World Applied Sciences Journal 19 (1): 153-158, 2012
ISSN 1818-4952;

DOI: 10.5829/idosi.wasj.2012.19.01.1597

New Method for Grid Computing Resource Discovery with Dynamic Structure of Peer-To-Peer Model Based on Learning Automata

[1]Ali Sarhadi, [2]Ali Yousefi and [3]Ali Broumandnia

[1]Department of Computer Engineering, Malayer Branch, Islamic Azad University, Malayer, Iran
[2]Department of Computer Engineering, Hamedan Branch, Islamic Azad University, Hamedan, Iran
[3]Department of Computer Engineering, Tehran South Branch, Islamic Azad University, Tehran, Iran

Abstract: The term "Grid" has become common parlance among parallel and distributed computer scientists to denote a middleware infrastructure for wide-area scientific and engineering computing. Information services are a vital part of any Grid software infrastructure, providing fundamental mechanisms for discovery and monitoring and thus for planning and adapting application behavior. Grid Information Service (GIS) stores information about the resources of a distributed computing environment and answer questions about it. So GIS has a key role in other grid computing services such as resource discovery service. Various kinds of solutions to organization of GIS have been suggested, in this paper we proposed dynamic structure of GIS based on peer-to-peer model using learning automata. Using learning automata is to increase efficiency of proposed model and generate dynamic structuring of peer-to-peer organization for GIS.

Key words: Grid computing · Resource discovery · peer-to-peer · Learning automata · GIS

INTRODUCTION

Grid computing is distributed over networks whose resources are managed, used, owned by several organizations and is dynamic in the sense that participants can join or leave the system at any time [1]. These systems facilitate uniform access to a large number of services and heterogeneous resources such as workstations, networks, storages and computing power that belong to several organizations and administrative domains [2].

Modern Grids are based on the service-oriented paradigm; for example, in the Globus Toolkit 4 based on the Web Services Resource Framework resources are offered through the invocation of Web services, which boast enriched functionalities such as lifecycle and state management. Grid computing has several services with grid resource discovery among the most important ones [3].

Resource discovery activity involves searching for the appropriate resource types that match the user's application requirements. Efficient Resource discovery mechanism is one of the fundamental requirements for Grid computing systems, as it aids in resource management and scheduling of applications. Resource discovery activity involves searching for the appropriate resource types that match the user's application requirements. Grid resource discovery and dissemination complement each other. The discovery is done by the application to find a suitable resource where as dissemination is initiated by the resource which needs to be discovered. The taxonomy of resource discovery and dissemination is shown in Fig. 1 [4].

Currently, there are two types of grid resource discovery systems, which can be classified into centralized and hierarchical systems. In centralized resource discovery systems, the information on resources (metadata) is indexed under a centralized node and users send their resource queries to that node. The resource providers update their resource status at periodic intervals using resource update messages. Condor system is an example of the centralized systems. In Condor model, the centralized node is Called Central Manager (CM), which collects information about the state of resources from resource providers. The resource providers are represented by Resource-owner Agent (RA), which is located in each resource provider. The CM then, receives users' tasks and matches them with the resources. In hierarchical systems however, the information on resources is indexed under a set of nodes

Corresponding Author: Ali Sarhadi, Department of Computer Engineering, Malayer Branch, Islamic Azad University, Malayer, Iran.

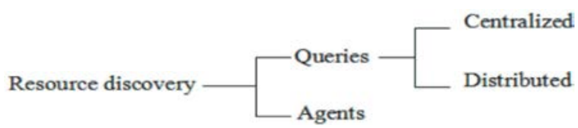

Fig. 1: Taxonomy of resource discovery

in a hierarchical manner (each chilled node indexes its metadata on its parent node). The Monitoring and Discovery Service (MDS) of Globus implements this model. It uses two services: a configurable information provider called Grid Resource Information Service (GRIS) and a configurable aggregate directory service called Grid Index Information Service (GIIS). A GRIS answers queries about the resources of a particular node. A GIIS combines the information provided by a set of GRIS services managed by a given Virtual Organization (VO). The Grid Information Service (GIS) is the infrastructure component responsible for collecting and distributing information about the Grid. It offers some tools to register resources, to query the data base, to remove lost nodes. The first implementations of a GIS used techniques based on *directories*, which are still used by Globus MDS-GT2 (LDAP). Directory-based systems suffer from a series of problems, including the fact that updated information does not propagate very quickly and that centralized servers may become bottle-necks or points of failure. GIS is a key component of resource discovery service, so the best organization of it can help to improve the efficiency of resource discovery service.

In recent years, there has been much research in peer-to-peer (p2p) [5] systems, in which the nodes of a system act as peers that create an overlay network to exchange information. Peer-to-peer systems have several desirable properties, including high scalability and reliability and the capability of self-healing the peer-to-peer overlay when nodes fail or join the network. Several researchers have observed that peer-to-peer and Grid paradigms have several goals in common and some peer-to-peer Grid information service have been developed. We proposed a new organization of peer-to-peer grid information service using learning automata have been proffered.

Resource Discovery: The basic service in Grid Computing is Resource Discovery. In large scale Grid environments, resource discovery is made challenging by potentially large number of resources and considerable heterogeneity in resource types and resource requests. Resource discovery can be defined as a directed to the spontaneous network's environment. Many resource discovery mechanisms have been proposed in the literature of Grid environments. Some of them, such as Metacomputing Directory Service (MDS2) and Data Grid Resource Discovery, are specific resource discovery services. The resource discovery mechanisms proposed for grids are either hierarchical or centralized. Most of the mechanisms retrieve data related to the machines that compose the grid (operating system used, CPU load and memory occupation, among others). Recently a lot of attempts have been made to use grid information service for resource discovery service. In this paper, we have tried to propose a new organization of traditional peer-to-peer for uses in grid resource discovery based on learning automata.

Grid Information Service: Information services traditionally advertise information about resources available on the Grid (machines and their capabilities, job submission mechanisms, services etc.). Owners of the resources or directly the resources usually publish such information about the resources into information service. Status monitoring can augment traditional information services by taking information published in them, checking validity and then publishing results in form of verified and thus authoritative information about resources using interface common to information services. Another problem with the Grid information services is that the most common information service-Globus MDS-also shows serious performance bottleneck when too many non-homogeneous information providers are subscribed to it (esp. with high percentage of "problematic" ones, like information providers behind firewalls or incompatible versions of software clients).we proposed a new organization of grid resource information service with peer-to-peer model has tried. Static nature of existing peer-to-peer model is very important problem. We try to use learning automata for restructuring of peer-to-peer organization at any time. Finding peers in proposed organization is based on learning automata.

Proposed Basic Peer to Peer Structure: A flat structure for organization of grid information service based on peer-to-peer model is proposed. All grid resource information is registered in grid information service. The Proposed structure is an efficient and extendable environment. It does create a dynamic network of directories. Organization of grid information service is based on peer-to-peer model. Peer to peer model for resource discovery is fully distributed. In this way information's nodes that participate in resource search are

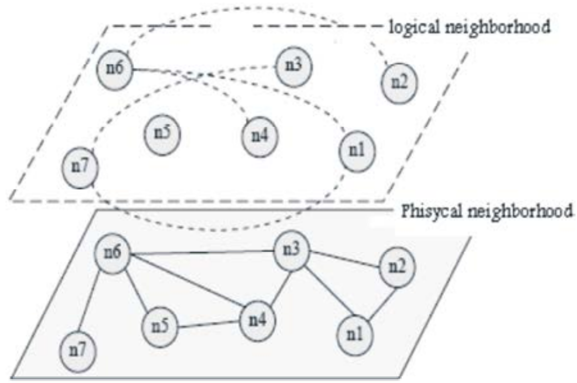

Fig. 2: Primary organization of information nodes based on peer-to-peer model

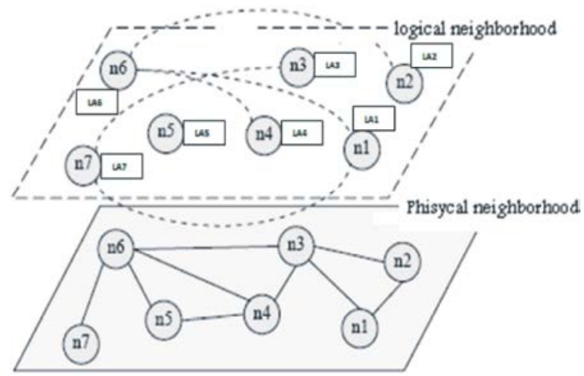

Fig. 3: Extended organization of information nodes based on peer-to-peer model using learning automata

equally in terms of importance. That means each information node in grid information service can process any resource discovery request and perform the discovery of resources. This structure is illustrated in Fig. 2. Grid from resource searching perspective is a set of information nodes that are geographically distributed. Each information node can be attached to other information nodes through contacting with them. In peer to peer search method each information node knows limited set of information nodes that associated with them. These set known as neighborhood of this node. These neighborhoods are logical neighbors' not physical ones. Note to Fig. 2 Users send their request to known nodes. If any information about the location of required resource in user request exists in local node, this nodes return information to user, otherwise request sends to one of the neighbors. This operation repeated until appropriate resource was found or TTL is valid. To limit the spread of requests through the network, each message header contains a time-to-live (TTL) field. This field indicates number of permitted hops in network. At each hop, the value of this field is decremented and when it reaches zero, the request is dropped. A Time-to-Live (TTL) value is used to manage the number of hop permitted in any resource discovery request.

New Organization of Basic Peer to Peer Structure Based on Learning Automata: We have tried to increase efficiency of proposed architecture by using of learning automata [6]. Proffered model of GIS with peer-to-peer structure in pervious section has several weaknesses. static organization of this model is One of the most important of them that means logical neighbors do not change at any time so it cause decrease efficiency of grid resource discovery process. Effective resource discovery service technique needs to have access to up to date information about grid resources status. Dynamic nature of grid environment causes grid resource information service to have a dynamic structure so that resource discovery performance is acceptable. As shown in Fig. 3 at proposed approach, we have equipped each grid information nodes or peers with learning automata. The role of learning automata is to select the best nodes for logical neighbors of this node according to received request [6]. Received request will be sent randomly to a number of nodes. The next choice of learning automata is to transfer the same request in future is suitability of selected node that contains this resource in terms of execution time of incoming job. In other words, peer-to-peer structure is changing at any time based on learning automata criteria. When a new request arrives, if the request is made before, based on previous experience of learning automata for the same request it will transfer to the best nodes of grid resource information nodes in peer-to-peer model but if the arrived request for resource is new, it is sent randomly to nodes and as mentioned earlier results are evaluated by learning automata and new structure of information nodes neighbors are formed.

For each incoming resource request

Begin
A. Select-randomly-information node ()
ψ *(i)* = Select-information node for logical neighbors for sending incoming task according to

Selecting policy
Restructured of peer-to-peer organization based on ψ *(i)*
Evaluate β according to selected mapping policy
A. Update (β)

When every incoming job is executed on the machine with the shortest execution time that registered in one of the information nodes or peers, it is defined as the ideal discovering which is represented as $\psi_{min}(i)$, $1\leq j \leq \tau$ and evaluated as shown in equation 1. For an arbitrary task s_i and an arbitrary machine m_j, $ETC(s_i,m_j)$ is the estimated execution time of s_i on m_j. Proposed method uses $\psi_{min}(i)$ criteria to select logical neighbors of this information node.

$$\psi_{\min}(i) = j \text{ such that } ETC(i,j) = \min_{1\leq q\leq \mu} ETC(i,q) \quad (1)$$

Using the following formulas, we can update action probabilities to design a general linear schema. Let action *i* be performed then:

Desired response:

$$p_i(n+1) = p_i(n) + a[1-p_i(n)] \quad (2)$$

$$p_j(n+1) = p(1-a)p_j(n);\ \forall j;\ j \neq i \quad (3)$$

Undesired response:

$$p_i(n+1) = (1-b)p_i(n) \quad (4)$$

$$p_j(n+1) = \frac{b}{1-r} = (1-b)p_j(n); \forall j; j \neq i \quad (5)$$

where *a* and *b* are reward and penalty parameters. When *a=b*, the automaton is called L_{RP}. If *b=0* the automaton is called L_{RI} and if *0<b<<a<1*, the automaton is called $L_{R\epsilon P}$. For more Information about learning automata the reader may refer to [6]. The proposed structure based on learning automata model, as is constructed by associating every information nodes n_i with a variable structure learning automaton, is represented by a 3-tuple ($\alpha(i)$, $\beta(i)$, $A(i)$). Each action of an automaton is associated with selecting the best information node for logical neighbors of this node. Therefore, for any incoming request for resource $\alpha(i) = n_1, n_2,...,n_\mu$ (n_i is the i^{th} information node) that is a set of information node which will be selected for logical neighbor and sending incoming request to it. And $\beta(i) \in [0,1]$, where $\beta(i)$ *closer* to 0 indicates that the action taken by the automaton of information node is favorable to the system and *closer* to 1 indicates an unfavorable response. Based on heuristic used for evaluating environment responses ($\beta(i)$'s), environment may be interpreted as a P-model, Q-model, or S-model.

Performance Evaluation: This section shows the behavior of our LA-based organization of peer-to-peer model for grid resource discovery service. The simulator used for the simulation is gridsim [7]. In these experiments, we check the behavior of proposed model by comparing it with conventional existing basic structure. To evaluate and compare proposed learning automata based on peer-to-peer model with its basic traditional peer-to-peer model a simulation environment known as Gridsim toolkit has been used. There are several grid simulators that allow evaluating a new grid scheduling algorithm, such as Bricks, MicroGrid and SimGrid. But Gridsim has some advantages which are listed below:

- It allows modeling of heterogeneous types of resources.
- Resource capability can be defined in the form of MIPS (Million Instructions per Second) as per SPEC (Standard Performance Evaluation Corporation) benchmark.
- There is no limit on the number of application jobs that can be submitted to a resource.
- It supports simulation of both static and dynamic schedulers.

Experimental Results: By using Gridsim toolkit, the proposed resource discovery mechanism using learning automata based on peer-to-peer structure with the basic peer-to-peer structure, is simulated.

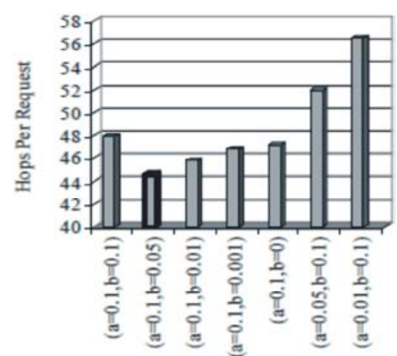

Fig. 4: Hops per request comparison with different a and b in S-model

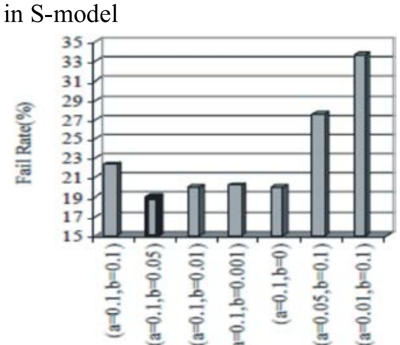

Fig. 5: Fail rate comparison with different a and b in S-model

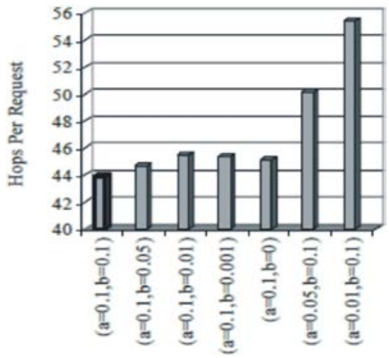

Fig. 6: Hops per request comparison with different a and b in P-model

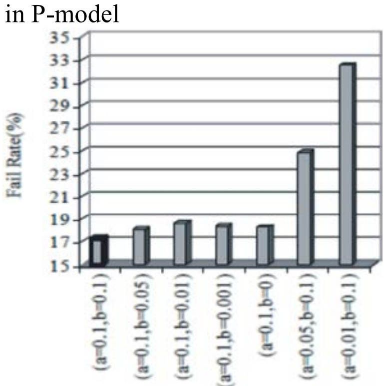

Fig. 7: Fail rate comparison with different a and b in P-model

In our first simulation, we studied the effectiveness of a and b parameters in S-model and P-model of learning automata. We compared different number of a and b to find the best value of them in terms of number of hops and fail rate parameters. We considered Heterogeneous environment to obtain the best result. In this experiments grid environment consist of 3000 nodes. Fig. 4 shows the number of hops and Fig. 5. Shows fail rate comparison and in S-model. Propose of these experiment is to select the best value of a and b. As Fig. 4 and Fig. 5 show, in S-model and heterogeneous environment a=0.1 and b= 0.05 produce a better result. Fig. 6 and Fig. 7 shows that in p- model and heterogeneous environment a= 0.1 and b=0.1 have better results.

RESULTS AND DISCUSSION

Based on above figures, we evaluate the performance of learning automata based on peer-to-peer structure for resource discovery service over a real grid environment. Proffered approach will be compared with conventional existing methods in terms of *success rate* that refer to the ratio of successful queries over the total number of queries generated. Another parameter for comparison is

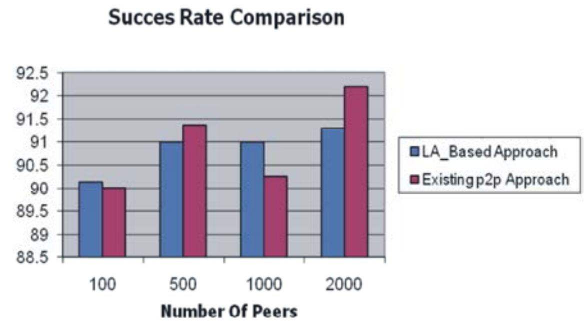

Fig. 8: Succes rate comparison between Proffered method and basic organization

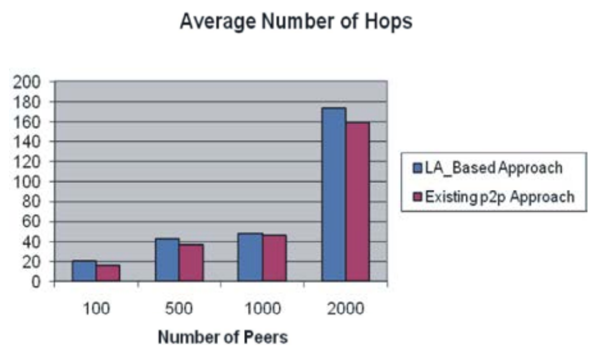

Fig. 9: Average number of hops comparison between Proffered method and basic organization

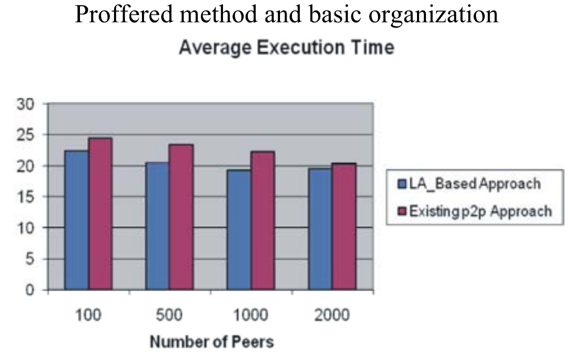

Fig. 10: Average execution time comparison between Proffered method and basic organization

Average number of hops per request that is used to specify the number of hops to discover the best resource for incoming query. And finally *Average execution time* parameter is selected for comparison of proffered structured and conventional existing method. As Fig. 8 shows proffered method and existing method have almost the same performance in success rate parameter comparison. Based on Fig. 9 in average execution time comparison, proffered approach has better performance than existing structure. And finally Fig. 10 shows increase in the number of hops in proffered method in comparison with exiting method that indicate obsession in selecting the best resource in proffered method.

CONCLUSION

In this investigation, the use of learning automata to generate dynamic grid information service has been presented. We have proposed learning automata based model for the new organization of grid information service in peer-to-peer model. Learning automata based peer-to-peer approach has been used in order to increase the efficiency of resource discovery service in grid computing. It has tow fundamental points. First, highly dynamic information system is needed. Second, this organization can be created using learning automata.

Finally, proffered approach has been compared with basic model of peer-to-peer in terms of success rate, average number of hops and execution time and results were presented in the pervious section. Results show the improvement in proffered resource discovery mechanism in comparison with existing method.

REFERENCES

1. Foster, I., C. Kesselman and S. Tuecke, 2008. The anatomy of the Grid: Enabling scalable virtual organizations, International Journal of Supercomputer Applications.
2. Ekmecic, I., I. Tartalja and V. Milutinovic, 1995. A taxonomy of heterogeneous computing, IEEE Computer, 28(12): 68-70.
3. Casavant, T.L. and J.G. Kuhl, 1998. A taxonomy of scheduling in general-purpose distributed computing systems, IEEE Transactions on Software Engineering, 14(2): 141-154.
4. Iamnitchi and I. Foster, 2001. On Fully Decentralized Resource Discovery in Grid Environments, IEEE International Workshop on Grid Computing, Denver, CO.
5. Domenico Talia, Paolo Trunfio and Jingdi Zeng, 2004. Peer-to-Peer Models for Resource Discovery in Large-Scale Grids: A Scalable Architecture.
6. Mirchandaney, R. and J.A. Stankovic, 1986. Using stochastic learning automata for Buyer scheduling in distributed processing systems, Journal of Parallel and Distributed Computing, pp: 527-551.
7. Buyya, R. and M. Murshed, 2002. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, Technical Report, Monash University, Nov. 2001. To appear in the Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, pp: 1-32.