7.1 INTRODUCTION

Cloud computing delivers on-demand services over the network. Cloud computing is a promising and a challenging domain of the future delivering on-demand services as a utility over the internet. Cloud computing provides various services to the users such as computation, software, data access, and storage. Based on the type of services provided, cloud computing can be classified into three categories. They are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Based on the deployment models, the cloud has been classified into private, community, public and hybrid clouds. A few examples of cloud are Amazon EC2, Google App Engine, Salesforce, Window Azure, IBM cloud. The features of cloud computing is that the user does not require knowledge of the physical location and configuration of the system that delivers the services. The users only pay for the amount of time they use the services that reduces the cost to a great extent. The main characteristics of the cloud are flexibility, scalability, location independence and reliability.

Many users tend to use the services provided by the cloud. But since multiple service providers may provide similar services in the cloud with different characteristics, the users find difficulty in identifying their best service based on
their requirements. The preferences also vary from one user to another based on their requirements. Selecting the best service based on user’s requirements is a complex process due to the multiple user requirements. In this chapter, an efficient novel service selection middleware adopting ELECTRE and PROMETHEE methodology for cloud environments has been proposed. Experimental analysis has been performed on the proposed middleware. The results show that the proposed middleware is efficient.

7.2 USER REQUIREMENTS THAT INFLUENCE SERVICE SELECTION IN CLOUD

It is obvious from the literature that many requirements such as cost, time, trust, responsiveness etc influence the selection of the service providers in the cloud. In fact, different service providers in a cloud can provide similar services with same functionality but can possess different criteria. Moreover, the requirements and the preferences of the users will also vary. In this section, a few factors that influence the selection of the service provider in a cloud have been provided.

**Turnaround Time** (Laiping Zhao et al, 2012) : Turnaround time is the time span from request submission by a user to request completion. It includes the service selection time, execution time of the job and also the network transmission time.

**Trust Degree** (Jing Zhou et al, 2011; Sheikh I. Ahmed, and Moushumi Sharmin, 2008; Yao Wang, and Julita Vassileva, 2007): More often, the user prefers higher trust relationships with the provider before exchange of any information from the user’s side. However, for some service requirements where confidentiality is not an important preference the user may select a lower level of trust degree.
**Service Cost** (Laiping Zhao et al., 2012; Gabor Kecskemeti et al., 2011): Service Cost is the cost for utilizing the services of the service provider. Most providers have fixed cost for their service.

**Capability** (Sheikh I. Ahmed et al., 2008): The number of services that the service provider will be able to handle at a particular instance of time.

**Responsiveness** (Patrick Wendell et al., 2010; Dabrowski C. et al., 2007): The time taken for the service provider to respond to the service request.

**Reliability** (Dimitrios Zissis, and Dimitrios Lekkas, 2012; Feng Zhu, 2007; Francoise Sailhan, and Valerie Issarny, 2005): The ability of the service provider to satisfy the requirements of the user without any service disruption.

**Flexibility** (Patrick Wendell et al., 2010; Laiping Zhao et al., 2012): The ability to meet the requirements of heterogeneous users.

The above factors influence the service selection in cloud environments. The requirements of each user may also vary from time to time. Many service providers in the cloud can also provide identical services. Therefore selecting a service based on user requirements is a challenge. The user can have multiple requirements. Therefore selecting the best service can be modeled as a multi-criteria decision making problem. Hence providing the user with the appropriate best service efficiently and effectively based on the user’s requirements and preferences is a major research problem in the cloud. Therefore an attempt has been made to propose a middleware that provides service to the user.
7.3 PROPOSED SERVICE SELECTION MIDDLEWARE IN CLOUD

The proposed service selection middleware for cloud environments is shown in Fig. 7.1. The Service Selection Middleware consists of Request Processing Unit, Service Selection Unit, Job Monitoring Unit and Service Delivery Unit.

The Request Processing unit helps the users to submit the request for the desired service along with the user requirements and preferences for the criteria. This unit assigns a job ID for the user request and stores as job request for further processing. Service Selection unit receives the job request, selects the best service provider in the cloud as per users’ requirement and assigns the job. The services in the cloud are obtained using Azure AppFabric. The service selection unit provides the information of the job details and the information of the selected service provider to the job monitoring unit. The execution of the job has been monitored. Once the job has been completed the information is sent to the service delivery unit, which also calculates the cost for the service and gives the information to the user.

Selection of the best service based on user requirements in a cloud is a multi criteria decision making problem. Along with the type of service, preference values of user for every Criterion plays a major role in service selection.
7.4 ALGORITHM FOR SERVICE SELECTION MIDDLEWARE

In this section, the algorithm for the Service Selection Middleware for Cloud environment is provided.

**Request Processor Algorithm**

Step 1: Each user submits *User_Request* message to Service Selection Middleware.
Step 2: For each user do
{
    Create User ID (U_ID)
    Retrieve and set user requirements (S_name, Tt, T, c, C, Tr, R, F)
    Create Job with Job ID (J_ID)
    Send User_Ack
}

Step 3: Submit Job Request to Service Selection Unit

Step 4: Retrieve the service parameters of each feasible service providers from Cloud and form an evaluation table

Step 5: Retrieve user requirements (S_name, Tt, T, c, C, Tr, R, F) and normalize to 1;

**Service Selection Algorithm using ELECTRE methodology (SSM_EC)**

Step 6: Get preference threshold $p$, indifference threshold $q$ and veto threshold $v$ for each user’s requirements

Step 7: For each feasible service provider based on pair wise comparisons do
{
    Compute Concordance Index $C(s_x, s_y)$
    Compute Discordance Index $D_i(s_x, s_y)$
    Compute Credibility Degree $CD(s_x, s_y)$
}

Step 8: Construct Credibility Degree Matrix

Step 9: Set Rank =1;

Step 10: Until all feasible service providers are ranked do
{
    Perform Ascending and descending distillations
    Compute the strength, weakness and final qualification
Identify the service provider with the highest qualification
Assign the identified Service Provider = Rank
Delete the identified service provider from the Credibility Degree Matrix
Increment the Rank by 1

Step 11: Submit the job request to the first ranked service provider.

(OR)

**Service Selection Algorithm using PROMETHEE methodology (SSM_PC)**

Step 6: For each feasible service provider do
{
Compute Preference function $PR_j(s_x, s_y)$ for each user requirements based on pair-wise comparison.
Compute aggregated preference indices $\pi(s_x, s_y)$
Compute Positive outranking flow $\phi^+(s_a)$ and negative outranking flow $\phi^-(s_a)$
Compute net outranking flow $\phi(s_a)$
}

Step 7: Rank the service providers in descending order based on the net out ranking flow.

Step 8: Submit the job request to the first ranked service provider.

Step 9: Go to Step 12

**Job Monitoring Algorithm**

Step 12: Set Alloted_service_log parameters (J_ID, S_ID, U_ID, SP_ID, Status) in the Job Monitoring unit.

Step 13: For every job do
Collect Job Status and Report at every check point
Verify and Update the Alloted_service_log
If ( Job Status == Complete)
{
    Generate Job_Completion report (J_ID,S_ID,U_ID,SP_ID, cost, delivery time, output, status)
    Send Job_Completion report to the service delivery unit
}

Service Delivery Algorithm
Step 14: For every Job Completed do
{
    Update Job_Status_Log
    Compute service cost.
    Deliver the Job result.
    Obtain the feedback parameter U_s and the payment from the user.
}

7.5 EXPERIMENTAL RESULTS AND DISCUSSION

The proposed middleware for cloud environment was simulated using Windows Azure in .NET Framework 4. Intel (R) Core (TM) 2 Duo CPU with 2.92GB RAM was used to simulate the environment. Fifty service providers were created in a cloud to provide various services. The proposed middleware was also
compared with SPSE (Laiping Zhao et al., 2012) and the performance was evaluated.

**Experiment 7.1: Analysis on Cloud Service Selection Time and Turn Around Time**

The users who require cloud services were permitted to provide their requirements such as service type and their preference values for the criteria. The number of criteria was varied by 2, 4, 6, 8, and 10. The service selection time taken to detect the best service as required by the user was determined by varying the feasible number of service providers in the cloud as 10, 20, 30, 40, and 50 for the proposed Middleware and is shown in Fig. 7.2. The proposed Middleware is also compared with SPSE for k=5 and is shown in Fig. 7.3.

![Fig. 7.2. Cloud Service Selection Time for SSM_EC](image)

**Fig. 7.2. Cloud Service Selection Time for SSM_EC**
Fig. 7.3. Comparison of Cloud Service Selection Time

It is observed from Fig. 7.2 that the Cloud service selection time increases with increase in number of service providers in a cloud for different Criteria k. This is due to the fact that when the number of service providers is high, the number of pair wise comparisons is higher and the number of computation of Concordance, Discordance Index is higher. This gives rise to bigger Credibility matrix and the computations for determining the rank also increases thereby increasing the cloud service selection time. It is observed from Fig. 7.3 that the Cloud service selection time for the SSM_PC is lower than the SSM_EC. This is due to the fact that the computations required for SSM_EC is higher than the SSM_PC due to the inherent nature of the algorithm. However, when compared to the already available algorithm SPSE, the performance of the proposed middleware is 57.81% and 32.51% better than SPSE in terms of cloud service selection time.
Table 7.1 Performance Improvement in terms of Cloud Service Selection time for SSM_PC, SSM_EC over SPSE

<table>
<thead>
<tr>
<th>Number of Service Providers</th>
<th>Cloud Service Selection Time (ms)</th>
<th>Improvement % of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSM_PC</td>
<td>SSM_EC</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>5.33</td>
</tr>
<tr>
<td>20</td>
<td>9.33</td>
<td>14.67</td>
</tr>
<tr>
<td>30</td>
<td>26</td>
<td>30.23</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>50</td>
<td>76</td>
<td>123</td>
</tr>
</tbody>
</table>

Average Improvement % 57.81 32.51

Fig. 7.4. Turnaround time for SSM_EC
The turn-around time experienced by the user for the proposed middleware and SPSE is shown in Fig. 7.4, Fig. 7.5. It was observed in Fig. 7.4 that the turn-around time for a particular job also increases with the increase in the number of service providers in the cloud. This is due to the fact that the turn-around time includes the service selection time, execution time of the job and also the network transmission time. It was also found that the number of requirements affect the cloud service Selection time as well as the turn-around time. It can be inferred from Fig. 7.5 and Table 7.2 that SSM_PC and SSM_EC respectively provide 50.74% and 37.40% performance improvement in terms of turn-around time when compared with SPSE.

Fig. 7.5. Comparison of Turnaround time
Table 7.2 Performance Improvement in terms of Turn around time for SSM_PC, SSM_EC over SPSE

<table>
<thead>
<tr>
<th>Number of Service Providers</th>
<th>Turn Around Time (ms)</th>
<th>Improvement % of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSM_PC</td>
<td>SSM_EC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>60.00</td>
<td>104.00</td>
</tr>
<tr>
<td>20</td>
<td>105.00</td>
<td>129.00</td>
</tr>
<tr>
<td>30</td>
<td>123.00</td>
<td>137.35</td>
</tr>
<tr>
<td>40</td>
<td>133.43</td>
<td>149.34</td>
</tr>
<tr>
<td>50</td>
<td>144.23</td>
<td>185.00</td>
</tr>
</tbody>
</table>

Average Improvement % 50.74 37.40

Experiment 7.2: Analysis on Service Selection Overhead

The users who were in need of the Cloud services were permitted to provide their requirements such as service type and their preference values for the criteria. The number of criteria was varied by 2, 4, 6, 8, and 10. The service selection overhead to detect the best service as required by the user was determined by varying the feasible number of service providers in the cloud as 5, 10, 15, 20, and 25 and is shown in Fig. 7.6. The service selection overhead is also compared with SPSE for k=5 and is shown in Fig. 7.7.
Fig. 7.6. Service Selection Overhead

Fig. 7.7. Comparison of Service Selection Overhead
It was observed from Fig. 7.6 that the average selection overhead increases with the number of service providers in the cloud. This is due to the fact that as the number of service providers in the cloud increases the number of computational messages required to identify the best service provider in the cloud increases.

**Table 7.3 Performance Improvement in terms of Selection Overhead for SSM_PC, SSM_EC over SPSE**

<table>
<thead>
<tr>
<th>Number of Service Providers</th>
<th>Average Selection Overhead</th>
<th>Improvement % of SSM_PC over SPSE</th>
<th>Improvement % of SSM_EC over SPSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>495</td>
<td>1015</td>
<td>51.23</td>
</tr>
<tr>
<td></td>
<td>584</td>
<td>1015</td>
<td>42.46</td>
</tr>
<tr>
<td>20</td>
<td>1995</td>
<td>4025</td>
<td>50.43</td>
</tr>
<tr>
<td></td>
<td>2364</td>
<td>4025</td>
<td>41.27</td>
</tr>
<tr>
<td>30</td>
<td>4500</td>
<td>9035</td>
<td>50.19</td>
</tr>
<tr>
<td></td>
<td>6534</td>
<td>9035</td>
<td>27.68</td>
</tr>
<tr>
<td>40</td>
<td>7900</td>
<td>16045</td>
<td>50.76</td>
</tr>
<tr>
<td></td>
<td>11004</td>
<td>16045</td>
<td>31.42</td>
</tr>
<tr>
<td>50</td>
<td>11000</td>
<td>25055</td>
<td>56.10</td>
</tr>
<tr>
<td></td>
<td>16004</td>
<td>25055</td>
<td>36.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average Improvement %</td>
<td>51.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>35.79</td>
</tr>
</tbody>
</table>

It is evident from Fig. 7.7 and Table 7.3 that that SSM_PC and SSM_EC respectively provide 51.74% and 35.79% improvement in performance in terms of selection overhead when compared with SPSE.
Experiment 7.3: Effect of Number of Users on Service Selection Time

The users who require cloud services were permitted to provide their requirements such as service type and their preference values for the criteria. The number of criteria was varied by 2, 4, 6, 8, and 10. The number of feasible service providers was set to 10. The service selection time to detect the best service in the cloud as required by the user was determined by varying the number of users as 200, 400, 600, 800, and 1000 and is shown in Fig. 7.8.

![Graph showing the effect of number of users on Service Selection Time](image)

**Fig. 7.8. Effect of number of users on Service Selection Time**

It is observed from Fig. 7.8 that as the number of users increases, the service selection time increases. However, for a maximum of 1000 users, the service selection time does not exceed beyond 400 ms when the number of criteria is 10. It is also observed for Fig. 7.9 and Table 7.4 that there is a performance improvement of 50.77 % for SSM_PC and 40.18% for SSM_EC over SPSE in terms of Service Selection time.
Fig. 7.9. Comparison of Service Selection time

Table 7.4. Performance Improvement in terms of service selection time for SSM_PC and SSM_EC over SPSE

<table>
<thead>
<tr>
<th>Number of Users</th>
<th>Service Selection Time (ms)</th>
<th>Improvement (%) of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSM_PC</td>
<td>SSM_EC</td>
</tr>
<tr>
<td>200</td>
<td>25.67</td>
<td>31</td>
</tr>
<tr>
<td>400</td>
<td>48.26</td>
<td>57.96</td>
</tr>
<tr>
<td>600</td>
<td>65.87</td>
<td>95</td>
</tr>
<tr>
<td>800</td>
<td>123.56</td>
<td>139.34</td>
</tr>
<tr>
<td>1000</td>
<td>154.28</td>
<td>179.56</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.6 CONCLUSION

In this Chapter, an efficient service selection middleware for cloud environments was proposed. The service selection was modeled as a multi-criteria decision making problem. Many criteria such as turnaround time, service cost, responsiveness, trust reliant, scalability, capability, flexibility, etc., influencing the service selection were considered. The experimental results show that the proposed middleware is efficient.