Journal of Systems Engineering and Electronics
Vol. 18, No. 2, 2007, pp.412-417
Available online at www.sciencedirect.com
ScienceDirect
www.sciencedirect.com/science/journal/10044132

Multiple QoS modeling and algorithm in computational grid*

Li Chunlin, Feng Meilai & Li Layuan
Dept. of Computer Science, Wuhan Univ. of Technology, Wuhan 430063, P. R. China
(Received February 6, 2006)

Abstract: Multiple QoS modeling and algorithm in grid system is considered. Grid QoS requirements can be formulated as a utility function for each task as a weighted sum of its each dimensional QoS utility functions. Multiple QoS constraint resource scheduling optimization in computational grid is distributed to two subproblems: optimization of grid user and grid resource provider. Grid QoS scheduling can be achieved by solving sub problems via an iterative algorithm.

Keywords: QoS modeling, Computational grid, Scheduling algorithm.

1. Introduction

Scheduling diverse applications on heterogeneous, distributed, dynamic Grid computing systems is a difficult problem. The requirements of applications are diverse and so are the Grid resources in terms of processor capabil-ities, data capabilities, network capabilities, instruments, and other services. Scheduling applications with QoS requirement to the right set of grid resources can have a dramatic effect on the performance of a Grid application. There are already some research efforts in this area. R. Al-Ali et al. Reference [2] extended the service abstraction in the OGSA for Quality of Service (QoS) properties. The realization of QoS often requires mechanisms such as advance or on-demand reservation of resources, varying in type and implementation, and independently controlled and monitored. Foster, I et al.[3] described a General-purpose Architecture for Reservation and Allocation (GARA) that supports flow-specific QoS specification, immediate and advance reservation. In Ref. [4], Atakan Dogan et al. considered the problem of scheduling a set of independent tasks with multiple QoS requirements. Chen Lee et al.[5] used resource-utility functions in a QoS management framework with a goal to maximize the total utility of the system. In this article, we consider scheduling finite resources to satisfy the QoS needs of various grid users with multiple dimensional QoS requirements.

2. Multiple QoS modeling in computational grid

Grid QoS modeling is based on some assumptions that simplify the problem formulation. In our study, the network resource QoS is primarily concerned with the capacity of bandwidth. The computation resource QoS is concerned with the computation cost, and the computation deadline. In modeling the QoS requirements, each user agent is assumed to associate a number of QoS requirements with its task agent. Each q_i^l is a finite set of quality choices for the ith task agent's lth QoS dimension; let M denote the number of QoS requirements of task agent i. $q_i^1, q_i^2, \cdots, q_i^M$ is the QoS dimensions associated with task agent i. $q_i = [q_i^1 \cdots\cdots q_i^M]$ defines an M dimensional space of the QoS choices of task agent i. Associated with each QoS dimension is a utility function, which defines the user's benefit in choosing certain value of QoS choices in that dimension. Formally, the utility function associated with the lth QoS dimension of

* The project was supported by the National Natural Science Foundation of China (60402028, 60672137) Wuhan Yonger Dawning Foundation (20045006071-15) and China Specialized Research Fund for the Doctoral Program of Higher Eduction (20060497015).

task agent i is $U_i^l(q_i^l)$. One dimension utility functions can express the task agent's benefits in individual QoS dimensions, but grid resource scheduling system needs multi-dimensional QoS requirements to evaluate the overall benefits of the task agents. Multi-dimensional QoS requirements can be formulated as a utility function for each task as a weighted sum of its each dimensional QoS utility functions. The utility function associated with task agent i is denoted by $U_i(q_i)$. Our model is composed of three dimensions: cost, deadline, and reliability. Assume each grid task agent needs computation resource owned by various computation resource agents, and network resources to complete its jobs. Specifically, we assume task agent i can buy bandwidth y_i^k from network agent k, and buy computation resources x_i^j from computation resource agent j. If the network resource agent has a total bandwidth s_k available to task agents, then the bandwidth allocations must obey $s_k \geqslant \sum_i y_i^k$. c_j is the capacity of computation resource represented by computation resource agent j, and the corresponding resource allocation constraint is therefore $c_j \geqslant \sum_i x_i^j$. The completion time for grid task agent i to complete its nth job is $t_i^n = f(x_i^j, y_i^k, b_{in}, d_{in})$, where, b_{in} is the computation quantity of the ith grid task agent's nth job, and d_{in} is the transmission quantity of the ith grid task agent's nth job. We assume that each grid user i can place an upper bound on the total completion time by $T_i \geqslant \sum_n^N t_i^n$, where, N is the number of user's jobs. Grid task agents compete for computation resources and network resource with finite capacity. The resource is allocated through resource market, where the partitions depend on the relative payments sent by the task agents. We assume that each task agent i submits payment v_i^k to the network resource agent k and u_i^j to the computation agent j. Then, $v^k = [v_1^k \cdots\cdots v_N^k]$ represents all payments of task agents for the kth network resource agent. The utility function associated with first dimension QoS is $U_i^1(q_i^1)$, which is related to the cost. $\sum_j u_i^j$ is the total payment of the ith task agent paid to computation resources, and $\sum_k v_i^k$ is the total payment of the ith task agent paid to network resources. w_i^1 denotes the weight assigned to the first QoS dimension of task agent i. The utility function associated with second dimension QoS is $U_i^2(q_i^2)$, which is related to the completion time. The completion time for grid task agent i includes two parts: computation time and transmission time. T_i is an upper bound on the total completion time of each grid task agent i. D denotes the delay time. w_i^2 denotes the weight assigned to the first QoS dimension of task agent i. The utility function associated with second dimension QoS is $U_i^3(q_i^3)$, which is related to the completion reliability. g is the number of times that the task has been successfully completed within the deadline, and f is the total number of invocations.

To provide the grid resource scheduler with a unique utility function, which maps the multi-dimensional QoS needs of the task to a benefit value, we can define the utility function of task agent as a weighted sum of the single-dimensional QoS utility function:

$$U_i(q_i) = w_i^1\Big(E_i - \sum_j u_i^j - \sum_k v_i^k\Big) + \\ w_i^2\Big(T_i - \sum_{n=1}^{n} \frac{b_{in}}{x_i^j} - \sum_{n=1}^{N} \frac{d_{in}}{y_i^k} - D\Big) + w_i^3 \frac{g}{f}$$

We will use these utility functions to define an overall system utility function, which is as a weighted sum of each task agent's QoS utility function. The grid resource scheduler's objective is to assign qualities and allocate resources to task agents, such that the system utility $\sum_{i=1}^N \omega_i U_i(q_i)$ is maximized. We now formulate the problem of grid scheduling optimization in computational grid as the following constrained non-linear:

$$\max \sum_{i=1}^{N} \omega_i U_i(q_i)$$

$$\text{Subject to} \quad \begin{aligned} & c_j \geqslant \sum_i x_i^j, S_k \geqslant \sum_i y_i^k \\ & T_i \geqslant \sum_i t_i^n, E_i \geqslant \sum_j u_i^j + \sum_k v_i^k \\ & x_i^j > 0, y_i^k > 0 \end{aligned}$$

ω_i is the priority weight assigned to task agent i by the Grid. The grid resource scheduler finds a possible task assignment that maximizes system util-