

# Fabric Level and Application Level QoS Guarantees in Grid Computing

Li CHUNLIN, Li LAYUAN

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

Received: October 2006

**Abstract.** In the paper, a cross-layer optimization between application layer and fabric layer is proposed. The aim is to optimize the end-to-end quality of the dynamic grid application as well as efficiently utilizing the grid resources. The application layer QoS and fabric layer QoS are closely interrelated in Grids since the upper layer service is based on the lower level's capabilities. A fabric level and application level QoS scheduling algorithm is proposed. We formulate the integrated design of resource allocation and user QoS satisfaction control into a constrained optimization problem. The optimization framework provides a layered approach to the sum utility maximization problem. The application layer adaptively adjusts user's resource demand based on the current resource conditions, while the fabric layer adaptively allocates CPU, storage and bandwidth required by the upper layer.

**Key words:** application layer, fabric layer, grid computing, distributed computing.

## 1. Introduction

Grid is a framework that enables the construction of dynamic, self-organizing environments that support global resource discovery and provide seamless, secure access to a wide range of grid resources on demand. Among the most challenging problems posed by grid systems are discovery and coordination, that is, how to find and orchestrate a set of services operating on heterogeneous resources under different administrative control in order to solve a given problem. Grids are dynamic and grid services may join and leave the system at any time by their own will or due to faults. The grid-layered architecture defines the purpose and functions of its components, while indicating how these components interact with one another (Foster *et al.*, 2001). From the bottom of the stack and moving upward, these layers are fabric layer, connectivity layer, resource layer, collective layer and application layer. Adaptive resource management has become a key technique for the grid system to provide desired services at appropriate QoS (Dong and Akl, 2006). Most of the current resource management algorithms are confined to a single layer of the layered grid architecture, which leads to an inferior performance. Cross layer design was proposed to address those challenges. The concept of cross layer design introduces interlayer coupling across multiple layers and allows the exchange of

necessary information between different layers. Although this concept can be employed in all communication networks, it is especially important in grid computing because of the dynamic, autonomous and the heterogeneous nature of the grid resources.

In this paper, a cross-layer optimization scheme between application layer and fabric layer is proposed. The aim is to optimize the end-to-end quality of the dynamic grid application as well as efficiently utilizing the grid resources. The application layer QoS and fabric layer QoS are closely interrelated in Grids since the upper layer service is based on the lower level's capabilities. A fabric level and application level QoS scheduling algorithm is proposed. We formulate the integrated design of resource allocation and user QoS satisfaction control into a constrained optimization problem. The optimization framework provides a layered approach to the sum utility maximization problem. The application layer adaptively adjusts user's resource demand based on the current resource conditions, while the fabric layer adaptively allocates CPU, storage and bandwidth required by the upper layer.

The rest of the paper is structured as follows. Section 2 discusses the related works. Section 3 presents fabric level and application level QoS guarantees in grid environment. Section 4 presents a fabric level and application level QoS scheduling algorithm. In Section 5 the experiments are presented and discussed. Section 6 concludes the paper.

## 2. Related Works

QoS resource management and scheduling is a hot research area. Buyya *et al.* (2000, 2002) have proposed and developed a distributed computational economy-based framework, called Nimrod/G, for resource allocation and to regulate supply and demand of the available resources. Ernemann *et al.* (2002) addresses the idea of applying economic models to the scheduling task. K. Cross layer design approaches have been applied to network architecture to make better use of network resources by optimizing across the boundaries of traditional network layers. In (Chiang, 2005), M. Chiang presents the overall communication network can be modeled by a generalized network utility maximization problem, each layer corresponds to a decomposed subproblem. In the case of the transport and physical layers, link congestion prices turn out to be the optimal layering prices. In (Jun *et al.*, 2005) Yuan *et al.* jointly considered Data routing, wireless medium contention and network coding to achieve the optimal network performance. The primal-dual solution method in the framework represents a cross-layer optimization approach. In (Chen *et al.*, 2005) Chen *et al.* considers optimal cross-layer congestion control, routing and scheduling design for ad hoc wireless networks. They formulate resource allocation in networks with fixed wireless channels or single-rate wireless devices that can mask channel variations as a utility maximization problem. In (Xiao *et al.*, 2004), Xiao *et al.* formulate the simultaneous routing and resource allocation problem as a convex optimization problem over the network flow variables and the communications variables. In (Nama *et al.*, 2005) Nama *et al.* propose a framework for cross-layer design across transport, network, and radio resource layers to find the optimal set of source rates, network

flows, and radio resource allocation that jointly maximizes the network utility and lifetime. In (Chen *et al.*, 2004) J. Chen *et al.* propose a joint cross-layer design for QoS content delivery. They propose a cross-layer design scheme, especially among application, MAC and physical layer, for wireless QoS content delivery. In (Choi *et al.*, 2006) Choi *et al.* propose a cross-layer optimization between application layer, data link layer, and physical layer for streaming video delivery in a multiuser wireless environment. In (Chunlin and Layuan, 2004a), the paper is to implement a uniform higher-level management of the computing resources and services on the Grid, and provide users with a consistent and transparent interface for accessing such services. In (Chunlin and Layuan, 2004b), 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 priori cooperation. In (Chunlin and Layuan, 2003), the paper presents an Agent-based Grid Service Management, which applies the concept of agents to computational grid. In (Chunlin and Layuan, 2006a), a distributed multiple dimensional QoS constrained resource scheduling optimization policy in computational grid is proposed. In (Chunlin and Layuan, 2004c), the paper provides a price-directed proportional resource allocation algorithm for solving the grid task agent resource allocation problem. Reference (Chunlin and Layuan, 2005) presents a distributed utility-based two level market solution for optimal resource scheduling in computational grid. A multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid is described in (Chunlin and Layuan, 2006b). The research objective of this paper is not as our previous works.

### 3. Fabric Level and Application Level QoS Guarantees

#### 3.1. System Model

In this section, we set up the mathematical models for fabric level and application level QoS guarantees in grid environment. The two major parties in grid computing, namely, resource users who submit various applications, and resources providers who own their resources, usually have different motivations when they join the grid. These incentives are presented by objective functions in scheduling. Grid users are basically concerned with the performance of their applications, for example the total cost to run a particular application, while resource providers usually pay more attention to the performance of their resources, for example the resource utilization in a particular period. The work proposes cross-layer optimization that provides a potential solution of joint optimization of the fabric and application layer. This model assumes that a user can get all of needed resources from resource provider with necessary payment. The grid scheduler have “societal goals” and attempt to maximize the average user utility and revenue of resource provider, as opposed to providers and grid users that have individualistic goals; each provider wishes to maximize its revenue, while each user wishes to maximize its utility and do so for as little cost as possible. The global grid system objective is to maximize the sum utility.

Maximizing sum utility serves as a common objective of joint fabric-application layer optimization to create incentives for grid user of application layer and resource provider to interplay.

$T_i$  is time limits given by the grid user  $i$  to complete its all jobs.  $x_i^j$  is resource allocation obtained by user  $i$  from resource provider  $j$ .  $C_j$  is capacity of resource provider  $j$ .  $t_i^n$  is the time taken by the grid user  $i$  to complete  $n$ th job.  $E_i$  is the budget of grid user  $i$ .  $P_i^j$  is the payments of the grid user  $i$  to the resource provider  $j$ .  $q_{in}$  is the size of  $i$ th user's  $n$ th job.  $pr_j$  is the price of the unit of resource provider  $j$ .

A separate utility function  $U_i$  is associated with each grid user  $i$ . The objective is to optimally allocate resource among the users, while choosing the best QoS satisfaction, in order to maximize the global grid sum utility. Expressed succinctly, the grid optimization problem is

$$\begin{aligned} & \text{Max } \sum_i U_i, \\ & \text{s.t. } C_j \geq \sum_i x_i^j, \quad T_i \geq \sum_n t_i^n, \\ & E_i > \sum_j P_i^j, \quad x_i^j > 0. \end{aligned} \quad (3.1)$$

In (3.1), the global grid optimization problem aims to maximize  $\sum U_i$  subject to the constraints of the application and the fabric layer. The constraint implies that the aggregate resource units do not exceed the total capacity of grid resource  $C_j$ , and grid user should complete all its jobs under time limits  $T_i$  and its budget  $E_i$ .

The Lagrangian technique can be used to solve the mixed integer-programming problem efficiently. In particular, the Lagrangian method leads to a decomposition of the utility maximization problem into two smaller subproblems at the application layer and the fabric layer, each of which may be solved independently. We can apply the Lagrangian method to solve the global grid optimization problem. Let us consider the Lagrangian form of Eq. 3.1:

$$\begin{aligned} L(\lambda, \beta, \gamma, x_i^j, P_i^j) = & \sum_i U_i - \lambda \left( \sum_i x_i^j - C_j \right) \\ & - \beta \left( \sum_i t_i^n - T_i \right) - \gamma \left( \sum_j P_i^j - E_i \right). \end{aligned} \quad (3.2)$$

In (3.2),  $\lambda, \beta, \gamma$  are the Lagrangian multipliers. The function consists of application layer variables  $P_i^j, t_i^n$ , and fabric layer variables  $x_i^j$ . Moreover,  $L$  can be separated into two maximization subproblems, namely a utility maximization problem, corresponding to user QoS satisfaction problem at the application layer.

$$AF = \text{Max} \left\{ \left( E_i - \sum_j P_i^j \right) + \left( T_i - \sum_{n=1}^N \frac{q_{in}}{x_i^j} \right) \right\} \quad \text{s.t. } T_i \geq \sum_n t_i^n, \quad (3.3)$$

and a resource allocation problem at the fabric layer.

$$RF = \text{Max} \sum P_i^j (\log x_i^j + 1) \quad \text{s.t.} \quad C_j \geq \sum_i x_i^j. \quad (3.4)$$

In (3.3), problem  $AF$  is conducted at the application layer; the grid user gives the unique optimal payment to resource provider under deadline constraint to maximize the user's satisfaction.  $(E_i - \sum_j P_i^j)$  represents the surplus of grid user, which is obtained by budgets subtracting the payments to resource providers.  $q_{in}$  is total resource requirement of grid user  $m$ 's  $n$ th job.  $x_i^j$  represents the fraction of service units allocated to grid user  $m$  by service provider  $i$ .  $(T_i - \sum_{n=1}^N q_{in}/x_i^j)$  represents the saving times for user, which is gotten by time limit subtracting actual spending time. So, the objective of problem  $AF$  is to get more surpluses of money and complete task for grid user as soon as possible. In (3.4), problem  $RF$  is conducted at the fabric layer, different resource providers compute optimal resource allocation for maximizing the revenue of their own, the objective of resource providers is to maximize  $P_i^j (\log x_i^j + 1)$  under the constraints of their provided amounts at the fabric layer.

Thus, the optimization framework provides a layered approach to the sum utility maximization problem. The application layer adaptively adjusts user's resource demand based on the current resource conditions, while the fabric layer adaptively allocates CPU, storage and bandwidth required by the upper layer. The interaction between the layers is now controlled through the use of the variable  $\lambda$ , which is the price charged from grid users by grid resource provider and coordinates the user demand and the fabric layer supply of resource.

### 3.2. Mathematical Solutions

We now describe efficient methods to solve the two subproblems, which together with the vertical decomposition described in the previous section, solve the overall utility maximization problem globally and efficiently.

The fabric layer problem is a resource allocation problem. Different resource providers compute optimal resource allocation for maximizing the revenue of their own under constraints of resource capacity.

$$RF = \text{Max} \sum P_i^j \log(x_i^j + 1) \quad \text{s.t.} \quad C_j \geq \sum_i x_i^j.$$

$\sum P_i^j \log(x_i^j + 1)$  presents the revenue obtained by grid resource provider  $j$  from grid users. Grid resource provider can't sell resource slice more than  $C_j$ , which is the upper of resource capacity. We take derivative and second derivative with respect to  $x_i^j$ :

$$RF'(x_i^j) = \frac{P_i^j}{x_i^j + 1}, \quad RF''(x_i^j) = -\frac{P_i^j}{(x_i^j + 1)^2}.$$

$RF''(x_i^j) < 0$  is negative due to  $0 < x_i^j$ . The extreme point is the unique value maximizing the revenue of grid resource provider. The Lagrangian for  $RF$  is  $L(x)$

$$\begin{aligned} L(x_i^j) &= \sum P_i^j \log(x_i^j + 1) + \sigma \left( C_j - \sum_i x_i^j \right) \\ &= \sum (P_i^j \log(x_i^j + 1) - \sigma x_i^j) + \sigma C_j, \end{aligned}$$

where  $\sigma$  is the Lagrangian constant. From Karush–Kuhn–Tucker theorem we know that the optimal solution is given  $\partial L(x)/\partial x = 0$  for  $\lambda > 0$ .

$$\frac{\partial L(x_i^j)}{\partial x_i^j} = \frac{P_i^j}{x_i^j + 1} - \sigma.$$

Let  $\partial L(x)/\partial x = 0$ .

$$x_i^j = \frac{P_i^j - \sigma}{\sigma}.$$

Using this result in the constraint equation, we can determine  $\sigma$  as

$$C_j = \frac{1}{\sigma} \sum_{i=1}^n P_i^j - n, \quad \sigma = \frac{\sum_{k=1}^n P_k^j}{C_j + n}.$$

We obtain

$$x_i^{j*} = \frac{P_i^j (C_j + n)}{\sum_{k=1}^n P_k^j} - 1.$$

$x_i^{j*}$  is the unique optimal solution to the fabric layer optimization subproblem. It means that grid resource providers allocate  $x_i^{j*}$  to grid user to maximize its revenue.

$$AF = \text{Max} \left\{ \left( E_i - \sum_j P_i^j \right) + \left( T_i - \sum_{n=1}^N \frac{q_{in}}{x_i^j} \right) \right\} \quad \text{s.t. } T_i \geq \sum_n t_i^n.$$

$q_{in}$  is resource requirement of grid user  $i$ 's  $n$ th job. Let  $pr_j$  denote the unit price of resource provider  $j$ . Let the pricing policy,  $pr = (pr_1, pr_2, \dots, pr_j)$ , denote the set of resource unit prices of all the resource providers at the fabric layer. The grid user  $i$  receives resources proportional to its payment relative to the sum of the resource provider's revenue. Let  $x_i^j$  be the fraction of resource units allocated to grid user  $i$  by resource provider  $j$ . The resource units  $x_i^j$  allocated to grid user  $i$  are

$$x_i^j = C_j \frac{P_i^j}{pr_j}.$$

The time taken by the  $i$ th grid user to complete  $n$ th job is  $t_i^n = \frac{q_{in}pr_j}{C_j P_i^j}$ . We reformulate  $AF$  problem as

$$\text{Max} \left\{ \left( E_i - \sum_j P_i^j \right) + \left( T_i - \sum_{n=1}^N \frac{q_{in}pr_j}{C_j P_i^j} \right) \right\}.$$

We take derivative and second derivative of  $AF$  with respect to  $P_i^j$ :

$$AF'(P_i^j) = \frac{d AF}{d P_i^j} = \sum_{n=1}^N \frac{q_{in}pr_j}{(P_i^j)^2 C_j} - 1,$$

$$AF''(P_i^j) = \frac{d^2 AF(P_i^j)}{d (P_i^j)^2} = - \sum_{n=1}^N \frac{q_{in}pr_j}{(P_i^j)^3 C_j}.$$

$AF''(P_i^j) < 0$  is negative due to  $0 < P_i^j$ . The extreme point is the unique value minimizing the grid user's payment under completed time limits. The Lagrangian for the grid user's utility is  $L(P_i^j)$ .

$$L(P_i^j) = \left( E_i - \sum_j P_i^j \right) + \left( T_i - \sum_{n=1}^N \frac{q_{in}pr_j}{C_j P_i^j} \right) + \lambda \left( T_i - \sum_{n=1}^N t_i^n \right),$$

where  $\lambda$  is the Lagrangian constant. From Karush–Kuhn–Tucker Theorem we know that the optimal solution is given  $\partial L(P_i^j)/\partial P_i^j = 0$  for  $\lambda > 0$ .

$$\frac{\partial L(P_i^j)}{\partial P_i^j} = -1 + \frac{q_{in}pr_j}{C_j (P_i^j)^2} + \lambda \frac{q_{in}pr_j}{C_j (P_i^j)^2}.$$

Let  $\partial L(P_i^j)/\partial P_i^j = 0$  to obtain

$$P_i^j = \left( \frac{(1 + \lambda)q_{in}pr_j}{C_j} \right)^{1/2}.$$

Using this result in the constraint equation, we can determine  $\theta = 1 + \lambda$  as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{k=1}^N \left( \frac{pr_k q_{ik}}{C_j} \right)^{1/2}}.$$

We obtain

$$P_i^{j*} = \left( \frac{q_{in}pr_i}{C_j} \right)^{1/2} \frac{\sum_{k=1}^N \left( \frac{q_{ik}pr_k}{C_j} \right)^{1/2}}{T_i}.$$

$P_i^{j*}$  is the unique optimal solution to the application layer optimization subproblem. It is the optimal payment of grid user  $i$  to resource provider  $j$  under completion time constraint to maximize the user's benefits.

#### 4. Fabric Level and Application Level QoS Scheduling Algorithm

The proposed fabric level and application level QoS scheduling algorithm is an iterative algorithm. In each iteration, at the application layer, the grid user computes the unique optimal payment to resource provider under completion time constraint to maximize the user's satisfaction. The grid user individually solves its fees to pay for needed resources to complete its all jobs, adjusts its grid resource demand under budget constraint and notifies the grid resource provider about this change. At the fabric layer, different resource providers compute optimal resource allocation for maximizing the revenue of their own. After the new grid resource demand is observed by the grid resource provider, it updates its price according to grid resource demands, and then sends the new prices to the grid user to allocate the resource for grid user, and the cycle repeats. The iterative algorithm that achieves fabric level and application level QoS scheduling is described as follows.

##### Algorithm 1. Fabric level and application level QoS scheduling algorithm

---

###### Subroutine 1: Grid user $i$ at application layer

---

Receives from the resource provider  $i$  the price  $pr_j^{(n)}$ ;

$$P_i^{j*} = \text{Max}\{AP(P_i^j)\} = \text{Max}\left\{\left(E_i - \sum_j P_i^j\right) + \left(T_i - \sum_{n=1}^N \frac{q_{in}}{x_i^j}\right)\right\};$$

//calculates  $P_i^{j*}$  to maximize  $\text{Max}\{AP(P_i^j)\}$ ;

---

If  $E_i \geq \sum_j P_i^j$

$$\text{Then } x_i^{j(n+1)} = P_i^{j*} / pr_j^{(n)};$$

// compute new resource demand

Return  $x_i^{j(n+1)}$  to resource provider  $i$ ;

Else Return Null;

---

###### Subroutine 2: Resource provider $j$ at fabric layer

---

Receives  $x_i^{j(n)}$  from grid user  $i$ ;

If  $C_j \geq \sum_i x_i^j$

Then

$$pr_j^{(n+1)} = \max\{\varepsilon, pr_j^{(n)} + \eta(\sum_i x_i^j - C_j)\};$$

Return new price  $pr_j^{(n+1)}$  to all grid users;

Else Return Null;

---



## 5. Simulation

In the simulation, three simulated entities were created: grid user, resource provider and grid scheduler. Grid scheduler implements joint application and fabric layer optimization scheduling algorithm. Once a job was submitted, the grid scheduler receives the job scheduling request, schedules the jobs to the host nodes, and then writes the scheduling records to the files for statistical analysis. The grid scheduler starts a listening thread that listens to the job requests. It receives the job requirements and puts them into the job queue. While the job queue is not empty, the grid scheduler starts the scheduling algorithm to find the right match. The joint application and fabric layer optimization simulator is implemented on top of the JAVASIM network simulator (JAVASIM). Network generator BRITE (BRITE) generates the computer network topology. BRITE is a random network topology generator used to generate the simulation test bed. With the BRITE, the user can configure the complexity of the network to specify number of nodes and connectivity, and it will generate a random network with the specified properties. The link propagation delays and bandwidth can be set freely. In order to simulate the dynamics and heterogeneity of the grid, all values of networks can be changed after topology generation. We simulate a simulated grid system consisting 100 clusters. The number of nodes of these clusters is a random variable normally distributed with the mean of 50 and the standard deviation of 20. The network bandwidth in the grid clusters is from 100Mbps to 1Gbps. Each node is characterized by a resource vector containing the CPU rate, the amount of main memory, and the disk capacity. Machine computing power of node is from 30% to 100%. For each set of clusters, the attributes of the cluster include the number of nodes, processor capacity, machine failure rate and machine computing power. This setup makes this grid environment non-dedicated, heterogeneous and dynamic. For example, the resource vector for a node with one 1.3 GHz CPU, 1 GB of memory, and a 30 GB disk is (1.3GHz; 1GB; 30GB). There are totally  $NU = 1500$  grid users and  $NR = 400$  grid resource providers. We neglect the network topology and the communication costs associated with it. Instead, we assume that each of the grid users can submit jobs to any of the resources. At each time step, grid user agents independently generate jobs at rate  $a = [0.1, 0.3]$ . To take into account the wide dispersion in the job sizes in real grid applications, the sizes of the jobs are taken randomly from the uniform distribution in the interval  $[1, 100]$ . It was assumed that the job size ranged from 20 Kbytes and 2 Mbytes, and the size of a job's result ranged from 10 Kbytes to 5 Mbytes. The capacities of the resources were also chosen uniformly in the interval  $[R_{min}; R_{max}]$ . We consider the dispersion in the resource capacities  $R_{max} - R_{min}$  is relatively large,  $R_{max} = 500$  and  $R_{min} = 50$ . The resource cost can be expressed in grid dollar that can be defined as processing cost per MIPS. Processor capacity varies from 220 to 580 MIPS. The initial price of resource is set from 10 to 500 grid dollars. The deadlines of users are chosen from 100ms to 800ms. The budgets of users are set from 100 to 1500 grid dollars.

The experiments study how performance of the proposed fabric level and application level QoS scheduling (FASQ) is affected by the change of different environment parameters. We use five metrics to evaluate the effectiveness of FASQ policy such as

QoS satisfaction ratio, the revenue for providers, user admission ratio, user utility and resource utilization. QoS satisfaction ratio measures the level of utility for satisfying job requests. It is computed as the proportion of jobs that required QoS are fulfilled out of all submitted jobs. A higher QoS satisfaction ratio represents better performance. The revenue for providers is the sum of revenues for all of its resource types. The user admission ratio is the number of admitted grid users over the number of all grid users. A user is admitted into the system when there is a grid provider able to allocate some of the resources requested by the grid user; otherwise the grid user is dropped. Resource utilization is the ratio of the resources consumed to the total resources available as a percentage. The behavior of FASQ policy is examined by three parameters: Job arrival rate ( $a$ ), Resource load ratio ( $r$ ) and Price increase ratio ( $p$ ). Job arrival rate is job arrival speed, which will affect the system load. The resource load ratio for a resource type is the ratio of the amount of resources requested by the grid users to the total capacity of resource providers for this resource type. Price increase ratio is the ratio of the increased price to initial unit price. The grid users do not provide the precise amount of resources needed, they only specify their utility functions.

How resource load ratio effects on QoS satisfaction ratio, revenue for providers, user admission ratio, user utility and resource utilization were illustrated in Fig. 1, Fig. 2, Fig. 3, Fig. 4 and Fig. 5, respectively. Resource load ratio  $r$  is set to 0.1, 0.25, 0.50, 0.75 and 1.0. As shown in Fig. 1, the user QoS satisfaction drops during the transient period and then converges to a steady value, the smaller is  $r$ , the earlier the system reaches the steady state and the higher is the user QoS satisfaction. When resource load ratio is low ( $r = 0.10$ ), the resources are enough to be allocated to users, user can choose appropriate resources to complete tasks under the deadline, and so user QoS requirement can be well satisfied. When resource load ratio increases, the available resources are not enough to be allocated, some users can't achieve the objective. Fig. 2 shows that there exists a resource load ratio ( $r = 0.5$ ) to maximize the revenue of provider. The maxim of the curve is the optimal revenue point for the resource provider. When  $r < 0.5$ , more user can afford the payment, and the unit price is not high. When  $r > 0.5$ , the unit price is too high, and less user can afford the payment. When  $r = 0.5$ , the unit price of resource is proper, so the revenue is maximal. Fig. 3 shows that when resource load ratio is low ( $r = 0.10$ ), the system is capable of handling all requests and the corresponding plots overlap with each other, during the steady state the user admission ratio is 1. When resource load ratio is high ( $r = 1.0$ ), some user requests are dropped and user admission ratio decreases to 0.5. As shown in Fig. 4 the smaller is  $r$ , the higher is the user utility. When  $r = 0.75$ , the user utility is as much as 35% less than that by  $r = 0.10$ . The resource load ratio is low, the resource is lightly loaded, the unit price of the resource is cheap; user can choose cheap resources to complete tasks under the deadline, so the benefit of the user (user utility) is high. When the system is heavily loaded, the unit price of the resource is expensive; the benefit of the user (user utility) is less. Considering the resource utilization, from the results in Fig. 5, as resource load ratio is higher, the resource utilization become lower. When  $r = 0.75$ , the resource utilization is as much as 30% less than utilization by  $r = 0.10$ . Because when resource load ratio increases quickly, the grid resource supply is

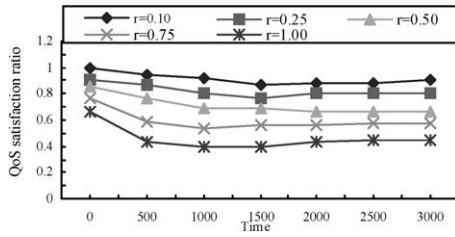


Fig. 1. QoS satisfaction ratio under various resource load ratio.

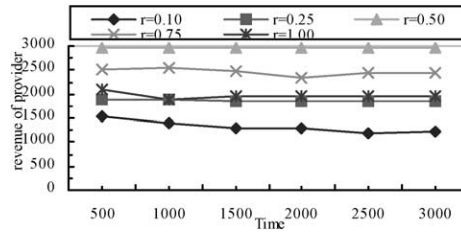


Fig. 2. Revenue of provider under various resource load ratio.

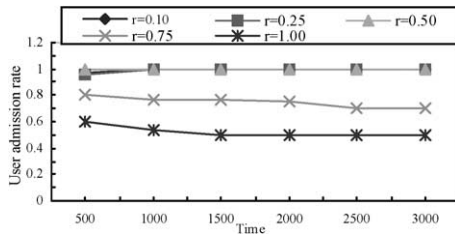


Fig. 3. User admission ratio under various resource load ratio.

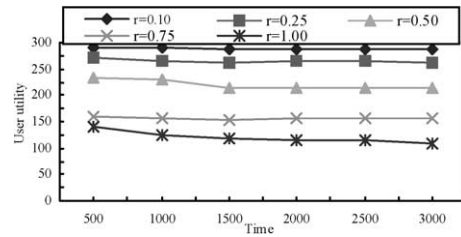


Fig. 4. User utility under various resource load ratio.

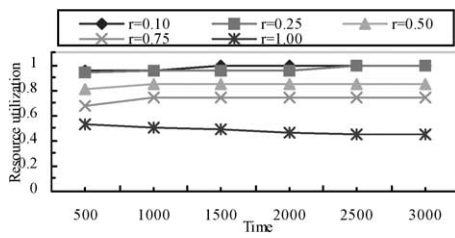


Fig. 5. Resource utilization under various resource load ratio.

not enough to be allocated to user, the unit price of the resource will increase, some users will not afford the grid resources.

The impacts of price increase ratio on QoS satisfaction ratio, revenue for providers, user admission ratio, user utility and resource utilization were illustrated in Fig. 6, Fig. 7, Fig. 8, Fig. 9 and Fig. 10 respectively. When  $p$  is set to 0.05, 0.15, 0.30, 0.45, 0.60 and 0.8, the smaller is  $p$ , the earlier the system reaches the steady state and the higher is the user QoS satisfaction as shown in Fig. 6. Because when price becomes high, users will afford more payment to obtain the grid resource, some tasks can't be completed before their deadlines. Price increasing quickly leads to some users with low budget can't be satisfied to fulfill their achievements. From the results of Fig. 7, we can find there exists a price increase ratio ( $p = 0.45$ ) to maximize the revenue of provider. The price variations adjust the revenue by means of the users' reaction depicted in FASQ by calculating its resource demand according to optimal payments for resource provider to maximize application layer's optimization objective function. There is also a dependence on the

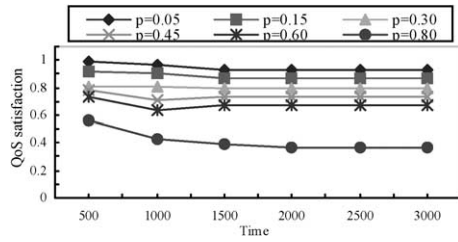


Fig. 6. QoS satisfaction under various price increase ratio.

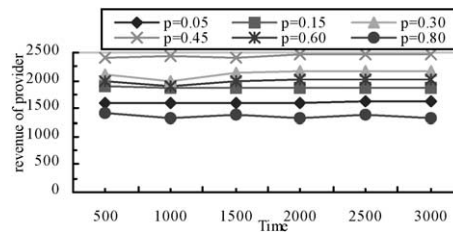


Fig. 7. Revenue of provider under various price increase ratio.

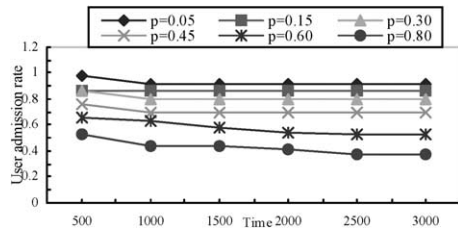


Fig. 8. User admission ratio under various price increase ratio.

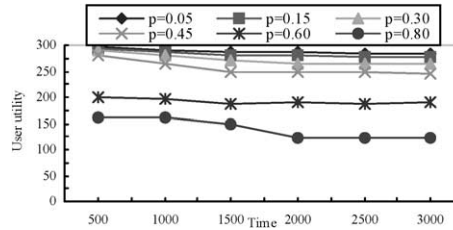


Fig. 9. User utility under various price increase ratio.

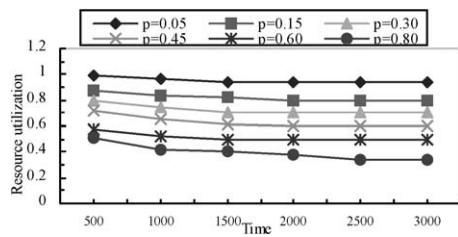


Fig. 10. Resource utilization under various price increase ratio.

resource provider's choices in assigning resource to the users. The maxim 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 resource allocation. Considering the user admission ratio, the results of Fig. 8 show that when increasing  $p$ , the user admission ratio become lower. Increasing prices of resource provider will prevent users from being admitted by the system. When increasing price increase ratio by  $p = 0.45$ , the user admission ratio is as much as 26% less than that with  $p = 0.05$ . When  $p = 0.8$ , the user admission ratio reduce to nearly 44%. The performance effects of varying the price increase ratio on the user utility are reported in Fig. 9. When the price increase ratio is increasing, user utility will decrease. When increasing price increase ratio by  $p = 0.45$ , the user utility is as much as 49% less than that with  $p = 0.05$ . When  $p = 0.8$ , the user utility reduce to nearly 28% compared with  $p = 0.15$ . Considering the resource utilization, from the results in Fig. 10, as price increase ratio is higher, the resource utilization become lower. When  $p = 0.60$ , the resource utilization is as much as 28% less

than utilization by  $p = 0.15$ . Because when price increases quickly, the users will afford more payment to obtain the grid resource, some tasks with low budget can't obtain the resources.

The impacts of job arrival rate on QoS satisfaction ratio, revenue for providers, user admission ratio, user utility and resource utilization were illustrated in Fig. 11, Fig. 12, Fig. 13, Fig. 14 and Fig. 15 respectively. Fig. 11 shows that the QoS satisfaction ratio increases during the transient period and then reaches a stable value. When  $a = 0.3$ , QoS satisfaction ratio is as much as 30% less than that by  $a = 0.10$ . The smaller is  $a$ , the more users can be admitted by resource providers with cheaper prices and these users experience higher QoS satisfaction. So the larger is  $a$ , the lower is QoS satisfaction ratio. Fig. 12 shows that the revenue of provider increases when job arrival rate increases. The larger is job arrival rate, the more user buy the resource to complete the jobs, and the more resources are allocated to users, so the higher is the revenue of resource provider. Fig. 13 shows that when  $a = 0.1$ ,  $a = 0.15$ , and  $a = 0.2$ , the user admission ratio is

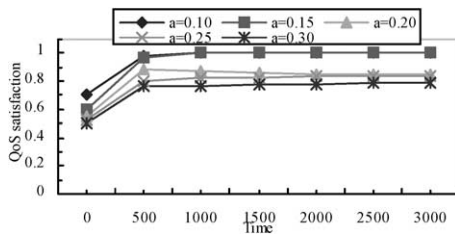


Fig. 11. QoS satisfaction ratio under various job arrival rate.

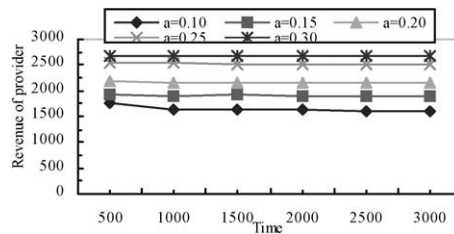


Fig. 12. Revenue of provider under various job arrival rate.

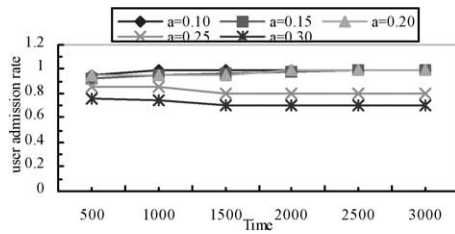


Fig. 13. User admission ratio under various job arrival rate.

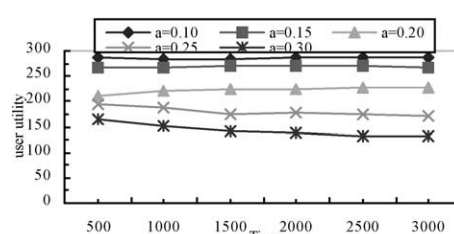


Fig. 14. User utility under various job arrival rate.

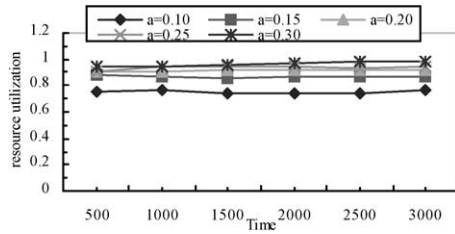


Fig. 15. Resource utilization under various job arrival rate.

approximately 1.0. When  $a = 0.3$ , as time goes on, the user admission ratio decreases. When  $a = 0.25$ , the user admission ratio is as much as 15% less than that with  $a = 0.10$ . Fewer users can be admitted into the system due to the increase of system burden. Fig. 14 shows when job arrival rate increases ( $a = 0.3$ ), user utility is as much as 21% less than that with  $a = 0.1$ . The user utility is larger when  $a$  is smaller. When job arrival rate increases, system load increases; some user's requirements can't be processed on time. The task with low budget can't be completed before deadline; this leads to low user's benefit (user utility). Fig. 15 shows as  $a$  increase, resource utilization ratio increases. When  $a = 0.3$ , the resource utilization is as much as 23% more than utilization by  $a = 0.10$ . When job arrival rate was very large, many tasks will be sent to system, resources are busier.

## 6. Conclusions

Adaptive resource management has become a key technique for the grid system to provide desired services at appropriate QoS. Most of the current resource management and scheduling algorithms are confined to a single layer of the layered grid architecture, which leads to an inferior performance. In the paper, we propose cross-layer optimization that provides a potential solution of joint optimization of the fabric and application layer, which combine both application centric and system-centric scheduling benefits. The algorithm aims to maximize the global utility as well as the QoS is guaranteed. We formulate the integrated design of into a constrained optimization problem. The formulation defines the interaction and cooperation between different components of the algorithm as well as the information exchange between the application layer and the fabric layer. The Lagrangian method leads to a decomposition of the utility maximization problem into two smaller subproblems, each of which may be solved independently. The interaction between the layers is now controlled through the use of the pricing variable, which coordinates the user demand at the application layer and supply of resources at the fabric layer.

## Acknowledgements

The authors thank the editors and the anonymous reviewers for their helpful comments and suggestions. The work is supported by national Natural Science Foundation of China under grants (60402028, 60672137) China Specialized Research Fund for the Doctoral Program of Higher Education under Grants (No.20060497015) and Wuhan Younger Dawning Foundation under grants (20045006071-15).

## References

BRITE, <http://www.cs.bu.edu/brite>

- Buyya, R., J. Giddy and D. Abramson (2000). An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications. In *Proc. of the 2nd International Workshop on Active Middleware Services (AMS 2000)*, Pittsburgh, USA, August. pp. 221–230.
- Buyya, R., D. Abramson, J. Giddy and H. Stockinger (2002). Economic models for resource management and scheduling in grid computing. *J. of Concurrency and Computation: Practice and Experience*, **14**(13–15), 1507–1542.
- Chen, J., T. Lv and H. Zheng (2004). Cross-layer design for QoS wireless communications. In *Proc. International Symposium on Circuits and Systems, ISCAS*. pp. 176–220.
- Chen, L., S. Low, M. Chiang and J. Doyle (2005). Optimal cross-layer congestion control, routing, and scheduling design in ad hoc wireless networks. Preprint.
- Chiang, M. (2005). Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control. *IEEE J. Sel. Areas Comm.*, **23**(1), 104–116.
- Choi, L.-U., W. Kellerer and E. Steinbach (2006). On cross-layer design for streaming video delivery in multiuser wireless environments. *EURASIP Journal on Wireless Communications and Networking*, 1–10.
- Chunlin, L., and L. Layuan (2003). Applying agents to build grid service management. *Journal of Network & Computer Applications*, **26**(4), 323–340.
- Chunlin, L., and L. Layuan (2004a). Agent framework to support computational grid. *Journal of Systems and Software*, **70**(1–2), 177–187.
- Chunlin, L., and L. Layuan (2004b). The use of economic agents under price driven mechanism in grid resource management. *Journal of Systems Architecture*, **50**(9), 521–535.
- Chunlin, L., and L. Layuan (2004c). Competitive proportional resource allocation policy for computational grid. *Future Generation Computer Systems*, **20**(6), 1041–1054.
- Chunlin, L., and L. Layuan (2005). A distributed utility-based two level market solution for optimal resource scheduling in computational grid. *Parallel Computing*, **31**(3–4), 332–351.
- Chunlin, L., and L. Layuan (2006a). A distributed multiple dimensional QoS constrained resource scheduling optimization policy in computational grid. *Journal of Computer and System Sciences*, **72**(4), 706–726.
- Chunlin, L., and L. Layuan (2006b). Multi economic agent interaction for optimizing the aggregate utility of grid users. In *Computational Grid, Applied Intelligence* (special issue in agent-based Grid computing), Springer-Verlag, Heidelberg, vol. 25/2. pp. 147–158.
- Dong, F., and S.G. Akl (2006). Scheduling algorithms for grid computing: state of the art and open problems. Technical Report. Queen's University School of Computing, January.
- Ernemann, C., V. Hamscher and R. Yahyapour (2002). Economic scheduling in grid computing. In *Proc. of 8th Workshop on Job Scheduling Strategies for Parallel Processing*, Edinburgh, Scotland, UK, July 2002. pp. 128–152.
- Foster, I., C. Kesselman and S. Tuecke (2001). The anatomy of the grid: enabling scalable virtual organizations. *International J. Supercomputer Applications*, **15**(3).
- JAVASIM, <http://jvasim.ncl.ac.uk>
- Jun, Y., L. Zongpeng, Y. Wei and L. Baochun (2005). A cross-layer optimization framework for multicast in multi-hop wireless networks. In *Proceedings of the First International Conference on Wireless Internet (WICON'05)*.
- Madan, R., S. Cui, S. Lall and A. Goldsmith (2005). Cross-layer design for lifetime maximization in interference-limited wireless sensor networks. In *Proc. IEEE INFOCOM*, March.
- Nama, H., M. Chiang and N. Mandayam (2005). Utility lifetime tradeoff in self regulating wireless sensor networks: A cross-layer design approach. *IEEE ICC*, Sept.
- Xiao, L., M. Johansson and S. Boyd (2004). Simultaneous routing and resource allocation via dual decomposition. *IEEE Transactions on Communications*, **52**, 1136–1144.

**L. Chunlin** received the ME in computer science from Wuhan Transportation University in 2000, and PhD degree in computer software and theory from Huazhong University of Science and Technology in 2003. She now is an associate professor of computer science in Wuhan University of Technology. Her research interests include computational grid, distributed computing and mobile agent. She has published over 20 papers in international journals.

**L. Layuan** received BE degree in communication engineering from Harbin Institute of Military Engineering, China in 1970 and ME 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, respectively. Since 1982, he has been with the Wuhan University of Technology, China, where he is currently a professor and PhD tutor of computer science, and editor in chief of the Journal of WUT. He is director of International Society of High-Technol. and paper reviewer of IEEE INFOCOM, ICCS and ISRSDC. He was the head of the Technical Group of Shaanxi Lonan P.O. Box 72, Ministry of Electrical Industry, China from 1970 to 1978. His research interests include high speed computer networks, protocol engineering and image processing. Professor Li has published over 150 technical papers and is the author of six books. He also was awarded the National Special Prize by the Chinese Government in 1993.

## **Tarpsluoksninis (uždavinių ir sistemų) resursų paskirstymo būdas GRID paskirstytų skaičiavimų tinkluose**

Li CHUNLIN, Li LAYUAN

Dauguma iš esamų resursų valdymo ir užduočių paskirstymo algoritmų GRID tinkluose apsiriboja vienu iš GRID architektūros sluoksnių, kas akivaizdžiai sumažina jų našumą. Straipsnyje nagrinėjama tarpsluoksninė optimizacija tarp taikomųjų uždavinių ir kompiuterinių sistemų sluoksnių. Vienu metu yra siekiama optimizuoti kaip GRID taikomojo uždavinio sprendimo kokybę, taip ir GRID tinklo resursų panaudojimą. Pasiūlomas tarpsluoksninis resursų paskirstymo algoritmas, kuris apjungia abiejų paskirstymo būdų pranašumus: orientuotus atitinkamai į taikomosius uždavinius ir kompiuterines sistemas. Siūloma integruota schema yra formalizuojama ir užrašoma kaip sąlyginės optimizacijos uždavinys. Jis apibrėžia sluoksnių sąveiką ir informacijos apsikeitimą. Taikomųjų uždavinių sluoksnis adaptyviai derina vartotojo resursų poreikį pagal prieinamus resursus ir tuo pačiu sistemų sluoksnis adaptyviai išskiria kompiuterinius resursus, reikalaujamas aukštesnio sluoksnio. Gauti GRID tinklo modeliavimo rezultatai demonstruoja pasiūlyto tarpsluoksninės optimizacijos paskirstymo pranašumą prieš tradicinius vienasluoksnius paskirstymo būdus.