV_Id, list P_Name, P_Value,URI to
user from table 2.
(iii) stop
Else
(i) Display a message no information about
resource R_Id available in our index
(ii) Stop

```

Retrieval Algorithm_2 (PN, PV)

Retrieval Algorithm_2 (PN, PV)

```

{
1 If (PN and PV exist in table 2) Then
(i) Display (P_Value, P_Name) to user
(ii) stop
Else
(i) No record found in our index
(ii) Stop
}

```

Retrieval Algorithm_3 (R_Id, PN, PV)

```

{
1 If R_id exist in table 1 Then
(i) Get C_Id of the corresponding R_Id
(ii) Get all PV_Id from table 2, where
P_Name=PN and P_Value=PV exist
(iii) Get all PV_ID from table 1 corresponding to
current R_Id
(iv) Now merge step (ii) and (iii) where
PV_ID match occur and Get list of VMs
(v) Display this list of VMs to user
(vi) stop
Else
(i) Display a message no information about
resource available in our index.

```

This semantic part constitutes the actual resource identifying mechanism and it accepts the user's requests. The request includes user's requirements for executing their task such as Operating System, Number of CPUs required, and Software libraries. The semantic algorithm relies on ontology reasoner, we have used jena reasoner. Information from the knowledge base will be retrieved by using Algernon inference engine. Hence, based on the user's resource request, the broker generates an Algernon query For Ex, if the user requests for a machine with RAM of 500MB and prefers for the OS of centos, then the query "RAM:500 OS:centos" will retrieve all resources with RAM of 500MB and with centos. The inference engine module parses the user query with the help of regular expression and stores the left sided tag and right sided tag in a vector form and converts the same into relevant Algernon query. Here in semantic part, we have created OWL ontologies for each and every registered resources using protégé ontology editor.

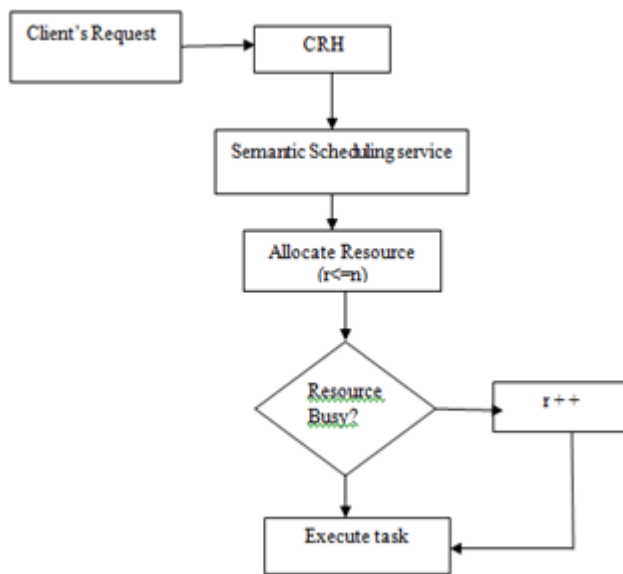


Fig 3: Inner Flow

5. Experiments and Results

The proposed work is evaluated by creating 20 to 40 resources in the ontology knowledge base and it has been compared. The aforesaid setup is built with the aid of two different private cloud (Eucalyptus and Open nebula), overseen by two diverse cloud virtualization techniques and semantic component which goes about as an interface between the diverse private clouds. In Eucalyptus based private cloud, we have created around 10 node controllers and in Open nebula based private cloud, 5 compute nodes were created. And we have created nearly 30 resources in knowledge base, from which semantic search is done for identifying the suitable resource. Obviously resource availability is increased as we access multiple cloud instead of single cloud. Thus availability rate is doubled for SICRA compared to single cloud architecture. Since we access multiple clouds, the best suited resource for our task can be identified to great extent.

If the submitted task finds the best potential resource, and starts execution, it means that the task has successfully executed. If the task has not found the suitable resource, ICD will go for next ranked resources, till the rank reaches five. Even after tracking all the five resources, and given task has not suited with any other resources, it is said to be failed. Hence execution rate can be related with the success rate. Execution rate is measured, when executing the task in single cloud, Interoperable Cloud Resource Allocation (ICRA) and in Semantic based Interoperable Cloud Resource Allocation (SICRA).

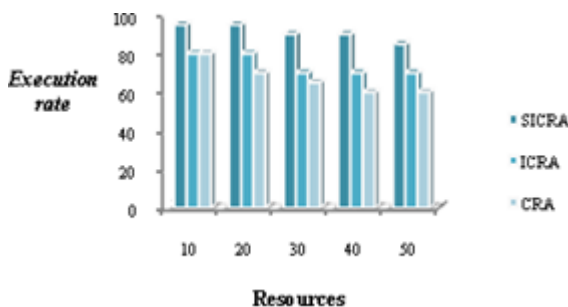


Fig 4: Execution Rate

The inference time for discovering the best suitable resources was calculated for traditional cloud infrastructure using keyword matchmaking technique and the cloud infrastructure that uses semantic technology for resource selection. The traditional keyword search takes lesser compared to semantic based resource discovery. This is because, when we go for semantic, entire ontology repository has to be traversed to find the best resource available. Even if the keyword search has lesser time than the semantic search in finding the best potential resources, the semantic has couple of pros than the former. They are

1. If provides the best suitable resources even if the user is unaware of resource details.
2. If particular resource is not found, semantic search is capable of providing at least the related result.

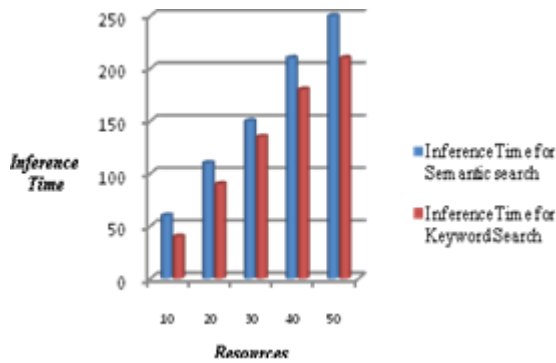


Fig 5: Inference Time

We also measure recall rate. Rate of number of retrieved relevant resources out of available relevant resources in index. Even if allotted resource is busy or fails, it can work with the next ranked resource in the index.

6. Conclusion and Future Work

This work focused on designing the architecture of Semantic based Interoperable Cloud Resource Allocation (SICRA) which utilizes the semantic innovation to settle the interoperability issues among different clouds. The combination of semantic technique and resource discovery service helps in identifying the potential cloud resources that are fit for virtual machine creation and task execution. The said framework improves the execution measures, such as, execution rate and success rate of tasks submitted to SICRA. As a future work SICRA will be extended to address load balancing and also in migrating the virtual machines from one private cloud datacenter to other. It also decreases the execution time of the application.

References

1. Suresh Kumar, Naresh Kumar, Manjeet Singh, Asok De, "OWL – Based Ontology Indexing and Retrieving Algorithm for Semantic Search Engine". Published in International Conference on Reliability, InfoCom Technology and Optimization (ICROTO 2010), November 1 -3, 2010 Lingayas University, Faridabad. Pp 392-399.
2. Yong Boem Ma, Sung Ho Jang and Jong Sik Lee, "Ontology Based Resource Management for Cloud Computing", Intelligent Information and Database Systems, Lecture Notes in Computer Science 2011, Springer Berlin Heidelberg, 978-3-642-20041-0, 343,6592
3. Nelson.V, Uma.V, "Semantic based Provisioning and Scheduling in Intercloud Environment", Proceedings of 12th International Conference on Recent Trends in Information Technology – 2012.
4. NikolaosLoutas, VassiliosPeristeras, ThanassisBouras, EleniKamateri DimitriosZeginis, "Towards a reference architecture for semantically interoperable clouds." IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), 2010.
5. Amshavalli, R.S., Kavitha, G., "Increasing the availability of cloud resources using broker with semantic technology" (2015) Proceedings of 2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014, art. no. 7019373, pp. 1578-1582. ISBN: 978-147993914-5 doi: 10.1109/ICACCCT.2014.7019373
6. David Bernstein, Deepak Vij, 'Using Semantic Web Ontology for Intercloud Directories and Exchanges', The 2010 International Conference on Internet Computing, Las Vegas, "V Jul 12- 15 2010.
7. JorgeEjarque, Marc de Palol, InigoGoiri, Ferran Julia, JordiGuitart, Jordi Torres and Rosa M. Badia, "Using semantics for resource allocation in computing service providers," Barcelona Supercomputing Center and Universitat Politecnica de Catalunya 2009.
8. David Bernstein Erik Ludvigson Krishna Sankar Steve "Blueprint for the inter-cloud – protocols and formats for internet and web applications and services." Diamond Monique Morrow Cisco Systems, Inc., ICIW '09.
9. Raja, S. Kanaga Suba, and T. Jebarajan. "Reliable and secured data transmission in wireless body area networks (WBAN)." European Journal of Scientific Research 82, no. 2 (2012): 173-184.
10. World Wide Web Consortium, "Resource Description Framework (RDF) Model and Syntax Specification", <http://www.w3.org/TR/rdf-syntax-grammar/2004>.
11. World Wide Web Consortium, "Resource Description Framework (RDF) Schema Specification 1.0", <http://www.w3.org/TR/rdfschema/2004>.
12. World Wide Web Consortium, "Web ontology Language (OWL) syntax and specification", <http://www.w3.org/TR/owl-features/>
13. Usha Kiruthika, Thamarai Selvi Somasundaram, S. Kanaga Suba Raja, (2020) 'Lifecycle Model of a Negotiation Agent: A Survey of Automated Negotiation Techniques', Group Decision and Negotiation, ISSN 0926-2644, Volume 29, Issue - 6, pp. 1239–1262.
14. VMware (2011). <http://www.vmware.com/Org/TR/rdf-schema>.

16. Typica API (20 1 1),<http://code.google.com/p/typicai>.
17. Protege-an ontologyeditor, available
at:<http://protege.stanford.edu/download/protege4/installanywhere/>
18. Pellet 2.3.0, at: <http://pellet 2.3.0.owlidl.com>
19. Jena 2.7.3- A Semantic Web Framework,
20. <http://downloads.sourceforge.net/jena/jena-2.7.3>