A Bidding-based Grid Resource Selection Algorithm Using Single Reservation Mechanism

Adil Yousif
Faculty of Computer Science
& Information System
Universiti Teknologi Malaysia

Abdul Hanan Abdullah
Faculty of Computer Science
& Information System
Universiti Teknologi Malaysia

Aboamama Atahar Ahmed
Faculty of Computer Science
& Information System
Universiti Teknologi Malaysia

ABSTRACT

The selection of resources that best fitting tasks in grid environments is an essential and critical factor to system performance. Current resource selection methods apply traditional mechanisms for the process of task allocation to resources, which cause performance problems. This paper proposes a resource selection algorithm in bidding based grid environment to minimize the total time for task completion in effective and efficient way. To avoid unexpected completion time arises when using non-reserved bidding process due to the competition between grid clients on resources; our algorithm 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. This algorithm uses single reservation mechanism to reserve the best resource for the task. Unlike traditional reserved algorithms this algorithm reserves only the best resource and hence it allows other resources to participate in other bidding processes. We believe the proposed algorithm can select the most fitting resources for tasks execution and achieve a good performance in terms of effectiveness and efficiency.

Keywords

Grid computing; Resource selection; Resource broker; locking; Bidding

1. INTRODUCTION

The demand for high computational power grows faster than the increase in hardware productivity and available individual computers [1].The existing individual computers cannot fulfill the rising number of intensive applications that need high computational power in order to be implemented [1]. To turn the tide we must look for techniques that result in significant computational gains. Grid technologies emerged in the middle of 1990s to satisfy the rising demand for bandwidth, storage, and computational resources [2].

Ian Foster in [2] defines grid computing as "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations". The sharing mentioned in the definition is not a simple file sharing but rather direct contact and access to resources and services in a secure manner[1-2]. A virtual Organization (VO) typically comprises of a set of individuals or organizations, those individuals or organizations physically dispersed, but they work as one general organization, with a general concern[3].

Grid technology contributes in solving several computational problems such as exploiting underutilized resources by executing an existing job on a different machine[4]. 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[5]. Furthermore grid computing allows access to additional resources and software rather than only those resources ans software available in the user organization; this has a great impact since it supports the user with expensive hardware and licensed software [5].

Resource management in grid environments is a great challenge; this is due to the heterogeneity of resources in grid environments and in addition to that, grid resources belong to diverse administrative domains and apply different management policies [6].

The resource allocation process in grid environments consists of three main phases; the first phase is resources discovery, the most important issues when dealing with resource discovery is how to publish the resource information by providers and how those resources can be discovered by grid clients [7-8]. This information is handled and supported by a resource registry[7]. The second phase is resource selection which regarding with the process of selecting the best resources to execute a certain task [9]. The last phase of resource allocation process is resource usage which is concerning with running the task on the selected resources and monitoring the execution [7]. In this paper we are focusing on resource selection phase and its issues.

This paper is organized as follows. Section 2 provides a review of the literature on contributions related to grid resource selection. Sections 3 review the existing resource selection models. The proposed algorithm of our resource selection mechanism is presented in Section 4. and section 5 summarizes the work and the future scope, followed by a list of references.

2. RELATED WORK:

Resource Management systems do not focus on the basic functionalities of the resources provided to the user, but rather they focus on the way those functions are achieved and controlled [4]. For Instance if there is a computational resource, the resource management system is not concerning on the primary functions of the computation process, rather it focuses on the management of those computation functions, such as when they start computing, how long is the duration of the computing processes and when they terminate.

Resource selection in grid systems is concerning with selecting 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 [7].

One of the most important components of Grid systems is resource brokering , which hides the complexity of grid middleware, and allows the grid clients to state the specifications and attributes of requested resources, and then the broker searches for resources that match the specifications and the attributes and chooses the appropriate resource for task execution [10].

Based on the scope of resource broker functions Elmroth and Tordsson [8] categorize two types of resource brokers. Centralized resource broker (Figure 1) handles all tasks submitted to the grid system [8]. While decentralized resource broker manages tasks submitted to the grid system by a single grid user. One of the advantages of centralized resource brokers they have knowledge and control over all grid resources and tasks submitted to the grid system and therefore they can provide a good management for tasks. However those brokers may become a performance bottleneck. Furthermore centralized resource brokers like other centralized systems usually represent a single point of failure [11]. Decentralized resource brokers provide scalable grid systems and do not represent single point of failure. However they suffer from the lack of knowledge about the global state of the system, and they do not have full control over the grid resources [8].

Decentralized resource broker (as described in figure 2) operates as follow: All resources join the grid have to register in one or more index servers [8]. The index server itself can register in other index servers. All users in the grid system have their own brokers, and they can use those brokers to access the available resources [12]. For each resource broker to discover which resources are available it communicates with one or more index servers. After discovering the available resources the resource broker can request additional information directly from the resource provider. Once the appropriate resources are selected by the user broker, user is informed to apply tasks submission and in some cases task control [8].

3. RESOURCE SELECTION MODELS:

3.1Matchmaking 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. 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[13]. 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[13-15].The matchmaker can be implemented by job manager[16] or by using centralized resource broker as discussed in [8].

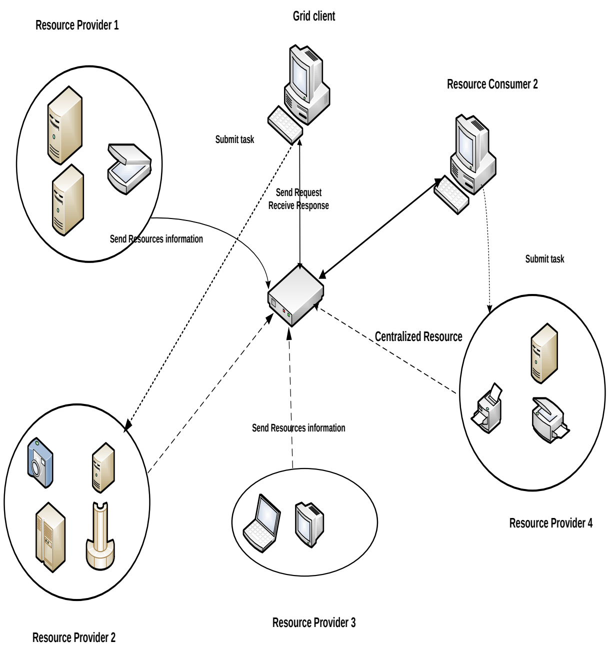

Figure 1: Centralized resource broker

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 [13, 16]. Following is an abstract description of bidding based model.

The bidding process starts when a grid client sends call-for-proposal (CFP) requests to all available resource providers; these requests include the job 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, order them and selects the resources that offer the best bid[13, 17-18].

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.

3.2.1Reserved 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[16]. However if the grid client later refuses the bid, that resource is squandered.

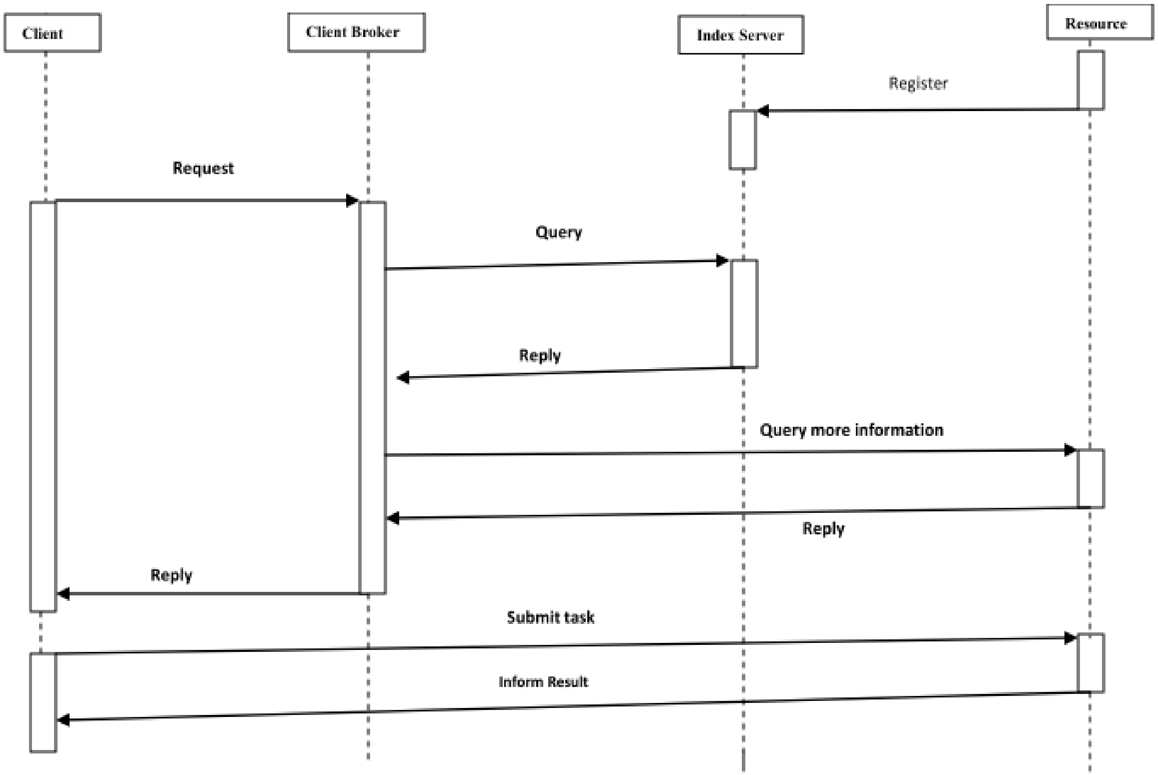

Figure 2: Decentralized resource broker

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[19-20].

3.2.2 Non-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 send 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 job execution time possibly will not be as estimated[16, 20].

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[13].

3.2.3Bidding process challenges:

Reserved bidding model wastes the resources, this is because more than one resource providers reserve their resources for a single bidding process and at the end, the grid client selects only one resource for task execution and hence other resources miss the opportunity to participate in other bidding processes and serves other grid clients before they are rejected by the original grid client. On the other hand non-reserved bidding process lead to unexpected completion time for tasks. These happen because the resource provider may participates in more than one bidding processes by the same resources simultaneously. This makes the grid clients compete on those resources and the competition may cause unexpected completion time for the submitted tasks.

4. THE 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, then the grid client sends a token to the old provider informs it to free the reserved resources for that bid and 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 and sends free token to the new bid provider.

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. The following is a brief description of the proposed algorithm:

4.1Algorithm steps in grid client side for each Task:

```
Start:
  Create an array  named  bid_array
  Specify task requirements in a CFP message.
  Send the CFP to all available resource providers
  i←1     // create and initialize counter
  wait for responses for a given time period T

  repeat
        received_bid←receive_bid_from_provider ( )
      if  i = 1
        best_ bid ← received_bid
      else
        if the received bid is better than best_ bid then
          send free_token to the best_ bid provider
          bid_array[i] ← best_ bid
          best_ bid ← received_bid
        else
          send free_token to the received_bid  provider
          bid_array[i] ←received_bid
        endif
      endif
   i  ← i +1
  until the given time period T finishes
  Submit the task to best_ bid Provider
  if submission fails then
      Sort(bid_array)
      i ←1
      repeat
        submit the task to bid_array[i] provider
        i ←i+1
      until the task is submitted or no appropriate providers
  are left to try
  end if
  end
```

4.2Algorithm steps in resource Provider side for each Task:

```
Start
Repeat
   receive CFP from  grid clients
  Compare CFP with current status of resources
   if capable to participate     then
       calculate the bid
       send the bid to the  grid client
       reserve the resource for that bid.
  else
      ignore the CFP.
      endif
if receive  a token free from a grid client
 free the reserved resource for that bid
if receive a task submission message
  if the resource is reserved for that task or the resource is free
     Submit the task to the resource
  endif
 else
ignore the message
endif
until  logout from grid system
End
```

5. CONCLUSION & FUTURE WORKS:

This paper presents a new resource selection algorithm in a bidding based grid environments to minimize the total time for task completion in effective and efficient way. Our algorithm proposes the single reservation mechanism to reserve the best resource for task execution as a commitment and hence guarantees the task completion time will be as expected. The proposed Algorithm tackles the problems arise when using traditional reserved and non-reserved resource bidding selection mechanisms. Now we are in the implementation phase and we plan to conduct an experiment in grid testbed using Globus Toolkit as a middleware to evaluate the algorithm. In future we plan to upgrade this model into dynamic reservation model.

REFERENCES:

[1] Foster, I. and C. Kesselman, *The grid: blueprint for a new computing infrastructure*. 2004: Morgan Kaufmann.

[2] Foster, I., C. Kesselman, and S. Tuecke, *The anatomy of the grid: Enabling scalable virtual organizations.* International Journal of High Performance Computing Applications, 2001. **15**(3): p. 200-222.

[3] Arafah, M.A., H.S. Al-Harbi, and S.H. Bakry, *Grid computing: A STOPE view.* International Journal of Network Management, 2007. **17**(4): p. 295-305.

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

[5] Ferreira, L., I.B.M.C.I.T.S. Organization, and I. ebrary, *Introduction to grid computing with globus*. 2003: IBM Corp., International Technical Support Organization.

[6] Foster, I., et al., *The physiology of the grid.* Grid computing: making the global infrastructure a reality, 2003: p. 217–250.

[7] Schnizler, B., *Resource Allocation in the Grid: A Market Engineering Approach*. 2007: Univ.-Verl. Karlsruhe.

[8] Elmroth, E. and J. Tordsson, *Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions.* Future Generation Computer Systems, 2008. **24**(6): p. 585-593.

[9] Malarvizhi, N. and V.R. Uthariaraj. *A Broker-Based Approach to Resource Discovery and Selection in Grid Environments*. in *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*. 2008.

[10] Haji, M.H., et al., *A SNAP-based community resource broker using a three-phase commit protocol: A performance study.* Computer Journal, 2005. **48**(3): p. 333-346.

[11] Roy, S., M. Sarkar, and N. Mukherjee. *Implementation of a Resource Broker for Efficient Resource Management in Grid Environment*. in *Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on*. 2007.

[12] Krauter, K., R. Buyya, and M. Maheswaran, *A taxonomy and survey of grid resource management systems for distributed computing.* Software: Practice and Experience, 2002. **32**(2): p. 135-164.

[13] Wang, C.-M., et al., *Dynamic resource selection heuristics for a non-reserved bidding-based Grid environment.* Future Generation Computer Systems, 2010. **26**(2): p. 183-197.

[14] Khanli, L.M. and M. Analoui, *An approach to grid resource selection and fault management based on ECA rules.* Future Generation Computer Systems, 2008. **24**(4): p. 296-316.

[15] Yu, H., X. Bai, and D.C. Marinescu, *Workflow management and resource discovery for an intelligent grid.* Parallel Computing, 2005. **31**(7): p. 797-811.

[16] Hongbo, Z., et al. *A virtual-service-domain based bidding algorithm for resource discovery in computational grid*. in *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*. 2005.

[17] Kakarontzas, G. and I.K. Savvas. *Agent-Based Resource Discovery and Selection for Dynamic Grids*. in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE '06. 15th IEEE International Workshops on*. 2006.

[18] Kolano, P.Z. *Surfer: an extensible pull-based framework for resource selection and ranking*. in *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*. 2004.

[19] Cheung, W.K., et al. *Dynamic Resource Selection For Service Composition in The Grid*. in *Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on*. 2004.

[20] Xueguang, C. and S. Haigang, *Further extensions of FIPA Contract Net Protocol: threshold plus DoA*, in *Proceedings of the 2004 ACM symposium on Applied computing*. 2004, ACM: Nicosia, Cyprus. p. 45-51.

[21] Maheswaran, M. and K. Krauter, *A parameter-based approach to resource discovery in Grid computing systems.* Grid Computing—GRID 2000, 2000: p. 363-385.