Computers & Operations Research 76 (2016) 167–182

Contents lists available at ScienceDirect

Computers & Operations Research

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

Acquisition planning and scheduling of computing resources

Chien-Nan Yang [a], Bertrand M.T. Lin [a,*], F.J. Hwang [b], Meng-Chun Wang [a]

[a] Institute of Information Management, Department of Information Management and Finance, National Chiao Tung University, Taiwan
[b] School of Mathematical and Physical Sciences, University of Technology Sydney, Ultimo 2007, Australia

ARTICLE INFO

Article history:
Received 23 November 2014
Received in revised form
11 March 2016
Accepted 17 June 2016
Available online 25 June 2016

Keywords:
Computing service
Acquisition planning
Scheduling
Heuristics
Tabu search
Genetic algorithm

ABSTRACT

Cloud computing has been attracting considerable attention since the last decade. This study considers a decision problem formulated from the use of computing services over the Internet. An agent receives orders of computing tasks from his/her clients and on the other hand he/she acquires computing resources from computing service providers to fulfill the requirements of the clients. The processors are bundled as packages according to their speeds and the business strategies of the providers. The packages are rated at a certain pricing scheme to provide flexible purchasing options to the agent. The decision of the agent is to select the packages which can be acquired from the service providers and then schedule the tasks of the clients onto the processors of the acquired packages such that the total cost, including acquisition cost and scheduling cost (total weighted tardiness), is minimized. In this study, we present an integer programming model to formulate the problem and propose several solution methods to produce acquisition and scheduling plans. Ten well-known heuristics of parallel-machine scheduling are adapted to fit into the studied problem so as to provide initial solutions. Tabu search and genetic algorithm are tailored to reflect the problem nature for improving upon the initial solutions. We conduct a series of computational experiments to evaluate the effectiveness and efficiency of all the proposed algorithms. The results of the numerical experiments reveal that the proposed tabu search and genetic algorithm can attain significant improvements.

1. Introduction

In the recent decade, cloud computing has become a popular topic in many research and application areas over the Internet. Armbrust et al. [2] gave a definition from an academic perspective: "Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services." It is an attractive solution for those companies that do not have the ability and capital to build a large ad hoc computing and/or data center. Acquiring computing resources from service providers instead of establishing private IT infrastructures saves both time and cost in many aspects. As the provider is also responsible for maintenance, the clients do not have to hire specialists to be in charge of the security or sustainability problem, thus reducing the personnel cost to a considerable extent. The convenience and stability brought forth by cloud computing have stimulated the development of technologies and service models for this booming business opportunity. One of the service models is called IaaS (Infrastructure as a Service), where the vendors sell their computing resources often referred to as "virtual machines" to the clients. Amazon Elastic Compute Cloud (http://aws.amazon.com/ec2/) and IBM SmartCloud (http://www.ibm.com/cloud/) are some prominent examples of IaaS.

In contrast to a wide variety of service modes, there is an obvious lack of the pricing schemes for adaptability. Nowadays the providers sell their computing resources in similar ways, which are called pay-as-you-go or pay-per-use. This usage-based pricing scheme stems from the utility service (e.g. electricity, natural gas and water) pricing concept. Besides, a group of different prices is set for processors according to the computing capacities, such as the CPU speeds or the memory sizes. There is no minimum fee or discount on purchasing multiple instances, and only a slight discount on purchasing an instance for a long time, e.g. a 1-year or a 3-year term. However, it is insufficient for the consumers who have different and special requirements. Thus, new pricing schemes are needed for attracting those consumers. Bakos and Brynjolfsson [5] studied the strategy of bundling distinct information goods and selling them for a fixed price. Analyses revealed that this strategy often yields higher profits than that from selling them separately. Sundararajan [25] indicated that fixed-fee unlimited-usage pricing and usage-based pricing schemes should be included in different stages of information markets. They suggested that a fixed-fee pricing scheme should be included in both

* Corresponding author.
E-mail address: bmtlin@mail.nctu.edu.tw (B.M.T. Lin).

http://dx.doi.org/10.1016/j.cor.2016.06.015
0305-0548/

early-stage and mature markets. With these studies, we come up with a new scheme to bundle the processors with different speeds together as packages and rate them at different prices according to their computing capacities. A fixed-charge time interval is also given since the unlimited-usage price may be too expensive for small- and medium-size enterprises. Within this time interval, consumers can fully utilize the resources in the purchased package without any extra fee. After the fixed-charge time interval, the usage-based pricing scheme applies to accumulate the expenses. This new pricing scheme is a win–win model to both parties. From the viewpoint of cloud computing service providers, the flexibility of adjusting processors makes them easily bundle processors for forming distinct packages. The providers can bundle the oldest/slowest machines and the newest/fastest machines together so that the performances could be balanced. The processors can be freely arranged to make multiple packages and fully satisfy various kinds of consumers. Also, this model can attain a large increase in profits as well as reduce the unnecessary waste of resources. From the standpoint of clients, there are more choices of packages and pricing schemes. The clients can hire packages according to their budgets and dispatch the jobs based on their urgency.

In this paper, we, from the perspective of a cloud service agent, develop a planning and scheduling model by regarding the cloud computing environments of IaaS as parallel machines. A cloud service agent acts as an intermediary between the cloud computing service providers and the clients to negotiate the contracts, bargains, and also to provide additional services. As the interest in cloud computing grows, a brokerage service is necessary for the clients. Like the real estate agents or stock agents, the cloud computing agents are the connections between service providers and clients. They help the clients to select the services they need, and purchase the services from the providers. They may have multiple jobs from various clients and also purchase computing resources from different providers. Therefore, the arrangement of resource dispatching becomes an important issue. The agents have to complete the jobs of consumers as soon as possible to earn their trust and the opportunities for future cooperation. On the other hand, they have to purchase sufficient resources to process the jobs and make their own profits. Thus, as a third-party business, the management of the cloud computing agents is directly related to both service providers and clients. For simplicity of description, we assume that the resources in the purchased packages can be fully controlled and managed by the agents, and allocation of jobs to processors and the job sequence on each processor are determined by the agents as well. Since the agent aims to lower the cost of purchasing computing resources and finish the jobs on time, we formulate the studied model as a parallel-machine scheduling problem with resources acquisition planning. Since the machine speeds are usually distinct in the environment of cloud computing, we are tackling precisely a scheduling problem on uniform parallel machines. The objective function is a linear combination of the total weighted tardiness ($\sum w_jT_j$) and the total package acquisition cost ($\sum \Psi_i(L_i)$). The acquisition cost of package i is calculated based on the schedule makespan L_i and the package pricing scheme proposed above. Using the three-field notation [23], the studied problem can be denoted by $Qm \parallel \alpha \sum w_jT_j + (1-\alpha)\sum \Psi_i(L_i)$, where Qm represents the uniform parallel machines, and α is the weighting parameter for normalizing the two types of costs.

The rest of this paper is organized as follows. A brief review of the related literature is given in Section 2. In Section 3, problem definition and notation are provided along with an illustrated example, and an integer programming model for the studied problem is formulated. In Section 4, several heuristics are used to attain initial solutions. Details of the meta-heuristics utilized for improving the initial solutions are described in Section 5. In Section 6, computational results and related analyses are reported. Finally, Section 7 concludes this research and provides suggestions for further research.

2. Literature review

Parallel-machine scheduling problems with different objectives and constraints have been extensively studied in the open literature. As Cheng and Sin [9] noted in their state-of-the-art review of major research results in parallel-machine scheduling problems, various job characteristics, machine configurations and performance criteria are of theoretical interest as well as practical significance. Minimizing the total tardiness with penalty weights is one of the commonly considered objectives. When the penalty weights are arbitrary positive numbers, the scheduling problem with identical parallel machines $Pm \parallel \sum w_jT_j$ is NP-hard in the strong sense [19,23]. When the penalty weights of all jobs are the same, the $Pm \parallel \sum T_j$ problem is at least binary NP-hard [17]. Several studies examined the properties that lie in the structures of an optimal schedule, and developed exact algorithms for $Pm \parallel \sum T_j$. Azizoglu and Kirca [3] presented some dominance properties for $Pm \parallel \sum T_j$ and proposed a branch-and-bound algorithm that can solve instances with up to 15 jobs and 3 machines. They also extended some of the properties for $Qm \parallel \sum T_j$. Yalaoui and Chu [29] developed more dominance properties and bounding rules, and showed that their branch-and-bound algorithm could obtain optimal solutions in some cases with 30 jobs and 2 machines. Shim and Kim [24] also provided dominance properties and lower bounds to show that the suggested algorithm could find optimal solutions for problems with up to 30 jobs and 5 machines in a reasonable time. As for the uniform parallel-machine problems, Dessouky et al. [10] presented algorithms for different objectives under the strong assumption that the jobs are identical, and proposed a dynamic programming algorithm for minimizing the total completion time subject to release dates.

Considering the non-classical objective functions, such as the job processing costs, several studies investigated bi-criteria scheduling problems. Leung et al. [20] addressed the bi-criteria consisting of one classical and one non-classical objective functions with two different bi-criteria structures in parallel machine scheduling. One is the hierarchical bi-criteria, i.e. optimizing the secondary objective among the schedules that the primary objective is minimized. The other is the linear combination of two objective functions, which is exactly the bi-criteria structure used in this paper. The considered classical objective is either the makespan or total completion time. Concerning a cost associated with the processing of a specific job on a particular machine, the addressed non-classical objective is the total machine assignment cost. They presented the complexity results for the considered problems. Lee et al. [18] studied the same problem with the hierarchical bi-criteria structure and developed approximation algorithms with worst-case performance analyses. A different job processing cost, which is determined by the time slots used by the job, is considered by Wan and Qi [27] for single-machine scheduling. The non-classical objective function considered in this paper is different from the above ones with respect to the following three characteristics: (i) The package acquisition cost function is a piecewise linear function; (ii) the incurred cost is associated with packages rather than jobs; (iii) the acquisition cost for each package is a function of the schedule makespan in the package.

Since the proposed $Qm \parallel \alpha \sum w_jT_j + (1-\alpha)\sum \Psi_i(L_i)$ model is an extension of the strongly NP-hard problem $Qm \parallel \sum w_jT_j$, developing exact solution methods, such as branch-and-bound algorithm, can only solve small or medium-size problem instances. Considering the large-scale instances in practical application,