Future Generation Computer Systems 26 (2010) 183–197

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Dynamic resource selection heuristics for a non-reserved bidding-based Grid environment

Chien-Min Wang [a], Hsi-Min Chen [b,*], Chun-Chen Hsu [c], Jonathan Lee [b]

[a] Institute of Information Science, Academia Sinica, Taipei, Taiwan

[b] Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan

[c] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 19 December 2008
Received in revised form
29 July 2009
Accepted 3 August 2009
Available online 7 August 2009

Keywords:
Grid computing
Resource selection
Resource management
Bidding
Matchmaking

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 Grids increases rapidly, selecting appropriate resources for jobs 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, we propose 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 jobs revealed by providers. We also present the results of experiments conducted to evaluate the performance of the proposed heuristics.

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]. 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 jobs 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. Thus, exploiting previous cluster-based scheduling heuristics [3–7] to allocate tasks through a centralized manager or mapper is not feasible.

In recent years, many matchmaking-based technologies have been proposed to address the issue of Grid resource management [8–15]. Fig. 1(a) presents an abstract matchmaking model generalized from these technologies. However, the matchmaking technique may cause a matchmaker overload problem. Since a resource matchmaker is responsible for registering all resource states advertised by providers and executing matching algorithms, an increase in the number of resources and the frequency of job requests creates a performance bottleneck. Moreover, resource states may change minute by minute due to requesters' activities or resource failures, so the matchmaking technique may fail to reflect the dynamic nature of Grid resources. This is because matchmaking is a push-based model in which a matchmaker does not learn about the changes in resource states until the resource providers publish their new states. In consequence, matchmaking may return inaccurate results.

To avoid single point of failure, matchmaker overload and expired resource information, bidding provides an alternative means of resource allocation in distributed systems [16–24]. Fig. 1(b) depicts the abstract process of the bidding model. A resource requester starts a bidding process by sending a set of call-for-proposal (CFP) requests, which contain job requirements, to resource providers. Then, based their resource utilization and policies, the providers decide whether or not to participate in the bidding process. If they join the bidding process, they return bids that describe the states of their resources to the requester. Finally, the requester evaluates and ranks the collected bids based on its selection strategy and submits the job to the provider that proposes the best-ranked bid. The bidding model has the following advantages over the matchmaking model. (1) Scalability: resource allocations between providers and requesters in the

* Corresponding author.
E-mail address: seeme@iis.sinica.edu.tw (H.-M. Chen).

0167-739X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2009.08.003

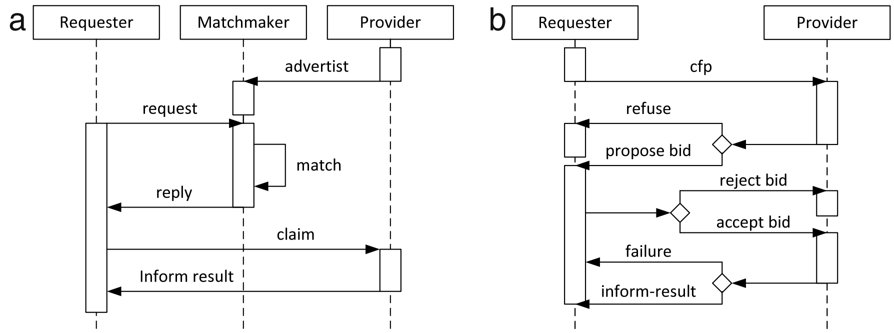

Fig. 1. (a) The abstract process of the matchmaking model. (b) The abstract process of the bidding model.

bidding model are fully distributed without the intervention of a centralized matchmaker/broker. (2) Autonomy: requesters themselves can determine which of the offered resources are best-suited to execute their jobs, while providers can contribute their resources according to their sharing policies and report up-to-date state information. (3) Reliability: if a resource fails during a job's execution, the requester can select other candidate resources from the received bids.

Under the bidding model, resource providers can usually choose between two bidding strategies, *reserved* and *non-reserved* bidding [25]. Providers who adopt the *reserved* strategy keep the resources for each bid as commitments to guarantee future resource states. However, if the requester subsequently rejects the bid, the reserved resources will be wasted. In this scenario, other requesters may be prepared to accept the bid before the original requester rejects it; hence, there is a high probability that the provider will miss the opportunity to serve other requesters with reserved resources. In contrast to the *reserved* strategy, resource providers who adopt the *non-reserved* option offer the same resource states to a set of requesters without reserving resources for each bid. This strategy enables providers to fully utilize their resources, but it does not guarantee the resource states. If requesters receiving the same bids submit jobs to the provider simultaneously, they will have to compete for the resources so that the job completion time may not be as expected.

In addition to the above strategies, the bidding model allows providers to reveal different levels of information about competing jobs to requesters. As shown in Fig. 1(b), after resource providers receive CFPs from requesters, they can simply reveal the capabilities of the provided resources, provide information about the number of competitors, or give even more complete information about the competitors. The level of information revealed is an important factor that affects the performance of resource selection for a job's execution.

In this paper, our objective is to minimize the turnaround time of jobs in a non-reserved bidding-based Grid environment. The turnaround time covers the period from the time a job arrives to the receipt of the executed result. In online systems, users are more sensitive to the turnaround time than the execution time, waiting time or makespan [26]. To minimize the turnaround time in this model, we propose a set of deterministic and probabilistic resource selection heuristics. In contrast to traditional centralized scheduling problems, the key challenge of resource selection in the bidding model is that there is no global information system to facilitate optimum decision-making; hence, requesters are only aware of partial information released by resource providers. Thus, we consider various levels of information about competitors in the proposed heuristics. We want to determine whether requesters could make better scheduling decisions if they have more information about the states of competing jobs.

We conduct experiments to evaluate the performance of the heuristics for various levels of information and the impact of non-cooperative requesters. The experimental results show that the performance of the Dissolve-P heuristic is superior to that of other heuristics when information about competitors is not provided. However, the MCT-D heuristics outperform the other heuristics when information about the execution times of competitors is provided. We also find that the level of information has a significant effect on the performance of the MCT-D based heuristics, but it does not influence the Dissolve-P based heuristics. Furthermore, requesters who adopt cooperative resource selection strategies achieve better results than those that use non-cooperative strategies.

The contributions of this paper are as follows. (1) To the best of our knowledge, this is the first study of the resource selection issue in an online non-reserved bidding-based Grid system that focuses on minimizing the turnaround time. (2) To address this issue, we propose a set of probabilistic and deterministic resource selection heuristics, as well as a pre-scheduling mechanism, and evaluate their performance. (3) The proposed heuristics consider various levels of information about competing jobs. (4) We examine the impact of cooperative requesters and non-cooperative requesters on the performance of Grid resource selection.

The remainder of the paper is organized as follows. Section 2 contains a review of the literature on resource selection. In Section 3, we formally define the problem considered in this research. Section 4 presents the proposed heuristics for the various levels of information revealed by providers. We describe the simulation setup and evaluate the performance of the proposed heuristics in Section 5. Then, in Section 6, we summarize our conclusions.

2. Related work

A number of resource management approaches have been proposed in various Grid projects. Globus Toolkit [27], the most popular Grid middleware, integrates distributed computing resources and provides a set of management tools, such as security, data management, information services and execution management. In addition, for resource management, Globus provides MDS (Monitoring and Discovery Service) [28] to support the discovery and monitoring of resources, services, and computations, and GRAM(Grid Resource Allocation and Management) [29] combined with RSL(Resource Specification Language) for resource allocation tasks. However, Globus only allows users to specify basic configurations, such as the file path, maximum CPU power, required memory, and wall clock time. It does not support job matching/scheduling at the global level; instead it leaves the task to the development of an upper-layer service. The bidding model and the proposed heuristics can be constructed as a high-level resource management service on top of Globus.

Condor matchmaker [8,11] is another well-known resource management framework designed for high-throughput computing