Grid Computing Systems and Combinatorial Optimization

Oleg Shcherbina[1] and Eugene Levner[2]

[1] University of Vienna, Vienna 1090, Austria,
Oleg.Shcherbina@univie.ac.at,
WWW home page: http://www.mat.univie.ac.at/∼oleg
[2] Holon Institute of Technology, Dept. of Computer Science,
Golomb St. 52 97502 Holon Israel

Abstract. The goal of this paper is to study the applicability of a combinatorial optimization model in grid resource optimization. The advent of grid computing and demand for QoS guarantees call for a need of advance reservation mechanisms in order to coordinate resource sharing between autonomous partners. This term means the guarantee of providing specific resources at a specific time. The paper assumes that grid resource sharing can greatly benefit from the application of combinatorial optimization for improving the advance reservation mechanisms. Specifically, the temporal knapsack problem is used for modeling the advance reservation. For improving the QoS guarantees and efficient advance reservation, a new methods based on dynamic programming and a decomposition of the temporal knapsack problem is developed.

Keywords: Grid computing, combinatorial optimization, temporal knapsack problem

1 Introduction

Grid computing is a form of parallel computing, whereby a network of loosely coupled computers act in concert to perform very large and complex tasks; these computers function together as a "virtual supercomputer". The tasks solved by grids need to process large amounts of data and require a great number of computer operations. The primary advantage of grid computing is that it can produce parallel computations similar to a multi-processor supercomputer, but at lower cost.

Grids are promising computing platforms that allow to aggregate distributed resources such as workstations and clusters to solve large-scale problems. The grid technology enables resource sharing and dynamic allocation of computational resources, thus increasing access to distributed data, promoting operational flexibility and collaboration, and allowing service providers to scale efficiently to meet variable demand.

With the emergence of grid technologies, the problem of scheduling tasks in heterogeneous systems has arisen. Resource management in highly dynamic grid environments is not only about scheduling large and compute-intensive applications, but also the manner in which resources are allocated, assigned, and accessed. In most systems, submitted jobs are initially placed into a queue if there are no available resources. Therefore, there is no guarantee as to when these jobs will be executed. This causes difficult problems in time-critical or parallel applications where jobs may have interdependencies.

In this paper we discuss discrete models of resource allocation in grids and focus on the issues related to enhancing their QoS using methods of combinatorial optimization.

2 Advance Reservation in Grid Computing

Unpredictable job execution environments pose a significant barrier to the widespread adoption of the grid paradigm, because of the innate risk of jobs failing to execute at the time specified by the user. Large-scale grids are complex systems composed of thousands of components from disjoined domains. Planning the capacity to guarantee *quality of service* (QoS) in such environments is a challenge because global *service-level agreements* (SLAs) depend on local SLAs. The advent of grid computing and demand for QoS guarantees call for a need of the advance reservation (AR) mechanisms in order to coordinate resource sharing between autonomous partners. In the most general sense, this term means the guarantee of providing specific resources at a specific time. AR in a scheduling system solves the above problem by allowing users to gain simultaneous and concurrent access to adequate resources for the execution of such applications. Currently, several Grid systems are able to provide AR functionalities, such as GARA and ICENI.

The goal of AR is to deliver higher levels of QoS. The advance reservation of grid resources can play a key role in enabling grid middleware to deliver on-demand resource provision with significantly improved QoS. Furthermore, advance reservation enables grid resource management to optimize different QoS parameters and enhance resource utilization with better capacity planning. However, in practice, concern about the AR has been insufficient mainly due to the dynamic grid behavior, under-utilization concerns, multi-constrained applications, and lack of support for agreement enforcement. Moreover, the support for advance reservations in grids has been restricted to processor reservations on clusters or bandwidth reservation on high speed LANs.

Whereas most of academic Grid schedulers such as GridWay (http://www.gridway.org) still lack the grid AR support, several commercial grid schedulers such as Moab Cluster Manager (http://www.clusterresources.com/pages/products.php) and LSF Multi-Cluster (http://www.platform.com/Products) can support Grid ARs.GridSim is a Grid AR simulation environment built in the context of Gridbus project (http://www.gridbus.org). GridSim can implement all Grid AR functionalities defined in the Grid AR framework except for advanced query functions.

To summarize, the AR in Grid computing is an important research area as it allows users to gain concurrent access to resources by allowing their applications to be executed in parallel. It also provides QoS guarantees on the availability of resources at the specified times in the future.

3 Combinatorial Optimisation for Task Scheduling in Grid Computing

Combinatorial optimisation approaches to task scheduling strategies for workflow-based applications in grids include using hypergraph representation, modeling advance resources reservation, taking the rationale of a grid resource manager that maximises its utility by choosing the optimal set of customers orders. Advance reservation will play a major role in Grid systems [1]. A resources manager manages some bounded resource, such as network bandwidth or computational nodes. Since the resources are limited per time unit, the provider may be unable to meet all demands, so the resources provider must choose which requests are to be accepted. The idea here is to maximise the utility of the resources provider by selecting an optimal subset of customers' requests.

Grid system receives the customers' priced requests and obtains the status of potential resources from the grid or directly from the resources manager. Grid system then computes an optimal selection of orders according to prices and penalties. This selection is forwarded to the manager who notifies customers of selection/rejection. Grid system acts as a particular web service used by the manager to optimize its business. There is no direct connection between customers and the grid system service. Customer orders use Start/End time; requested QoS, i.e., amount of resources per time unit; price proposed for the service, and penalty if QoS is not satisfied. This advance reservation will be modeled via integer programming as a temporal knapsack problem (TKP)[3] [2], where the (hard) constraints ensure that QoS is maintained given bounded resources. In the TKP a resource allocator is given bids for portions of a time-shared resource (e.g. CPU time or communication bandwidth) or a space-shared resource (such as computer memory, disk space etc.). Each bid specifies the amount of resource needed, the time interval throughout which it is needed, and a price offered for the resource. The resource allocator will, in general, have more demand than capacity, so it has the problem of selecting a subset of the bids that maximizes the total price or total utility obtained.

Here we consider optimal AR problems, where requests must be dynamically allocated to limited resources in order to maximize profit.

In the TKP a resource allocator is given bids $j = 1, \ldots, n$ for portions of a time shared resource – such as CPU time, or a shared space resource such as computer memory, disk space, or communication bandwidth. Each bid specifies the amount of resource q_j needed, time interval $[\alpha_{\mathrm{j}},\ \beta_{\mathrm{j}}]$ throughout which it is needed, and profit c_{j} obtained for the resource. At each time instant t, the resource allocator will, in general, have a larger amount of resource demands than capacity $b_t,\ t = 1, \ldots, T$, so it has the problem of selecting a subset of the bids that maximises the total profit obtained.

[3] This model named *advance reservation model* was earlier proposed in [6].

Let us assume that the total number of bids, n, is known. Let us introduce decision variables x_{j}, $j = 1, \ldots, n$: $x_j = 1$ if bid j is selected, $x_j = 0$ otherwise.

$$\sum_{j=1}^{n} c_j x_j \to \max$$

$$\sum_{j \in F_t} q_j x_j \le b_t,\ t = 1, \ldots, T,$$
$$x_j = 0,\ 1, \quad j = 1, \ldots, n$$

where $F_t = \{j : \alpha_j \le t \le \beta_j\}$. The constraint matrix of the above problem is a Petrie matrix [4].

To meet the challenge of solving large scale TKP problems in reasonable time, we propose to apply for solving the AR problems structural decomposition techniques developed in [17], exact routing methods with logical constraints [21] and highly efficient approximation methods with performance guarantees earlier developed in [15]. Structural decomposition algorithms compute global information using local computations (i.e., computations of information about elements of neighbourhood of variables or constraints – usually, solving subproblems)). Among structural decomposition techniques are known nonserial dynamic programming and its modifications: bucket elimination, tree decomposition method, hypertree and hinge decomposition.

References

1. Smith, W., Foster, I., and Taylor, V.: Scheduling with advanced reservations. In: Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'00). Cancun, Mexico, May 1-5, 2000 (2000)
2. Bartlett, M. et al.: The Temporal Knapsack Problem and its solution. In: Second International Conference on the Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR), pp. 34-48 (2005)
3. Mejia-Alvarez, P., Levner, E., Mosse, D.: Adaptive scheduling server for power-aware real-time tasks. ACM Transactions on Embedded Computing Systems. 3(2), 284-306 (2004)
4. Shcherbina, O.: Graph-based local elimination algorithms in discrete optimization. In: Foundations of Computational Intelligence Volume 3. Global Volume 3. Global Optimization Series: Studies in Computational Intelligence, Vol. 203, P. 235-266 / Abraham, A.; Hassanien, A.-E.; Siarry, P.; Engelbrecht, A. (Eds.). Springer Berlin / Heidelberg (2009)
5. Adelson-Velsky, G.M., Levner, E.: Finding extremal paths in AND-OR graphs: A generalization of Dijkstra's algorithms. Mathematics of Operations Research 27(3), 504-517 (2002)
6. Shcherbina, O.: A unimodular problem of integer programming, USSR Computational Mathematics and Mathematical Physics 26, Issue 4, 88-90 (1986)

[4] Recall that the Petrie matrix is a finite matrix whose elements are either zeros or ones such that the ones in each column occur consecutively.