IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012
ISSN (Online): 1694-0814
www.IJCSI.org

A New Bidding Based Algorithm for Resource Selection in Grid Environment

Esmaeil Pourazarm-Hajiaga[1], Saeed Parsa[2]

[1] **Department of Computer, Tabriz Branch, Islamic Azad University, Tabriz, Iran**

[2] **Department of Computer, University of Iran Science and Technology (IUST), Tehran, Iran**

Abstract

A Grid system is comprised of large sets of heterogeneous and geographically distributed resources that are aggregated as a virtual computing platform for executing large-scale scientific applications. As the number of resources in Grid increases rapidly, selecting appropriate resources for tasks has become a crucial issue. The selection of resources that best fitting tasks in Grid environments is an essential and critical factor to system performance. To avoid single point of failure server overload problems, bidding provides an alternative means of resource selection in distributed systems. This paper proposes a resource selection algorithm in bidding based Grid environment to minimize the total time for task completion time arises when using non-reserved bidding algorithms. The result of Evaluation shows the task average completion time less than task average completion time in non-reserved bidding model.

Keywords: *Grid Computing, Resource Selection, Matchmaking Model, Bidding Model, Resource Reservation.*

1. Introduction

With the rapid growth in the number of PCs and clusters, Grid computing technologies have emerged to facilitate resource sharing and the coordination of problem solving in distributed systems [1, 2, 3]. Such systems consist of large sets of heterogeneous and geographically distributed resources that are aggregated as a virtual computing platform for executing large-scale scientific applications. As the number of resources in Grids increases rapidly, selecting appropriate resources for tasks has become a crucial issue. In essence, Grid resources are heterogeneous and managed independently by different organizations, and resource providers can specify their own access policies for sharing resources and joining/leaving Grids dynamically [3-8].

Grid technology contributes in solving several computational problems such as exploiting underutilized resources by executing an existing task on different machines [9]. Furthermore Grid technology allows implementing parallel CPU capacity by dividing the Application and sending the small parts to a number of parallel CPUs and hence the program finishes in shorter time. Furthermore Grid computing allows access to additional resources and software's. However resource management in Grid environments is a great challenge; this is due to heterogeneity of resources in Grid environments and in addition to that, Grid resources belong to diverse administrative domains and apply different management policies.

The resource allocation process in Grid environments consist of three main phases; the first phase is resource discovery, the most important issues when dealing with resource discovery is how to publish the resource information by providers and how the resources can be discovered by Grid clients. The second phase is resource selection which regarding with the process of selecting the best resources to execute a certain task. The last phase of resource allocation process is resource usage which is concerning with running the task on the selected resources and monitoring the executing [9]. In this paper we are focusing on resource selection phase and its issues.

The reminder of the paper is organized as follows: Section 2 contains a review of the literature on resource selection. In Section 3, review the existing resource selection models. The proposed algorithm of our resource selection mechanism is presented in Section 4. We describe the simulation setup and evaluate the performance of the proposed algorithm in Section 5. Then, in Section 6 we summarize the work and the future scope.

2. Related Work

Resource selection in Grid system is concerning with selection a resource or a set of resources to perform the submitted tasks. There are many mechanisms to select the appropriate resources for specific task. These mechanisms often depend on the user and the application requirements [10]. In [9] to avoid unexpected completion time arises when using non-reserved bidding model, proposes the single reservation mechanism to reserve the best resource for the task as a commitment and hence guarantees the task completion time will be as expected. [11] Proposed a resource selection framework that is intended to identify an optimal resource for a given application by considering the reliability characteristics of available resources.

As the number of resources in Grids increases rapidly, selecting appropriate resources for tasks has become a crucial issue. To avoid single point of failure and server overload problems, bidding provides an alternative means of resource selection in distributed systems. However, under the bidding model, the key challenge of resource selection is that there is no global information system to facilitate optimum decision-making; hence requesters can only obtain partial information revealed by resource providers. To address this problem, [3] proposed a set of resource selection heuristics to minimize the turnaround time in a non-reserved bidding-based Grid environment, while considering the level of information about competing tasks revealed by providers.

[12] Design and implemented a new data mining–based Grid resource broker service for selection resources on Grid environment. The role of this resource broker service is using learning method to find the best nodes according to the requirements of the task and the distributed computing resources on the Grid. Due to Grid computing is enabled by an infrastructure that allows users to locate computing resources and data dynamically during a computation, one of the main challenges in Grid computing is efficient selection of resources to the tasks submitted by users. In order to locate resources dynamically in Grid environment, a Grid application consults a broker or matchmaker agent that uses keywords and ontology's to specify Grid services. Moreover, any successful selection mechanism should be highly distributed and robust to the dynamic property of Grid environment. However, [13] believes that keywords and ontology's cannot be defined or interpreted precisely enough to make matchmaking between agents sufficiently robust in a truly distributed, heterogeneous computing environment. To this end, examines a simple algorithm for distributed resource selection that meets the above requirements. Also to address this problem, [14] studied a minimalist decentralized algorithm for resource selection in a simplified Grid-like environment that meets the above requirements. For this purpose considers a system consisting of large number of heterogeneous learning automata connected to tasks that select best resources for their computational needs. There is no communication between the learning automata: the only information that learning automata's received is the (expected) completion time of a task it submitted to a particular resource and which serves as a reinforcement signal for the learning automata.

3. Resource Selection Models

In recent years many attempts have been made on the resource selection in the Grid environment and the following models have been presented as an approach to select resource.

3.1 Matchmaking Model

Matchmaking model emerged to address several resource management issues in Grid systems; the function of resource matchmaker is registering the status of all resources announced by resource providers and running matching programs (Figure1). The raise in numbers of resources and the regularity of task demands generate overload problems hence the matchmaking methods might lead the matchmaker to performance problems [9, 11].

Furthermore the matchmaker information often are out of dates, this is because the status of resources changes frequently and the matchmaker does not learn about the resources status until the resources advertise their new status to the matchmaker [9].

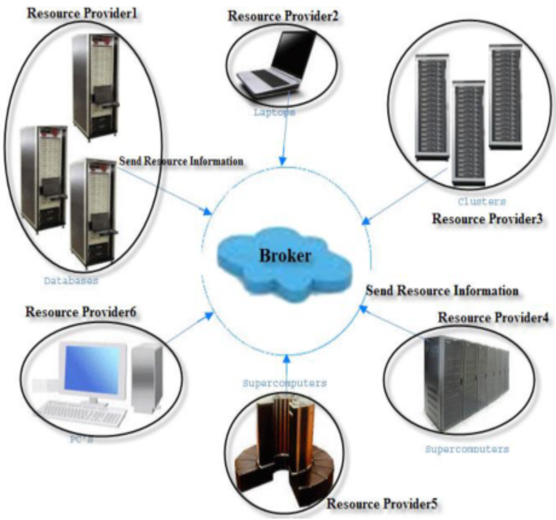

Fig.1 Broker and Resource Providers in Matchmaking Model.

3.2 Bidding Based Resource Selection

Bidding-based model emerged to address the expired information and the performance problems that appear when using matchmaking model [9].

The bidding process starts when a Grid client sends call-for-proposal (CFP) requests to all available resource providers; these requests include the task requirements. According to their characteristics and status of their resources, resource providers determine if they can join the bidding process or not. If a resource provider participates in a bidding process, it sends a bid that expresses the status of its resources to Grid clients. The Grid client evaluates the bids received from providers, orders them and selects the resources that offer the best bid [9, 10]. Based on the commitments of resources for tasks and the utilization of resources we can categories two types of bidding model, reserved bidding model and non-reserved bidding model.

3.2.1 Reserved Bidding Model

In the reserved bidding model when a resource provider decides to join a bidding process, it reserves the resource for that bid to guarantee resource status in the future [9]. However if the Grid client later refuses the bid, that resource is squandered. Under these circumstances, other Grid users possibly are ready to agree that bid prior to the original Grid client refuses it; therefore, the likelihood for resource provider to lose the chance for participation in those bidding processes and support those Grid users is high [9].

3.2.2 None-Reserved Bidding Model

On the contrary to the reserved bidding model, in non reserved bidding model the resource providers do not reserve the resource for each bid. This approach permits the resource providers to completely utilize and employ their resources; however it does not agree a commitment for the status of the resources to any bid. If more than one Grid user sends CFP to the same resources in the same resource provider simultaneously, the resource provider will participate in all bidding processes but will not guarantee the states of resources for each user and hence the task executing time possibly will not be as estimated [9].
The bidding approach permits resource providers to reveal information concerning competing tasks to Grid clients such as the characteristics of the offered resources, the number of competitor and other characteristics of competitors. The amount of information exposed is a significant aspect that influences and affects the bidding process [9, 11].

4. Proposed Algorithm

We propose a resource selection algorithm to address the performance problems raised when implementing traditional reserved and non–reserved bidding models, following is an abstract description for our algorithm.
In the proposed algorithm the bidding process starts when a grid client sends call-for-proposal (CFP) requests to all available resource providers; these requests include the task requirements. Based on its characteristics and the status of its resources, resource provider determines if it can join the bidding process or not. If a resource provider participates in a bidding process, it sends a bid that expresses the status of its resources to the grid client and reserves the resources for that bid.
The evaluation of received bids by the grid client starts when the client receives the first bid. At that point the client considers the first received bid as the best bid. When the client receives the next bid, it compares the current received bid with the best bid; if the new bid is better than the best bid, consider the new bid as best bid. Moreover the client creates a sorted array for the received bids and assigns the old bid to the first element in the array. But if the best bid is better than the recently received bid, the client simply allocate the new bid in the first location in the sorted bid array.
This process is repeated each time the client receives a new bid. When the given time period T finishes; the client submits the task to the provider with the best bid. If the submission of the task fails then the client tries to allocate the task to the first provider in the sorted array of bids, if this also fails then the client tries the next and so on. If all providers in the array fail then the grid client starts a new bidding process.

Example:

It is assumed that resource providers or RPs, participated in the Bidding, are RP1, RP2 and RP3. The grid client sends CFP to all these RPs. At the beginning, the first bid is given to the grid client by RP1 and grid client considers RP1 as the best bid (Best_Bid), means that:

Best_Bid = RP1

During the bidding process, RP2 and RP3 also give their bids in order to participate in this bidding, respectively and it is assumed that RP3 is better than RP1 and RP1 is better than RP2. Therefore, the content of Best_Bid and the given array called (Bid_Array) are as follows:

Best_Bid = RP3

Bid_Array:

RP1	RP2			

It is clear that after the ending of bidding time period T, the task will be submitted to RP3 in order to execute. In the case of any problem or fault by RP3, the task is given to other available resources within array where they have order of priority.
The following is a brief description of the proposed algorithm:

1. Start;
2. if the first bid received, consider the Best_Bid;
3. After receiving the next bid, compare this bid with Best_Bid;
4. If the received bid is better than Best_Bid then store the Best_Bid in an array (named Bid_Array) and consider the received bid the Best_Bid;
5. Else store the received bid in Bid_Array;
6. Repeat this process when received a new bid;
7. If T is finished, submit the task to Best_Bid;
8. If submission fails then sort the array (descending) and submit the task to Resource Provider that placed in first location of Bid_Array;
9. If all Resource Providers in Bid_Array fail, start a new bidding Process;
10. End;

Fig.2 Algorithm Steps in Grid client side for each task.

And the proposed algorithm in resource providers is as follows for each task:

1. Start;
2. Receive CFP from grid clients;
3. Compare CFP with current status of resources;
4. If capable to participate then calculate the bid and send it to the grid client;
5. Else ignore the CFP;
6. If received a task submission message
   If the resource is reserved for that task or the resource is free then submit the task to the resource;
7. Else ignore the message;
8. Repeat this process until logout from grid system;
9. End;

Fig.3 Algorithm Steps in Resource Provider side for each task.

5. Evaluation

As described in section 3, the bidding model is divided into two categories: Reserved bidding model and non-Reserved bidding model. So it would be reasonable to compare the proposed algorithm presented in this article with the non-reserved model in terms of task waiting time and the task completion time. In order to do this, we will compare the proposed algorithm with the proposed algorithm in [3] which uses non-reserved method to select resource in terms of task waiting time and task completion time. For this purpose, we will use the experimental results of Taiwan UniGrid [15] resource selection that is also presented in [3].

According to this tool, 20 machines are considered as resource providers and 20 tasks no matter big or small (in terms of time) are submitted to these machines. Of all 20 tasks we choose 10 tasks to evaluate the algorithm. It is presumed that running time of bidding is 15 minutes and grid client waiting time to receive an answer (proposal) from the resource provider is 20 seconds, means that if a proposal submitted from the resource provider takes more than 20 seconds, that proposal will be ignored by the grid client. The arriving rate of the tasks is 50 seconds, that is on average one task is submitted to grid system every 50 seconds. We assumed that of these 10 tasks, the first 5 tasks and the next 5 tasks are submitted to recourse provider by CFP1 and CFP2, respectively.

According to non-reserved bidding model, the resource provider participates at both CFPs and then submits tasks to its resources to execute. Since grid environment is dynamic, with participation of resource provider in other biddings, the tasks completion time which equals to executing time + task waiting time in the queue (Network delay ignored), will be unpredictable. In Table1 all the primary information needed to evaluate the proposed algorithm is given.

Table 1: Information needed for evaluating the proposed algorithm

Parameter	*Value*
Number of Resource Providers	20 RPs
Number of tasks	10 tasks
Task arrival rate	50 seconds
Deadline for waiting bids	20 seconds
Bidding period	15 minutes

In Table2, Table3 the required time to execute tasks and their delivery time to the system are given.

Table 2: The average time needed to executing tasks and arrival time of tasks in CFP1

Tasks	*Time needed for task executing in CFP 1*	*Arrival time of tasks in CFP1 (Second)*
Task 1	122.08 s	≅0
Task 2	13.74 s	50
Task 3	136.63 s	100
Task 4	14.20 s	150
Task 5	130.39 s	200

Table 3: The average time needed to executing tasks and arrival time of tasks in CFP2

Tasks	*Time needed for task executing in CFP 2*	*Arrival time of tasks in CFP2 (Second)*
Task 6	18.51 s	250
Task 7	130.27 s	300
Task 8	16.00 s	350
Task 9	159.47 s	400
Task 10	16.05 s	450

If we presume that task scheduler adapt the FCFS (First Come First Serve) algorithm to select tasks [16], in this case, the average waiting time of tasks and average executing time of tasks in the CFP1 would be like this:

- The average waiting time of tasks = 63.40 seconds
- The average executing time of tasks = 83.40 seconds

Thus:

- The average completion time of tasks will be 146.80 seconds.

And average waiting time of tasks and average executing time of tasks in the CFP2 would be like this:

- The average waiting time of tasks = 64.85 seconds
- The average executing time of tasks = 68.06 seconds

Thus:

- The average completion time of tasks will be 132.91 seconds.

Now, if by allocating resources to submitted tasks in CFP1, the resource provider also participates in CFP2, then the average waiting time of tasks and the average executing time of tasks would be like the following:

- The average waiting time of tasks = 128.25 seconds
- The average executing time of tasks = 151.46 seconds

Thus:

- The average completion time of tasks will be 279.71 seconds.

Briefly, if we want to compare average completion time of tasks in the Non-Reserved bidding model with the average completion time of tasks in the proposed algorithm, the result will be:

- The average completion time of tasks in Non-Reserved bidding model = 279.71 seconds.
- The average completion time of tasks in proposed algorithm (Reserved bidding model) =146.80 seconds.

In fact, in the proposed algorithm the resource provider completes the tasks nearly within given time in CFP by grid client via sending proposal to one Call for Proposal (CFP) and ignoring the other CFPs and also guarantees future resource statues. While in Non-Reserving model, the resource provider delays the completion time of tasks via participating in other CFPs. Resource provider can participate in various CFPs via adopting the Non-Reserving strategy and considering the fact that grid environment is a dynamic. Since the number of these CFPs and consequently the number of submitted tasks in every CFP is already indefinite (especially during the submitting of the task to the grid system by the client), hence the time of task completion would be unpredictable.

In Figure.3, the comparison of completion time of task in Non-Reserved model and the model proposed in [3] is shown and in Figure.4 the time of task completion in proposed algorithm is shown. The data is related to Table2 and network delay time and other delay times have been ignored.

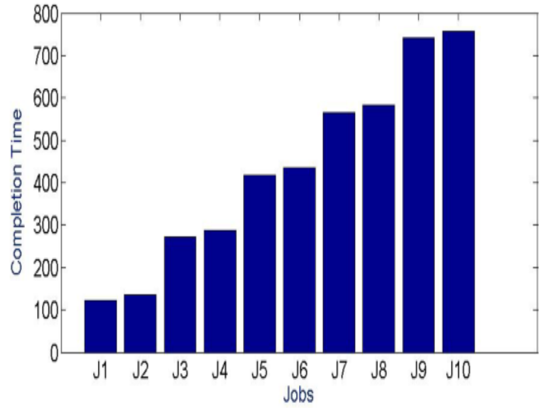

Fig.4 Task completion time in proposed [3]

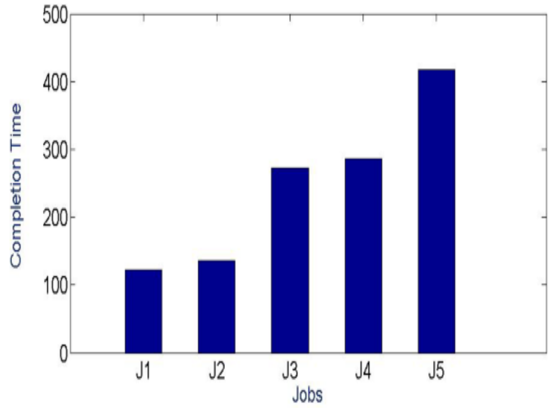

Fig.5 Task completion time in our proposed algorithm

6. Conclusion and Future Works

This paper presents a new resource selection algorithm in a bidding based grid environments to minimize the total time of task completion. The obtained result of evaluation shows that in the proposed algorithm, the tasks are completed nearly within given time in CFP by grid client. According to this fact that in the reserved bidding model the resource utilization rate is low (because, resources are reserved by resource provider for a bid), we can monitor the number of available resources and their features and also the number of requests sent by the grid clients to take the resource and release the resources from the lowest priority and let their resource provider to participate it in other biddings. By repeating this process over definite time, we can control and manage the failure of resource to some extent that is currently executing the tasks with other reserved resources. We can also take high advantage of the available resources in the grid environment.

References

[1] I. Foster, C. kesselman, S. Tuecke, "The anatomy of the Grid: Enabaling scalable virtual organizations", The International Journal of High Performance Computing 15 (3) (2001).

[2] W.E. Johnston, "Computational and data Grids in large-scale science and engineering", Future Generation Computer System 18 (8) (2002).

[3] Wang,C.-M., et al., " Dynamic resource selection heuristics for a non-reserved bidding-based Grid environment. Future Generation Computer System", 2010.26(2):p. 183-197.

[4] D.A Menasce, D. Saha, S.C. da Silva Porto ,V.A.F. Almedia, S.K. "Tripathi, Static and dynamic processor scheduling disciplines in heterogeneous parallel architectures", Journal of Parallel and Distributed Computing 2 (1) (1995).

[5] R. Nossal, "An evolutionary approach to multiprocessor scheduling of dependent tasks", Future Generation Computer Systems 14 (5_6) (1998) 383-392.

[6] S. Salleh, A.Y. Zomaya, "Multiprocessor scheduling using mean-field annealing", Future Generation Computer Systems 14 (5_6) (1998) 393_408.

[7] F. Seredynski, J. Koronacki, C.Z. Janikow, "Distributed multiprocessor scheduling with decomposed optimization criterion", Future Generation Computer Systems 17 (4) (2001) 387_396.

[8] M.J. Litzkow, M. Livny, M.W. Mutka, Condor: "A Hunter of Idle Workstations", in: Proceedings of the 8th International Conference on Distributed Computing Systems, 1988.

[9] Adil Yousef, Abdul Hanan Abdullah and Aboamama Atahar Ahmed, " A Bidding-based Grid Resource Selection Algorithm Using Single Reservation Mechanism", International Journal Of Computer Applications (0975-8887) Volume 16- No.4, Feburay 2011.

[10] De Roure, D., et al., "The evolution of the Grid. Grid computing: making the global infrastructure a reality", 2003. 13: p. 14-15.

[11] Zhiling Lan and Yawei Li, "Failure-Aware Resource Selection for Grid Computing" 2006.

[12] Asgarali Bouyer, Mohammadbager Karimi, Mansour jalali and Mohd Noor MD SAP, " A new Approach for selecting Best Resource Nodes By Using Fuzzy Decision Tree in Grid Resource Broker", International journal of Grid and Distributed Computing 2008.

[13] Zhengi Zahi, "Grid Resource Selection Based on Reinforcement Learning", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010).

[14] Ali Sarhadi and Mohammad Reza Meybodi, "New Algorithm to Resource Selection in Computational Grid Using Learning Automata", International journal of Intelligent Information Technology Application, 2009, 2(5):204-208.

[15] P.C shih, H.M. chen, Y.C. Chung, C.M. Wang, R.S. Chang, C.H. Hsu, K.C. Huang, C.T. Yang, Middleware of Taiwan UniGrid, in:Proceedings of the 2008 ACM symposium on Applied computing 2008. Available at http://www.unigrid.org.tw/.

[16] Volker Hamscher, Uwe Schwiegelshohn, Achim Streit, and Ramin Yahyapour. "Evaluation of Job-Scheduling Strategies for Grid Computing", 2000.

Esmaeil Pourazarm-Hajiaga received the B.Sc. degree in Computer Engineering, Software engineering, from Islamic Azad University, Tabriz branch, Tabriz, Iran, in 2007, the M.Sc. degree in Computer Engineering, Software engineering from Islamic Azad University, Shabestar branch, Shabestar, Iran, in 2011. His research interests include Distributed systems, Computational Grid and Cloud Computing.

Saeed Parsa received the BS in mathematics and computer Science from Sharif University of Technology, Iran, the MS degrees in computer science from the University of Salford at England, and the Ph.D. degree in computer science from the University of Salford at England. He is an associated professor of computer science at Iran University of Science and Technology. His research interests include software engineering, soft Computing and algorithms.