Computing and Informatics, Vol. 29, 2010, 1117–1140, V 2010-Dec-9

SIMULTANEOUS OPTIMIZATION OF APPLICATION UTILITY AND CONSUMED ENERGY IN MOBILE GRID

Chunlin Li, Layuan Li

State Key Laboratory of Software Development Environment, Beijing University of Aeronautics and Astronautics, Beijin, 100083, P.R. China
&
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 11 November 2008; revised 2 November 2009
Communicated by Boleslaw Szymanski

Abstract. Mobile grid computing is aimed at making grid services available and accessible anytime anywhere from mobile device; at the same time, grid users can exploit the limited resources of mobile devices. This paper proposes simultaneous optimization of application utility and consumed energy in mobile grid. The paper provides a comprehensive utility function, which optimizes both the application level satisfaction such as execution success ratio and the system level requirements such as high resource utilization. The utility function models various aspects of job, application and system. The goal of maximizing the utility is achieved by decomposing the problem into a sequence of sub-problems that are then solved using the NUM optimization framework. The proposed price-based iterative algorithms enable the sub-problems to be processed in parallel. The simulations and analysis are given to study the performance of the algorithm.

Keywords: Mobile grid, utility, scheduling

1 INTRODUCTION

Mobile grid combines mobile computing and grid computing, it expands grid computing to mobile devices and supports grid services of diverse devices. Mobile grid may be constructed on current network infrastructure, integrate continually developing wireless network technologies; enrich network contents and software platform function. Mobile grid means that movable wireless devices are integrated into traditional wired grid through wireless channel to share grid resources (CPU power, storage capacity, instrument, devices, data, software, etc.), meanwhile mobile devices can provide service or resource to grid users, such as PDAs, cellular phones, handsets or wearable computers, laptops with GPS service, mobile service, etc. [1]. Mobile grid includes various kinds of mobile devices, and then leads to the grid system more complicated than wired grid system due to mobile grid node dynamical behavior in the grid system. However, mobile grid will bring users a more flexible and scalable computing environment. Wireless and mobile devices incorporated into Grid system can act as either service/resource consumer or service/resource provider. Unfortunately, in wireless environment these devices have some inherent characteristics: limited energy, lower and variable bandwidth, and intermittent connection. These hinder mobile grid feasibility and practicality. Energy conservation and improvement of system utility for mobile grids have become increasingly important issues.

This paper proposes simultaneous optimization of application utility and consumed energy in mobile grid. The paper provides a comprehensive utility function, which optimizes both the application level satisfaction such as execution success ratio and the system level requirements such as high resource utilization. The utility function models various aspects of job, application and system. The goal of maximizing the utility is achieved by decomposing the problem into a sequence of sub-problems that are then solved using the NUM optimization framework. The paper proposes a utility and energy optimization algorithm in mobile grid. The price-based iterative algorithms enable the sub-problems to be processed in parallel. The simulations and analysis are given to study the performance of the algorithm.

The rest of the paper is structured as follows. Section 2 discusses the related works. Section 3 presents simultaneous optimization of application utility and consumed energy in mobile grid. Section 4 presents utility and energy optimization algorithm in mobile grid. In Section 5 the simulations and analysis are given. Section 6 gives the conclusions to the paper.

2 RELATED WORKS

Energy efficiency for high performance computing and communication system has recently become a hot research area. Many works have been carried out on conserving energy, but those considering energy in grid computing are few. Y. Huang et al. [2] present techniques for exploiting intermittently available resources in grid infrastructures to support QoS-based multimedia applications on mobile devices. They integrate power aware admission control, grid resource discovery, dynamic load-balancing and energy adaptation techniques to enable power deficient devices to run distributed multimedia applications. Ziliang Zong et al. [3] design energy efficient scheduling algorithms for parallel applications running on clusters, they propose a scheduling strategy called energy efficient task duplication schedule, which can significantly conserve power by judiciously shrinking communication energy cost when allocating parallel tasks to heterogeneous computing nodes. Tarek A. AlEnawy et al. [4] propose to minimize the number of dynamic failures while remaining within the energy budget. They propose techniques to statically compute the speed of the CPU in order to meet the (m, k) -firm deadline constraints. Tao Xie et al. [5] address the issue of allocating tasks of parallel applications in heterogeneous embedded systems with an objective of energy-saving and latency-reducing. They proposed BEATA (Balanced Energy-Aware Task Allocation), a task allocation scheme considering both energy consumption and schedule length, developed to solve the energy-latency dilemma. Kyong Hoon Kim et al. [6] provide power-aware scheduling algorithms for bag of tasks applications with deadline constraints on DVS enabled cluster systems in order to minimize power consumption as well as to meet the deadlines specified by application users. Eunjeong Park et al. [7] designed an entire process of multimedia service composition for mobile computing. Their approach adapts the composition graph and the use of service routing for the context of mobile devices with the support of monitoring components. Network utility maximization (NUM) framework has recently gained much attention to make better use of network resources by optimizing across the boundaries of traditional network layers. In [17], the paper extends the distributed network utility maximization (NUM) framework to consider the case of resource sharing by multiple competing missions in a military-centric wireless sensor network (WSN) environment. They exploit joint-utility functions and multicast dissemination of sensor data. H. Nama et al. [18] 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. The cross-layer optimization problem decomposes vertically into three separate problems – the joint transport and routing problem, the radio resource allocation problem, and the network lifetime problem, which interact through the link and node-battery prices. Lin Xiao et al. [19] formulate the simultaneous routing and resource allocation problem as a convex optimization problem over the network flow variables and the communications variables. They exploit this separable structure by dual decomposition. The method attains the optimal coordination of data routing in the network layer and resource allocation in the radio control layer via pricing on the link capacities. The works [9–13] mainly deal with resource allocation, QoS optimization in the computational grid and do not consider energy consumption for mobile grid.

Motivated by the above works on energy efficient high performance computing and network utility maximization design method, we propose simultaneous optimization of application utility and consumed energy in mobile grid. The main differences between other peoples' works and our work are from three aspects. Firstly, the paper addresses the problem of simultaneous optimization of application utility and consumed energy in mobile grid. We investigate both energy minimization for mobile devices and grid utility optimization problem. Secondly, we solve the problem to find a system wide optimization by using a utility decomposition method. Thirdly, the paper adopts a pricing based iterative algorithm for energy constraint scheduling in mobile grid. The above three contributions do not appear in other related works.

3 SIMULTANEOUS OPTIMIZATION OF APPLICATION UTILITY AND CONSUMED ENERGY IN MOBILE GRID

In the mobile grid, for any mobile device $m_i \in M$ there are grid jobs arriving at m_i. The jobs are assumed to be computationally intensive, mutually independent, and can be executed at any mobile device. As soon as a job arrives, it must be assigned to one mobile device for processing. When a job is completed, the executing mobile device will return the results to the originating mobile device or ordinary fixed grid node of the job. We use J to denote the set of all jobs generated by grid application i, $J_i = \{J_i^1, J_i^2 \dots J_i^n\}$. Each grid job can be described as $J_i^n = (t_i^n, e_i^n)$, in which t_i^n stands for the time taken by the i^{th} grid application to complete n^{th} job, e_i^n stands for energy dissipation of n^{th} job. There are no dependencies among the jobs, so the submission order and completion order will not impact on the execution result. A user application set is represented as $A = \{A_1, A_2 \dots A_i\}$, for $1 \leq i \leq N$, grid application A_i submits a job, together with parameters including: T_i, which is the deadline limit of job completion time, B_i, which is the expense budget limit for all jobs, and E_i, which is a limited energy budget for all jobs. The system model is shown in Figure 1.

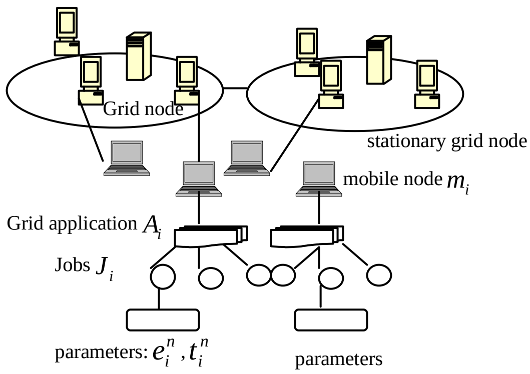

Fig. 1. System model

Energy consumption rate of each node in the system is measured by Joule per unit time. Let e_i^n be an energy dissipation caused by grid application i's n^{th} job, t_i^n be the execution time of job n of grid application i on the grid node. er is the energy consumption rate of energy resource l. If the energy consumption is proportional to execution time of job n, as is the case with battery energy, the energy dissipation of grid application i's n^{th} job can be written as follows:

$$e_i^n = er.t_i^n.$$

We assume that the mobile grid has heterogeneous nodes with different system performance rates and network conditions. This means that the energy consumption of the mobile device can vary with the response time of the application and the network bandwidth. We denote by e_i^l the consumed energy fraction of the energy l (e.g. a battery) by grid application i. Total consumed energy of all grid applications $\sum_{i=1}^{I} e_i^l$ does not exceed the total capacity Ce_l of energy l. We define the energy consumption of each application by A_i as the sum of the energy consumed by N grid jobs $\sum_{n=1}^{N} e_i^n$. The energy consumption of all grid jobs of each application A_i should be lower than the available resources of e_i^l which is the limited energy budget of grid user application i.

Now, we formulate the problem of simultaneous optimization of application utility and consumed energy in mobile grid as constraint optimization problem, the total utility U_{total} is defined as the sum of grid application utilities. The utility function for application A_i depends on allocated resources x_i^j, y_i^k and consumed energy e_i^l. e_i^l is the energy obtained by grid application i from the energy l. x_i^j is CPU allocation obtained by grid application i from the computing resource provider j. y_i^k is bandwidth allocation obtained by grid application i from the network resource provider k. The objective of the simultaneous optimization of application utility and consumed energy is to maximize the utility of the system U_{total} without exceeding the resource capacity, the energy budget, expense budget and the deadline. We formalize the problem using nonlinear optimization theory; the simultaneous optimization of application utility and consumed energy can be formulated as follows:

$$\begin{gathered} \max U_{\text{total}} \\ \text{s.t.} B_i \geq \sum_{l=1}^{L} Pe_i^l + \sum_{j=1}^{J} Pc_i^j + \sum_{k=1}^{K} Pn_i^k, \sum_{i=1}^{I} e_i^l \leq Ce_l \\ T_i \geq \sum_{n=1}^{N} t_i^n, \mathrm{C}n_k \geq \sum_{i=1}^{I} y_i^k, \sum_{n=1}^{N} e_i^n \leq e_i^l, \mathrm{Cc}_j \geq \sum_{i=1}^{I} x_i^j. \end{gathered} \tag{1}$$

In the problem (1), the first type of constraints is related with different resource capacity. The QoS constraint implies that the aggregate network resource units $\sum_{i=1}^{I} y_i^k$ do not exceed the total capacity Cn_k of network resource provider k, aggregate consumed energy of all grid applications $\sum_{i=1}^{I} e_i^l$ does not exceed the total Ce_l of energy l, aggregate computing power $\sum_{i=1}^{I} x_i^j$ does not exceed the total resource Cc_j of the computing resource provider j. The second type of constraints is related with grid application expense budget. Grid application needs to complete a sequence of jobs in a specified amount of time, T_i, while the payment overhead accrued cannot exceed B_i, which is the expense budget of grid application i. Pe_i^l, Pc_i^j, Pn_i^k are the payments of the grid application i to the energy storage provider l, computing resource provider j and network resource provider k. The total payments of the grid application i $\sum_{l=1}^{L} Pe_i^l + \sum_{j=1}^{J} Pc_i^j + \sum_{k=1}^{K} Pn_i^k$ do not exceed B_i. The total energy consumed by all jobs of grid application i $\sum_{n=1}^{N} e_i^n$ cannot exceed the energy budget e_i^l which is the available energy obtained by grid application i from the energy storage l.

Let us consider the Lagrangian form of simultaneous optimization of application utility and consumed energy in mobile grid:

$$\begin{aligned} L \;=\; & \textstyle\sum_{i=1}^{I} U_i\left(e_i^l, x_i^j, y_i^k\right) - \lambda_i\left(\sum_{i=1}^{I} e_i^l - Ce_l\right) - \beta_i\left(\sum_{i=1}^{I} x_i^j - Cc_j\right) \\ & - \varphi_i\left(\textstyle\sum_{i=1}^{I} y_i^k - Cn_k\right) - \gamma_i\left(\sum_{l=1}^{L} Pe_i^l + \sum_{j=1}^{J} Pc_i^j + \sum_{k=1}^{K} Pn_i^k - B_i\right) \\ & - \mu_i\left(\textstyle\sum_{n=1}^{N} e_i^n - e_i^l\right) - \alpha_i\left(\sum_{n=1}^{N} t_i^n - T_i\right) \end{aligned} \tag{2}$$

where λ_i, β_i and φ_i are the Lagrange multipliers of grid application with their interpretation of energy price, computing resource capacity price, and network resource capacity price, respectively. Since the Lagrangian is separable, this maximization of Lagrangian over (x_i^j, y_i^k, e_i^l) can be conducted in parallel at each application A_i. In problem (1), although the allocated resources x_i^j, y_i^k and consumed energy e_i^l are coupled in their constraints, respectively, they are separable. Given that the grid knows the utility functions U of all the grid applications, this optimization problem can be mathematically tractable. However, in practice, it is not likely to know each application's utility, and it is also infeasible for mobile grid environment to compute and allocate resources in a centralized fashion. To derive a distributed algorithm to solve problem (1), we decompose the problem into subproblems.

In the paper, maximization formulation of the grid system utility adopts a network utility maximization (NUM) framework [14] in which each application has an associated utility function. In [14], an optimization framework leads to a decomposition of the overall system problem into a separate problem for each user, in which the user chooses a charge per unit time that the user is willing to pay, and one for the network. The network's optimization problem leads to two classes of algorithm, which may be interpreted in terms of either congestion indication feedback signals or explicit rates based on shadow prices. It was shown that a system optimum is achieved when users' choices of charges and the network's choice of allocated rates are in equilibrium.

The grid total utility denoted as the sum of grid application utility can be defined as follows (3):

$$\begin{aligned} U_{\text{total}} \;=\; & \left(B_i - \textstyle\sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k\right) + \left(T_i - \sum_{n=1}^{N} t_i^n\right) \\ & + \left(e_i^l - \textstyle\sum_{n=1}^{N} e_i^n\right) + \sum_{i=1}^{I}\left(Pe_i^l \log e_i^l + Pc_i^j log x_i^j + Pn_i^k log y_i^k\right). \end{aligned} \tag{3}$$

Grid system utility functions are maximally optimized with specific constraints. In (3), $Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k$ present the revenue of energy storage resource, computing power and network resource provider. We could have chosen any other form for the utility that increases with x_i^j, y_i^k, e_i^l; but we chose the log function because the benefit increases quickly from zero as the total allocated resource increases from zero and then increases slowly. Moreover, log function is analytically convenient, increasing, strictly concave and continuously differentiable. The benefits of grid resource provider are affected by payments of grid applications and allocated resources. It means that the revenue increases with increasing allocated resources and increasing payment.

The Lagrangian form of the problem (1) can be reformulated as follows (4):

$$\begin{aligned} L \quad = \quad & \left(B_i - \sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k\right) + \left(T_i - \sum_{n=1}^{N} t_i^n\right) \\ & + \left(e_i^l - \sum_{n=1}^{N} e_i^n\right) - \lambda_i \left(\sum_{i=1}^{I} e_i^l - Ce_l\right) \\ & + \sum_{i=1}^{I} \left(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k\right) - \beta_i \left(\sum_{i=1}^{I} x_i^j - Cc_j\right) \\ & - \varphi_i \left(\sum_{i=1}^{I} y_i^k - Cn_k\right) - \gamma_i \left(\sum_{l=1}^{L} Pe_i^l + \sum_{j=1}^{J} Pc_i^j + \sum_{k=1}^{K} Pn_i^k - B_i\right) \\ & - \mu_i \left(\sum_{n=1}^{N} e_i^n - e_i^l\right) - \alpha_i \left(\sum_{n=1}^{N} t_i^n - T_i\right). \end{aligned} \tag{4}$$

The system model presented by (1) is a nonlinear optimization problem with N decision variables. Since the Lagrangian is separable, the maximization of the Lagrangian can be processed in parallel for grid user applications and grid resource providers. From (4), the resource allocation $\{\ e_i^l,\ x_i^j,\ y_i^k\ \}$ solves problem (1) if and only if there exists a set of nonnegative shadow costs $\{\lambda_i, \beta_i, \varphi_i\}$. Generally, solving such a problem by typical algorithms such as steepest decent method and gradient projection method is of high computational complexity, which is very time costing and impractical for implementation. In order to reduce the computational complexity, we decompose the utility optimization problem (1) into two subproblems for grid user applications and grid resource providers so that the computational complexity is reduced. The shadow costs suggest a mechanism to distribute the resource optimization between the grid applications and the grid system. The problem (1) maximizes the utility of grid applications on the energy price, computing power capacity price, and network resource capacity price; $\sum_{i=1}^{I} U_i(e_i^l, x_i^j, y_i^k)$ is the total utility of mobile grid system, $\beta_i \sum_{i=1}^{I} x_i^j$ is the computing power cost, $\lambda_i \sum_{i=1}^{I} e_i^l$ is the energy cost, $\varphi_i \sum_{i=1}^{I} y_i^k$ is the network resource cost. By decomposing the Kuhn-Tucker conditions into separate roles of consumer and supplier at grid market, the centralized problem (1) can be transformed into a distributed problem. Grid application's payment is collected by the resource providers. The payments of grid applications paid to resource providers are the payments to resolve the optimality of resource allocation in the grid market. We decompose the problem into the following two subproblems (5), namely grid user application QoS optimization problem and (6) which is grid resource providers optimization problem, seek a distributed solution where the grid resource provider does not need to know the utility functions of individual grid user application. Equations (5), (6) derived from the distributed approach are identical to the optimal conditions given by the centralized simultaneous optimization of application utility and consumed energy (1). This means if two subproblems converge to its optimal points, then a globally optimal point is achieved. Total user application benefit of the mobile grid is maximized when the equilibrium prices, obtained through the two subproblems (5) and (6), equal the Lagrangian multipliers λ_i, β_i and φ_i, where λ_i, β_i and φ_i are the optimal prices charged by resource providers including energy, computing power and network resource to grid applications. Two maximization subproblems correspond to grid user application QoS optimization problem as denoted by (5):

Sub 1:

$$\begin{aligned} \max U_{\mathrm{GA}} &= \left(B_i - \sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k\right) + \left(T_i - \sum_{n=1}^{N} t_i^n\right) \\ &\quad + \left(e_i^l - \sum_{n=1}^{N} e_i^n\right) \\ &= \sum_{n=1}^{N} e_i^n \le e_i^l, T_i \ge \sum_{n=1}^{N} t_i^n, B_i \ge \sum_{l=1}^{L} Pe_i^l + \sum_{j=1}^{J} Pc_i^j \\ &\quad + \sum_{k=1}^{K} Pn_i^k \end{aligned} \tag{5}$$

Sub 2:

$$\begin{aligned} \max U_{\mathrm{RP}} &= \sum_{i=1}^{I} \left(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k\right) \\ &= \sum_{i=1}^{I} e_i^l \le Ce_l, \mathrm{Cc}_j \ge \sum_{i=1}^{I} x_i^j, \mathrm{C}n_k \ge \sum_{i=1}^{I} y_i^k. \end{aligned} \tag{6}$$

In Problem Sub 1, the grid application gives the unique optimal payment to resource provider under the energy budget, expense budget and the deadline constraint to maximize the user's satisfaction. $(B_i - \sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k)$ represents the money surplus of grid application i, which is obtained by expense budgets subtracting the payments to various resource providers. $(T_i - \sum_{n=1}^{N} t_i^n)$ represents the saving times for grid application i, which is the result of time limit subtracting actual spending time. $(e_i^l - \sum_n e_i^n)$ represents the energy surplus of grid application i which is obtained by the energy budgets subtracting actual energy dissipation. So, the objective of Problem Sub 1 is to get more surpluses of money and more energy savings, and simultaneously complete task for grid user application as soon as possible. In Problem Sub 2, different resource providers compute optimal resource allocation for maximizing the revenue of their own. Grid application i submits the payment Pe_i^l to the energy resource provider l, Pn_i^k to network resource provider k and Pc_i^j to computing power provider j. $Pe_i^l \log e_i^l$ presents the revenue obtained by energy resource l from grid application i. $Pc_i^j \log x_i^j$ presents the revenue obtained by computing power j from grid application i. $Pn_i^k \log y_i^k$ presents the revenue obtained by network resource k from grid application i. The objective of resource providers is to maximize $Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k$ under the constraints of their provided resource amounts. Grid resource providers can't sell the resources to grid applications more than total capacity.

In Problem Sub 1, the grid application adaptively adjusts its payments to computing power, network resource and energy based on the current resource conditions, while in Problem Sub 2, the grid resource provider adaptively allocates energy, CPU and bandwidth required by the grid application in the Problem Sub 1. The interaction between two sub-problems is controlled through the use of the price variable λ_i,

β_i and φ_i, which is the energy price, computing power price, and network resource price charged from grid applications by grid energy resource, computing power and network resource. The interaction between two sub-problems also coordinates the grid application's payment and the supply of grid resource providers.

Simultaneous optimization of application utility and consumed energy problem involves variables from grid applications and resource providers. Lagrange relaxation and gradient optimization can be applied to decompose such an overall optimization problem into a sequence of two sub-problems, each only involving variables from the grid application and resource providers, respectively. Interactions between the two sub-problems are through optimal price variables.

In Problem Sub 1, grid application maximizes its satisfaction and gives the unique optimal payment to resource provider under the energy budget, expense budget and the deadline constraint. Grid application optimization problem can be written as

Sub 1:

$$\begin{aligned}\max U_{\mathrm{GA}} &= \left(B_i - \textstyle\sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k\right) + \left(T_i - \textstyle\sum_{n=1}^{N} t_i^n\right)\\ &\quad + \left(e_i^l - \textstyle\sum_{n=1}^{N} e_i^n\right)\\ &= \textstyle\sum_{n=1}^{N} e_i^n \le e_i^l, T_i \ge \sum_{n=1}^{N} t_i^n, B_i \ge \sum_{l=1}^{L} Pe_i^l + \sum_{j=1}^{J} Pc_i^j\\ &\quad + \textstyle\sum_{k=1}^{K} Pn_i^k.\end{aligned}$$

Theorem 1. There exist $Pe_i^{l^*}$, $Pc_i^{j^*}$, $Pn_i^{k^*}$ which are optimal payments of grid application i paying for energy resource l, computing power j and network resource k to execute grid jobs under completion time constraint.

The proof is in Appendix.

In problem Sub 2, different resource providers compute optimal resource allocation for maximizing the revenue of their own under constrains of resource capacity $Ce_l, \mathrm{Cc}_j, \mathrm{C}n_k$. The objective of resource providers is to maximize $Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k$ under the constraints of their resource capacity.

Sub 2:

$$\begin{aligned}\max U_{\mathrm{RP}} &= \textstyle\sum_{i=1}^{I} \left(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k\right)\\ &= \textstyle\sum_{i=1}^{I} e_i^l \le Ce_l, \mathrm{Cc}_j \ge \sum_{i=1}^{I} x_i^j, \mathrm{C}n_k \ge \sum_{i=1}^{I} y_i^k\end{aligned}$$

Theorem 2. There exist e_i^{l*}, x_i^{j*}, z_i^{l*} which are the unique optimal resource allocation to grid application i for maximizing the revenue of energy provider l, computing power provider j and network resource provider k.

The proof is in Appendix.

4 UTILITY AND ENERGY OPTIMIZATION ALGORITHM IN MOBILE GRID

Utility and energy optimization in mobile grid is targeted to maximize the utility of the grid system. The proposed algorithm decomposes simultaneous optimization of application utility and consumed energy problem into a sequence of sub-problems via an iterative algorithm. Figure 2 shows the activities of different parts of the algorithm. The iterative algorithm that achieves utility and energy optimization in mobile grid is described as follows.

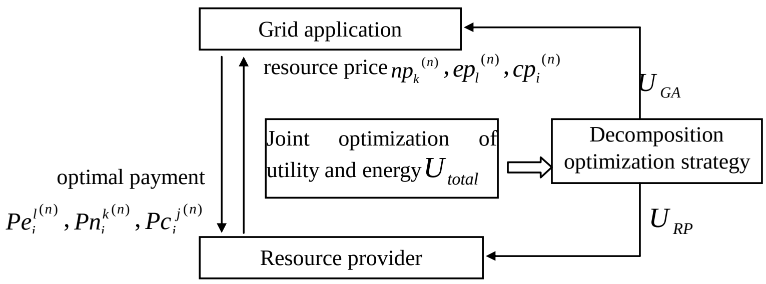

Fig. 2. Operations in Algorithm 1

Algorithm 1 Utility and Energy Optimization Algorithm in Mobile Grid (UEOA)

Grid Application i behavior

Receives the new price ep_l from the energy provider l;
$Pe_i^{l*} = Max\{U_{app}\left(Pe_i^l, Pc_i^j, Pn_i^k\right)\}$; // Calculates Pe_i^{l*} to maximize $U_{app}\left(Pe_i^l, Pc_i^j, Pn_i^k\right)$
If $B_i \geq \sum_j Pc_i^j + \sum_k Pn_i^k + \sum_l Pe_i^l$
Then Return Pe_i^{l*} to energy resource l;
Else Return Null;
Receives the new price cp_j from the computing power j;
$Pc_i^{j*} = Max\{U_{app}\left(Pe_i^l, Pc_i^j, Pn_i^k\right)\}$; //calculates Pc_i^{j*} to maximize $U_{app}\left(Pe_i^l, Pc_i^j, Pn_i^k\right)$
If $B_i \geq \sum_j Pc_i^j + \sum_k Pn_i^k + \sum_l Pe_i^l$
Then Return Pc_i^{j*} to computing power j;
Else Return Null;
Receives the new price np_k from the network resource provider k;
$Pn_i^{k*} = Max\{U_{app}\left(Pe_i^l, Pc_i^j, Pn_i^k\right)\}$; // Calculates Pn_i^{k*} to maximize $U_{app}\left(Pe_i^l, Pc_i^j, Pn_i^k\right)$
If $B_i \geq \sum_j Pc_i^j + \sum_k Pn_i^k + \sum_l Pe_i^l$
Then Return Pn_i^{k*} to network resource k;
Else Return Null;

Computing power j, network resource k and energy resource l
Receives optimal payments Pe_i^{l*}, Pc_i^{j*}, Pn_i^{k*} from grid application i;
If $Ce_l \geq \sum_{i=1}^{I} e_i^l$
Then
$e_i^{l(n+1)*} = Max\{U_{resource}(e_i^l, x_i^j, y_i^k) = \sum_{i=1}^{I}(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k)\}$;
// Calculates its optimal energy resource $e_i^{l(n+1)*}$
$ep_l^{(n+1)} = \max\{\varepsilon, ep_l^{(n)} + \eta(e^l ep_l^{(n)} - Ce_l)\}$; // Computes a new price
// $e^l = \sum_{i=1}^{I} e_i^l$, $\eta > 0$ is a small step size parameter, n is iteration number.
Return energy resource price $ep_l^{(n+1)}$ to all grid applications;
Else Return Null;
If Cc$_i \geq \sum_{i=1}^{I} x_i^j$
Then
$x_i^{j(n+1)*} = Max\{U_{resource}(e_i^l, x_i^j, y_i^k) = \sum_{i=1}^{I}(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k)\}$;
// Calculates its optimal computing power $x_i^{j(n+1)*}$
$cp_j^{(n+1)} = \max\{\varepsilon, cp_j^{(n)} + \eta(x^j cp_j^{(n)} - Cc_j)\}$; // Computes a new price
// $x^j = \sum_i x_i^j$, $\eta > 0$ is a small step size parameter, n is iteration number
Return computing power price $cp_j^{(n+1)}$ to all grid applications;
Else Return Null;
If $Cn_k \geq \sum_{i=1}^{I} y_i^k$
Then
$y_i^{k n+1)*} = Max\{U_{resource}(e_i^l, x_i^j, y_i^k) = \sum_{i=1}^{I}(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k)\}$;
// Calculates its optimal network resource demand $y_i^{k(n+1)*}$
$np_k^{(n+1)} = \max\{\varepsilon, np_k^{(n)} + \eta(y^k np_k^{(n)} - Cn_k)\}$; // Computes a new price
// $y^k = \sum_i y_i^k$, $\eta > 0$ is a small step size parameter, n is iteration number
Return network resource price $np_k^{(n+1)}$ to all grid applications;
Else Return Null;

5 SIMULATIONS

In this section, we present the performance evaluation of our utility and energy optimization algorithm in mobile grid (UEOA). Our simulator supports a topology of multiple LANs connected through wired nodes and wireless LANs, and bandwidth monitoring. The proposed simulator considers mobile grid environment parameters such as the battery (power) state, the network state (latency and bandwidth), and the system loading state (CPU and memory). The grid simulator is implemented on top of the JAVASIM network simulator [15]. In order to simulate the dynamics and heterogeneity of the Grid, all values of networks can be changed after topology generation. Network generator BRITE [16] generates the computer network topology. BRITE is a random network topology generator used to generate the simulation testbed. In the simulator, different agents are used, namely resource provider agents, user agents and grid scheduler agent which implements UEOA algorithm. The grid scheduler receives the task request, schedules the tasks 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 task requests. It receives the task requirements and puts them into the task queue. While the task queue is not empty, the grid scheduler starts the scheduling algorithm to find the right match. When resource agent updates its price, the resource agent forwards the price to user agents; the resource price is put in a packet. Whenever the new price packet passes to user agent, the user agent calculates the utility. According to the algorithm, if the price becomes higher than its maximum willingness to pay, user agent does not buy grid resource. The user agent can be informed about the price for the next iteration by the next price packets.

We simulate a grid environment containing 10 grid domains. To simulate grid domain, we profile each node in domain with a group of parameters to represent arrival rate, machine computing power, energy state and communication bandwidth. We assume that each grid application can use any of grid resources including computation, communication and energy resources. Processor capacity varies from 220 to 580 MIPS. The wireless network bandwidth is from 10 Kbps to 1 Mbps. The main memory is set by 128 M, 256 M, 512 M, and 2 G. The disk capacity is set by 80 G, 30 G, 20 G. The selective grid applications for simulation are computation-intensive applications such as image processing applications and mpeg players. The simulator leaves each application on the mobile device or delegates it to a grid node. There are a total of 150 resources and 600 applications are taken for experimental evaluation of the system. Energy consumption is represented as a percentage of the total energy required to meet all job deadlines. Assume that the maximum power, $P_{\max}$, corresponds to running all jobs with the maximum processing frequency. The maximum frequency is assumed to be $f_{\max} = 1$ and the maximum frequency-dependent power is $P_{\max} = 1$. When the energy budget for each interval is limited, we can only consume a fraction of $P_{\max}$ when processing requests during a given interval. Jobs arrive at each site s_i, $i = 1, 2, \ldots, n$ according to a Poisson process with rate α. 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 $[1, 100]$. The capacities of the resources were also chosen uniformly in the interval $[50, 500]$. The resource cost can be expressed in grid dollar that can be defined as unit processing cost. The initial price of resource is set from 10 to 500 grid dollars. Users submit their jobs with varying deadlines. The deadlines of users are chosen from 100 ms to 400 ms. The budgets of users are set from 100 to 1 500 grid dollars. Each measurement is run 6 times with different seeds. Simulation parameters are listed in Table 1.

The experiments are conducted to compare our utility and energy optimization algorithm in mobile grid (UEOA) with low-energy earliest deadline-first (LEDF) scheduling algorithm [8] proposed by Vishnu Swaminathan et al. The reason for choosing LEDF as the comparison is that both our work and LEDF deal with energy and QoS constrained scheduling. Vishnu Swaminathan et al. [8] studied scheduling workloads containing periodic tasks in real-time systems. The proposed

Simulation Parameter	Value
Total number of applications	600
Total number of resource providers	150
Reschedule Interval	600 ms
Initial price of computing power (grid dollar)	$[10, 500]$
Deadline	$[100\,\mathrm{ms}, 400\,\mathrm{ms}]$
Expense Budget	$[100, 1\,500]$
Energy Budget	$[0.1, 1.0]$
Resource capacities	$[50, 500]$
Load factor	$[0.1, 0.9]$

Table 1. Simulation parameters

approach minimizes the total energy consumed by the task set and guarantees that the deadline for every periodic task is met. They present a mixed-integer linear programming model for the NP-complete scheduling problem. They proposed a low-energy earliest deadline-first (LEDF) scheduling algorithm. The process of the low-energy earliest deadline first (LEDF) is as follows. LEDF maintains a list of all released tasks, called the *ready list*. When tasks are released, the task with the nearest deadline is chosen to be executed. A check is performed to see if the task deadline can be met by executing it at the lower voltage (speed). If the deadline can be met, LEDF assigns the lower voltage to the task and the task begins execution. During the task's execution, other tasks may enter the system. These tasks are placed automatically on the *ready list*. LEDF again selects the task with the nearest deadline to be executed. As long as there are tasks waiting to be executed, LEDF does not keep the processor idle. This process is repeated until all the tasks have been scheduled.

In the simulation, we compare utility and energy optimization algorithm in mobile grid (UEOA) with low-energy earliest deadline-first (LEDF) scheduling algorithm by varying load factor and price to study how they affect the performance of two algorithms. The performance metrics include energy consumption ratio, resource utilization, execution success ratio and allocation efficiency. Energy consumption ratio is defined as the percentage of consumed energy among total available energy resources. Execution success ratio is the percentage of tasks executed successfully before their deadline. Resource utilization is the ratio of the consumed resources to the total resources available as a percentage, commonly refers to the percentage of time a resource is busy. Allocation Efficiency is a measure of the efficiency of the allocation process, which is computed using the number of all requests and number of accepted requests. System load is defined as the ratio of the total number of requests arrived in one interval over the number of requests that can be handled by the system within one interval. The value of system load expresses the extent to which the whole system is busy. If in a certain period of time the number of jobs submitted to the system is low, the system load is light; otherwise, the system load is heavy. System load influences the performance of scheduling inherently.

Load factor (LF) varies from 0.1 to 0.9. The resource price (p) is set from 10 to 500 grid dollars.

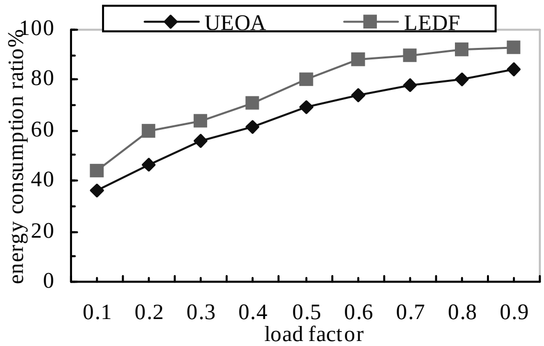

Fig. 3. Energy consumption ratio under various load factor

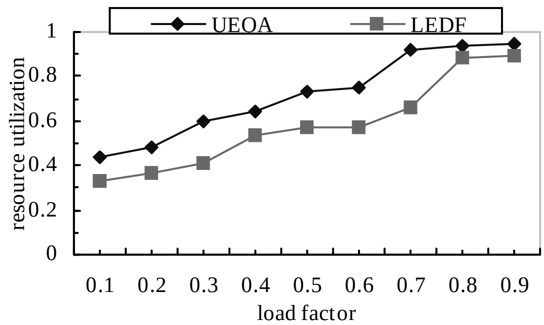

Fig. 4. Resource utilization under various load factor

The impacts of different load factor on energy consumption ratio, resource utilization, execution success ratio and allocation efficiency were illustrated in Figures 3–6, respectively. Load factor varies from 0.1 to 0.9. Figure 3 shows the effect of load factor (LF) on the energy consumption ratio. When the load factor increases, more requests need to be processed within one interval and the energy consumption ratio increases. When increasing the load factor by $LF = 0.7$, the energy consumption ratio of UEOA is as much as 21 % more than $LF = 0.2$. Under the same load factor ($LF = 0.6$), the energy consumption ratio of UEOA is 19 % less than that of LEDF. Figure 4 shows as load factor increases, resource utilization ratio increases. When $LF = 0.5$, the resource utilization of UEOA is as much as 44 % more than the utilization by $LF = 0.10$. When load factor was very large, many jobs will be sent to system, resources are busier. Compared with LEDF, the resource utilization of UEOA decreases slower than LEDF when the load factor decreases. When load factor is 0.1 ($LF = 0.1$), resource utilization of LEDF decreases to 34 %, resource

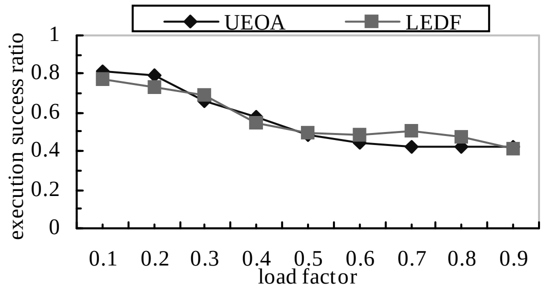

Fig. 5. Execution success ratio under various load factor

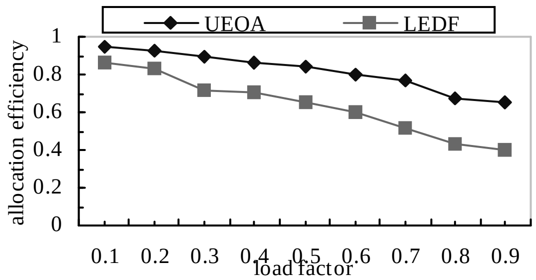

Fig. 6. Allocation efficiency under various load factor

utilization of UEOA decreases to 47 %. Figure 5 shows that the execution success ratio decreases when load factor increases. When $LF = 0.5$, execution success ratio of UEOA is as much as 17 % lower than that by $LF = 0.10$. The smaller LF, the lower system load; grid resources are available for grid users. The requirements of the users can be processed on time and these users experience higher execution success ratio. So the smaller LF, the higher execution success ratio. Under the same load factor ($LF = 0.5$), UEOA has 19 % higher execution success ratio than LEDF. Figure 6 shows when load factor increases ($LF = 0.5$), allocation efficiency of UEOA is as much as 21 % less than that with $LF = 0.1$. The allocation efficiency is larger when the load factor LF is smaller. The value of LF is low, the system is lightly loaded, the unit price of the resource is cheap; user application can choose more resources to complete tasks under the deadline, so the allocation efficiency is high. When the system is heavily loaded, the unit price of the resource is expensive; the allocation efficiency is lower. Compared with LEDF, the allocation efficiency of UEOA decreases slower than LEDF when the load factor increases. When the load factor is 0.6 ($LF = 0.6$), the allocation efficiency of LEDF decreases to 49 %, the allocation efficiency of UEOA decreases to 70 %.

The impacts of the price on resource utilization, energy consumption ratio, execution success ratio, allocation efficiency were illustrated in Figures 7–10, respectively. The resource price (p) is set from 10 to 500 grid dollars. From the results

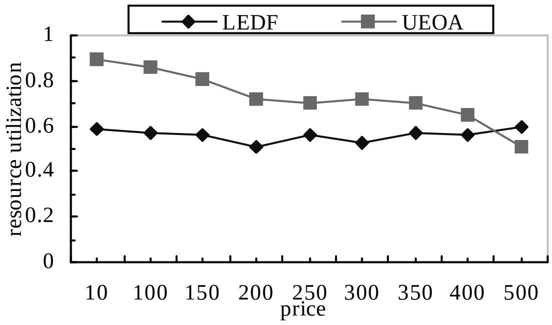

Fig. 7. Resource utilization vs. price

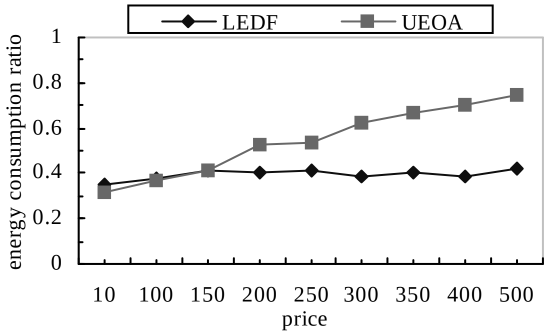

Fig. 8. Energy consumption ratio vs. price

in Figure 7, as the price is higher, the resource utilization becomes lower. When $p = 500$, the resource utilization of UEOA is as much as 28 % less than utilization by $p = 100$, because when the price increases quickly, the users with low expense budget will be prevented from obtaining resources. The smaller p, the lower the

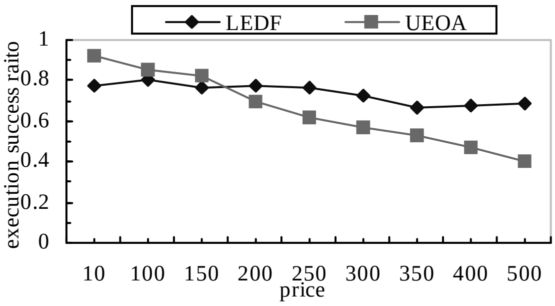

Fig. 9. Execution success ratio vs. price

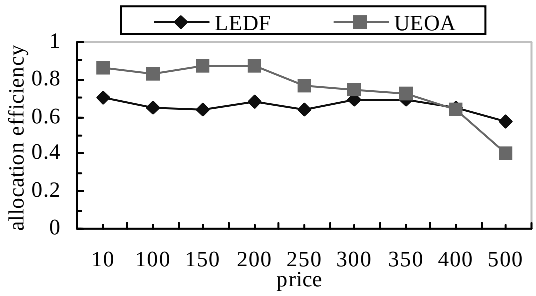

Fig. 10. Allocation efficiency vs. price

energy consumption ratio (cf. Figure 8). When price becomes high, users will afford more payment to obtain energy-consuming resource, some tasks cannot be completed before their deadlines. Price increasing quickly results in that some users with low budget cannot be satisfied to fulfil their achievements. When $p = 500$, energy consumption ratio of UEOA is as much as 34 % more than that by $p = 100$. Considering the execution success ratio, the results of Figure 9 show that when increasing price values, the execution success ratio becomes lower, because when price becomes high, grid users will afford more payment to obtain the grid resource, some users with low budget will not complete tasks before their deadlines. When price increases ($p = 500$), execution success ratio of UEOA is as much as 39 % less than that with $p = 10$. Considering the allocation efficiency, the results of Figure 10 show that when increasing p, the allocation efficiency become lower. Increasing prices of resource provider will prevent users from being admitted by the system, fewer users will exploit the resources. When $p = 500$, the allocation efficiency is reduced to nearly 42 % compared with $p = 10$.

6 CONCLUSIONS

This paper proposes simultaneous optimization of application utility and consumed energy in mobile grid. The paper provides a comprehensive utility function, which optimizes both the application level satisfaction such as execution success ratio and the system level requirements such as high resource utilization. Utility functions are used to express grid users' requirements, resource providers' benefit function and system's objectives. Dynamic programming is used to optimize the total utility function. A distributed algorithm in mobile grid environment is proposed which decomposes mobile grid system optimization problem into sub-problems. The simulations and analysis are given to study the performance of the algorithm.

Acknowledgement

The authors thank the editor and the anonymous reviewers for their helpful comments and suggestions. The work was supported by the Open Fund of the State Key Laboratory of Software Development Environment under Grant No. SKLSDE-2009KF-2-02, Beijing University of Aeronautics and Astronautics, and by the National Natural Science Foundation of China (NSF) under grants (No. 60773211, No. 60970064), and by Program for New Century Excellent Talents in University, China (NCET-08-0806), and by the National Science Foundation of HuBei Province under Grant No. 2008CDB335. Fok Ying Tong Education Foundation, China (Grant No. 121067), Fundamental Research Funds for the Central Universities (2010-II-003, 2010-ZX-031) and the Scientific and Technological Project in WuHan under Grant No. 201010621207. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

7 APPENDIX

7.1 Proofs for Theorem 1

We assume that grid application i submits Pe_i^l to energy resource l, Pc_i^j to computing power j and Pn_i^k to network resource k. Then, $Pe_i = [Pe_i^1 \dots Pe_i^l]$ represents all payments of grid applications for energy resource l, $Pc_i = [Pc_i^1 \dots Pc_i^j]$ represents all payments of grid applications for computing power j, $Pn_i = [Pn_i^1 \dots Pn_i^k]$ represents all payments of grid applications for the network resource k. Let $m_i = \sum_l Pe_i^l + \sum_j Pc_i^j + \sum_k Pn_i^k$, m_i be the total payment of the i^{th} grid application. N grid applications 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 applications. Let ep_l, cp_j, np_k denote the price of the resource unit of energy resource l, the price of the resource unit of computing power j and network resource k, respectively. Let the pricing policy, $ep = (ep_1, ep_2, \dots, ep_l)$, denote the set of resource unit prices of all the energy resources in the grid, $cp = (cp_1, cp_2, \dots, cp_j)$, denote the set of resource unit prices of all the computing powers, $np = (np_1, np_2, \dots, np_k)$ be set of network resource unit prices. The i^{th} grid application receives the resources proportional to its payment relative to the sum of the resource provider's revenue. Let e_i^l, x_i^j, y_i^k be the fraction of resource units allocated to grid application i by energy l, computing power j and network resource k.

$$x_i^j = Cc_j \frac{Pc_i^j}{cp_j}, \quad e_i^l = Ce_l \frac{Pe_i^l}{ep_l}, \quad y_i^k = Cn_k \frac{Pn_i^k}{np_k}$$

The time taken by the i^{th} grid application to complete the n^{th} job is:

$$t_i^n = \frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}$$

The energy dissipation used by the i^{th} grid user to complete the n^{th} job is:

$$e_i^n = er.t_i^n = er.\left(\frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}\right)$$

Problem Sub 1 can be reformulated as

$$\begin{aligned}\max U_{\mathrm{GA}} \quad = \quad & \left(B_i - \sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k\right) \\ & + \left(T_i - \sum_{n=1}^{N}\left(\frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}\right)\right) \\ & + \left(e_i^l - \sum_{n=1}^{N} er\left(\frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}\right)\right)\end{aligned}$$

The Lagrangian for the grid application's utility is $L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)$.

$$\begin{aligned}L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right) \quad = \quad & \left(B_i - \sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k\right) \\ & + \left(T_i - \sum_{n=1}^{N}\left(\frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}\right)\right) \\ & + \left(e_i^l - \sum_{n=1}^{N} er\left(\frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}\right)\right) \\ & + \nu_i\left(B_i - \sum_{l=1}^{L} Pe_i^l - \sum_{j=1}^{J} Pc_i^j - \sum_{k=1}^{K} Pn_i^k\right) \\ & + \sigma_i\left(T_i - \sum_{n=1}^{N}\left(\frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}\right)\right) \\ & + \varepsilon_i\left(e_i^l - \sum_{n=1}^{N} er\left(\frac{cq_i^n cp_j}{Cc_j Pc_i^j} + \frac{bq_i^n np_k}{Cn_k Pn_i^k} + \frac{eq_i^n ep_l}{Ce_l Pe_i^l}\right)\right)\end{aligned}$$

where $\varepsilon_i, \sigma_i, \nu_i$ is the Lagrangian constant. We know from Karush-Kuhn-Tucker Theorem that the optimal solution is given $\frac{\partial L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)}{\partial Pe_i^l} = 0$ for $\varepsilon_i, \sigma_i, \nu_i > 0$.

$$\begin{aligned}\frac{\partial L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)}{\partial Pe_i^l} \quad = \quad & -1 - \nu_i + \frac{eq_i^n ep_l}{Ce_l\left(Pe_i^l\right)^2} + er\frac{eq_i^n ep_l}{Ce_l\left(Pe_i^l\right)^2} \\ & + \sigma_i\frac{eq_i^n ep_l}{Ce_l\left(Pe_i^l\right)^2} + \varepsilon_i.er\frac{eq_i^n ep_l}{Ce_l\left(Pe_i^l\right)^2}\end{aligned}$$

Let $\frac{\partial L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)}{\partial Pe_i^l} = 0$ to obtain

$$Pe_i^l = \left(\frac{\left(1 + er + \sigma_i + \varepsilon_i.er\right) eq_i^n ep_l}{\left(1 + \nu_i\right) Ce_l}\right)^{1/2}$$

Using this result in the constraint equation, we can determine $\theta = \frac{(1+er+\sigma_i+\varepsilon_i.er)}{1+\nu_i}$ as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{m=1}^{N}\left(\frac{ep_m eq_i^n}{Ce_m}\right)^{1/2}}$$

We obtain Pe_i^{l*}

$$Pe_i^{l*} = \left(\frac{eq_i^n ep_l}{Ce_l}\right)^{1/2} \frac{\sum_{m=1}^{N}\left(\frac{eq_i^n ep_m}{Ce_m}\right)^{1/2}}{T_i}$$

It means that grid application wants to pay Pe_i^{l*} to energy resource l for needed energy consumed to execute grid jobs under completion time constraint.

$$\begin{aligned}\frac{\partial L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)}{\partial Pc_i^j} &= -1 + \frac{cq_i^n cp_j}{Cc_j\left(Pc_i^j\right)^2} + er_i^n \frac{cq_i^n cp_j}{Cc_j\left(Pc_i^j\right)^2} - \nu_i + \sigma_i \frac{cq_i^n cp_j}{Cc_j\left(Pc_i^j\right)^2} \\ &\quad + \varepsilon_i.er \frac{cq_i^n cp_j}{Cc_j\left(Pc_i^j\right)^2}\end{aligned}$$

Let $\frac{\partial L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)}{\partial Pc_i^j} = 0$ to obtain

$$Pc_i^j = \left(\frac{(1 + er + \sigma_i + \varepsilon_i.er)\, cq_i^n cp_j}{(1+\nu_i)\, Cc_j}\right)^{1/2}$$

Using this result in the constraint equation, we can determine $\xi = \frac{(1+er+\sigma_i+\varepsilon_i.er)}{1+\nu_i}$ as

$$(\xi)^{-1/2} = \frac{T_i}{\sum_{m=1}^{N}\left(\frac{cp_m cq_i^n}{Cc_m}\right)^{1/2}}$$

We obtain Pc_i^{j*}

$$Pc_i^{j*} = \left(\frac{cq_i^n cp_j}{Cc_j}\right)^{1/2} \frac{\sum_{m=1}^{N}\left(\frac{cq_i^n cp_m}{Cc_m}\right)^{1/2}}{T_i}$$

It means that grid application wants to pay Pc_i^{j*} to computing power j for needed resource to execute grid jobs under completion time constraint.

$$\begin{aligned}\frac{\partial L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)}{\partial Pn_i^k} &= -1 + \frac{bq_i^n np_k}{Cn_k\left(Pn_i^k\right)^2} + er_i^n \frac{bq_i^n np_k}{Cn_k\left(Pn_i^k\right)^2} - \nu_i \\ &\quad + \sigma_i \frac{bq_i^n np_k}{Cn_k\left(Pn_i^k\right)^2} + \varepsilon_i \frac{bq_i^n np_k}{Cn_k\left(Pn_i^k\right)^2}\end{aligned}$$

Let $\frac{\partial L_1\left(Pe_i^l, Pc_i^j, Pn_i^k\right)}{\partial Pn_i^k} = 0$ to obtain

$$Pn_i^k = \left(\frac{(1 + er + \sigma_i + er.\varepsilon_i)\, bq_i^n np_k}{(1+\nu_i)\, Cn_k}\right)^{1/2}$$

Using this result in the constraint equation, we can determine $\tau = \frac{(1+er+\sigma_i+er.\varepsilon_i)}{1+\nu_i}$ as

$$(\tau)^{-1/2} = \frac{T_i}{\sum_{m=1}^{N}\left(\frac{np_m bq_i^n}{Cn_m}\right)^{1/2}}$$

We obtain $Pn_i^{k^*}$

$$Pn_i^{k^*} = \left(\frac{bq_i^n np_k}{Cn_k}\right)^{1/2} \frac{\sum_{m=1}^{N}\left(\frac{bq_i^n np_m}{Cn_m}\right)^{1/2}}{T_i}$$

It means that grid application wants to pay $Pn_i^{k^*}$ to network resource k for needed resource to execute grid jobs under completion time constraint.

7.2 Proofs for Theorem 2

We take derivative and second derivative with respect to x_i:

$$U'_{\mathrm{RP}}\left(e_i^l\right) = \frac{Pe_i^l}{e_i^j} U''_{\mathrm{RP}}\left(e_i^l\right) = -\frac{Pe_i^l}{e_i^{l2}}$$

$U''_{\mathrm{RP}}\left(e_i^l\right) < 0$ is negative due to $0 < e_i^l$.The extreme point is the unique value maximizing the revenue of energy provider. The Lagrangian for Problem Sub 2 is $L_2\left(e_i^l, x_i^j, y_i^k\right)$.

$$\begin{aligned}
\mathrm{L}_2\left(e_i^l, x_i^j, y_i^k\right) &= \sum\left(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k\right) \\
&\quad + \lambda_i\left(Ce_l - \textstyle\sum_i e_i^l\right) + \beta_i\left(Cc_j - \textstyle\sum_i x_i^j\right) + \varphi_i\left(Cn_k - \textstyle\sum_i y_i^k\right) \\
&= \sum\left(Pe_i^l \log e_i^l + Pc_i^j \log x_i^j + Pn_i^k \log y_i^k - \lambda_i e_i^l - \beta_i x_i^j - \varphi_i y_i^k\right) \\
&\quad + \lambda_i Ce_l + \beta_i Cc_j + \varphi_i Cn_k
\end{aligned}$$

where λ_i, β_i and φ_i, is the Lagrangian constant. We know from Karush-Kuhn-Tucker Theorem that the optimal solution is given $\frac{\partial L_2\left(e_i^l, x_i^j, y_i^k\right)}{\partial e_i^l} = 0$ for $\lambda_i, \beta_i, \varphi_i > 0$.

Let $\frac{\partial L_2\left(e_i^l, x_i^j, y_i^j\right)}{\partial e_i^l} = 0$ to obtain $e_i^l = \frac{Pe_i^l}{\lambda_i}$.

Using this result in the constraint equation $Ce_l \geq \sum e_i^l$, we can determine λ_i as

$$\lambda_i = \frac{\sum_{d=1}^{n} Pe_i^d}{Ce_l}$$

We substitute λ into e_i^l to obtain

$$e_i^{l*} = \frac{Pe_i^l Ce_l}{\sum_{d=1}^{n} Pe_i^d}$$

e_i^{l*} is the unique energy allocation for maximizing the revenue of energy provider l.

Using the similar method, we can solve computing power allocation optimization problem.

Let $\frac{\partial L_2\left(e_i^l, x_i^j, y_i^k\right)}{\partial x_i^j} = 0$ to obtain $x_i^j = \frac{Pc_i^j}{\beta_i}$.

Using this result in the constraint equation $Cc_j \geq \sum x_i^j$, we can determine β_i as

$$\beta_i = \frac{\sum_{d=1}^{n} Pc_i^d}{Cc_j}$$

We substitute β into x_i^j to obtain

$$x_i^{j*} = \frac{Pc_i^j Cc_j}{\sum_{d=1}^{n} Pc_i^d}$$

x_i^{j*} is the unique optimal computing power allocation for maximizing the revenue of computing power provider j.

Using the similar method, we can solve network resource allocation optimization problem.

Let $\frac{\partial L_2\left(e_i^l, x_i^j, y_i^k\right)}{\partial y_i^k} = 0$ to obtain $y_i^k = \frac{Pn_i^k}{\varphi_i}$.

Using this result in the constraint equation $Cn_k \geq \sum y_i^k$, we can determine φ_i as

$$\varphi_i = \frac{\sum_{d=1}^{n} Pn_i^d}{Cn_k}$$

We substitute φ into y_i^k to obtain

$$y_i^{k*} = \frac{Pn_i^k Cn_k}{\sum_{d=1}^{n} Pn_i^d}$$

y_i^{k*} is the unique optimal network resource allocation for maximizing the revenue of network resource provider k.

REFERENCES

[1] Jameel, H.—Kalim, U.—Sajjad, A.—Ee, S.—Jeon, T.: CityMobile-to-Grid Middleware: Bridging the Gap Between CityMobile and Grid Environments. P.M.A. Sloot et al. (Eds.): EGC 2005, LNCS 3470, pp. 932–941, 2005.

[2] Huang, Y.—Mohapatra, S.—Venkatasubramanian, N.: An Energy-Efficient Middleware for Supporting Multimedia Services in Mobile Grid Environments. IEEE International Conference on Information Technology, 2005.

[3] Zong, Z.—Qin, X.: Energy-Efficient Scheduling for Parallel Applications Running on Heterogeneous Clusters. International Conference on Parallel Processing (ICPP 2007), IEEE Press, 2007.

[4] AlEnawy, T. A.—Aydin, H.: Energy-Constrained Scheduling for Weakly-Hard Real-Time Systems. Proceedings of the 26th IEEE International Real-Time Systems Symposium (RTSS '05), IEEE Pres, 2005.

[5] Xie, T.—Qin, X.—Nijim, M.: Solving Energy-Latency Dilemma: Task Allocation for Parallel Applications in Heterogeneous Embedded Systems. Proceedings of the 2006 International Conference on Parallel Processing (ICPP '06), IEEE Press 2006.

[6] Kim, K. H.—Buyya, R.—Kim, J.: Power Aware Scheduling of Bag-of-Tasks Applications with Deadline Constraints on DVS-enabled Clusters. Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid, IEEE Computer Society, Washington, DC, USA, pp. 541–548, 2007.

[7] Park, F.—Shin, H.: Multimedia Service Composition for Context-Aware City-Mobile Computing. T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4352, Part II, pp. 115–124, 2007.

[8] Swaminathan, V.—Chakrabarty, K.: Real-Time Task Scheduling for Energy-Aware Embedded Systems. Journal of the Franklin Institute, Vol. 338, September 2001, pp. 729–750.

[9] Li, C.—Li, L.: Joint QoS Optimization for Layered Computational Grid. Information Sciences, Vol. 177/15, August 2007, pp. 3038–3059.

[10] Li, C.—Li, L.: Agent Framework to Support Computational Grid. Journal of Systems and Software, Vol. 70/1-2, 2004, pp. 177–187.

[11] Li, C.—Li, L.: Multi Economic Agent Interaction for Optimizing the Aggregate Utility of Grid Users in Computational Grid. Applied Intelligence, Vol. 25/2, 2006, pp. 147–158.

[12] Li, C.—Li, L.: A Distributed Utility-based Two Level Market Solution for Optimal Resource Scheduling in Computational Grid. Parallel Computing, Vol. 31/3-4, 2005, pp. 332–351.

[13] Li, C.—Li, L.: Utility Based QoS Optimisation Strategy for Multi-Criteria Scheduling on the Grid. Journal of Parallel and Distributed Computing, Vol. 67/2, 2007, pp. 142–153.

[14] Kelly, F.—Maulloo, A.—Tan, D.: Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. J. of Operational Res. Soc., Vol. 49, 1998, pp. 237–252.

[15] JAVASIM, `http://javasim.ncl.ac.uk`.

[16] BRITE, `http://www.cs.bu.edu/brite`.

[17] Eswaran, S.—Misra, A.—la Porta, T. F.: Utility-Based Adaptation in Mission-Oriented Wireless Sensor Networks. SECON 2008, pp. 278–286.

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

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

Chunlin Li received the M. E. in computer science from Wuhan Transportation University in 2000, and Ph. D. degree in computer software and theory from Huazhong University of Science and Technology in 2003. She now is a 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.

Layuan Li received B. E. degree in communication engineering from Harbin Institute of Military Engineering, China in 1970 and 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 PhD 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 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. He 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.