CHAPTER 3

A SEMANTIC NETWORK BASED INFRASTRUCTURE SERVICE DISCOVERY

3.1 INTRODUCTION

This chapter presents a new cloud broker framework, which supports the cloud user to discover the infrastructure type of services such as computing and storage from various providers. Here, users are allowed to specify their service requirements in terms of numerical representation. A cloud ontology has been constructed by the broker to know the service availability. Further, a semantic network based service representation is put forth by the proposed broker that enables the user to discover the appropriate services. The rest of the chapter is organized as follows: Section 3.2 defines the problem; Section 3.3 explores the functionalities of a proposed broker and a prototype model has been constructed in Section 3.4 to demonstrate the novelty of the proposal. The experimental results are presented in Section 3.5 and Section 3.6 concludes the chapter.

3.2 PROBLEM STATEMENT

Service discovery (Zeshan et al. 2017) in cloud computing meant for finding a suitable cloud service that essentially meets out the cloud user requirement in the aspect of functional, non-functional and the budgetary constraints. Without considering these factors, it is inappropriate to discover right services from the multiple cloud providers. Unfortunately, many of the
service discovery schemes are unable to perform the right discovery of services due to the issues such as, (i) incorrect / invalid service details, (ii) handling of uncertainties in service specification, (iii) fails to make interaction between the multiple cloud providers, (iv) effectiveness in handling feedback of trusted third parties, (v) user / service based discovery, (vi) dynamic adaptation of cloud user’s feedback, and (vii) issues with respect to the service upgradation for future usage. Though many of the works (Damiani & Fugini 1991; Bosc et al. 2001; Noor et al. 2013; Wang et al. 2006, 2015) are attempted to address these issues in the phase of service discovery, certain challenges still exist.

3.3 SEMANTIC NETWORK BASED CLOUD SERVICE DISCOVERY

The semantic network based broker contributes the cloud service discovery in the following manner:

(i) Constitutes an end-user portal to define the service requirements very precisely in terms of numerical values.

(ii) Processing of user requirements and constructs the service ontology.

(iii) Discovering the appropriate IaaS type cloud services with the constructed cloud ontology and represent them in the form of semantic networks. By performing the intersection search, the appropriate services are discovered.

(iv) A simulated platform is designed and developed to analyse the broker performance with the existing approaches.
The proposed semantic network based broker architecture for the discovery of cloud infrastructure is shown in Figure 3.1.

![Figure 3.1 Intelligent cloud broker for service discovery](image)

### 3.3.1 User Interface

The proposed broker architecture enables the cloud user to interact through user interface. The portal obtains service requirements in terms of numerical value with the aid of ‘user interface’ and then pre-process the input values with respect to the providers’ service specification details. The following example shows the input requirements furnished by the cloud users through the portal for the services such as computing and storage (Table 3.1).
Table 3.1 Requirements furnished through user interface

<table>
<thead>
<tr>
<th>Cloud user (CU)</th>
<th>Computing services (GHz)</th>
<th>RAM (GB)</th>
<th>HDD (TB)</th>
<th>Expected budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU1</td>
<td>3.2</td>
<td>4</td>
<td>1</td>
<td>$0.25</td>
</tr>
<tr>
<td>CU2</td>
<td>3.0</td>
<td>2</td>
<td>1</td>
<td>$0.15</td>
</tr>
<tr>
<td>CU3</td>
<td>3.4</td>
<td>8</td>
<td>2</td>
<td>$0.30</td>
</tr>
<tr>
<td>CU4</td>
<td>3.2</td>
<td>16</td>
<td>4</td>
<td>$0.33</td>
</tr>
<tr>
<td>CU5</td>
<td>3.1</td>
<td>2</td>
<td>1</td>
<td>$0.25</td>
</tr>
</tbody>
</table>

3.3.2 Service Crawler

After receiving the service specification, the service crawler initiates searching process with the service repository and extends the process to cloud service providers. The resulted information is termed as ‘service concepts’ and further forwarded as input for the cloud ontology construction stage. In addition to allowing cloud service providers to register their service details with broker’s service repository, service crawler is also used to build and maintain registry with needed IaaS service instances.

3.3.3 Cloud Ontology

A cloud ontology is a description of different cloud concepts with its relationships to facilitate the reasoning among all types of cloud services. It enables the users/machines to process the information more precisely and conveniently. A typical cloud ontology for the IaaS type of services is shown in Figure 3.2.
With respect to this scenario, the broker constructs a cloud ontology for effective representation of IaaS types of cloud services with reasonable budget. By receiving the posted requirements through the user interface, it constructs the cloud ontologies by referring the service descriptions obtained from various cloud service providers. A typical ontological representation constructed by the broker for the posted requirements is shown in Figure 3.3.
3.3.4 Service Discovery

The service discovery is constrained by functional, technical specifications of the services, and budgetary constraints with appropriate security policies. The proposed broker adopts a semantic network based service representation technique for the task of service discovery. A semantic network is a knowledge representation technique to epitomise semantic relations among concepts. Here, the service entities are represented as a set of nodes connected to each other by a set of labelled arcs for representing the relationship among the discovered services. Further, the broker applies an intersection search to discover the similarities among the identified services. The matching between the posted requirements and the provider’s service description are determined finally. The algorithm for service discovery adopted by broker is shown below.

Algorithm 1: Service discovery

**Input:** User_requirements \( U_{req} \)

**Output:** Discovered_services \( D_s \)

1. Obtain the service requirements \( U_{req} \) from the cloud user
2. If \( U_{req} \) are in numerical terms then
   2.1 Check for ontological representation
3. Else
   3.1 Construct the ontology and update with Broker’s Service-Repository
4. End if
5. Represent the service details with semantic network
   5.1 Identify the similarity measures to find the best service
   5.2 Check for the right computing and storage services
   5.3 Assign the rank values before the aggregation process
6. Aggregate the discovered services \( (D_s) \) from the semantic network
7. Invoke Recommendation \( (D_s) \) for broker decision for top ranked list
A typical semantic network constructed by the proposed broker is shown in Figure 3.4.

![Semantic Network Diagram]

**Figure 3.4 Semantic representation of discovered services**

Let $A(x)$ be the service requirements of a cloud user and $B(x)$ be the available services of some cloud providers, then the similarities between $A$ and $B$ are determined by,

$$S_{A,B} = \frac{|A(x) \cap B(x)|}{A(x)} \quad (3.1)$$

In another way, the service similarities can also be determined by,

$$S_{B,A} = \frac{|B(x) \cap A(x)|}{B(x)} \quad (3.2)$$

By considering the service requirements for the computing services of cloud user ‘CU1’, the similarities among the services are represented as:

$S$ (Computing) $\Rightarrow$ $S(3.0 \text{ GHz})=2$, $S(3.1 \text{ GHz})=2$, 
$S(3.2 \text{ GHz})=2$, $S(3.3 \text{ GHz})=2$, $S(3.4 \text{ GHz})=3$, $S(3.5 \text{ GHz})=3$ 
$\Rightarrow S(3.2 \text{ GHz}, 3.3 \text{ GHz}) = 2/2 \quad (3.3)$
Likewise, for the RAM service, the similarities are represented as

\[
S \text{ (RAM)} \rightarrow S(4 \text{ GB}) = 4, S(8 \text{ GB}) = 4, S(16 \text{ GB}) = 4, S(32 \text{ GB}) = 4
\]

\[
\rightarrow S(4 \text{ GB}, 8 \text{ GB}, 16 \text{ GB}, 32 \text{ GB}) = 4/4 \tag{3.4}
\]

Then the broker lists out the possible cloud services among the providers which are very closer to the posted requirements for user CU1 is shown in Table 3.2.

**Table 3.2 Discovered services for the user CU1**

<table>
<thead>
<tr>
<th>Cloud Service Provider (CSP)</th>
<th>Computing services (GHz)</th>
<th>RAM (GB)</th>
<th>HDD (TB)</th>
<th>Exact budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSP1</td>
<td>3.2</td>
<td>4</td>
<td>2</td>
<td>$0.25</td>
</tr>
<tr>
<td>CSP2</td>
<td>3.2</td>
<td>2</td>
<td>1</td>
<td>$0.20</td>
</tr>
<tr>
<td>CSP3</td>
<td>3.3</td>
<td>4</td>
<td>1</td>
<td>$0.20</td>
</tr>
</tbody>
</table>

The best selection is up to the decision of intelligent broker by considering the general merits and cost factors of the service provider. Accordingly, providers ‘CSP1’ and ‘CSP3’ have been identified with the aid of constructed ontology, the ‘CSP1’ has been recommended to the user as a first choice by considering the budgetary requirements of the user. Though ‘CSP1’ service cost stands above the stated budgetary requirements of the user, their desirable service feature (2 TB of HDD) has been weighted and recommended to the user (Table 3.3). The *service recommendation* algorithm shows the broker decision about the top most service selection from the discovered services.
Algorithm 2: Service Recommendation

**Input:** Discovered_services $D_s$, User_requirements $U_{req}$

**Output:** Recommended_services $Rec_{service}$

1. List out the Discovered_services $D_s$
2. If $D_s$ fulfils the cloud user requirements $U_{req}$ && Add-on service packs are enabled then
3. For each Discovered_service $D_s$
   3.1 Evaluate the budget information of service instances
   3.2 If budget is adequate and Add-on service is good then
       3.2.1 Assign rank = 1
   3.3 Else if budget is adequate and the Add-on service is satisfactory then
       3.3.1 Assign rank = 2
   3.4 Else if budget is adequate and the Add-on service is not satisfied then
       3.4.1 Assign rank = 3
   3.5 End if
4. End for
5. End if
6. List out the service instances with the rank = 1
7. Create Recommended_services list $Rec_{service}$
8. Publish the $Rec_{service}$ description details in to the IaaS Service repository.
### Table 3.3 Recommended IaaS services with features

<table>
<thead>
<tr>
<th>Instance Details</th>
<th>Instance features with add-on services</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cloud Service Provider</strong></td>
<td><strong>CSP3</strong></td>
</tr>
<tr>
<td>CPU</td>
<td>3.3</td>
</tr>
<tr>
<td>RAM</td>
<td>4</td>
</tr>
<tr>
<td>HDD</td>
<td>1</td>
</tr>
<tr>
<td>Customer support</td>
<td>√</td>
</tr>
<tr>
<td>Trustworthiness</td>
<td>√</td>
</tr>
<tr>
<td>Availability</td>
<td>√</td>
</tr>
<tr>
<td>Service updates</td>
<td>√</td>
</tr>
<tr>
<td>Feedback responses</td>
<td>√</td>
</tr>
<tr>
<td>Budget</td>
<td>$0.20</td>
</tr>
</tbody>
</table>

In this perspective, the proposed intelligent broker performs service recommendations besides service matching. In the given example, the proposed intelligent broker discovers and recommends services from a single cloud provider. In addition, the proposed broker periodically investigates the services and creates a service log to ensure the service effectiveness and their business levels. Such a self-healing attitude plays a vital role in performing effective customer relationship management.

### 3.3.5 IaaS Service Repository

All kinds of IaaS cloud services with its service descriptions are published with the service repository. Whenever, a service request comes from the cloud user, the broker checks for the repository after constructing the cloud ontology. From the cloud service provider’s perspective, the repository plays a vital role by incorporating the different type of services. In another way, the cloud broker is responsible to identify the appropriate services on behalf of the cloud user.
3.4 PROTOTYPE DESIGN

A .NET-based prototype model is developed to demonstrate the cloud broker functionality with respect to the discovery of cloud infrastructure services. The developed model simulates the functioning with the sample data set, which contains 100 service instances from 10 cloud service providers. Each service instance is represented as a set of facilities such as computing, RAM and Hard Disk Drive (HDD). The tests were conducted using a HP Pavilion Laptop, with 8GB of RAM, 1 TB of Hard drive and a Intel (R) Core (TM) i7-6500 CPU (2.59 GHz) processor. Figure 3.5 shows the broker interface for supporting the cloud provider and user for service publishing and discovery process. Whenever the new services are emerged, service repository of the cloud broker got updated immediately (Figure 3.6). Hence, the cloud user can avail the updated service instances in a timely manner. Similarly, as per the posted service requirements of cloud user ‘CU1’, the discovered services with their ranking is presented in Figure 3.7. The broker recommendations about top ranked services with respect to budgetary constraints is also shown. Instead of just recommending the services to the user, the proposed broker continuously assists them in all aspects with respect to the mutual binding of cloud service provider.

![Figure 3.5 User interface of the proposed cloud broker](image-url)
Apart from the service recommendation, the proposed broker offers various value-added features such as evaluation of trustworthiness, inferring the availability, service updations and performing responses to user queries.

Figure 3.6 Service publication by the cloud service provider

Figure 3.7 Service recommendation by the cloud broker
3.5 EXPERIMENTAL RESULTS

In Chen et al. (2017), the functionally equivalent services are resulted as dissimilar because of the proposed matching process. Also, the opposite relationship among the service concepts are calculated based on subjective assessment instead of reasoning process, which may be inaccurate. Similarly, the construction of abstract network by Li et al. (2017) increases the complexity of similarity identification and resulted in delaying the process of service selection and composition.

In this proposal, the semantic network is used to represent the similar services with the aid of constructed ontology. Hence, the proposed system improves the overall performance by reducing the time needed to construct the ontology and semantic network based service representation. From the experimental work, it has been identified that the increased service instances with multiple copies without reasoning process resulted the performance degradation in the existing methods. In addition, the construction of ontology with the service concepts and the discovery of services through the developed broker certainly improves the system performance. The differences in the service discovery process is shown in Figure 3.8.

![Figure 3.8 Performance based on service matching](image)
In the final stage, the semantic network representation of recommended services certainly reduces the memory requirements with respect to the service details management. The usage of intersection search in the process of service representation improves the overall performance and is shown in Figure 3.9.

![Figure 3.9 Performance comparison with existing work](image)

### 3.6 CONCLUSION

In this chapter, an intelligent cloud broker is proposed for discovering the IaaS type of cloud services from multiple cloud service providers. The proposed system obtains the service requirements in numerical form and constructs cloud ontology for reasoning purpose. First, the broker discovers the appropriate services as per the posted service requirements and represent them using semantic network. Second, the intersection search is carried out to recommend the ranked services for the cloud user. The algorithms are proposed to support the service discovery and recommendation process. The next chapter presents the fuzzy logic based cloud broker for the selection and provisioning of cloud infrastructure services.