Resource discovery algorithm based on hierarchical model and Conscious search in Grid computing system

Nasim Nickbakhsh¹*, Mohammad Reza Sultan Aghaei²

(1) Department Of Computer, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran
(2) Department Of Computer, Islamic Azad University Isfahan (Khorasgan) Branch

Copyright 2017 © Nasim Nickbakhsh and Mohammad Reza Sultan Aghaei. 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
The distributed system of Grid subscribes the non-homogenous sources at a vast level in a dynamic manner. The resource discovery manner is very influential on the efficiency and of quality the system functionality. The “Bitmap” model is based on the hierarchical and conscious search model that allows for less traffic and low number of messages in relation to other methods in this respect. This proposed method is based on the hierarchical and conscious search model that enhances the “Bitmap” method with the objective to reduce traffic, reduce the load of resource management processing, reduce the number of emerged messages due to resource discovery and increase the resource according speed. The proposed method and the “Bitmap” method are simulated through Arena tool. This proposed model is abbreviated as “RNTL”.

Keywords: Grid computing, Distribution system, Hierarchical model, Conscious search.

* Corresponding Author. Email address: nasim.nikbakhsh@gmail.com, Tel: +989131023143.
1 Introduction

Resource discovery is one of the essential services in resource management. The number of resource search is influential in efficiency and the quality of the system functionality [1 and 2]. The objective here is to propose an optimized model in resource discovery in the distributed Grid system in a format where the distributed messages in the Tree would not generate heavy traffic and the resource message would not encounter the failure point. The hierarchical and conscious search models are applied in this method. In conscious search, the member nodes in the network seek to accumulate and store data specific to their neighbors and these resources [2 and 3]. Any node of the Tree is a resource manager and manages its own resources in its territory. To reduce messages emerged from resource search, the source manager must be equipped with the data from other managers [3 and 5], while high volume of data slows the process down, requires more storage space and more data up-dating; consequently, a model can be proposed which would reduce traffic and the managers’ processing load in resource discovery [6 and 7]. Based on the available studies, the following assumptions are evident:

- Implementing resource to apply centralized manner is simple but it would encounter an increase in resource number and requirements and failure points; thus’ it is necessary to apply decentralized models for this purpose [8, 9 and 10]. The major advantage of hierarchical model: it can be considered in different levels of different planning polices while the centralized manager would lead to complete system failure [11, 12 and 13].
- Provided, that the resource manager is equipped with data from domestic resources of other managers resource discover would take place in a rapid manner, thus it is better to apply conscious search [14 and 15]

The proof of the following hypothesis is sought in this article:

- By applying the hierarchical model and conscious search the resource managers would not encounter the failure point; thus, a reduction in their possessing load
- By determining the search messages’ time-to-live, they will not be transmitted in all levels, but the levels where the users’ requirements would be met
- This proposed method would improve the “Bitmap” method
- Provided that the resource managers are distributed in a manner where at any level their number is decreased against which the Tree height is increased, thus, will be less messages due to resource discovery

A “Bitmap” method is presented based on hierarchical mode and conscious search by [12] where there is no limit for search message advances towards the leaf nodes and the search message is transmitted to all nodes with the same considered source. Since the search messages advances to the leaf nodes, more resource are reserved and the waits time for the next number of messages increases for the release of the reserved resources.

In this proposed method, any resource manager saves some data related to the managers at its lower level and, the available number of resources of concern at every level of the Tree is determined. The search messages’ time-to-live (TTL) is determined based on the number of resources at each level; the search message is not distributed to all nodes; the search message transmission is restricted. In determining the search messages’ time-to-live value, closeness of the resource manager and the number of requested resources are of concern. Here, the proposed and “Bitmap” methods are simulated through the Arena tool. Results obtained here, when compared with that of the “Bitmap” methods’ reduce the nodes processing load level, the number of the search messages and the network traffic; therefore, the Grid system efficiency and resource access speed increases. This proposed method is abbreviated as “RNTL”.
2 literature Review

The management and resource discovery models are classified in five in [16]: Central, Distributed, peer-to-peer, hierarchical and agent based. The central models where a server is available as the resource manager, is addressed in [12 and 17]. The central manager collects the data related to the resource status, which is shared in a dynamic manner. If the member users of Grid need a resource, their request is presented to the central manager. The flooding method is assessed in [4 and 5] for resource discovery. Here, the users’ request and resource discovery is accomplished through Cascade algorithms. In flooding method, the search messages are transmitted to all member of network and every network member transmits the search message to all of its connected neighbors. The “MMO” method is introduced in [18], where the resource managers are distributed based on “peer-to-peer” and they use the resource information for searching. The Super Peer model is assessed in [12 and 17] where in every organization the resource manager is implemented in a hierarchical or centralized manner. Here, from every organization one node is selected as the Super-Peer and all Super-Peers become connected in a (PP) pattern. The two “footprint” and “Bitmap” are presented in [11 and 13] where the resource managers are organized according to hierarchical structure and the resource discovery takes place through conscious search.

a. Resource discovery in Grid through hierarchical method by “Footprint”

The “Footprint” method is introduced in [13] where the hierarchical method is adopted in resource management and discovery. In this method all edges are codified. Every resource manager has two “Counter Bitmap” and “Path Bitmap” records.

The “Counter Bitmap” Record determines the count of the available resources in the child node and the local resource data are stored in the “Path Bitmap” record.

<table>
<thead>
<tr>
<th>XP</th>
<th>Linux</th>
<th>Unix</th>
<th>Vista</th>
<th>MacOS</th>
</tr>
</thead>
</table>

Figure 1: A sample of “counter Bitmap” record, the operating system

```
0 0 0 0 0 0 2 3 0 0
```

Figure 2: A sample of “Path Bitmap” record, with “vista” operating system

Provided that, both the “path Bitmap” and “counter Bitmap” here a node like the one in Figure 1 and 2, that is, the child node with code 11 has the Vista system agent and 2 indicates the count of the necessary bits to exhibit the node binary codes of the concerned resources. A sample of resources search Tree, where the “Footprint” method is adopted is shown in Figure 3.
Here, an example is assessed regarding Figure 3. Assume that a user from node 4 requires the “MACOS” resource; node “4” which is the resource manager does not have this resource and transmits the request to the Father node. In the root node, the “Path Bitmap” number of “MACOS” resource is 5 and 17. The number 5 is the binary code length 17 that is 10001. Through the binary code, the node containing the required resource becomes accessible in a direct manner. In “Footprint” method, the managers processing load is sizable due to “Path Bitmap” calculation. The extent of the resource reserved for the applicant is small, thus making the Tree updating difficult and generate an increase in network traffic.

b. Resource discovery in Grid through hierarchical method by “Bitmap”

“Bitmap” method is presented in [11]. The local resource information of every node is stored in the “local resource bitmap” records. The “attribute counter” record stores the count of available resource in child nodes and the “index” record saves the child resource information. The “index” record with index “Bitmap[i] =0” indicates that the resource “R” with “i” feature is absent among the child of this node. A sample of the structured Tree, based on this method is drawn in Figure 4.

According to available studies, it can be deduced that the cascade algorithm distributes the request to all nodes (even the nodes lacking the resource of concern). The search speed in cascade method is very low in relation to “Bitmap”, “MMO” and “Footprint” methods and it generates heavy traffic. The “MMO” algorithm applies the “Point-to-point” structure to organize the resource manager with the same feature
like that of the cascade algorithm, but since here “Point-to-point” is applied, it is of higher speed. The “Footprint” algorithm applies the hierarchical and conscious search models and transmits the requests to only the intermediate nodes with the resource of concern in the local resource or its children, while here the processing load of the managers is high. The processing load of the managers in “Bitmap” is lower than that of the “Footprint” method and the number of target nodes receiving the requests is low in “Bitmap” compared to other methods. This newly proposed method improves the “Bitmap” method.

3 Methodology

The developed model here is based on the hierarchical and conscious search model that performs according to the count of resource at any level of Resource Number at each Tree. This newly introduced method is abbreviated as the “RNTL” (resource number at any tree level). Many organizations share their resources through Grid. In any organization there exists one member as the local manager. The users of every organization can have access to Grid resource through this manager. In this method the nodes are connected to constitute one node of the Tree; that is, the resource managers are. When a user becomes a member of the organization it has a resource request, this message is given to the local manager and if the organization lacks the resource of concern or the resource is occupied, according to the Tree structure, the request is diverted to the child node for search purposes. If the same happens here, the request is returned to the father node. This process can continue and it is possible to reach the root. To transmit request from a level to a higher one only one message would suffice; to transmit request from a higher to a lower level generates of many messages and an increases the network traffic. The purpose here is to reduce the number of messages transmitted from higher to lower levels. Accordingly, the requested message should not be transmitted to all the children, thus, by holding the least data from children resources, unnecessary transmission of search message to all of them would be eliminated.

a. Preparing the resource search Tree

Every resource manager has a resource data table and the total presented number from the resource of concern is hold at every sub-level. For example, in the Tree, Fig. 5, node 1 must have the number of concerned resources of concern presented at levels one, two and three. To accomplish this, every node, transmits its number of local resources to the Father node. This pattern continues to the root. The first and the last stages of tree preparation are shown in Figure 5 and 6.

![Diagram of resource search tree]

Figure 5: Preparing the resource search Tree and transmitting the provided number from the resource from level 3 to level 2
b. User request respond pattern

The record saved by every node for the resource of concern is named “R”. The R[0] determines the number of resources of concern in the local resources of this node. The related record length of the data of a resource in every node is “h-L” where, “h” is the height and “L” is the number of the level where the node is situated in. Upon issuance of the user request, first, should be determined that to which level of the Tree the search message is distributed to. When the local manager of the organization receives the request message and does not have the resource of concern, first, it adds a section named “TTL” to the request message and distributes the resource message in the Tree, with respect to remaining time-to-live time. The resource manager determines the “TTL” value based on the number provided from the resource of concern at every sub-level. It is assumed that in a Tree structure, the resource manager “X” is located at “LX” level, which is the data record of the resource of concern, the “RX”. Node “X” must assess the value of “RX [L-LX]” to know the count provided from the source of concern at level “L”. Every node, upon receiving a message from the Father node, begins to search from the resource of concern in its local resource and deducts one unit from message TTL. If TTL value=0 Father does not transmit search message to its child and if TTL>0 he would transmit search message to all children at the next level, Figure 7.
A user requests 4 operative windows operating system resource from node 1 at level 0, Fig. (8). Upon receiving the request, node 1 begins to assess the local resources, that is, \( R[0] \), knowing that there exists one window operating system in its local resource, it is not able to respond; therefore, it determines value of the message TTL and transmit the message to the required levels. As to \( R \) record, if the request message advances to level 2 which can reserve 4 resources for the user and send a message to the applicant as a notification. Every node reduces one search message unit upon receiving a search massage and the case TTL ≠ 0 it transmits the search message to all child nodes and at next level, observe Figure 6, 9 and 10.

Figure 7: User request respond pattern in a node
Figure 8: Receiving a request by node 1 and transmitting search message to all child nodes at level 1

Figure 9: Receiving the search message by nodes at level 1 and transmitting the same to all child nodes at level 2
c. Updating the resource search Tree
A node that releases or attributes a resource should update the record therefore, a root approaching process. For example, if the node as a Tree leaf is to update a resource record, the count of the messages transmitted for updating in $h-L$, that is, the most message count for updating is $h-L$.

4 Experimental Studies

The proposed method here is being presented in accordance with that [11] and is assessed and compared by Arena tool.

The resource search Tree is simulated through both the methods. Every Tree sample is simulated based on the hypotheses, subject to uniform conditions by applying “RNL” and “Bitmap” algorithms, the obtained results of which are compared at the end.

a. The simulation of hypotheses
• The resource search Tree heights 3 and 4 fold
• The network delay is ignored
• For better illustration every Tree is exhibited for one type of resource
• The user, in any request message, request only one of the resources of concern
• Requests of node are issued in a random manner

The three Trees exhibited in Figure 11, 12 and 13 are simulated through the “RNTL” and “Bitmap” methods.
Every Tree is simulated by applying the Arena tool; for example, simulation of node 2 from Tree number 2 is shown in Figure 14, the nodes are connected to one another according to the exhibited structure.
### Table 1: The test run for Tree number 1

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Number of request massage made by the Tree nodes</th>
<th>Percentage of available resource of concern in search Tree in relation to total requested resource</th>
<th>Resource request count through node 1</th>
<th>Resource request count through node 2</th>
<th>Resource request count through node 3</th>
<th>Resource request count through node 4</th>
<th>Resource request count through node 5</th>
<th>Resource request count through node 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>10%</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>25%</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>50%</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>70%</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>100%</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>10%</td>
<td>40</td>
<td>10</td>
<td>30</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>25%</td>
<td>40</td>
<td>10</td>
<td>30</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>50%</td>
<td>40</td>
<td>10</td>
<td>30</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>70%</td>
<td>40</td>
<td>10</td>
<td>30</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>100%</td>
<td>40</td>
<td>10</td>
<td>30</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

### Table 2: The test run for Tree number 2

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Number of request massage made by the Tree nodes</th>
<th>Percentage of available resource of concern in search Tree in relation to total requested resource</th>
<th>Resource request count through node 1</th>
<th>Resource request count through node 2</th>
<th>Resource request count through node 3</th>
<th>Resource request count through node 4</th>
<th>Resource request count through node 5</th>
<th>Resource request count through node 6</th>
<th>Resource request count through node 7</th>
<th>Resource request count through node 8</th>
<th>Resource request count through node 9</th>
<th>Resource request count through node 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>40</td>
<td>10%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
<td>25%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>50%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>40</td>
<td>75%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>40</td>
<td>100%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>100</td>
<td>10%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>17</td>
<td>100</td>
<td>25%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>50%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>19</td>
<td>100</td>
<td>75%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>100%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Table 3: The test run for Tree number 3

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Request message made by the Tree nodes</th>
<th>Percentage of available resource of concern in search Tree in relation to total requested resource</th>
<th>Resource request count through node 1</th>
<th>Resource request count through node 2</th>
<th>Resource request count through node 3</th>
<th>Resource request count through node 4</th>
<th>Resource request count through node 5</th>
<th>Resource request count through node 6</th>
<th>Resource request count through node 7</th>
<th>Resource request count through node 8</th>
<th>Resource request count through node 9</th>
<th>Resource request count through node 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>10%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>25%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>50%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>75%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>100%</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>10%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>27</td>
<td>25%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>50%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>29</td>
<td>75%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>100%</td>
<td>40</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Not that in the second test it is assumed that 40 of the source request message is issued on the nodes part and 25% of the requested amount for the source of concern is available in the Tree search resource, that is, to begin with, there exist 10 free sources of concern in the subject Tree. The details of the simulated Trees in test 2, through “RNTL” and “Bitmap” algorithms are presented in Figure 15 and 16.

Figure 15: The simulated Tree through “Bitmap” algorithm in test number 2
The results obtained from comparing these two models through tests 1-5 indicate that, the node number due to the search for 40 resources are met and distinguished, Figure 17.

The sample of the Trees simulated in Test number 8, are shown in Figure 18 and 19, when it is assumed that the request number is 100, of which 50% is available in the Tree, that is, initially, there exist 50 free resources in the whole Tree, Figure 20. the results obtained from simulating Tree number 2 for 40 resource search messages, Figure 21.
Figure 19: The simulated Tree through “RNTL” algorithm in test number 8

Figure 20: The results obtained from simulating Tree number 1 for 100 source search messages

Figure 21: The results obtained from simulating Tree number 2 for 40 source request messages

The details of the simulated Trees in test number 17 through both “RNTL” and “Bitmap” algorithms are shown in Figure 22 and 23, when it is assumed that the request number is 100, of which 25% is available in the Tree, that is, initially, there exist 25 free resources in the whole tree, Figure 24.
Figure 22: Simulation of Tree in test number 17 through “Bitmap” algorithms

Figure 23: Simulation of Tree in test number 17 through “RNTL” algorithms

Figure 24: The results obtained from simulating Tree number 2 for 100 resource search messages

The results obtained from simulating tree number 3 are shown in Figure 25 and 26.
The results obtained from simulating trees number 2 and 3 through “RNTL” method are expressed in Figure 27. In this comparison with the number of resource managers is equal in both the Trees. The managers of Tree number 2 is organized with a height of 4 and the same for Tree number 3 with a height of 3. The number of resource request message through every resource manager is the same in both the Trees. By comparing the Trees of 3 and 4 heights it could be claimed that, provided that the number of resource managers remain constant, the number of transmitted messages in the Tree with height 4 is less than that of the Tree with height 3 and the generated traffic in the Tree with Height 4 is lower.
Based on the results obtained in Figure 28, 29, 30 and 31, increasing the number of available resources in the tree, has reduced the number of the Search messages.

Figure 28: Comparison of the 3 Trees for 40 resources request message when 10% of the requested number from the resource of concern is available in the Tree

Figure 29: Comparison of the 3 Trees for 40 resources request message when 100% of the requested number from the resource of concern is available in the Tree

Figure 30: Comparison of the 3 Trees for 100 resources request message when 10% of the requested number from the resource of concern is available in the Tree
Figure 31: Comparison of the 3 Trees for 100 resources request message when 100% of the requested number from the resource of concern is available in the Tree

5 Conclusion

The grid system is distributed at a vast level and for resource management, this is not an appropriate central model and as the system expands the central manager faces failure and is forced to adopt models where the resource managers are distributed. The hierarchical model is one of the most important models that distribute the resource managers at different levels. The systems that apply conscious search method to discover resource have one multi service directory which collects data related to the resource status and the manager apply these data to attribute tasks to resources or discover resources. Therefore, models based on conscious search methods are more rapid than the ones not based on conscious search and increase the network efficiency.

This proposed method is based on hierarchical and conscious search model and concentrates on reducing the message number due to resource search. By determining the message time-to-live, the search message advance level among the children becomes limited and prevents the transmission of unnecessary messages. In this “RNTL” model lower counter of resource managers are met and the managers receive less search message; hence, speed processing user requests and resource discovery increase. By adopting this method, less resource is reserved and message search waiting time is reduced for the next search in making the reserved resources free. Based on the run assessments “RNTL” algorithms improve the “Bitmap" algorithm.

For future studies, a combination of hierarchical and P-P models could be applied in a manner where for a small or medium sized organization a central or hierarchical model is applied. Next, one node is selected as the main manager in every organization which are connected together based on “Point-to-Point” structure. The cost of requested resource can be determined based on TTL message, when discovering resources.

References

https://doi.org/10.1109/ICMCS.2012.6320221


https://doi.org/10.1109/MC.2012.363


http://dx.doi.org/10.1016/j.jnca.2013.09.013

https://doi.org/10.1016/j.parco.2008.07.001

https://doi.org/10.1109/dexa.2005.166

https://doi.org/10.1109/cmc.2009.74

https://doi.org/10.1109/MIC.2001.968826