Journal of Computer & Robotics 12 (2), 2019, 57-66

Computer & Robotics

# Weighted-HR: an Improved Hierarchical Grid Resource Discovery

Mohammad mehdi Gilanian sadeghi [a] ; Mahdi MollaMotalebi [b, *]

[a] *Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[b] *Islamic Azad University of Buinzahra*

**Abstract**

Grid computing environments include heterogeneous resources shared by a large number of computers to handle the data and process intensive applications. The required resources must be accessible for Grid applications on demand, which makes the resource discovery a critical service in Grid environments. In recent years, diverse techniques are provided to index and discover the Grid resources. The response time and message load during the search process highly affect the efficiency of resource discovery. This paper proposes a new technique to forward the queries based on the resource types which are accessible through each branch in hierarchical Grid resource discovery approaches. The proposed technique is simulated in GridSim and the experimental results indicated that it is able to reduce the response time and message load during the search process especially when the Grid environment contains a large number of nodes.

*Keywords:* Grid computing; Hierarchical; Weight-table; Query forwarding; Resource discovery.

## 1.Introduction

Grid computing aims to handle data and process intensive applications. Grid environments typically include a large number of nodes each of which owns one or more resources to be shared and used by the Grid applications [1, 2]. Grid nodes and resources belong to wide range of heterogonous types and applications. Generally, the scale of grid networks is large in terms of the number of resources and users as well as geographical spread [3]. With regard to the continuous increasing of networks' bandwidth and resource varieties, Grid computing is emerging technology as the next-generation computing platform in government, science, and business[4]. The Grid resources usually are spread over the areas belonging to independent organizations and they are utilized to handle multiple segments of an application.

Resource discovery is the process that takes the execution requirements of Grid applications, searches the network for required resources and returns a set of Grid nodes including resources matched to the application requests [5, 6]. Grid environments are inherently large-scale and dynamic; as a result, Grid resource discovery is a time and also message consuming process which affects the efficiency of the whole Grid [7, 8] . The message load and response time are two main parameters to evaluate the resource discovery efficiency. The response time shows the time between requesting a resource and returning the resource owner addresses. The message load indicates the number of transmitted messages per second between the Grid nodes in search process [9, 10]. In general, there are two approaches to search the resources: Blind and Informed. Two popular schemes, Gnutella [11] and Random-walks [12] use the Blind search. The Blind search works by the flooding algorithm, where Grid nodes do not have information about existing resources, nodes and their location. Consequently, the resource requests broadcast to several Grid nodes to find the required resources. Broadcasting generates large amounts of

* Corresponding Author. Email: motalebi@gmail.com

message load particularly in large scale Grid environments.

In an Informed search schmes such as CAN [13] and Chord [14], Grid nodes know the current nodes and their location. Therefore, the resource queries directly send on the network using the indexed information. As a result finding the required resources is faster and has lower message loads. Consequently, the informed search schemes are preferred in large-scale Grid environments. Nowadays, most of the resource discovery techniques act as informed; however, the maintenance costs will be increased particularly, when the conditions are dynamic.

The hierarchical is a prevalent resource discovery approach in Grid environments. This approach keeps the resource information of Grid nodes in multiple indexing nodes as hierarchical in order to limit the search domain and increase it gradually as needed. Some other approaches are based on peer-to-peer [11, 13], agent-based [20], super-peer [6] and centralized [15] structures.

This paper proposes a new weight-based technique in order to improve the hierarchical resource discovery approach. The remainder of this paper is organized as follows. Section 2 represents works related to hierarchical resource discovery approaches. The proposed technique is described and evaluated in Section3 and Section4 respectively. Section 5 concludes the paper.

## 2. Related Work

With regard to the significant impacts of resource discovery on the efficiency of Grid computing systems, different techniques and structures (such as Centralized, Hierarchical, Super-peer, Peer-to-peer, etc.) have been proposed in recent years in order to improve the indexing and discovery of Grid resources [16]. This paper proposes a weight-based technique in order to improve the efficiency of hierarchical solution regarding the response time and message load.

In the hierarchical structure, resource information is recorded in multiple servers as a tree organization. The resource owners are located at the leaves of the tree and indexing nodes at every level can keep and forward the resource information to their parent [9]. A node located at upper levels is busier, and it keeps information of more resources. Every node in the hierarchical structure should answers the queries which are received from its descendants. This solution may encounter high message loads in large-scale environments especially on upper levels of the hierarchy. It typically deals with potential bottleneck in the root node [17, 18]. In the following paragraphs, some hierarchical-based schemes for resource discovery in Grid environments are discussed.

The authors in [19] proposed a resource discovery system with a model describing Grid resource information regarding the computing and storage elements. In this case, a module called Information-Finder needs to determine the required resources, and then retrieve the details of resource information. Upon a query received, Information-Finder does the resource locating by using querying index servers as hierarchical. When the resources matched to the query found, it then obtains the complete up-to-date resource information, so as to whole static and dynamic information of the resources retrieved.

Here, a thread pool controls the parallel mechanism of resource discovery and information retrieval in order to avoid the overloads. Furthermore, a time limited cache is applied to retain the information of frequently requested resources, and then use them in the following searches to decrease the resource discovery overheads.

In the hierarchical method proposed by Ramos and Melo [20], inside different virtual organizations of Grid environment, there are some hierarchical linked master nodes containing several slave nodes. The master nodes are responsible to update resource database, and slave nodes also gather resource information from Grid nodes. The slave nodes address by using a description file associated to them. The description file is an XML document saved in the master node, and contains IP addresses and all slave nodes name associated to existing master node. Upon a slave node obtains a resource query, it then checks the requested resource satisfaction. Next, it returns the result to its master node. After receiving answers from all slave nodes, the searching module in master node processes and returns them to the requester.

This technique improved scalability by dividing the resource management between different master nodes; however, leaving master nodes cause some slave nodes be inaccessible to the system.

Ma et al. [14] propose a hierarchical resource discovery scheme which is based on the small-world network. Here, each node in a small-world network has diverse contacts with close and far nodes. The query forwarding holds by long ranged contacts via message passing among numerous nodes.

The Grid in this technique is mapped to a hierarchical tree structure where the upper nodes supervise their descendant nodes. Moreover, the upper level nodes have higher bandwidth than lower ones, and then, they deal with much more processes. There are different levels; and they have their own administration node to meet the autonomous managements. Furthermore, it uses redundancy in order to overcome the single point of failure, and keeping the data stable and consistent.

The researchers showed that usually requests are handled by short ranged nodes, and traversing far nodes is not needed. Generally, two strategies, generic and senior, use to search the required resources. In generic scheme, users are aware of clusters' resources, and they can access them directly. Thus, it eliminates the investigation of clusters, which are not related to the current request. Senior scheme enables more attributes for queries, and also uses shortcuts in small-world network. Hence, it decreases the query distribution costs. This technique is more scalable than centralized ones. It also does not cause the bottleneck problems which increases its reliability.

Huedo et al. [21] proposed a grid architecture containing diverse layers of meta-schedulers which is organized in a hierarchical structure. Also, every target grid is held as a resource in a recursive approach by means of the equivalent interfaces for resource management. By using, this architecture a straightforward federation of grid infrastructures will be allowed, and then it will observe organizational boundaries. It uses the Globus toolkit services in a recursive way in order to create the required virtualization technology. It also provides a powerful abstraction of the fundamental grid resource management services. This technique decentralizes the control of resources; hence, the operation in every level of the hierarchical tree is autonomous, and has different policies for scheduling, resource sharing and user access control. Here, there is no need to deploy new services, by using the Globus interfaces. Moreover, there exist fewer entities at every level, as well as information dissemination or scheduling take the advantage of spatial locality. On the other hand, this architecture has high overhead and cannot be useful in dynamic environments.

Yan et al. [22] suggested a hierarchical resource discovery system scheme which is based on the small-world network. In fact, in a small-world network, every node has differently ranged contacts where there are near and far nodes. The query forwarding is done by long ranged contacts via message passing among several nodes. The Grid in this technique is mapped to a hierarchical tree structure which the upper level nodes supervise their descendant nodes. Moreover, the upper level nodes have more bandwidth in comparison with lower ones, and they perform more processes.

The proposed technique utilizes redundancy in order to overcome the issue of single point of failure, and to keep data consistency. Integrating the small-world networks into clusters distribute the messages faster than usual networks. The researchers revealed that the most requests are answered by near nodes without traversing in the far nodes. Two approaches, senior and generic are utilized in order to do search for the resources.

In generic approach, usually users are aware of clusters' resources, and then they have access to them directly. Therefore, it eliminates the inquiry of clusters that are not concerned to the existing request. Senior approach makes more attributes for queries. It also uses the shortcuts in small-world networks that lead to a decrease in the query distribution costs.

Chang and Hu [23] presented a resource discovery technique which uses tree architecture in order to do indexing and searching for the resource information. Each indexing node indexes the resource information of its own children nodes in terms of a bitmap form. Each indexing node has a local bitmap to record the local resource information and one index bitmap to record its own children's resource information. Here, the resources with higher values would return as the response, if a resource with a particular value could not be found.

This technique has high message load particularly in large scales, because when the number of resources increases, most of the bitmaps on upper level indexing nodes will be masked. This causes the issued queries to be forwarded to all or most of the children indexing nodes that would increase the message load. Ebadi and Khanli [24] presented a hierarchical scheme for resource discovery in grid environments. The scheme improves fault tolerance and reduces response time of the resource discovery process. In this case, sibling nodes in every level directly send resource requests to each other.

To provide the fault tolerance, the resource discovery process continues to find more resources after finding the first one. It also accelerates the replacement functions if it is necessary. On the other hand, it suffers from traffic overheads, and that is inefficient when Grid-scale increases quickly.

Zhongping Zhang et al. [25] proposed a grid resource discovery model using combination of the block and chessboard distances. It organized the overlay network by combination of hierarchical and tree structures. The service nodes are placed at the bottom, and each resource constitutes one physical resource. The grid resources are classified based on their type as multiple layers.

The structure is in two levels, overlay network as the upper layer, and physical resources as the lower layer. The nodes in every layer are assigned to a domain based on their type and upper layers manage the lower ones. Each domain is managed by a resource router that keeps service information of domain nodes.

Here, when the number of nodes in a domain exceeds a threshold value, the domain would be split into two or more smaller ones providing small granularity of resource type. Also, a domain is merged to other one when its nodes count is reduced to a defined limit. Consequently a multilayer overlay network is formed. This model provides fault tolerance and partly scalability. This also is able to handle dynamicity of grid environments with almost short response times.

Our paper proposes a weight-based resource discovery technique that forwards the queries to appropriate branches in hierarchical structure based on the number of resource types that are accessible through each branch. The proposed technique is described and analyzed in the following sections.

## 3. The Proposed Technique

The Weight-based Hierarchical (Weighted-HR) technique aims to decrease the message load and response time in hierarchical resource discovery by limiting the children nodes receiving the forwarded queries. It is performed by adding a weight table to indexing nodes indicating the weight of their branches to be chosen in query forwards. In each indexing node, the weight table indicates the number of resource types that are accessible through each branch. By this way, when a query is received by an indexing node, it forwards the query to the limited number of branches with highest weights for the requested resource type. The weight-based technique has been evaluated in our previous research [26] for super-peer based Grid environments. The resource information of superpeer nodes cannot be collected during the join or update processes in Super-peer structure, but such information is available in the hierarchical structure. Also, the peer-to-peer communication existing in super-peer structure is different from Hierarchical one. In this section, the proposed weight-based technique for forwarding of Grid resource requests in hierarchical indexing structure is evaluated.

Figure 1 illustrates the indexing structure of Weighted-HR. The regular nodes consisting of the resources are located in the leaves of the hierarchical tree. These nodes are linked to indexing nodes at level N-1 of the tree. The indexing nodes in different levels of the tree keep the resource information of descendant nodes. There is also a weight table that contain row entities corresponding to branches and column entities indicating the number of resources from different types that are accessible through each branch. Once the resources change in regular nodes, the new information send to the parent of indexing nodes, which is placed at level N-1 of the hierarchical tree. The parent nodes update their resource information and modify the weight table corresponding to the resource types that are accessible through the branches. Then they forward the new resource information to their parent at level N-2 of the tree. This process goes continuously, until all ancestors could update their weight table with the changed information.

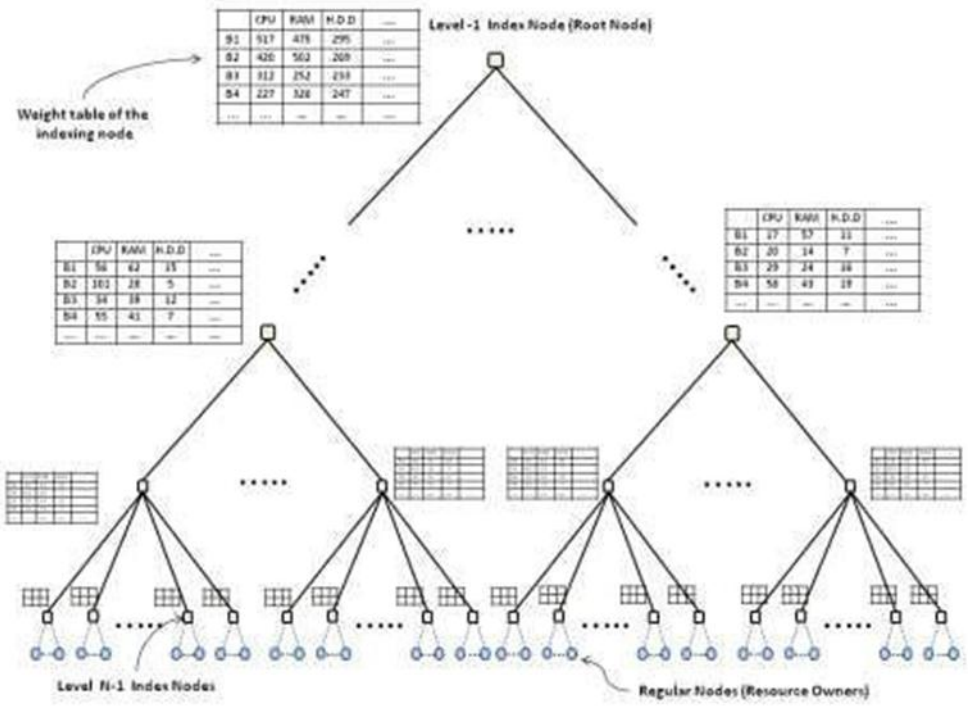


Fig. 1. The Indexing Schema of Weighted-HR Resource Discovery Technique

The columns in the weight table of indexing nodes refer to existing resource types in the Grid environment. The value of each resource type indicates the number of nodes that shares this type of resources in descendant nodes. This value is calculated by sum of the values for that resource type in all descendant indexing nodes of the corresponding branch. It is updated upon join and leave events as well as allocation of the resources.

The procedure of resource discovery initiates by a regular node. The regular node, in the last level of indexing tree, transmits the query to its parent. At first, the parent examines its local resource information index in order to find the requested resource. If the resource found, then, it would return the resource holder address to that particular node. Otherwise,

the parent transmits the query to the parent of indexing node placed at level N-2 of the tree.

Once the parent receives the query from its child, it extracts the requested resource type from the received query. Then it checks its weight table to find the branches consisting of the requested resource type. It forwards the query to the specified percentages of branches with highest number of the requested resource type. After forwarding the query to the branches of tree, if no result found, the query would returned to the parent. This process goes continuously until catching the requested resource or getting to the root indexing node. Once an indexing node finds the requested resource, it forwards the resource holder address to the initial resource requester. If the query reached to the root node and the requested resource could not be found, the root node terminates the search process.

The indexing tree retains resource information in five levels. Also, the number of current regular nodes shows the capacity of indexing nodes on the last level. As it mentioned earlier, the proposed technique uses hierarchical structure, and similar to our proposed method in [27], the indexing nodes require regular nodes which are indexed on their descendant nodes. Here, n shows the total number of Grid nodes, and b express the number of branches at each level.

Also, the level of indexing node is shown by l. If we assume L is the last level of the tree, then the number of indexing nodes in the lower levels of a node is calculated as

$$I_1 = \sum_{i-1}^{L-1} b^i \ .$$

For each experiment, 1000 resource requests are issued. The resources could join or leave the network, and hence they would transfer more messages. . When a required resource found, it would allocated to the requester node. Changing the resource information which is affected by allocation events should also be considered. These update messages are measured by the equation (1), where s express the success rate of queries, and r shows the total number of issued resource requests. Also, k depicts the rate of the multi-attribute query issues which call two resources.

$$n_1 = 2(k+1)*s*r \quad (1)$$

$$n_2 = h_1 * n \quad (2)$$

The join or leave events' messages calculates by equation (2), where $h_1$ is the rate of the join or leave events, and *n is* the number of Grid nodes. Moreover, some messages transmitted for the random changing events. That is applied by the system on resources. Equation (3) reckons these messages, where *R* shows the maximum number of resources in a Grid node, and h2 represent the rate of changing events on the resources.

$$m_c = n1 + n2 + \sum_{i=1}^{R} (h_2 * n * i) \quad (3)$$

With regard to forwarding the received queries to different percentages of the branches, it affects the probability of finding a requested resource in the descendant nodes. The demanded resource can be achieved by the probability of $1/n$ in a node of the last level of the tree. Therefore, in level l, the probability to find a resource can be estimated by the equation (4). In this equation, P represents the percentage of branches which are chosen to forward the queries.

$$P_1 = m_1 * \frac{1}{n} * P \quad (4)$$

Also, the number of resource requests at level l of the tree is calculated by the equation (5), where r is the number of requested resource, and k is the rate of multi-attribute queries.

$$m_r = \sum_{i=L}^{l} (b^{(i-1)} * 2(k+1) * r * (1 - p_i)) \quad (5)$$

$$m = \sum_{i=1}^{L} ((b-1) * \left( \frac{m_c + m_r}{b^{(i-1)}} \right)) \quad (6)$$

Here, $b^{(i-1)}$ represents the number of indexing nodes at each level. Hence, the total transmitted messages is estimated by the equation (6), where b shows the number of branches in the indexing nodes, and $m_r$ represent the messages transferred for resource requests. Also, $m_c$ is the messages transferred for resource change information.

## 4. Analysis of the simulation results

The proposed technique is implemented in Java under the simulation toolkit GridSim version 5.2. The hardware platform includes CPU Intel Core 2 Duo 2.93 GHZ and 4 GB of RAM. The experimental results are executed by diverse number of Grid nodes i.e. 1000, 5000, 10000, and 20000.

Moreover, in order to investigate the impacts of the forwarded branches, three values i.e. 25, 50, and 75 of the forward percentages are considered for the indexing nodes. Also, 1000 resource requests are used in each experiment, and then the average of the message load and response times are computed. In order to compare and evaluate the results, an existing hierarchical [25] technique is simulated with the similar software and hardware platform. This technique is referred as HR in the rest of this paper.

The message load in the performed experiments is measured by dividing the number of messages transmitted in the search process to the time spent for issuing

all the requests. Also, the response time is taken by calculating the average value of all the requests' response time. The response time of each request is the time spent between issuing the request and finding the result. The message load of Weighted-HR for diverse number of Grid nodes, and forwarding percentage values are shown in Table 1.

Table 1
The Message Load of Weighted-HR for Diverse Number of Grid Nodes and Forwarding Percentage Values

| | Number of Grid Nodes | | | |
|---|---|---|---|---|
| | 1000 | 5000 | 10000 | 20000 |
| Pr =25% | 18.1 | 47.23 | 79.3 | 137.5 |
| Pr =50% | 22.7 | 62.31 | 112.1 | 203.7 |
| Pr =75% | 25.33 | 72.9 | 133.2 | 243.9 |
| HR | 27.5 | 80.2 | 148.1 | 274.1 |

As shown in Table 1, the obtained results indicate that Weighted-HR reduces the message load of resource discovery relative to HR. It is because the queries are forwarded to a specified percent of the branches at each step of traversing the tree. The message load reduction rate depends on the percentage of chosen branches to forward the queries. It is expected to achieve higher rates of the message load drops for lower percentage values of the forwarded branches. The lessening rates of the message load in Weighted-HR for diverse number of Grid nodes and forwarding percentages are calculated and shown in Table 2.

As mentioned above, the message load reduction rates depend on the percentage of branches to forward the received query such that choosing higher values of the forwarding percentages causes Weighted-HR to act more similar to the pure hierarchical techniques. As it is expected, the results of Table 2 indicate that choosing higher percent of branches to forward the queries could decrease the message load of the resource discovery with the lower rates. Also, the reduction rates of message load are not equal to the excluded forwarding branches since the weight tables are used only to choose the forwarding branches for received queries, and the update messages are transferred regardless to weight tables' information.

Moreover, the message load decreases with the higher rates, when there are large number of nodes in Grid environment. The weight tables' information updates during the join or leave events of the Grid nodes. Therefore, more number of Grid nodes' join or leave events enriches the information of the weight tables. Afterward, it increases the message load reduction rates, because the information of the weight tables will be more useful and the requested resources found with less number of messages.

Table 2
The Message Load Reduction Rates of Weighted-HR for Diverse Forwarding Percentages

| | Number of Grid Nodes | | | | Overall Reduction Rate |
|---|---|---|---|---|---|
| | 1000 | 5000 | 10000 | 20000 | |
| Pr =25% | 34.1% | 41.1% | 46% | 49.7% | 42.7% |
| Pr =50% | 17.4% | 22.3% | 24.2% | 25.6% | 22.3% |
| Pr =75% | 7.8% | 9.1% | 10% | 11% | 9.4% |

The response time of the resource discovery is also affected by the weight-based forwarding of the queries however it does not depend to the percent of chosen branches. It is because the response time is computed by the time difference between issuing the request and receiving the response. Thus, it is not dependant to the number of forwarding branches.

Table 3
The Average Response Time of HR and Weighted-HR for Diverse Number of Grid Nodes

| | Number of Grid Nodes | | | | Overall Reduction Rate |
|---|---|---|---|---|---|
| | 1000 | 5000 | 10000 | 20000 | |
| Weighted-HR | 0.29 | 0.41 | 0.48 | 0.54 | |
| HR | 0.31 | 0.45 | 0.53 | 0.60 | |
| Reduction Rate | 6.4% | 8.8% | 9.4% | 10% | 8.65% |

The average response time of HR and Weighted-HR for diverse number of Grid nodes is shown in Table 3. The table specifies when the response time decreases more, the number of Grid nodes increases. This is because the larger number of Grid nodes can enrich the weight tables' information and it subsequently increases the probability of discovering the requested resources with less delays.

In order to choose the appropriate configuration of Weighted-HR and evaluate its improvements related to other techniques, its success rate is investigated. The success rate is calculated by dividing the number of successful queries to the number of all issued queries.

Table 4 presents the success rate of Weighted-HR for diverse number of Grid nodes and percentages of forwarding branches.

Table 4
The Success Rate of Weighted-HR for Diverse Number of Grid Nodes and Query Forwarding Percentages

| | Number of Grid Nodes | | | | Overall success rate | Reduction rate |
|---|---|---|---|---|---|---|
| | 1000 | 5000 | 10000 | 20000 | | |
| Pr=25% | 38.6% | 41.5% | 43% | 44.2% | 41.8% | 50.2% |
| Pr=50% | 65.1% | 68.7% | 69% | 69.4% | 68.05% | 23.95% |
| Pr=75% | 85% | 86.4% | 87.9% | 88.6% | 86.9% | 5.1% |

The success rate of Weighted-HR is less than HR in all cases because it excludes some branches from investigation during the search process. With regard to the results presented in Table 2 and
Table 4, and success rate of HR which is 92%, the success rate is reduced more than the message load when the queries are forwarded to less than 75% of branches. For instance, in the case of choosing 50% for the forwarding branches, the message load is reduced 22.3% while the success rate is reduced 23.95%. But when the query forwarding percentage is chosen as 75%, the message load is reduced 9.4% while the success rate is reduced 5.1%. It is because the branches are chosen as ordered based on the number of accessible resource types.

When the query is forwarded to 75% of the branches, it means that 25% of the branches with lowest number of resource types are excluded from the search domain. Thus, it is expected that the excluded branches do not impact significantly on success rate of the searching process. In connection with the obtained results, forwarding the queries to 75% of the branches eventuates to higher rates of the message load decrease relative to the success rates.

As a result, reducing the branches to forward the queries in the indexing nodes of hierarchical structure can decrease the message load and response time of the resource discovery when the branches are chosen as ordered in terms of the number of accessible resource types. Also, the rate of chosen branches to forward the queries

affects the success rate of the resource discovery. The results of Weighted-HR showed that choosing more than 75% of the branches to forward the queries reduces the message load and response time with the higher rates relative to the success rate reductions.

## 5. Conclusion

The resource discovery service which is responsible to discover the resources required by Grid applications has an important role in Grid computing systems. This paper proposed a weight-based technique in order to improve the hierarchical and super-peer-based Grid resource discovery solutions in message load and response time. In the proposed technique, each indexing node keeps a weight table consisting of its branches and the number of different resource types that are accessible through each branch. Such information is collected during the resource's join/update processes in hierarchical structure.

However this technique is able to decrease the message load and response time of the searching process, but the success rates are reduced. It is because this technique excludes a specified percent of branches form investigation during the search process. The experimental results of the proposed technique indicated that it reduces the message load and response time during the searching process. With regard to the contents of the weight

tables, choosing higher percentage of branches to forward the queries causes the system to act similar to conventional hierarchical systems. On the other hand, choosing very low percent of branches excludes many resource owners from the search domain. The results showed that choosing more than 75% of branches could result to appropriate efficiency of the message load and response time while keeping the success rates high.

## References


[1] T. Ma, S. Shi, H. Cao, W. Tian, and J. Wang, "Review on Grid Resource Discovery: Models and Strategies," *IETE Technical Review*, vol. 29, no. 3, pp. 213-222, 2012.

[2] P. Suri and S. Rani, "Resource Management in Grid Computing: A Review," *Global Journal of Computer Science and Technology*, vol. 13, no. 17-E, pp. 7, 2013.

[3] C. Comito, D. Talia, and P. Trunfio, "Grid services: principles, implementations and use," International Journal of Web and Grid Services, vol. 1, no. 1, pp. 48-68, 2005.

[4] M. R. Islam, M. T. Hasan, and G. Ashaduzzaman, "An architecture and a dynamic scheduling algorithm of grid for providing security for real-time data-intensive applications," International Journal of Network Management, vol. 21, no. 5, pp. 402-413, 2011.

[5] M. Hauswirth and R. Schmidt, "An Overlay Network for Resource Discovery in Grids," in *Database and Expert Systems Applications, 2005. Sixteenth International Workshop on*, 2005, pp. 343-348.

[6] M. Mollamotalebi, R. Maghami, and A. S. Ismail, "Resource Discovery Approaches for Grid Environments," *International Journal of Networks and Communications*, vol. 3, no. 2, pp. 53- 61, 2013.

[7] Hameurlain, D. Cokuslu, and K. Erciyes, "Resource discovery in grid systems: a survey," Int. J. Metadata Semant. Ontologies, vol. 5, no. 3, pp. 251-263, 2010.

[8] V. Reinhard and J. Tomasik, "A centralised control mechanism for network resource allocation in grid applications," *International Journal of Web and Grid Services*, vol. 4, no. 4, pp. 461-475, 2008.

[9] P. Trunfio *et al.*, "Peer-to-peer models for resource discovery on grids," in *Proceeding of the 2nd CoreGRID Workshop on Grid and Peer to Peer Systems Architecture*, 2006.

[10] D. T. P. Trunfio, P. Fragopoulou, H. Papadakis, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, S. Haridi, "Peer-to-Peer Models for Resource Discovery on Grids," Institute on System Architecture, CoreGRID - Network of ExcellenceMarch, 2006.

[11] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like p2p systems scalable," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 407-418: ACM.

[12] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer- to-peer networks," presented at the Proceedings of the 16th international conference on Supercomputing, New York, New York, USA, 2002.

[13] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content- addressable network," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 161-172, 2001.

[14] R. Stoica, Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," Sigcomm Comput. Commun. Rev., vol. 31, no. 4, pp. 149-160, 2001.

[15] D. Cokuslu, A. Hameurlain, and K. Erciyes, "Grid resource discovery based on centralized and hierarchical architectures," *International journal for Infonomics*, vol. 3, no. 1, pp. 227- 233, 2010.

[16] M. MollaMotalebi, A. S. B. H. Ismail, and A. A. Ahmed, "A New Model for Resource Discovery in Grid Environment," in *International Conference*

*on Informatics Engineering and Information Science*, 2011, pp. 72-81: Springer.

[17] B. Yang and H. Garcia-Molina, "Improving search in peer-to-peer networks," in *Distributed Computing Systems, 2002. 22nd International Conference on*, 2002, pp. 5-14.

[18] D. Puppin, S. Moncelli, R. Baraglia, N. Tonellotto, and F. Silvestri, "A grid information service based on peer-to-peer," in *European Conference on Parallel Processing, Conference on*, 2005, pp. 454-464: Springer.

[19] E. Elmroth and J. Tordsson, "An Interoperable, Standards-Based Grid Resource Broker and Job Submission Service," presented at the Proceedings of the First International Conference on e-Science and Grid Computing, 2005.

[20] T. G. Ramos and A. C. M. A. d. Melo, "An Extensible Resource Discovery Mechanism for Grid Computing Environments," presented at the Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, 2006.

[21] E. Huedo, R. S. Montero, and I. M. Llorente, "A recursive architecture for hierarchical grid resource management," *Future Generation Computer Systems*, vol. 25, no. 4, pp. 401-405, 2009.

[22] M. Yan, G. Bin, and Z. Lida, "Resource Discovery Algorithm Based on Small-World Cluster in Hierarchical Grid Computing Environment," in *Grid and Cooperative Computing, 2008. GCC '08. Seventh International Conference on*, 2008, pp. 110-116.

[23] R.-S. Chang and M.-S. Hu, "A resource discovery tree using bitmap for grids," *Future Generation Computer Systems*, vol. 26, no. 1, pp. 29-37, 2010.

[24] S. Ebadi and L. M. Khanli, "A new distributed and hierarchical mechanism for service discovery in a grid environment," *Future Generation Computer Systems*, vol. 27, no. 6, pp. 836-842, 2011.

[25] L. H. Zhongping Zhang, Chao Zhang, "Grid Resource Discovery Algorithm Based on Distance " *JOURNAL OF SOFTWARE*, vol. 9, no. 11, pp. 2966-2973, 2016.

[26] M. Mollamotalebi, R. Maghami, and A. S. Ismail, "A Weight-based Query Forwarding Technique for Super-peer-based Grid Resource Discovery," *Engineering, Technology & Applied Science Research*, vol. 7, no. 1, pp. pp. 1398-1404, 2016.

[27] M. Mollamotalebi, R. Maghami, and A. S. Ismail, "THRD: Threshold-based hierarchical resource discovery for Grid environments," *Computing*, vol. 97, no. 5, pp. 439-458, 2015.