Resource discovery in Grid computing using Fuzzy Logic and Tabu Table

Afsaneh Zargar Nasrollahi and Ali Asghar Pourhaji Kazem*,

Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

Summary
Resource discovery is a very important task in Grid environments. Grid technologies enable us to share the distributed resources. For using these resources what will be needed are the resource management systems. The resource management discovers the resources to obtain some information about the resources and therefore, because of the complex and dynamic nature of the Grid resources, the issue of resource discovery and sharing has been emerged out. In this paper, an approach has been introduced for the resource discovery with implications for searching efficient resources using fuzzy logic and taboo table. The resource discovery starts with the network request to find appropriate resources in the Grid. The discovery process is very vital and critical in the management and allocation of resources. Proposed approach have been simulated in Matlab environment and the comparison of obtained results to other approaches indicates the efficiency of proposed resource discovery.
Key words:
Resource Discovery, Resource Sharing, Efficient Resources, the Fuzzy Logic

1. Introduction

The computational Grid system is a hardware and software infrastructure that provides reliable, stable, universal and cheap access to the facilities of other people. The Grid computing system is linked with a collection of heterogeneous resources such as personal computers, work stations, clusters, etc. in a wide-scale scenario. In recent years, Grid computational systems have emerged out as an alternative in increasing the memory and processing capacity of integrated resources through sharing their resources. Grid computational systems make possible the integration of computers and resources such as software packages, data and peripheral devices through an inter-related network. On a computational Grid system a collection of computational resources appear in the form of a virtual powerful computer system. So the resource discovery in modern distributed systems plays a critical role. The resource discovery allows the system to be aware of what resources are added to the system as well as the situation, features and the capabilities of available resources [1].

Resources can be computers, computer clusters, direct devices, storage space, data, and applications. In a Grid computing network, the existence of several different operating systems and different management areas and the lack of portability among database-dependent approaches make resource discovery even harder. Hence, implementing a suitable resource discovery is an important aspect of Grid environments. The success of a computational Grid network mainly depends on locating suitable resources for a particular task. In fact, Grid computing is a way toward distributed processing which considers not only the geographical position but also the architecture of machines struggling with software constraints. Hereby it provides any connected person to the Grid with an unlimited computational power. A Grid computational system consists of a group of machines and a communicational network among these machines. As mentioned before, each machine can be a personal computer, computational resource, database, etc.

With the development of computational Grid systems and the importance of finding different resources for users, time and space saving will be more critical which in turn highlights the great importance of resource discovery algorithms. Nowadays, with the development of Grid environments and the increased number of resources as well as the geographical distribution of resources implementing a resource discovery algorithm which could provide the required resources of users in a very short time is one of the main tasks in Grid environments [2]. Therefore, in order to achieve this aim in finding the appropriate resources for Grid users, in this paper, a new Grid resource discovery mechanism is proposed. In the proposed mechanism, fuzzy logic (multi-criterion decision making) and taboo table for optimum discovery and scheduling of resources are used. The multi-criterion decision making is the most common modelling method in the decision making issues and it tries to model a decision making problem while the number of goals or decision making attributes are more than one. The multi-criterion decision making is considered as a rule-based structure of preferential relations for evaluating a set of options in the presence of several indicators. The resource discovery acts as a connector between the resource holder and the resource requester and as a support for the scheduling operations. In the rest of this paper, in section 2, previous works for resource discovery in Grid environments are

Manuscript received September 5, 2016
Manuscript revised September 20, 2016
*Corresponding author

introduced. In section 3 the proposed mechanism is outlined and the simulation results are evaluated in section 4. Finally, section 5 concludes the paper.

2. Related Works

In [4] a new algorithm with the help of a weighted tree to discover resources and a bitmap is used for studying some of resources that locate in special positions. In fact, in each node of the tree there will be signs or tracks of existing resources in the generated children of the node. So when the user sends his/her request for each node if these requests exist in the associated children of that node, then direct access to the request-holding nodes will be possible without any reference to additional nodes and creating extra traffic. This algorithm implies that the number of nodes met in resource discovery algorithm is less than those of other algorithms .With the notable increase in the number of nodes the updating cost will reduce as well [4]. The resource discovery algorithm which uses flooding-based and random methods in order to explore the resources in the Grid environment lowers the system performance due to the transmitted user request to unnecessary routes and generated traffic. The tree algorithm of resource discovery [5] drastically eliminates many disadvantages of the previous methods which were related to additional traffic and heavy loads and reduces the updating cost. But still the user's renewed requests pass through unnecessary and create extra traffic which in turn reduces the system performance in a Grid environment with a lot of nodes. In this method a weighted tree structure "Footprint Resource Tree" is introduced. In this way [5] one bitmap with different content named as the "route bitmap "will replace several bitmaps. For updating, a super bitmap "bitmap counter" will be used. In the current method to any type of resources two bitmap states (one of the counter type and the other of footprint type) are dedicated therefore the user's request should change according to this. When sending requests to the nodes, if the target node contains the requested resources it means that the desired node is found, and if it is not within its resources the query will be sent to the parent node. So if the node does not contain the requested resource it is understood that it would be found inside one of the children of this node. The mentioned node can be directly found without reference to additional nodes using the information stored in the nodes and edge weights. So in this algorithm unnecessary referrals to other nodes, creation of additional traffic and increase of extra loads are all prevented. Thereafter the time is saved, the resource discovering ability is improved and updating cost is reduced. In the algorithm [6] resource discovery in dynamic Grid is investigated on the basis of re-routing table. With the resource discovery problem checked in a dynamic Grid model based on a Grid routing model it shows that the Grid is seen as a formed environment by the router and resources in which each router is responsible for its local resources. The problem of discovering resources on the Grid is seen as a problem of finding suitable resources for a particular request in that environment. Efforts for solving the resource discovery problem are currently ongoing .Various mechanisms have been introduced so far. One of these mechanisms is the routing table which can guarantee that appropriate solutions are found for a particular request in a static Grid environment in which the resources are always online and connected to the Grid. In this method the effect of the mechanism of routing table is checked, which in turn can guarantee that the appropriate resources are found in a Grid dynamic environment where resources can be disconnected and therefore become offline. In the paper [7] a resource discovery algorithm in the Grid context uses methods based on flooding or random sampling methods to send the requests of users. In this case a request may pass many unnecessary routes causing additional traffic and reducing system performance. In this method the tree structure is introduced for discovering resources and it shows all resource specifications by using bitmap format. The user's request for finding the appropriate resources is sent to the server (indicator). If the server finds the under-request resources in the node, the request will be sent to the children of that node. If the desired resources do not exist, the search in the tree is carried on until reaching the root. Hence using the tree structure can reduce the unnecessary traffic and improve the efficiency of resource discovery.

3. The proposed algorithm

In multi criteria decision making procedures which have found popularity among researchers in recent decades, instead of using a measure of optimality several measures are used. The Multi Criteria Decision Making (MCDM) models are divided into two major categories: Multi Objective Decision Making (MODM) models and Multi Attribute Decision Making (MADM) models. Generally speaking, the Multi Objective Decision Making models are used for design purposes while the Multi Attribute Decision Making models are used for the selection of superior options. The main difference of the Multi Objective Decision Making models with the Multi Attribute Decision Making models is that the first type is defined in continuous space while the latter one is defined in discrete space respectively. Generally in the mentioned problems, three tasks must be done over the engaged criteria:

- Converting qualitative criteria to quantitative ones
- Normalizing criteria
- Determining the relative weights of criteria

The multi criteria decision making is the most common method in decision making problems and tries to model a decision making problem while the number of decision's objectives or attributes are more than unity. The multi criteria decision making can be seen as a rule-based structure of preferential relations for evaluating a set of options in the presence of several indicators. The aim from MCDM techniques is to design and to help the decision making in finding the most appropriate solution from the decision maker's viewpoint toward the problem. MCDM problems are usually divided into two categories, the Multi Attribute Decision Making (MADM) and the Multi Objective Decision Making (MODM). In the Multi Attribute Decision Making the concern of decision maker is to prioritize and choose from among several available options and generally the number of options is limited and known, and the operation is done in discrete space while multi objective problems are engaged for designing purposes, they have a structure of mathematical programming and their goal is the simultaneous optimization [8]. The architecture used in the proposed method has a hierarchical structure that its details are as follows: The architecture is hierarchical and consists of several clusters so that each cluster has a large number of sites. In this model, there is a central coordinator and local coordinators interact with the central coordinators for allocating data to the requesters. In the case that the requested resource does not exist inside the own cluster, the request is transmitted to the central coordinator. Figure 1 displays the architecture of proposed method with three clusters.

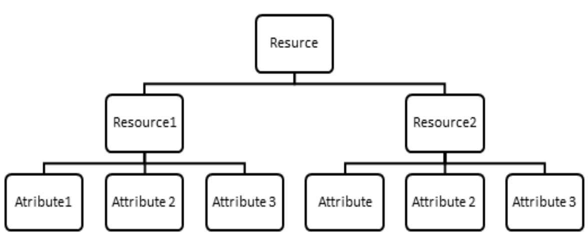

Fig1: The hierarchical architecture of the proposed approach

In the proposed architecture, each cluster consists of many resources and also an efficiency super-factor monitors all machines to assure from the proper operating of them. Of course, for having a comprehensive and overall view from the entire system, the remote surveillance of the required parameters of all operations must be possible. When monitoring is possible, the applied policies should guarantee the system in reaching at its maximum value of efficiency .As each resource has its own features, all these features must be heeded duly. The resource factor acts as a local coordinator and in the hierarchal model these local coordinators interact with a central coordinator (resource requester) for allocating resources. If the requested resource does not exist inside the own cluster, the request is transmitted to the central coordinator and the central coordinator replies to this request.

The multi criteria decision making indicates a resource evaluation process which should be assessed by a few criteria and can be divided into two classes of the Multi Attribute Decision Making and the Multi Objective Decision Making. In this paper according to the terms of decision making, the Multi Attribute Decision Making method was engaged for the selection of widely used data .Its stages are as follows:

- Identification and selection of effective criteria in resources (the resource features should be studied as each resource may have several features).
- Determining the weight of resource features inside the Grid by using the peer comparisons method.
- Ranking the available options and procuring the fitness of each resource and specifying the best resource, and if the best resource is not found, suggesting a resource with adequate weight.

First, based on the studies and research done in the field the determinant criteria for the stage "the identification and selection of effective criteria "are identified and selected and then the weights of the criteria are determined. One of the what-computing methods is the method of Professor Saati in the form of paired comparisons.

This method starts with peer-to-peer comparisons between decision elements (Figure2) and through the allocation of quantified scores which represent the preference or importance between the two decisions (Figure3). After the peer comparisons matrix has been formed, several methods may be applied for calculating of the relative weight and in the proposed method we used the arithmetic mean method from the approximate methods. In this case the procedure is as follows: In the resulted matrix the sum of figures in each column is calculated, and then each column entry is divided by the sum of the figures in that column. The new obtained matrix is called the normalized comparisons matrix. Finally the mean value of each row of the normalized comparisons matrix is calculated. The final resulted matrix is a column matrix which its entries are the compared criterion weights [8].

Fig2: The matrix representation of criteria's preference in Grid resources

	Source1	Source 2	Source3	Source 4
Source1	1	The two priority of service / The first priority of service	Thet hird priority of service / The first priority of service	The four priority of service / The first priority of service
Source2	The first priority of service / The second priority of service	1	Thet hird priority of service / The second priority of service	The four priority of service / Thesecond priority of service
Source3	The first priority of service / The third priority of service	The second priority of service / The third priority of service	1	The four priority of service / The third priority of service
Source4	The first priority of service / The forth priority of service	The second priority of service / The forth priority of service	Thet hird priority of service / The forth priority of service	1

Matrix A:

	Source 1	Source 2	Source3	Source 4
Source1	1	The two priority of service / The first priority of service	Thet hird priority of service / The first priority of service	The four priority of service / The first priority of service
Source 2	The first priority of service / The second priority of service	1	Thet hird priority of service / The second priority of service	The four priority of service / Thesecond priority of service
Source 3	The first priority of service / The third priority of service	The second priority of service / The third priority of service	1	The four priority of service / The third priority of service
Source 4	The first priority of service / The forth priority of service	The second priority of service / The forth priority of service	Thet hird priority of service / The forth priority of service	1
Sum column	S1	S2	S3	S4

Matrix B:

	Source 1	Source 2	Source 3	Source 4	Average row
Source 1	A_{11}/S_1	A_{12}/S_2	A_{13}/S_3	A_{14}/S_4	D_1
Source 2	A_{21}/S_1	A_{22}/S_2	A_{23}/S_3	A_{34}/S_4	D_2
Source 3	A_{31}/S_1	A_{32}/S_1	A_{33}/S_3	A_{34}/S_4	D_3
Source 4	A_{41}/S_1	A_{42}/S_2	A_{43}/S_3	A_{44}/S_4	D_4

	Source 1	Source 2	Source 3	Source 4	Total lines
Source1	$B11/D1$	$B12/D1$	$B13/D1$	$B13/D1$	First characteristic weight
Source 2	$B21/D2$	$B22/D2$	$B23/D2$	$B24/D2$	The second characteristic weight
Source 3	$B31/D3$	$B32/D3$	$B33/D3$	$B34/D3$	The third characteristic weight
Source 4	B_{41}/D_4	B_{42}/D_4	$B43/D4$	$B44/D4$	The fourth characteristic weight

Now the weight matrix is obtained.

Definition	Importance degree
Identical Importance	1
Rather Preferred	3
High Importance	5
High popularity	7
Very High Importance	9
In-between Values	2,4,6,8

Fig3: A scale for the peer-to-peer comparisons (Saaty, 1980)

The last stage of decision making is to rank the resources on the basis of specified features. To determine the ranking, for each option per all its resources, the option value (xij) in multiplied by the criterion weight (WI), and they all are gathered together (Σwjxij). A figure is obtained for each option which is called the decision value. Based on the decision values, the resource ranking is completed. In the proposed method, starting from the highest ranks one-third of the total important resources are star-red .the same number of resources (1/3) are ticked starting from the lowest ranks to be used in next stages and for generating rules. They are added to the prohibited table in order not to be used in renewed resource requests and to prevent from long delay. For achieving this goal they are added to the Taboo Table. In the first phase starred factors are the factors which should take part in the negotiation.

Factor weighting in the design phase is calculated based on the peer-to-peer comparisons. The resources weighting in different applications changes resource priorities and causes access to favourable conditions. Assume that the

following table is the result of the designer's paired comparisons based on the Figure 3. To calculate the resources' weight the following stages must be followed:
Adding the entries of each column.
Dividing each matrix entry by the total sum of all entries in the same column (Normalization). Calculating the average of the entries in each row.
Therefore, the factor weights are achieved according to Figure 2.After selecting resources in the central coordinator, the options should be ranked and the resources should be scheduled, after sending information from resources to resource requesters the final data is sorted in an ascending manner. Now the most appropriate source must be selected. Therefore first the obtained values should be normalized. A simple normalization method which often researchers prefer it is the division of all figures of each column by the largest figure in that column. Following this we face with a kind of normalization in which the largest value in each column will be converted into 1. After normalization, according to the normalized values and resource weights the decision value is calculated for each resource. Worth noting that the weight of factors bearing negative impact is used in the complement form of ;that is ;in such a problem like this the data size criterion has a negative impact on the decision .For example instead the weight =0.45, 1-0.45=0.955 (the weight = 955.0) is replaced. The decision making value for each data item s found using relationship (1), for example, for data 1 we have:
D(1)= Min {max (1-0.955,0.46), max (1-0.265,0.311), max (1-0.571,0.658), max(1-0.119,0.474)} = Min {0.46,0.735,0.658,0.881}= 0.46 (1)
Finally, the data are ranked based on the decision values and starting from the highest ranks. One-third of the total data items are marked or starred as the best selection. The same number of lowest ranks is marked too in order to be removed.
The pseudo code of proposed algorithm

- Initialized to the number of tasks (processes)
- create tasks with Exponential
- Initialized process time
- Time service to any business or process

Loop (When the service to be provided to any work completed)
1. The system is set to the Grid or used in Grid computing
2. Create systems with random function
3. The number of Grid smadeor created
4. Initialize the system, the characteristics of each system is initialized.
5. Set the character and needs of each business or process
6. Weighted according to the importance of the process or the importance of the quality required in the process
7. For each standard has different unit renormalized metrics?
8. Get fitness function works so that we can keep them as long as the system or Grid comprised of fitness function system over the process.
9. Calculate the fitness function of process or the resources needed to run each process
10. If the system does or fails to meet the needs of process in this case,the Grid system or the forbidden or tabu is added to the table
11. The work or process which has been allocated to each Grid.
End loop

4. Evaluation and comparison of simulation results

In this section, we compare the results of our simulation with the recommended algorithm [5], the flooding method, the proposed algorithm in [8,9]. In the first simulation experiment 400 requests were tested. In our proposed algorithm, as each system executes a process fitted with its performance and powerful systems are dedicated to very complex processes and any process is run on a proper machine or system, therefore little traffic will appear and the processes not only will wait short time in the queue but also their run time will be proportional. And the experiments were repeated with 300 and 1000 requests as well. As mentioned before the traffic caused by the Fuzzy algorithm is better than those of other techniques.
The simulation was performed at MATLAB environment and the simulation characters are expressed in the following table. Figure 5 shows the number of met nodes in 180 requests. As seen , in the proposed algorithm since all the criteria are of a certain degree of importance and their fitness amount are calculated from Fuzzy computations , so the number of met nodes is less and less this value is , lower the traffic will be in Grid. Also this test has been performed with limited number of servers.

Parameters	value
The number of servers	10-15-20-25-30
Grid size	15-40-85-156-256-400
The number of queries	180-300- 200-1000
The number of nodes visited	0-8000
Server update	15-40-85-156-256-400

Fig 4: The simulation parameters

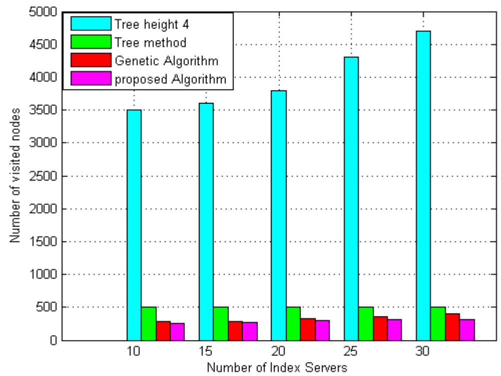

Fig5: The number of met nodes for 180 sent queries

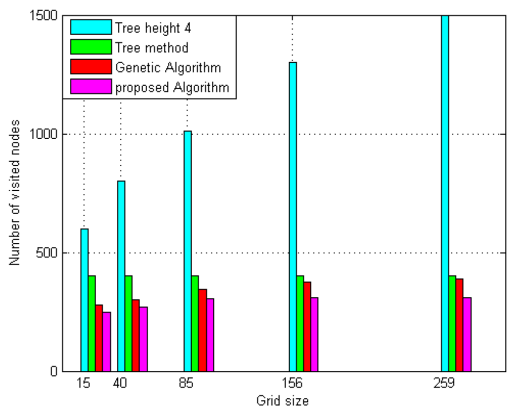

Fig 6: the number of nodes which have sent query

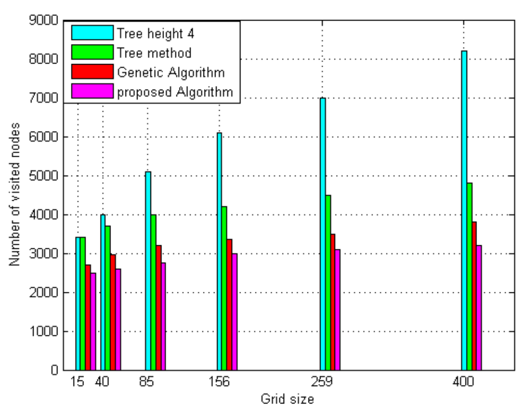

Fig 7: The total number of nodes has been met in resource discovery and updating

In the next experiment, we compared our method with the flooding method, MMO and the resource discovery tree algorithm. In the current test 300 queries are assumed, as shown in Figure 8 .In our proposed algorithm the average number of nodes which send query is less other methods.

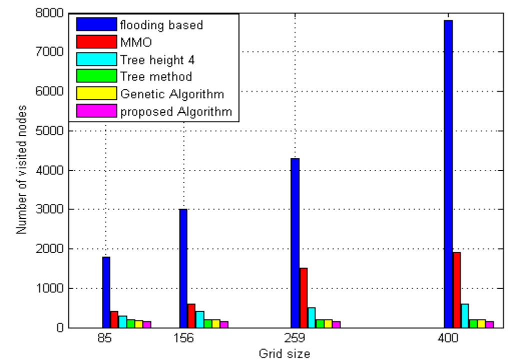

Fig 8: The average number of nodes which send query using different methods

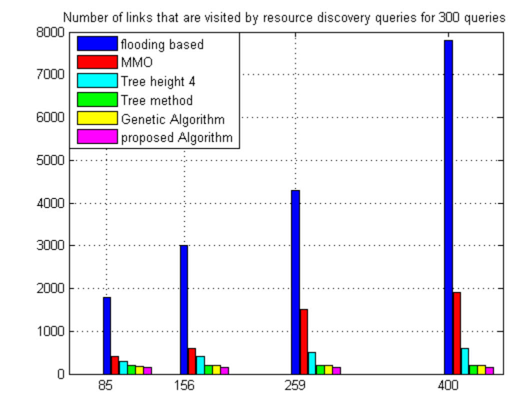

Fig 9: the number of links has been met by resource discovery request for 300 queries

In Figure 10 the number of requests is 300 queries which show that in our proposed algorithm the entire traffic of Grid is less and this in turn increases the system performance. The processes are distributed in a balanced manner. Worth mentioning that proper processes will be allocated to individual Grids to allow them to run their processes with appropriate conditions.

The proposed algorithm was compared with the genetic algorithm. In Figure 10 with increase in the number of servers, we examined the number of met nodes. As shown with increasing the number of servers the number of nodes has been incremented too but in the proposed algorithm according to the fact that all parameters have certain weights and if the process needs be not met, the Grid aggregation is utilized so the number of nodes is not incremented drastically. But in the genetic algorithm as cross-over and mutation operations are used it may be trapped in local optimality and may not satisfy the required fitness level. Therefore it has lower efficiency than the proposed algorithm.

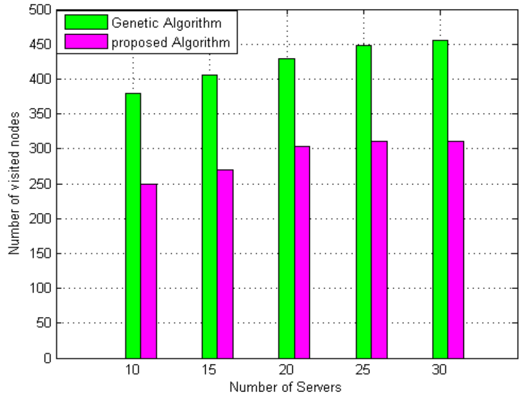

Fig 10: The comparison of the proposed algorithm with the genetic algorithm with increase in the number of servers

Figure 11 shows the average waiting time in the queue and it is the mean time duration each process waits in the queue to get service. As in our algorithm it endeavors to assign a system to a process which has a close fitness to the required process fitness so the waiting time in queue has been decreased in quantity.

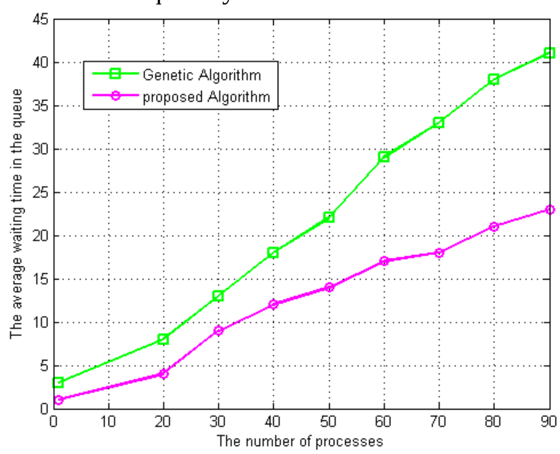

Fig 11: The comparison of the proposed algorithm with the genetic algorithm regarding the average waiting time in queue

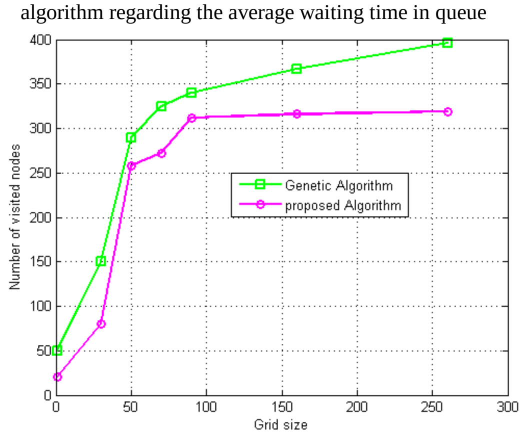

Fig 12: The comparison of the proposed algorithm with the genetic algorithm regarding the number of met nodes in the query

5. Conclusions

In tree methods all nodes should be met but surface scanning is time-consuming. In our method, a Fuzzy Logic is introduced for discovering resources. Peer to peer comparisons turns out that what state is the best one and it prioritizes the parameters i.e. which feature is more important and it per se selects the best state which is closer to our request. In our algorithm we find the good resource simply by adding some figures without any need for a long scan. Our algorithm is better because the weighting and attributes are important, it is of Fuzzy type, it has detection power and it is intelligent. In our algorithm the number of integrated nodes with together is almost identical, because we have made a system Grid which matches our requests and criteria. The simulations showed that the number of nodes and links met in resource discovery algorithm after the resource discovery is much less than other algorithms and it is anticipated that in the near future this algorithm will be evaluated and developed to cover the security edge of Grids or to mix our algorithm with the imperialistic algorithm or to combine the genetic algorithm with the Fuzzy algorithm which can be implemented on very modern servers and albeit with much higher techniques.

References

[1] I. Foster, C. Kesselman and S. Tuecke, “The anatomy of the Grid: Enabling scalable

[2] Ian Foster, Carl Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers Inc., San Francisco, CA, (2003).

[3] Anju Sharma, Seema Bawa Computer Science & Engineering Department Thapar University, Patiala"Comparative Analysis of Resource Discovery Approaches in Grid Computing"JOURNAL OF COMPUTERS, VOL. 3, NO. 5, MAY (2008).

[4] Leyli Mohammad Khanli a, Saeed Kargar b" FRDT: Footprint Resource Discovery Tree for Grids", Future Generation Computer Systems 27 (2011) 148–156

[5] R.-S. Chang, M.-S. Hu, A resource discovery tree using bitmap for Grids, Future Gener. Comput. Syst. 26 (2011) 29–37.

[6] Konstantinos I. Karaoglanoglou, Helen D. Karatza"Resource Discovery in a dynamical Grid based on Re-routing Tables"Simulation Modelling Practice and Theory 16 (2008) 704–720.

[7] Chang, R.-S. & Hu, M.-S. (2010). A resource discovery tree using bitmap for Grids, Future Gener. Comput. Syst. 26 29–37

[8] Leyli Mohammad Khanli1,Saeed Kargar2 and Ali Kazemi Niari2" A New Approach to Resource Discovery in Grid Computing" Published in print edition May, 2012.

[9] Marzolla CA.M, M. Mordacchini, S. Orlando, Peer-to-peer systems for discovering resources in a dynamic Grid, Parallel Comput. 33 (4–5) (2007) 339–358.

Afsaneh Zargar Nasrollahi received the B.S. and M.S. degrees in Computer Engineering from Islamic Azad University, Tabriz Branch in 2013 and 2015, respectively. Her research interests include Distributed Systems and Grid Computing.

Ali Asghar Pourhaji Kazem was born in 1974 in Marand, Iran. He received his B.S. in computer engineering, software engineering from University of Isfahan, Isfahan, Iran, in 1996; the M.S. in computer engineering, software engineering from Shahid Beheshti University, Tehran, Iran, in 2003 and the Ph.D. degree in computer engineering, software engineering from Islamic Azad University, Science and Research branch, Tehran, Iran, in 2013. He is currently an assistant professor in the Department of Computer Engineering at Islamic Azad University, Tabriz branch, Tabriz, Iran. He has published more than 40 papers in different journals and conference proceedings. His research interests include Distributed Systems, Grid Computing, Cloud Computing, Social Networks, Computational Intelligence, Evolutionary Computing, and Database Systems.