Computing and Informatics, Vol. 26, 2007, 605–626

A DISTRIBUTED ITERATIVE ALGORITHM FOR OPTIMAL SCHEDULING IN GRID COMPUTING

Chunlin Li, Layuan Li

Department of Computer Science
Wuhan University of Technology
Wuhan 430063, P. R. China
e-mail: `chunlin74@tom.com, jwtu@public.wh.hb.cn`

Manuscript received 21 November 2005; revised 22 November 2006
Communicated by Marian Vajteršic

Abstract. The paper studies a distributed iterative algorithm for optimal scheduling in grid computing. Grid user's requirements are formulated as dimensions in a quality of service problem expressed as a market game played by grid resource agents and grid task agents. User benefits resulting from taking decisions regarding each Quality of Service dimension are described by separate utility functions. The total system quality of service utility is defined as a linear combination of the discrete form utility functions. The paper presents distributed algorithms to iteratively optimize task agents and resource agents functioning as sub-problems of the grid resource QoS scheduling optimization. Such constructed resource scheduling algorithm finds a multiple quality of service solution optimal for grid users, which fulfils some specified user preferences. The proposed pricing based distributed iterative algorithm has been evaluated by studying the effect of QoS factors on benefits of grid user utility, revenue of grid resource provider and execution success ratio.

Keywords: Scheduling, grid, iterative algorithm, simulation

1 INTRODUCTION

Qualities of service (QoS) and resource scheduling are hot issues in grid. QoS is a key technology that determines whether the grid can provide grid resources on demand efficiently. Enforcing QoS in the grid is complicated by the unpredictable grid characteristics and grid resource dynamic consumption. Two types of QoS are mainly studied in the grid infrastructure, which reflects quantitative and qualitative characteristics of the grid environment. Qualitative characteristics refer to elements such as service reliability and user satisfaction regarding service. Quantitative characteristics refer to elements such as networks, CPUs, or storage. The specification of the QoS requirements of grid applications should be described in a high-level manner. A good mechanism is needed to map the high-level requirements into low-level QoS parameters. These parameters specify the amount of resources to be allocated, such as amount of memory, and network bandwidth. Due to the highly dynamic grid environment, any attempt at QoS provisioning should be adaptive in nature. It is necessary to consider the changes in resource availability, network topology, and network bandwidth and latency, so that the grid can provide the best possible QoS to the application. In the grid environment, grid users may specify the tasks that should be performed at certain QoS level. When the grid scheduler receives a request from the user, the resources and QoS requirements are expressed in the request; through mapping, converting and negotiating the QoS parameters, the scheduler can implant the user's requirement about QoS in the process of resource scheduling. The grid scheduler contacts the grid resource provider to determine whether or not this request can be satisfied, given the current usage and allocation of resources.

The paper studies a pricing based distributed iterative algorithm for QoS scheduling on the grid. Grid user requirements are formulated as dimensions in a quality of service problem expressed as a market game played by grid resource agents and grid task agents. User benefits resulting from taking decisions regarding each Quality of Service dimension are described by separate utility functions. The total system quality of service utility is defined as a linear combination of the discrete form utility functions. Dynamic programming is used to optimize the total system utility function in terms of an iterative algorithm. The paper presents algorithms to iteratively optimize task agents and resource agents functioning as sub-problems of the quality of service grid resource scheduling optimization. Such constructed resource scheduling algorithm finds a multiple quality of service solution optimal for grid users, which fulfils some specified user preferences. The proposed pricing based distributed iterative algorithm has been evaluated by studying the effect of QoS factors on benefits of grid user utility, revenue of grid resource provider and execution success ratio.

The rest of the paper is structured as follows. Section 2 describes the related works. Section 3 presents distributed iterative algorithm for QoS scheduling on the grid. In Section 4 the experiments are presented and discussed. Section 5 concludes the paper.

2 RELATED WORKS

Recently, QoS provisioning for computational grid and its applications has received considerable attention. R. Buyya et al. [1] propose a deadline and budget constrained

(DBC) scheduling algorithm, which uses economy driven method for allocating resources to application jobs in such a way that the grid users' requirements are met. Chen Lee et al. [2, 3] use resource-utility functions in a QoS management framework with the goal to maximize the total utility of the system. They propose two approximation algorithms, and compare the run-times and solution quality with an optimal solution. In [4], Atakan Dogan et al. consider the problem of scheduling a set of independent tasks with multiple QoS requirements, which may include timeliness, reliability, security, version, and priority, in a grid computing system in which resource prices can vary with time during scheduling time intervals. Dong Su Nam et al. [5] propose a quorum based resource management scheme; the resource quorum includes middleware entity and network entity, both can satisfy requirements of application QoS. They suggest a heuristic configuration algorithm in order to optimize performance and usage cost of the resource quorum. R. Al-Ali et al. [6–8] extend 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. Yutu Liu et al. [10] presented an open, fair and dynamic QoS computation model for web services selection through implementation of and experimentation with a QoS registry in a hypothetical phone service provisioning market place application. Kavitha S. Golconda et al. [11] compare five QoS-based scheduling heuristics from the literature, in terms of three performance parameters, namely the number of satisfied users, makespan and total utility of the meta-task. Tarek F. Abdelzaher et al. [12] propose, implement, and evaluate a novel communication server architecture that maximizes the aggregate utility of QoS-sensitive connections for a community of clients even in the case of overload. Liangzhao Zeng et al. [13] presents a middleware platform that addresses the issue of selecting web services for the purpose of their composition in a way that maximizes user satisfaction expressed as utility functions over QoS attributes. In [14], the paper presents economic agent based grid resource management. A system model is described that allows agents representing various grid resources and grid users to interact without assuming a priori cooperation. In [15], the paper presents an Agent-based Grid Service Management, which applies the concept of agents to computational grid. In [16], the paper designed and implemented a mobile agent platform based on tuple space coordination. In [17], the paper provides a price-directed proportional resource allocation algorithm for solving the grid task agent resource allocation problem. In [18], a distributed utility-based two level market solution for optimal resource scheduling in computational grid is presented. In [19], the paper is to implement a uniform higher-level management of the computing resources and services on the Grid, and to provide users with a consistent and transparent interface for accessing such services. In [22], the paper integrates software agents and CORBA to allocate resource in computational grid. Li Layuan et al. [20, 21] discuss the multicast routing problem with QoS constraints such as delay, delay jitter, bandwidth and packet loss metrics, and describe a network model that is suitable to search such routing problem, and presents a QoS-guaranteed multicast routing protocol (QGMRP). The research objective of this paper is to study multiple QoS constrained grid resource scheduling, which is not the same as in the above works [14–22, 26].

3 OPTIMAL SCHEDULING ON THE GRID: MATHEMATICAL MODEL AND ALGORITHMS

3.1 Mathematical Model

The dynamics, autonomy and heterogeneity of grid system are considered and solved in our proposed QoS scheduling optimization model. Since the dynamics of a grid computing system are difficult to model, the grid system is modeled as a market economy and hence the past research in the field of economics can be put to good use here. The various resources in the grid system (e.g. CPU, bandwidth etc.) are modeled as hypothetical resource producers, who sell their resources to hypothetical resource consumers. The incoming job requests of the user come with a budget, which is a measure of the grid user perceived value of the job. They may have to wait longer, or are starved if prices of resources are too high for their budget. Grid schedulers base their decisions on the state of economy as suggested by the price of different resources. The state of equilibrium in economy is when the demand for the resource is the same as the supply of it. The price of a resource, as in a real market, reflects its relative worth only if the economy is in a state of equilibrium. Therefore if the market cannot be brought to equilibrium, decisions will be poor. The proposed grid can be modeled as multi economic agents. Both users and resources can be viewed as autonomous agents, having control of their own behavior. This autonomy gives rise to inherent uncertainty, since an individual cannot predict how another one will respond to changing situations. Whenever a new grid task agent is created, it is first given an endowment of electronic cash to spend to complete its task. Before a job can be executed on the computational grid, some attributes have to be set properly. A job can be characterized by time limit, budget, and data size and runtime requirements. We assume that when a task agent purchases a portion of resources owned by the resource agents, it is guaranteed that the task agent continues to receive resource without interruption from the resource agent until its task is completed.

Assume each q_i^l is a finite set of quality choices for the i^{th} task agent's l^{th} QoS dimension; let M denote the number of QoS requirements of task agent i. $q_i^1, q_i^2, \ldots, q_i^M$ is the QoS dimensions associated with task agent i. $q_i = [q_i^1, \ldots, q_i^M]$ defines an M-dimensional space of the QoS choices of task agent i. Associated with each QoS dimension is a utility function, which defines user's benefit in choosing certain value of QoS choices in that dimension. Formally, the utility function associated with the l^{th} QoS dimension of task agent i is $U_i^l(q_i^l)$. One dimension utility function can express task agent' benefits in individual QoS dimensions, but grid resource scheduling system needs multi-dimensional QoS requirements to evaluate 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 $U_i(q_i)$, the function $U_i(q_i)$ can be defined as a weighted sum of $U_i^l(q_i^l)$. We have constructed a QoS model that includes system and process categories. Our model is composed of three dimensions: cost, deadline, and reliability. Cost (C) represents the cost associated with the execution of grid tasks. Task cost is the cost incurred when a task t is executed; it can be broken down into two parts, which include computation resource cost and bandwidth resource cost. Deadline (D) is a common and universal measure of performance. Task deadline corresponds to the overall time a task is processed in the grid. The task deadline can be broken down into two parts that include process time and delay time. The task reliability (R) is defined to be the probability that the task can be completed successfully. Each user may specify a degree of reliability that is acceptable for its task, in order to minimize the adverse effects of failures. Cost (C), Deadline (D), Reliability (R)are considered as the QoS dimensions of a task. As a result, the QoS model of task agent i can be formulated as $q_i = [C, D, R]$. 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 $s_k \geq \sum_i y_i^k$. c_j is the capacity of computation resource represented by computation resource agent j, the corresponding resource allocation constraint is therefore $c_j \geq \sum_i x_i^j$. The completion time for grid task agent i to complete its n^{th} job is $t_i^n = f(x_i^j, y_i^k, b_{in}, d_{in})$ where b_{in} is the size of computation data of the i^{th} grid task agent's n^{th} job, d_{in} is the amount of transferring data of the i^{th} grid task agent's n^{th} job. We assume that each grid user i can place an upper bound on the total completion time by $T_i \geq \sum_n^N t_i^n$ 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 computation agent j. Then, $v^k = [v_1^k, \ldots, v_N^k]$ represents all payments of task agents for the k^{th} network resource agent.

Let us consider the utility function associated with three dimensions QoS of the task agent. The utility function associated with first dimension QoS is $U_i^1(q_i^1)$, which is related with the cost. In $U_i^1(q_i^1)$, $\sum_j u_i^j$ is the total payment of the i^{th} task agent paid to computation resources, $\sum_k v_i^k$ is the total payment of the i^{th} 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 with the completion time. In $U_i^2(q_i^2)$, 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 second QoS dimension of task agent i. The utility function associated with third dimension QoS is $U_i^3(q_i^3)$, which is related with the completion reliability. In $U_i^3(q_i^3)$, g is the number of times that the task has been successfully completed within the deadline, and f is the total number of invocations. w_i^3 denotes the weight assigned to the third QoS dimension of task agent i.

$$U_i^1\left(q_i^1\right) = w_i^1\left(E_i - \sum_j u_i^j - \sum_k v_i^k\right)$$

$$U_i^2\left(q_i^2\right) = w_i^2\left(T_i - \sum_{n=1}^{N}\frac{b_{in}}{x_i^j} - \sum_{n=1}^{N}\frac{d_{in}}{y_i^k} - D\right)$$

$$U_i^3\left(q_i^3\right) = w_i^3\frac{g}{f}$$

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 single-dimensional QoS utility function:

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

Each task agent has a utility function that measures the value it puts on quality assignments. The overall system's QoS utility is a linear combination of $U_i(q_i)$. 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:

$$U_{system} = \sum_{i=1}^{N}\omega_i U_i(q_i).$$

Grid resource scheduler' objective is to assign qualities and allocate resources to task agents, such that the system utility U_{system} is maximized.

We now formulate the problem of grid scheduling optimization in computational grid as the following constrained non-linear optimization problem. In U_{system}, ω_i is the priority weight assigned to task agent i by the Grid. Grid resource scheduler finds a possible task assignment that maximizes U_{system} subject to users' QoS constraints. Computation resource units are allocated to task agent i by x_i^j that computation resource agent j allocates, and y_i^k is the network resource obtained by grid task agent i from network resource agent k. The QoS constraint implies that the aggregate network resource units do not exceed the total capacity of resource s_k, aggregate computation resource units do not exceed the total resource c_j, and grid task agent should complete all its jobs under time limits. Grid task agent needs to complete a sequence of jobs in a specified amount of time, T_i, while the cost overhead accrued cannot exceed the budget E_i.

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

$$c_j \geq \sum_i x_i^j, S_k \geq \sum_i y_i^k$$

Subject to:

$$T_i \geq \sum_i t_i^n, E_i \geq \sum_j u_i^j + \sum_k v_i^k$$

$$x_i^j > 0, y_i^k > 0$$

We can apply Lagrangian method to solve such a problem [23–25]. Let us consider the Lagrangian form of this optimization problem:

$$\begin{aligned} L\left(\lambda_i, \beta_i, \varphi_i, \gamma_i\right) = \sum_i U - \lambda_i \left(\sum_i y_i^k - S_k\right) - \beta_i \left(\sum_i x_i^j - c_j\right) \\ - \varphi_i \left(\sum_j u_i^j + \sum_k v_i^k - E_i\right) - \gamma_i \left(\sum_i t_i^n - T_i\right) \end{aligned}$$

where λ_i, β_i, γ_i is the Lagrangian multiplier of grid task agent i. Thus, given that the grid knows the utility functions U of all the grid task agents, this optimization problem can be mathematically tractable. However, in practice, it is not likely to know all the U, and it is also infeasible for computational grid to compute and allocate resources in a centralized fashion. Solving the objective function $\mathrm{Max} \sum_{i=1}^{N} \omega_i U_i(q_i)$ requires global coordination of all grid users, which is impractical in distributed environment such as the computational grid. In order to achieve a distributed solution, we decompose the problem into the following two sub-problems; seek a distributed solution where the grid provider does not need to know the utility functions of individual grid user. For a completed time, the task agent optimization problem $\mathrm{Max} U_i(q_i)$ can be written as follows:

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

In resource market, computation resource agent and network resource agent acted as suppliers to maximize their benefits. The grid resource agent, given the amounts that the grid task agents are willing to pay, attempts to maximize the function $\sum \left(u_i^j \log x_i^j + v_i^k \log y_i^j\right)$. So the grid resource provider's optimization problem can be formulated as follows. y_i^k is the network resource sold to the task agent i by network resource agent k, x_i^j is the computation resource sold to task agent i by computation resource agent j. $\sum \left(u_i^j \log x_i^j + v_i^k \log y_i^j\right)$ presents the revenue obtained by computation resource agent j and network resource agent k from the task agents. Computation agent or network agent cannot sell the resources to task agent more than c_j or s_k, which is the upper limit of resource presented by resource agents.

$$\mathrm{Max} \sum \left(u_i^j \log x_i^j + v_i^k \log y_i^j\right)$$

$$\text{s.t. } c_i \geq \sum_j x_i^j, s_k \geq \sum_k y_i^k$$

QoS constraint resource scheduling optimization in computational grid is distributed to two subproblems: optimization of task agent and resource agent in resource market.

Firstly, consider task agent's optimization.

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

$$\text{s.t. } T_i \geq \sum_i t_i^n$$

We assume that each task agent submits u_i^j to the computational resource agent and v_i^k to network resource agent. Then, $u_i = [u_i^1, \ldots, u_i^j]$ represents all payments of grid task agents for the j^{th} computation resource agent, $v_i = [v_i^1, \ldots, v_i^k]$ represents all payments of grid task agents for the k^{th} network resource agent. Let $m_i = \sum_j u_i^j + \sum_k v_i^k$, m_i be the total payment of the i^{th} task agent. N grid task agents compete for grid resources with finite capacity. The resource is allocated using a market mechanism, where the partitions depend on the relative payments sent by the grid task agents. Let pc_j, pn_k denote the price of the resource unit of computation resource agent j and network resource agent k, respectively. Let the pricing policy, $pc = (pc_1, pc_2, \ldots, pc_n)$, denote the set of computational resource unit prices of all the computation resource agents in the grid, $pn = (pn_1, pn_2, \ldots, pn_k)$ is set of network resource unit prices. The i^{th} task agent receives resources proportional to its payment relative to the sum of the resource agent's revenue. Let x_i^j, y_i^k be the fraction of resource units allocated to task agent i by computation resource agent j and network resource agent k.

The task agent's sub-problem can be reformulated as

$$\begin{aligned} \text{Max} U_{task} &= w_i^1 \left(E_i - \sum_j u_i^j - \sum_k v_i^k \right) \\ &\quad + w_i^2 \left(T_i - \sum_{n=1}^{N} \frac{b_{in} pc_j}{c_j u_i^j} - \sum_{n=1}^{N} \frac{d_{in} pn_k}{s_k v_i^k} - D \right) + w_i^3 \frac{g}{f}. \end{aligned}$$

The Lagrangian for the task agent's utility is $L(u, v)$.

$$\begin{aligned} L\left(u_i^j, v_i^k\right) &= w_i^1 \left(E_i - \sum_j u_i^j - \sum_k v_i^k \right) \\ &\quad + w_i^2 \left(T_i - \sum_{n=1}^{N} \frac{b_{in} pc_j}{c_j u_i^j} - \sum_{n=1}^{N} \frac{d_{in} pn_k}{s_k v_i^k} - D \right) + w_i^3 \frac{g}{f} \end{aligned}$$

$$+ \lambda \left(T_i - \sum_{i=1}^{N} t_i^n \right)$$

where λ is the Lagrangian constant. From Karush-Kuhn-Tucker Theorem [9] we know that the optimal solution is given $\frac{\partial L(u,v)}{\partial u} = 0$ for $\lambda > 0$.

$$\frac{\partial L(u_i^j, v_i^k)}{\partial u_i^j} = -w_i^1 + w_i^2 \frac{b_{in} p c_j}{c_j (u_i^j)^2} + \lambda \frac{b_{in} p c_j}{c_j (u_i^j)^2}$$

Let $\frac{\partial L\left(u_i^j, v_i^k\right)}{\partial u_i^j} = 0$ to obtain

$$u_i^j = \left(\frac{(w_i^2 + \lambda)\, b_{in} p c_j}{w_i^1 c_j} \right)^{1/2}.$$

Using this result in the constraint equation, we can determine $\theta = \frac{w_i^2 + \lambda}{w_i^1}$ as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{m=1}^{N} \left(\frac{m p c b_{im}}{c_m} \right)^{1/2}}.$$

We substitute θ to obtain $u_i^{j^*}$

$$u_i^{j^*} = \left(\frac{b_{in} p c_j}{c_j} \right)^{1/2} \frac{\sum_{m=1}^{N} \left(\frac{b_{im} p c_m}{c_m} \right)^{1/2}}{T_i}.$$

$u_i^{j^*}$ is the unique optimal solution to the optimization problem *task agent*. It means that grid task agent wants to pay $u_i^{j^*}$ to computation resource agent j for needed resource under completion time constraint.

Using the similar method, let $\frac{\partial L\left(u_i^j, v_i^k\right)}{\partial v_i^k} = 0$

$$\frac{\partial L\left(u_i^j, v_i^k\right)}{\partial v_i^k} = -w_i^1 + w_i^2 \frac{d_{in} p n_k}{s_k (ikv)^2} + \lambda \frac{d_{in} p n_k}{s_k (ikv)^2} = 0.$$

We can get

$$v_i^k = \left(\frac{(w_i^2 + \lambda) d_{in} p n_k}{w_i^1 s_k} \right)^{1/2}.$$

Using this result in the constraint equation, we can determine $\theta = \frac{w_i^2 + \lambda}{w_i^1}$ as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{m=1}^{N} \left(\frac{m p n d_{im}}{s_m} \right)^{1/2}}$$

We obtain $v_i^{k^*}$

$$v_i^{k^*} = \left(\frac{d_{in} p n_k}{s_k}\right)^{1/2} \frac{\sum_{m=1}^{N} \left(\frac{d_{im} p n_m}{s_m}\right)^{1/2}}{T_i}.$$

It means that grid task agent wants to pay $v_i^{k^*}$ to network resource agent k for needed resource under completion time constraint.

Resource agent's optimization is solved as follows:

$$\text{Max} \sum \left(u_i^j \log x_i^j + v_i^k \log y_i^j\right)$$

$$\text{s. t. } \ c_i \geq \sum_j x_i^j, s_k \geq \sum_k y_i^k$$

$$U_{resource}\left(x_i^j, y_i^k\right) = \sum \left(u_i^j \log x_i^j + v_i^k \log y_i^j\right).$$

The Lagrangian for resource agent subproblem is $L(x, y)$

$$L\left(x_i^j, y_i^k\right) = \sum \left(u_i^j \log x_i^j + v_i^k \log y_i^k\right) + \lambda \left(c_j - \sum_i x_i^j\right) + \beta \left(s_k - \sum_i y_i^k\right)$$

$$= \sum \left(u_i^j \log x_i^j + v_i^k \log y_i^k - \lambda x_i^j - \beta y_i^k\right) + \lambda c_j + \beta s_k$$

where λ, β is the Lagrangian constant. From Karush-Kuhn-Tucker Theorem we know that the optimal solution is given $\frac{\partial L(x,y)}{\partial x} = 0$ for $\lambda > 0$.

$$\frac{\partial L\left(x_i^j, y_i^k\right)}{\partial x_i^j} = \frac{u_i^j}{x_i^j} - \lambda$$

Let $\frac{\partial L(x,y)}{\partial x} = 0$ to obtain

$$x_i^j = \frac{u_i^j}{\lambda}.$$

Using this result in the constraint equation $c_j \geq \sum_j x_i^j$, we can determine λ as

$$\lambda = \frac{\sum_{m=1}^{n} u_m^j}{c_j}.$$

We substitute λ to obtain $x_i^{j^*}$

$$x_i^{j^*} = \frac{u_i^j c_j}{\sum_{k=1}^{n} u_k^i};$$

$x_i^{j^*}$ is the unique optimal solution to the optimization problem of computation resource agent.

Let us consider network resource agent's optimization problem, using the similar method.

Let $\frac{\partial L(x,y)}{\partial y} = 0$

$$\frac{\partial L\left(x_i^j, y_i^k\right)}{\partial y_i^k} = \frac{v_i^k}{y_i^k} - \beta = 0.$$

We can get

$$iky = \frac{v_i^k}{\beta}.$$

Using this result in the constraint equation $s_k \geq \sum_k y_i^k$, we can determine β as

$$\beta = \frac{\sum_{m=1}^n v_m^k}{s_k}.$$

We obtain $y_i^{k^*}$

$$y_i^{k^*} = \frac{v_i^k s_k}{\sum_{m=1}^n v_m^k};$$

$y_i^{k^*}$ is the unique optimal solution to the optimization problem of network resource agent. It means that network resource agent acting as provider wants to allocate $y_i^{k^*}$ to grid task agent to maximize its revenue.

3.2 Distributed Iterative Algorithm for Optimal Scheduling

Optimal scheduling algorithm uses dynamic programming to optimize the total system utility function in terms of an iterative algorithm. The paper presents algorithms to iteratively optimize task agents and resource agents utility functions as sub-problems of the quality of service for grid resource scheduling optimization. Such constructed grid QoS scheduling algorithm finds a multiple quality of service solution optimal for grid users, which fulfils some specified user preferences. In each iteration, the task agent individually computes its optimal payment for grid resource agents, adjusts its computation resource demand and network resources demand and notifies the grid about this change. After the new computation resource and network resource demand are observed by the computation resource agent and network resource agent respectively, they update their prices accordingly and communicate the new prices to the grid task agent, and the cycle repeats. The distributed iterative algorithm that implements QoS scheduling is then given as follows.

Grid task agent
Receives the price pc_j from the computation resource agent j;
$u_i^{j^*} = \mathrm{Max}U\left(u_i^{j^*}\right)$; // calculates $u_i^{j^*}$ to maximize $U(u_i^j)$
If $E_i \geq \sum_j u_i^j + \sum_k v_i^k$
Then $x_i^j(n+1) = \frac{(n)u_i^{j^*}}{pc_j^{(n)}}$; // Calculates its optimal computation resource demand $x_i^j(n+1)$
Return $x_i^{j(n+1)}$ to computation resource agents;
Else Return Null;
Receives the price pn_k from the network resource agent k;
$v_i^{k^*} = \mathrm{Max}U(v_i^{k^*})$; // Calculates $v_i^{k^*}$ to maximize $U(v_i^k)$
If $E_i \geq \sum_j u_i^j + \sum_k v_i^k$
Then $y_i^k(n+1) = \frac{(n)v_i^{k^*}}{pn_k^{(n)}}$; // Calculates its optimal network resource demand $y_i^k(n+1)$
Return $y_i^{k(n+1)}$ to network resource agents;
Else Return Null;

Grid resource agent
Receives grid computation demand x_i^j, y_i^k from grid task agents;
If $c_i \geq \sum_j x_i^j$
Then
$pc_j^{(n+1)} = \max \epsilon, pc_j^{(n)} + \eta(x^j pc_j^{(n)} - c_j)$; // Computes a new price
// $x^j = \sum_i x_i^j$, $\eta > 0$ is a small step size parameter, n is iteration number.
Return new price $pc_j^{(n+1)}$ to all grid task agents;
Else Return Null;
If $s_k \geq \sum_i y_i^k$
Then
$pn_k^{(n+1)} = \max \epsilon, pn_k^{(n)} + \eta(y^k pn_k^{(n)} - s_k)$; // Computes a new price
// $y_k = \sum_i y_i^k$, $\eta > 0$ is a small step size parameter, n is iteration number.
Return new price $pn_k^{(n+1)}$ to all grid task agents;
Else Return Null;

4 SIMULATION STUDY

To evaluate the performance of distributed iterative QoS scheduling algorithm, a series of experiments are conducted to study the effect of QoS factors such as task deadline, price, reliability, budget and payment on benefits of grid user utility, revenue of grid resource provider and execution success ratio. The overview of the system environment is given in Table 1. The descriptions of grid users and grid resource providers are listed in Tables 2 and 3. Network generator BRITE generates the computer network topology. In order to simulate the dynamics and heterogeneity of the grid, all values of networks can be changed after topology generation. The simulated grid was defined to simulate a WAN consisting of 10 LANs. Nodes of each LAN range from 10 to 50. Therefore, there are a total of 500 nodes in the simulation. The bandwidth between WAN nodes was defined to be 1 Gbps, and the bandwidth between LAN nodes was set to be 100 Mbps to 1 Gbps. All tasks had different execution times. In the simulation setup, the task execution time ranged from 50 ms to 2 000 ms. Task arrival for scheduling followed an exponential distribution. As the number of submitted tasks increased, the computing load on the entire network increased as well. In the simulation, different numbers of tasks were tested in order to observe the impact on the grid scheduling performance; totally, there were 500 tasks. The resource cost can be expressed in grid dollars that can be defined as processing cost per MIPS. Processor capacity varies from 220 to 580 MIPS. Initial price of computing power is from 10 to 500 grid dollars.

Parameter	Value
Reschedule Interval	600
Number of tasks	500
Arrival time (ms)	200
System load	0.3
Request interval	100
Task agent number	300

Table 1. Simulation parameters

Grid users	1	2	3	4	5	6	7	8
Deadline (ms)	100	200	300	400	100	300	200	300
Budget	1000	1 500	2 000	500	1 500	1 000	500	1 500

Table 2. Description of the grid users

Grid resource provider	1	2	3	4
Processor Capability (MIPS)	370	370–380	220	510–580
Unit Price (grid dollar)	300∼500	200∼500	10∼100	100∼500

Grid resource provider	5	6	7	8
Processor Capability (MIPS)	340–390	370	510	220
Unit Price (grid dollar)	20∼200	200	300	100

Table 3. Description of the grid resource provider

First, the experiments aimed at evaluating the effect of the QoS metrics such as reliability, price, task deadline, and budget on the revenue of resource provider.

In Figure 1, when reliability of the grid resource is high, the revenue of the grid resource provider is also high, because grid user tends to choose the service with high reliability, and pay more for resource provider with high reliability. Figure 2 represents the effect of the price on the revenue. The maximum of the curve is the optimal revenue point for the resource provider. The highest value of the revenue is determined by both acceptable price and suitable grid resource allocation. Figure 3 represents the effect of task budget on the revenue of the resource provider earned from the grid users. When task budget is high, the revenue of the resource provider is also high, because grid user tends to choose expensive resource, and pay more for resource provider with high performance; then grid resource providers can achieve high revenue from grid users. Figure 4 shows the effect of task deadline on the revenue. The revenue increases first, then decreases as deadline increases. When grid users have high deadline, they can choose cheaper grid resources to complete tasks, so the resource providers will get less revenue from grid users. Figure 5 shows that increasing the payment leads to higher revenue for grid resource providers.

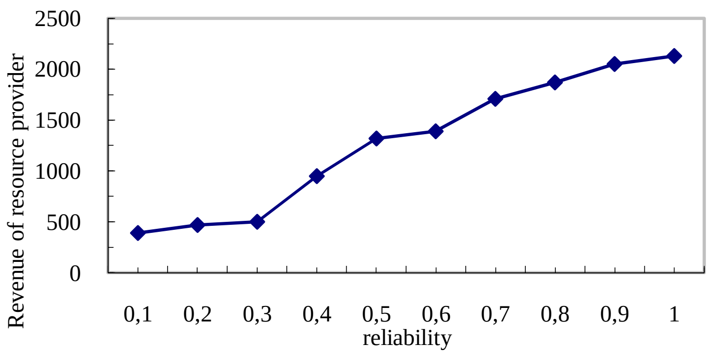

Fig. 1. Revenue of resource provider vs. reliability

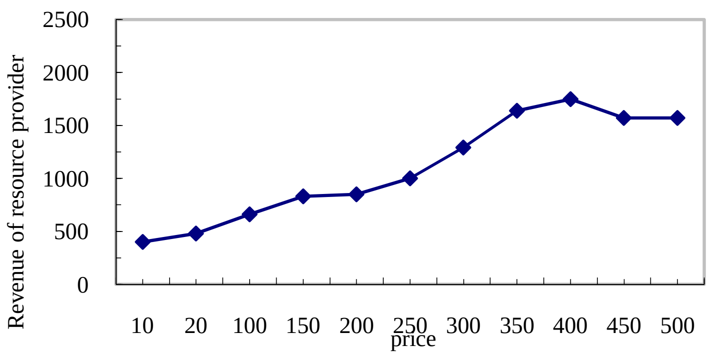

Fig. 2. Revenue of resource provider vs. price

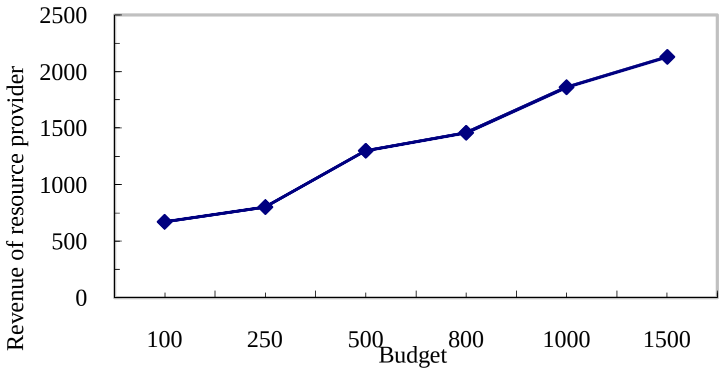

Fig. 3. Revenue of resource provider vs. budget

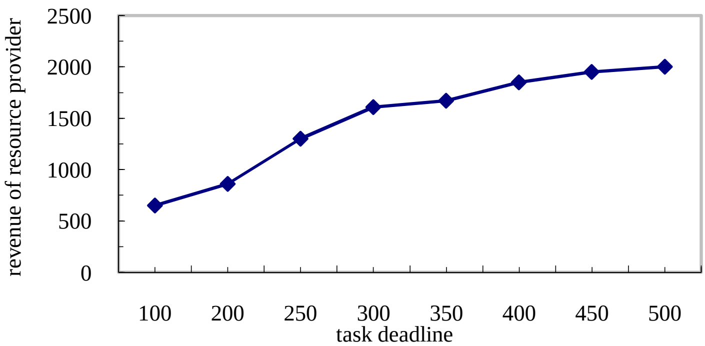

Fig. 4. Revenue of resource provider vs. task deadline

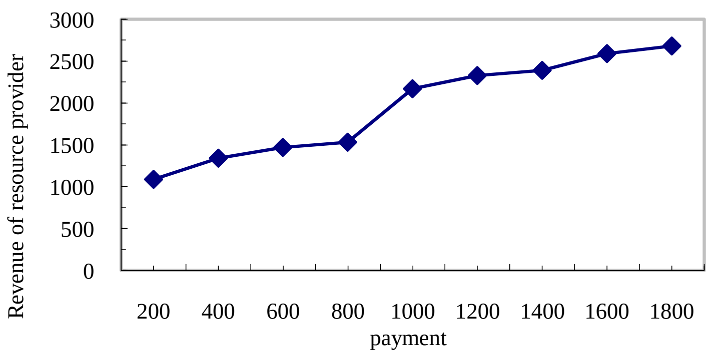

Fig. 5. Revenue of resource provider vs. payment

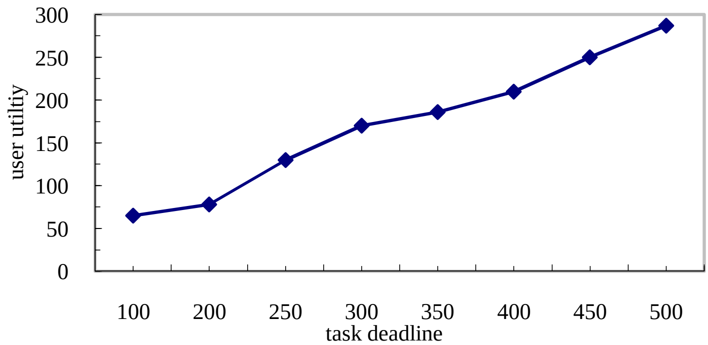

Fig. 6. User utility vs. task deadline

Second, the experiments aimed at evaluating the effect of the QoS metrics such as task deadline, payment, price, reliability, and budget on the grid user utility. From the results in Figure 6, when the deadline is low, there is intensive demand for the resources in a short time, so grid user chooses more expensive resources to process the tasks. However, when the deadline changes to higher one, it is likely that tasks can be completed before deadline, so grid user considers using the cheaper resources to complete tasks to maximize the utility. Figure 7 shows that increasing the payment leads to lower user utility. Figure 8 shows that when reliability increases, the user utility increases quickly. In Figure 9, the X-axis shows changes in resource price values, price value varies from 50 to 500 grid dollars. From the result of Figure 9, the utility of grid users becomes lower as the price values increase. Because the price increases, users will pay more to get grid resources, and some user with low budget cannot afford payment to get the needed grid resource. Figure 10 represents the impact of different budget constraint on the user utility. When the budget is small, user utility is low, because user cannot buy expensive and efficient resources to complete task. When the budget increases, user utility grows quickly, because users can afford more expensive resources. So most users like to choose proper resources to achieve their goals. Larger budgets enable grid users to afford more expensive resource to maximize the user utility.

Third, the experiments aimed at evaluating the effect of task deadline, reliability, number of task and budget on the execution success ratio. In Figure 11, The X-axis shows changes in task number values, task number value varies from 10 to 300. It can be observed from Figure 11, as the number of tasks increased, the grid scheduling performance worsened. This is easily understood. As more tasks were submitted, less computational resources were available for the tasks to share. This was reflected by a decrease in the percentage of tasks that met their deadlines. Figure 12 shows the effect of task deadline on execution success ratio. When the task deadline is low, execution success ratio is low. When increasing deadline, execution success ratio becomes higher. Figure 13 shows the effect of budget on execution success ratio. When increasing budget values, the execution success ratio becomes higher.

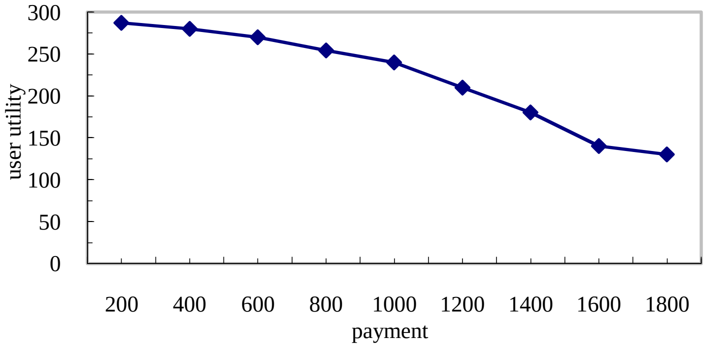

Fig. 7. User utility vs. payment

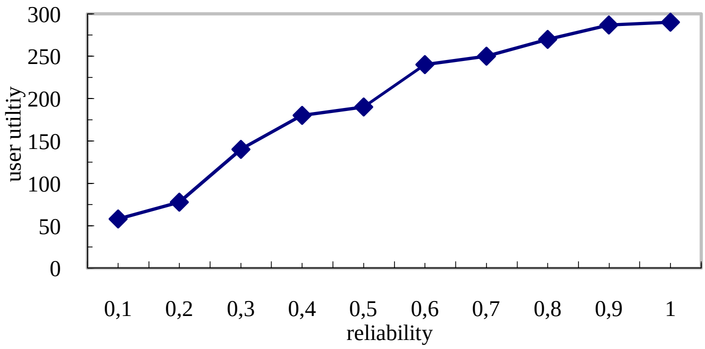

Fig. 8. User utility vs. reliability

A larger budget enables grid user to afford more expensive resources to complete the task before its deadline. From the results in Figure 14 when the reliability of grid resource is high, execution success ratio is also high. Since the reliability of grid resource is high, the resources selected can guarantee being available when the grid task needs to be executed, user will complete all tasks before its deadline.

5 CONCLUSIONS

The paper presents grid QoS scheduling, based on a mathematical QoS model and a distributed iterative algorithm. Grid user requirements are formulated as dimensions in a quality of service problem expressed as a market game played by grid resource agents and grid task agents. User benefits resulting from taking decisions regarding each Quality of Service dimension are described by separate utility functions. The total system quality of service utility is defined as a linear combination of the discrete form utility functions. Dynamic programming is used to optimize the total system utility function in terms of an iterative algorithm. The paper presents

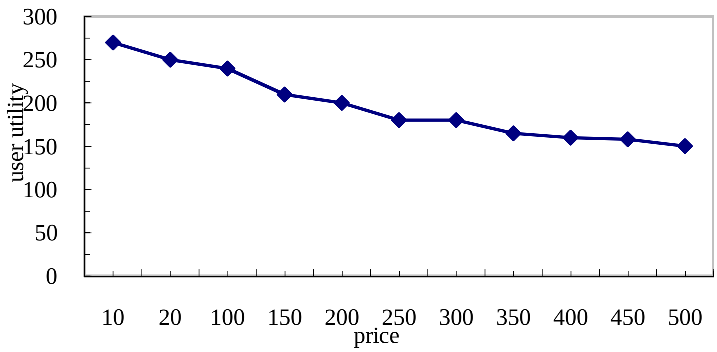

Fig. 9. User utility vs. price

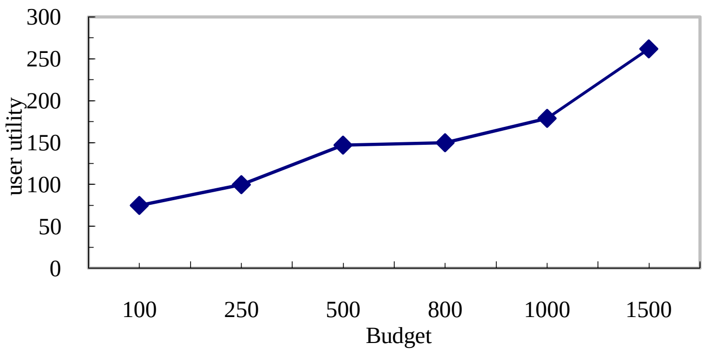

Fig. 10. User utility vs. budget

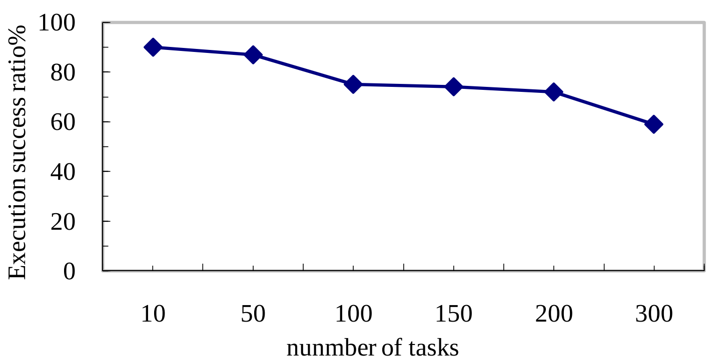

Fig. 11. Execution success ratio vs. number of task

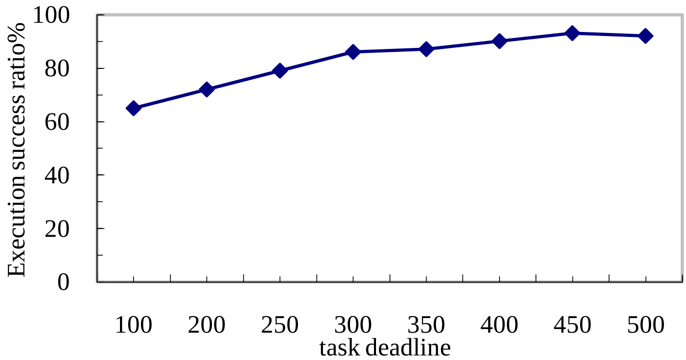

Fig. 12. Execution success ratio vs. task deadline

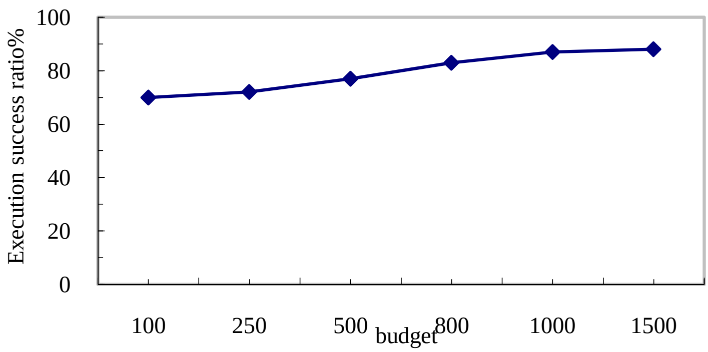

Fig. 13. Execution success ratio vs. budget

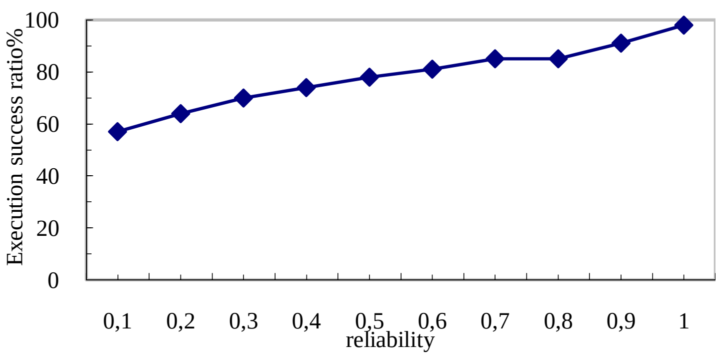

Fig. 14. Execution success ratio vs. reliability

algorithms to iteratively optimize task agents and resource agents functioning as sub-problems of the quality of service grid resource scheduling optimization. Such constructed resource scheduling algorithm finds a multiple quality of service solution optimal for grid users, which fulfils some specified user preferences. The proposed pricing based distributed iterative algorithm has been evaluated by studying the effect of QoS factors on benefits of grid user utility, revenue of grid resource provider and execution success ratio. In the future, we will consider more QoS metrics such as availability that is defined as the fraction of time that resource is available for use, and achieve QoS global optimization. More experiments are conducted to compare the performance of the proposed distributed algorithms with more related works.

Acknowledgements

The work is supported by National Science Foundation of China under grants No. 60402028 and 60773211, Wuhan Younger Dawning Foundation under grant No. 20045006071-15.

REFERENCES

[1] Abramson, D.—Buyya, R.—Giddy, J.: A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker. Future Generation Computer Systems, Vol. 18, 2002, No. 8, pp. 1061–1074, Elsevier Science, The Netherlands, October 2002.

[2] Chen Lee—Lehoczky, J.—Siewiorek, D.—Rajkumar, R.—Hansen, J.: A Scalable Solution to the Multi-Resource QoS Problem. 20^{th} IEEE Real-Time Systems Symposium, December 1999.

[3] Chen Lee—Lehoczky J.—Rajkumar, R.—Siewiorek, D.: On Quality of Service Optimization with Discrete QoS Options. In Proceedings of the IEEE Real-time Technology and Applications Symposium, June 1999.

[4] Dogan, A.—Ozguner, F.: Scheduling Independent Tasks with QoS Requirements in Grid Computing with Time-Varying Resource Prices. In Proceedings of GRID 2002, Springer-Verlag, LNCS 2536, 2002, pp. 58–69.

[5] Dong Su Nam—Chan-Hyun Youn: QoS-Constrained Resource Allocation for a Grid-Based Multiple Source Electrocardiogram Application. ICCSA 2004, LNCS 3043, 2004, pp. 352–359.

[6] Al-Ali, R.—Hafid, A.—Rana, O.—Walker, D.: QoS Adaptation in Service-Oriented Grids. In Proceedings of the 1^{st} International Workshop on Middleware for Grid Computing (MGC2003) at ACM Middleware 2003, Brazil 2003.

[7] Al-Ali, R.—Shaikhali, A.—Rana, O.—Walker, D.: Supporting QoS-Based Discovery in Service-Oriented Grids. In Proceedings of IEEE Heterogeneous Computing Workshop (HCW '03), Nice, France 2003.

[8] Al-Ali, R.—Rana, O.—Walker, D.—Jha, S.—Sohail, S.: G-QoSM: Grid Service Discovery Using QoS Properties. Computing and Informatics, Special Issue on Grid Computing, Vol. 21, 2002, No. 4, pp. 363–382.

[9] Kuhn, H. W.—Tucker, A. W.: Non-Linear Programming. In Proc. 2^{nd} Berkeley Symposium on Mathematical Statistics and Probability, Univ. Calif. Press, 1961, pp. 481–492.

[10] Yutu Liu: QoS Computation and Policing in Dynamic Web Service Selection. WWW 2004, New York, IEEE Computer Society, May 17–22, 2004, pp. 66–73.

[11] Golconda, K. S.—Ozguner, F.: A Comparison of Static QoS-Based Scheduling Heuristics for a Meta-Task with Multiple QoS Dimensions in Heterogeneous Computing. Proceedings of the 18^{th} International Parallel and Distributed Processing Symposium (IPDPS '04), 2004.

[12] Abdelzaher, T. F.—Shin, K. G.—Bhatti, N.: User-Level QoS-Adaptive Resource Management in Server End-Systems. IEEE Transactions on Computers, Vol. 52, 2003, No. 5, pp. 678–685.

[13] Zeng, L.: QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering, Vol. 30, 2004, No. 5, pp. 311–327.

[14] Li, C.—Li, L.: The Use Of Economic Agents Under Price Driven Mechanism in Grid Resource Management. Journal of Systems Architecture, Elsevier, Vol. 50, 2004, No. 9, pp. 521–535.

[15] Li, C.—Li, L.: Applying Agents to Build Grid Service Management. Journal of Network and Computer Applications, Elsevier, Vol. 26, 2003, No. 4, pp. 323–340.

[16] Li, C.—Lu, Z.—Li, L.—Zhang, S.: A Mobile Agent Platform Based on Tuple Space Coordination. Journal of Advances In Engineering Software, Vol. 33, 2002, No. 4, pp. 215–225.

[17] Li, C.—Li, L.: Competitive Proportional Resource Allocation Policy for Computational Grid. Future Generation Computer Systems, Elsevier, Vol. 20, 2004, No. 6, pp. 1041–1054.

[18] Li, C.—Li, L.: A Distributed Utility-based Two Level Market Solution for Optimal Resource Scheduling in Computational Grid. Parallel Computing, Vol. 31, 2005, No. 3–4, pp. 332–351.

[19] Li, C.—Li, L.: Agent Framework to Support Computational Grid. Journal of Systems and Software, Vol. 70, 2004, No. 1–2, pp. 177–187.

[20] Li, L.—Li, C.: A QoS-Guaranteed Multicast Routing Protocol. Journal of Computer Communications, Vol. 27, 2004, No. 1, pp. 59–69.

[21] Li, L.—Li, C.: A Distributed QoS-Aware Multicast Routing Protocol. Acta Informatica, Vol. 40, 2003, No. 3, pp. 211–233.

[22] Li, C.—Li, L.: Integrate Software Agents and CORBA in Computational Grid. Journal of Computer Standards and Interfaces, Vol. 25, 2003, No. 4, pp. 357–371.

[23] Chiang, M.: Balancing Transport and Physical Layers in Wireless Multihop Networks: Jointly Optimal Congestion Control and Power Control. IEEE J. Sel. Areas Comm., Vol. 23, 2005, No. 1, pp. 104–116.

[24] Xiao, L.—Johansson, M.—Boyd, S.: Simultaneous Routing and Resource Allocation via Dual Decomposition. IEEE Transactions on Communications, Vol. 52, 2004, pp. 1136–1144.

[25] Nama, H.—Chiang, M.—Mandayam, N.: Utility Lifetime Tradeoff in Self Regulating Wireless Sensor Networks: A Cross-Layer Design Approach. IEEE ICC, September 2005.

[26] Li, C.—Li, L.: Multi Economic Agent Interaction for Optimizing the Aggregate Utility of Grid Users in Computational Grid. Applied Intelligence, Special Issue on Agent-Based Grid Computing, Vol. 25, 2006, No. 2, pp. 147–158.

Chunlin Li received her M. E. degree in computer science from Wuhan Transportation University in 2000, and her Ph. D. degree in computer software and theory from Huazhong University of Science and Technology in 2003. She is now an associate professor of computer science at Wuhan University of Technology. Her research interests include computational grids, distributed computing and mobile agents. She has published over 15 papers in international journals.

Layuan Li received his B. E. degree in communication engineering from Harbin Institute of Military Engineering, China in 1970 and his M. E. degree in communication and electrical systems from Huazhong University of Science and Technology, China in 1982. He academically visited Massachusetts Institute of Technology, USA in 1985 and 1999. Since 1982, he has been with the Wuhan University of Technology, China, where he is currently a Professor and Ph. D. tutor of computer science, and Editor in Chief of the Journal of WUT. He is the director of International Society of High-Technology, and paper reviewer of IEEE INFOCOM, ICCC and ISRSDC. He was the head of the Technical Group of Shaanxi Lonan PO Box 72, Ministry of Electrical Industry, China between 1970 and 1978. His research interests include high speed computer networks, protocol engineering and image processing. Professor Li has published over one hundred and fifty technical papers and is the author of six books. He also was awarded the National Special Prize by the Chinese Government in 1993.