A method for optimized arrangement of service providers in elastic cloud platform

Samira Talebi1*, Hassan Khotanlou2, Mansour Esmaeilpour1

(1) Department of Computer Engineering, Hamedan Branch, Islamic Azad University, Hamedan, Iran.
(2) Department of Computer Engineering, Bu-Ali Sina University, Hamedan, Iran.

Copyright 2018 © Samira Talebi, Hassan Khotanlou and Mansour Esmaeilpour. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract
Nowadays considering developments in computing area, cloud computing is regarded as a new appearing technology in high level computations as well as a storage system in which the service suppliers receive money from users based on the amount of using their services. One of the most important necessities of a distributed system, like cloud, is employing an efficient method for discovery of a service which is in harmony with users' needs that helps managing services and timing users' activities. In this paper, an algorithm is introduced for minimizing the number of service registry message in unstructured peer to peer network of cloud platform for execution of which, the genetic algorithm has been benefited of discover an efficient arrangement of service providers as well as exploiting the breadth first search to discover connective arrangements. The execution is done on different arrangements of service providers the results of which show that the mentioned minimizing algorithm has reached to the most efficient arrangement which leads the least service registry messages.

Keywords: Cloud Computing, Service Discovery, Unstructured Peer To Peer Network, Elastic Cloud Platform.
1 Introduction

Cloud computing, depend on the method of resource distribution, consists of tree layer of Infrastructure as service (IAAS), Platform as service (PAAS) and Software as service (SAAS). Each layer is responsible for presenting a determined service to its users. At the lowest level which is called infrastructure as service, processor, memory and hardware components are presented by service provider as a service. Platform layer as service is the host of different environments of software development for service providing. At last the highest layer is software as service access to which is via web service and web browsers.

The process of finding suitable service which is in harmony with users' needs is called service discovery. One of the most important necessities of a distributed system, like cloud, is employing an efficient method for discovery of a service which is in harmony with users’ needs that helps managing services and timing users’ activities. Service discovery process in cloud systems requires service registry and also searching for suitable resource with the request of the user. Up to the moment, some mechanisms such as service description and discovery using semantic, discovery base on agent, discovery based on service quality structured and unstructured peer to peer networks as well as combined method have been introduced.

2 Related Works

By ever increasing development of web services, efficiency and scalability of service discovery has become an important issue in cloud computing.

2.1. Centralized Service Discovery

Centralize architecture is one of ancient methods of service discovery which includes a centralized index server in which there is a service registry operated by means of UDDI registry. In federated UDDI is offered for service query in high-scale environments. This method includes different pools which are periodically. Federated UDDI is an efficient and favorites solution for service discovery in distributed service networks. The cost of repeating updated information in such service is very high and as a result using this method for supporting dynamic data discovery directly is difficult. [1]-[2]

2.2. Decentralized Service Discovery

In decentralized way, there is no centralized index in decentralized architecture and service discovery is performed in a fully distributed environment in which there are several providers. Performing applications in high scale such as internet is one of the characteristics of decentralized architecture. [3]-[5]

2.3. Hybrid Service Discovery

Hybrid service discovery the combination of centralized and decentralized approaches which utilizes both approaches. Each group has a local registry which itself is managed by global registry. Execution of application in high-scale is one of the characteristics of decentralized architecture.

3 Suggested Method

As it mentioned, the search for intended service of users is successful only if service registry is done in the best way. This system consists of two parts. In the first part, elastic cloud platform together with all of its components and also providers' peer to peer network are introduced. intelligent method that were called minimization algorithm for service registry message will be explained afterwards which is the main subject of this article.
3.1. Elastic Cloud Platform

Introduced cloud platform in [6] is continually trying to provide cloud services and competes in business world. This notion that only one cloud platform covers all needs of users is completely impossible. In practice, the user's requests are limited and are cloud services as well. Elastic cloud platform in able to provide unlimited services to the users by outsourcing. Figure 1 shows an overview of elastic cloud platform. According to the picture, three main roles in elastic cloud platform are as follows [20]:

- Service Requester or User
- Cloud Platform
- Web Service Resource

![Elastic Cloud Platform Diagram]

Figure 1: Schema of elastic cloud platform with outsourcing scenario [6]

3.2. Algorithm of minimizing service registry messages in unstructured peer to peer network of cloud platform

In unstructured peer to peer network the more the number of neighbor nodes have the information of their neighbor nodes, the more optimized the task of search for a specific service in the network will be. Node arrangement of service provider's network should be in the way that produced the minimum number of service registry message. Arrangement of service providers in cloud platform peer to peer network that intended in [20] is a random arrangement in which the most service registry message will be produced. The aim of this article is to propose an algorithm for making an optimized arrangement of service providers in cloud platforms peer to peer network which leads to the least number of massages for service registry stage. For accomplishing the article for minimizing the number of service registry the genetic algorithms is employed. Definitions are as fallow:

**Definition 3.1. Chromosome and Population**

*Chromosome is an M * M matrix in which "M" is service provider’s node, genes is considered in binary state in which "1" Means the relationship between two service provider nodes and "0" means the lack of relationship between tow nodes.*

**Definition 3.2. Fitness Function**

*The aim of this article’s algorithm is to minimize the number of needed messages for service registry phase and is introduced as the relation (3.1):*
\( Z = \sum_{i=1}^{m} \left( \sum_{i} \text{Registration Packet Count} \right) \)  

In relation 1, "M" indicates the number of service providers and \( \sum_{i} \text{Registration Packet Count} \), Registration Packet count is the sum of sent messages from \( (i) \) service providers of the network. The aim of connective matrices or produced chromosomes by algorithm is intended to be breadth first search.

**Definition 3.3. Selection Operation**

Selection operation in a suggested algorithm are the methods of tournament and random.

**Operation 3.1. Cross over Operation**

Cross over operation in suggested algorithm is uniform and one point method.

**Definition 3.4. Mutation Operation**

**4 Assessment**

**A: The Number of Service Registry Messages and Iteration**

In this stage of execution, the effects of producing random numbers by machine and also by Iteration on the Number of Service Registry Messages will be examined. Execution is accomplished in two phases of Tournament and Random Operation. The results are shown as Table 1. Regarding the results, it is clear that the numbers reached by Random Selection in two states of Random Algorithm and also by employing Minimizing Algorithm by use of iteration are smaller. As it seen, in the Figure 2, according to the results, we achieved better results by increase iteration.

**Table 1: Implementation based on Iteration with and without execution algorithm**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Without Minimization Algorithm</th>
<th>With Minimization Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tournament Selection</td>
<td>Random Selection</td>
</tr>
<tr>
<td>100</td>
<td>( 5 \times 10^3 )</td>
<td>( 3.6 \times 10^3 )</td>
</tr>
<tr>
<td>150</td>
<td>( 5 \times 10^3 )</td>
<td>( 3.5 \times 10^3 )</td>
</tr>
<tr>
<td>200</td>
<td>( 5 \times 10^3 )</td>
<td>( 3.4 \times 10^3 )</td>
</tr>
</tbody>
</table>

**Figure 2: Number of service registry messages and Iteration with and without execution algorithm**

**B: The number of service registry messages and service providers**

In this part of execution, each time we have changed the number of service. The numbers of providers in execution were 80, 100, 120, 140 and 160. Execution has shown in Table 2. Based on the results shown in
Figure 3, the less number of service registry messages are produced by minimizing algorithm of tournament and random than the number of messages produced using only random algorithm.

Table 2: Implementation based on service providers with and without execution algorithm

<table>
<thead>
<tr>
<th>Node number</th>
<th>Without Minimization Algorithm</th>
<th>With Minimization Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tournament Selection</td>
<td>Random Selection</td>
</tr>
<tr>
<td>100</td>
<td>$4.9 \times 10^5$</td>
<td>$5 \times 10^5$</td>
</tr>
<tr>
<td>120</td>
<td>$8.6 \times 10^5$</td>
<td>$8.5 \times 10^5$</td>
</tr>
<tr>
<td>140</td>
<td>$11.2 \times 10^5$</td>
<td>$11.6 \times 10^5$</td>
</tr>
</tbody>
</table>

C: The Number of Service Registry Messages and TTLr

In this stage of execution, the effects of TTLr on the Number of Service Registry Messages we be examined. Execution with quantities of TTLr is considered as equal to 1, 2 and 3. The results of execution by different TTLr, Tournament and Random selection are shown in the Figure 4. We will have a significant decrease in Number of Service Registry Messages by use of Minimizing Algorithm.

Table 3: Implementation based on TTLr with and without execution algorithm

<table>
<thead>
<tr>
<th>TTLr</th>
<th>Without Minimization Algorithm</th>
<th>With Minimization Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tournament Selection</td>
<td>Random Selection</td>
</tr>
<tr>
<td>1</td>
<td>$1.4 \times 10^8$</td>
<td>$2.6 \times 10^8$</td>
</tr>
<tr>
<td>2</td>
<td>$3.6 \times 10^9$</td>
<td>$5 \times 10^9$</td>
</tr>
<tr>
<td>3</td>
<td>$5 \times 10^9$</td>
<td>$5 \times 10^9$</td>
</tr>
</tbody>
</table>
5 Conclusion and Future Works

As it can be seen in the results, the more Number of Service Providers in Peer to Peer Network of Cloud Platform will be, the more Number of Service Registry Messages should be produced. Also the more we consider TTL_r, because of increasing dispatching levels in network, the more Service Registry Messages will be produced. The issue aroused is that increasing TTL_r leads to increase Service Registry Messages which results in more network traffic. On the other hand, decreasing TTL_r also leads reaching less Service Registry Messages to the neighbors. So the best solution to prevent this issue is decreasing or increasing TTL_r based on network traffic.

Considering the results, with different parameters such as Service Provider Numbers, TTL_r and Iteration, it is observed that in case of using Minimizing Algorithm with regard to different parameters, the least Service Registry Messages in Unstructured Peer to Peer Cloud Platform Network in comparison with the time we use random Algorithm.

For more researches, one can use other revolutionary Algorithms such as Particle Crowding, Ant Colony and other algorithms rather than Genetic Algorithm and compare speed and accuracy of other proposed algorithms, in finding better answer, with the results of this article. Also Peer to Peer network parameters, traffic level of service provider nodes and connection links traffic can be used in optimization time.

References
https://dl.acm.org/citation.cfm?id=963713

https://doi.org/10.1109/NAS.2008.16

https://doi.org/10.1109/ICWS.2004.1314739

https://doi.org/10.1007/978-3-540-24655-8_32

https://doi.org/10.1002/cpe.2993