This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON COMPUTERS

On the Design of Mutually Aware Optimal Pricing and Load Balancing Strategies for Grid Computing Systems

Qin Zheng and Bharadwaj Veeravalli, *Senior Member, IEEE*

Abstract—Managing resources and cleverly pricing them on computing systems is a challenging task. Resource sharing demands careful load balancing and often strives to achieve a win-win situation between resource providers and users. Toward this goal, we consider a joint treatment of load balancing and pricing. We do not assume static pricing to determine load balancing, or vice versa. Instead, we study the relationship between the price that a computing node is charged and the load and revenue that it receives. We find that there exists an optimal price which maximizes the revenue. We then consider a multi-user environment and explore how the load from a user can be balanced on processors with existing loads. Finally, we derive an optimal price that maximizes the revenue in the multi-user environment. We evaluate the performance of the proposed algorithms through simulations.

Index Terms—Pricing, load balancing, cost, revenue, response time

1 Introduction

In this paper, we consider a grid computing system where compute nodes are heterogeneous and their prices vary dynamically. We do not assume that they are fully cooperative and unselfish. Further, we consider a multi-user environment where the current load needs to be balanced onto nodes with existing loads. Finally, we consider different objectives of users and providers where the user wants to minimize the cost and response time while the provider wants to maximize the revenue. This paper first studies how load balancing and pricing influence each other where the load on nodes and their charged prices are dynamic. We find out that the provider's revenue is maximized when its node is charged at a certain optimal price. This optimal price can be determined given the output of the underlying load balancing approach. Then the load is re-balanced with respect to these new optimal prices for the current job. Therefore, pricing and load balancing are "mutually aware".

1.1 Related Works on Load Balancing

The purpose of load balancing is to improve the performance of applications by allocating them properly on the computing nodes. Common performance metrics considered are response time or cost to applications, which are representative parameters for the service level agreement (SLA). Existing load balancing approaches are categorized in [8] as: 1) global approach, 2) cooperative approach, and 3) non-cooperative approach. In the global approach, a centralized decision maker minimizes the expected response time of the system. Many existing approaches fall into this category. In the cooperative approach, the computing nodes cooperate in making the decisions such that each of them will operate at its optimum. This is implemented by a cooperative game theory [13]. In the non-cooperative approach, each decision maker attempts to minimize its own response time by playing a non-cooperative game with other decision makers. The equilibrium reached eventually is called *Nash equilibrium* [2], which is a strategy profile with the property that no decision maker can benefit by changing its own decision unilaterally. Existing works in this category include [5], [8], [17], [11].

Load balancing approaches to minimize job response time or cost in distributed systems or Grids [1] have been developed in [4]-[10]. Queuing models have been widely used to model systems with shared resources, in order to give estimates of the performance. By modeling each node as an M/M/1 queuing system, the expected response time at each node can be determined. Consequently, the optimal load distribution to minimize the overall response time for user jobs is derived [4], [5], [6], [7], [8]. We broadly define this approach as *MinRT*. Following this approach and assuming a constant k that maps the execution time to the amount of resources required, the optimal load distribution to minimize the overall cost for user jobs

- *Qin Zheng is with the Computing Science Department, Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore 138632. E-mail: qinzheng.sg@gmail.com*
- *Bharadwaj Veeravalli is with the Communications & Networks Laboratory, Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576. E-mail: elebv@nus.edu.sg*

Digital Object Indentifier 10.1109/TC.2013.56

is derived [9], [10]. We broadly define this approach as *MinC*. Both approaches assume that prices of all nodes are given and fixed.

These two approaches consider one objective, that is, minimizing the response time or minimizing the cost. As a result, MinC prefers low-price nodes and when the system load is high, these nodes may become hot spots leading to long response time. On the other hand, MinRT prefers nodes with short response time without considering their prices, which may result in high cost. Finally, these two approaches benefit the users by providing them load balancing solutions with the minimum response time or cost. However, the objective of a resource provider, to maximize its revenue, is not taken into account. We note that it conflicts with the objective of MinC, which is to minimize the cost on behalf of the user.

1.2 Related Works on Pricing

Economic models have been proposed for distributed systems and Grids in recent studies [14], [15], [16], [17], [18], [19]. According to [19], fewer resources are wasted, while excess capacity and overloading are averaged over a very large number of providers and consumers. Further, they comment that in an economic model, all participants are considered self-interested. The resource providers are trying to maximize their revenues. The consumers want to obtain the maximum possible resources for the minimum possible price. Indeed, the various computers within a Grid site may not even cooperate with each other [17], for example, the noncooperation among departments in a large organization. It is also noted for the case of Internet that providers simply seek to maximize their own profit by charging users for access to their service [20].

A competition model is presented in [14], which describes the possible interactions between the competing resources in the Grid as service providers and ubiquitous applications as subscribers. It assumes that demand function is linear in all QoS parameters and investigates when the reduction in price leads to increase or decrease in revenue. In [15], the authors estimate (by aggregating historical information) the accuracy of predictions by the resource provider on when resources will be available and their price. Their scheduling and admission control policies are extended to factor in both resource-uncertainty and risk. To aid the users in deciding how much funding their jobs would need to complete, the authors of [16] develop a suite of price prediction models and tools by analyzing historical data. A hierarchical game-theoretic model of the Grid is presented in [17], which takes into account nodes selfishness and analytically derives the Nash equilibrium and optimal strategies. The authors of [18] consider pricing by adding a utilization-based price on top of a base price and the ratio between these two is the ratio of the maximum capacity over the free capacity (after serving this job and other jobs). A simulation study is reported in [19], which validates their choice of utility, price, and satisfaction function, to get some intuition regarding the transient and the steady state behavior of their economic models, assuming that all providers reveal their capacity and pricing parameters to the broker. Recently, as demonstrated on an experimental resource market at Google [3], by allowing prices to fluctuate using the proposed auction mechanisms, the authors observe an efficient transition of users from more congested resource pools to less congested resources.

Our work is different from existing pricing algorithms as we consider how owners can price their resources given the load balancing decision. We also investigate how pricing may influence load balancing decisions and the provider's revenue. Further, instead of estimating the price using *historical* data, we formulate it as an optimization problem and derive the optimum price that maximizes the revenue. This optimum price takes into account aggregated information on prices and service rates of all nodes in the system. Finally, we explain how to balance the load on nodes with existing loads and how to price nodes when the user load is balanced using a global approach or a greedy approach.

2 Problem Formulation

There is a resource broker in the system and it manages the computing nodes and acts as the intermediary between them and external users. Figure 1 shows a system with a broker and n heterogeneous computing nodes. Suppose there is a user with an arrival rate ϕ. Nodes are heterogeneous and each node i is characterized by its average service rate μ_i (load serviced per time unit) and the price per unit service rate p_i (the same price definition is used in [10]). We use a vector p to represent the prices of all nodes. Let β_i denote the fraction of ϕ allocated to node i and β denote a vector representing the fractions allocated to all nodes. The broker will determine a subset of nodes for load balancing and their arrival rate fractions, based on the arrival rate ϕ and service rates and prices of all nodes. Consequently, the expected cost and response time for this job are known. The broker also facilitates the owners on pricing their resources to maximize their revenues. Note that the price for a node should also be decided taking into account the user arrival rate and service rates and prices of all other nodes.

The objective of load balancing is to find a vector β such that the response time or cost to the user is minimized. Formally, for the case of cost minimization,

$$\min C(\beta, p), \tag{1}$$

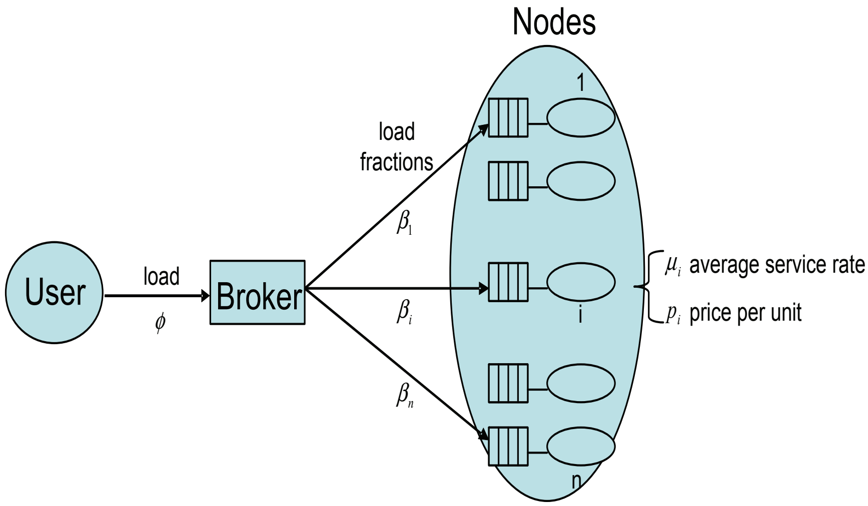

Fig. 1. The system model.

subject to $0 \leq \beta_i \leq \mu_i$ and $\sum_{1 \leq i \leq n} \beta_i = \phi$. We denote this problem as the optimal load balancing problem. It can be solved by the MinC theory, which derives the optimal arrival rate fractions minimizing the cost. We will discuss them in details in Section 3.1. Finally, a user may have a maximum total cost that he is willing to pay. If the overall cost is higher than his budget, he may not use the resources.

On the other hand, the objective of the owner of node i is to maximize the revenue, that is

$$\max R_i(\beta_i, p_i), \tag{2}$$

where $R_i(\beta_i, p_i) = p_i \beta_i$.

We denote this problem as the optimal pricing problem and its solution is the price that leads to the maximum revenue. Note that this optimal revenue helps an owner decide whether to participate considering its cost. Readers are referred to [22] for a detailed discussion on costs associated with providing resources for grid computing. Also, we note that an owner may own multiple computers while in this paper the revenue that each computer generates is maximized individually.

Note that the optimal load balancing problem and the optimal pricing problem have somehow conflicting objectives. However, they are two coupled problems as the optimal price depends on the arrival rate fractions and vice versa. Therefore, it is beneficial for them to be mutually aware. Further, it is important to consider both of them in an integrated manner in order to optimize the objectives of users and providers. Toward this goal, we will develop pricing algorithms that interact with the underlying load balancing approach.

3 An Integrated Optimal Load Balancing and Optimal Pricing Framework

The pricing issue and the load balancing issue are interrelated. The distribution of the arrival rate on nodes depends on their prices. On the other hand, the optimal prices also depend on the fractions of the arrival rate assigned to the nodes. Therefore, both are dynamic variables. In our design, they are mutually aware and integrated. In this section, we first explain the MinC approach, which determines the optimal distribution of the arrival rate given the prices of all nodes. Then we explain our pricing theory, which determines the optimal price given the arrival rate fractions of all the nodes.

3.1 Optimal Load Balancing

We use MinC as the underlying load balancing approach for our pricing work. In MinC, each node is modeled as an M/M/1 queuing system and hence the expected response time at node i is $\frac{1}{\mu_i - \beta_i}$. Recall that a constant k_i is assumed to map the execution time to the amount of resources consumed at node i. The objective of this approach is to minimize the overall job cost, that is

$$\min \sum_{1 \leq i \leq n} \frac{k_i p_i \beta_i}{\mu_i - \beta_i}. \tag{3}$$

Note that we could also have the arrival rate ϕ on the denominator so that the cost is averaged over the arrival rate. However, it will not affect the load balancing decision. For simplicity, in the remaining derivations, we let k_i equal to 1. The optimal arrival rate fraction to the above optimization problem, as given in [9], [10], is:

$$\beta_i = \mu_i - \sqrt{\mu_i p_i} \frac{S_{i,1} + \mu_i - \phi}{S_{i,2} + \sqrt{\mu_i p_i}} \tag{4}$$

Where

$$S_{i,1} = \sum_{1 \le j \le n, j \ne i} \mu_j, \quad S_{i,2} = \sum_{1 \le j \le n, j \ne i} \sqrt{\mu_j p_j}.$$

The optimal arrival rate allocation may not always be practical since β_i is not guaranteed to be non-negative. β_i is negative if

$$\sqrt{\frac{\mu_i}{p_i}} < \frac{\sum_{1 \le i \le n} \mu_i - \phi}{\sum_{1 \le i \le n} \sqrt{p_i \mu_i}}. \quad (5)$$

Therefore, in [9], [10], nodes are sorted in a list in non-decreasing order of their $\frac{\mu_i}{p_i}$ values ($\frac{\mu_1}{p_1} \le \frac{\mu_2}{p_2} \le ... \le \frac{\mu_n}{p_n}$). If there exists a node m ($1 \le m < n$) with $\sqrt{\frac{\mu_m}{p_m}} < \frac{\sum_{m \le i \le n} \mu_i - \phi}{\sum_{m \le i \le n} \sqrt{\mu_i p_i}}$, then $\beta_i = 0$ ($1 \le i \le m$). That is, m is the largest i for which the inequality holds.

An example is shown in Figure 2 where 16 nodes are sorted in a non-decreasing order of their $\frac{\mu_i}{p_i}$ values. The blue bar shows the service rate of the node and four of them (with high $\frac{\mu_i}{p_i}$ values) are used to process the current arrival rate (where the portion in red denotes β_i). It can be observed that β_i does not follow a non-decreasing order as it is also determined by its service rate as can be seen in Equation 4.

We note that after the current arrival rate, the nodes in the list still have their $\frac{\mu_i}{p_i}$ values in order. It can be proved as follows. For a node i that is selected to process the current arrival rate ($\beta_i > 0, m < i \le n$), its effective service rate will be $\mu_i - \beta_i = \sqrt{\mu_i p_i}\,\frac{S_{i,1}+\mu_i-\phi}{S_{i,2}+\sqrt{\mu_i p_i}}$ and its $\frac{\mu_i - \beta_i}{p_i} = \sqrt{\frac{\mu_i}{p_i}}\,\frac{S_{i,1}+\mu_i-\phi}{S_{i,2}+\sqrt{\mu_i p_i}}$. Therefore, the order is maintained among these nodes with $\beta_i > 0$. Further, this new value is larger than that of a node i with $\beta_i = 0$ ($1 \le i \le m$) due to Equation 5.

If $S_{i,1} \le \phi$, it implies that in this case node i is *dominant*. That is, if $S_{i,1} < \phi$, the current arrival rate cannot be assigned without node i. Or, if $S_{i,1} = \phi$, the current arrival rate can be balanced but will use up resources at all other nodes (besides i). Finally, note that the above arrival rate fraction is derived assuming that prices of all nodes are given and fixed.

3.2 Optimal Pricing Theory

In the following, we study the relationship between the revenue that an owner receives and the price that it charges. As an owner wants to maximize its revenue, it may use its price as a strategy and overcharge for its resource. However, we will show that increasing its price may decrease its arrival rate fraction monotonically. Further, there exists an optimal price that maximizes the revenue received by each node, if this node is not dominant.

When the owner of i increases its price, suppose that the prices of other nodes remain the same, the fraction of arrival rate ϕ that it receives decreases monotonically. This can be observed by rewriting Equation 4 in the following form:

$$\beta_i = \mu_i - \frac{S_{i,1} + \mu_i - \phi}{1 + \frac{S_{i,2}}{\sqrt{\mu_i p_i}}}. \quad (6)$$

Therefore, when an owner increases its price, its revenue is not necessarily increased. The following theorem formally determines when an owner's revenue is increasing or decreasing with its price, if prices of all other nodes remain the same.

Theorem 1. *i) If node i is dominant, owner i's revenue is monotonically increasing with its p_i from $p_i = 0$. ii) Otherwise, owner i's revenue is monotonically increasing with its p_i from $p_i = 0$ until p_i^*, when its maximum revenue (denoted as R_i^*) is achieved; after that, its revenue is monotonically decreasing to 0; and $\sqrt{p_i^*}$ is the (only) positive root ($\frac{-B-\sqrt{B^2-4AC}}{2A}$) of a quadratic function with*
$A = -2\mu_i(S_{i,1} - \phi)$
$B = \sqrt{\mu_i} S_{i,2}\,[\mu_i - 3(S_{i,1} - \phi)]$
$C = 2\mu_i(S_{i,2})^2$

Proof: We have

$$R_i(\beta_i, p_i) = p_i \beta_i = p_i(\mu_i - \sqrt{\mu_i p_i}\,\frac{S_{i,1} + \mu_i - \phi}{S_{i,2} + \sqrt{\mu_i p_i}}). \quad (7)$$

The last step is because we substitute β_i using Equation 4. It can be observed that now R_i is solely determined by p_i. Further, R_i equals to 0 when p_i is 0. In the following, we consider its derivative and continue from Equation 7, we have

$$\frac{\partial R_i(\beta_i, p_i)}{\partial p_i} = \frac{1}{2(\sqrt{\mu_i p_i} + S_{i,2})^2}\left\{A(\sqrt{p_i})^2 + B\sqrt{p_i} + C\right\} \quad (8)$$

$A = -2\mu_i(S_{i,1} - \phi)$
$B = \sqrt{\mu_i} S_{i,2}\,[\mu_i - 3(S_{i,1} - \phi)]$
$C = 2\mu_i(S_{i,2})^2$

It can be observed that on the right side of the equation, the part inside the braces is actually a *quadratic function* of $\sqrt{p_i}$. As $2(\sqrt{\mu_i p_i} + S_{i,2})^2$ must be positive, whether the derivative of $R_i(\beta_i, p_i)$ is positive or negative or zero is determined by the quadratic function. It can be observed that C must be positive. The parabola opens upward or downward depending on whether $S_{i,1} - \phi$ is negative or positive, respectively.

When $S_{i,1} - \phi$ is negative, A is positive and the parabola opens upward. Further, B must be positive. Recall that C is positive. Therefore, when p_i increases from 0, the quadratic function will strictly increase from 0. As a result, the derivative of the owner's revenue will remain positive and hence the revenue will increase monotonically. For the special case that $S_{i,1} - \phi$ is 0, it becomes a linear function. The revenue

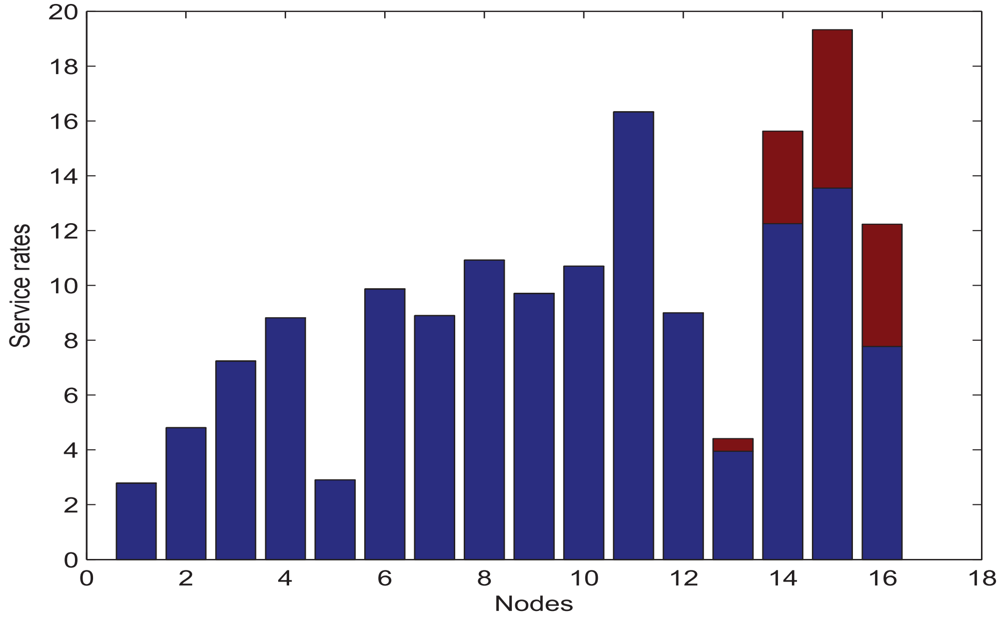

Fig. 2. An example for MinC on 16 nodes.

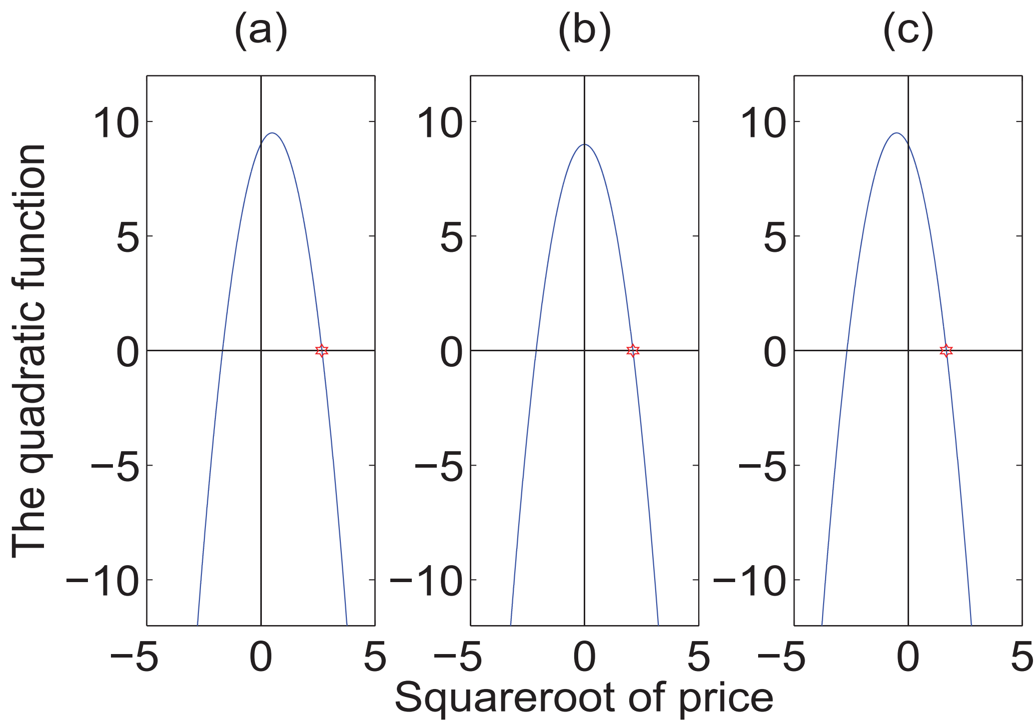

Fig. 3. The quadratic function ($A(\sqrt{p_i})^2 + B\sqrt{p_i} + C$) vs. the square root of price ($\sqrt{p_i}$) where subfigures (a), (b) and (c) correspond to the cases where B is positive, zero or negative, respectively.

will also increase monotonically with $\sqrt{p_i}$. Therefore, when $S_{i,1}-\phi$ is non-positive, i.e., node i is dominant, its owner can receive more revenue by increasing its price.

Otherwise, i.e., when $S_{i,1}-\phi$ is positive, the parabola opens downward. It can be observed that B can be positive, negative or zero depending on whether μ_i is greater than, smaller than or equal to $3(S_{i,1}-\phi)$. The positive case is shown in the left graph in Figure 3. We will explain this case and the same result can be obtained for the zero and negative cases, as shown in the middle and right graphs in the figure. When p_i increases from zero till it reaches the positive root of the quadratic function, the quadratic function, and hence the derivative of the owner's revenue is positive. After this point, the derivative becomes negative. Therefore, the optimal revenue (denoted as R_i^*) is reached when $\sqrt{p_i}$ is at the positive root. This optimal value for $\sqrt{p_i^*}$ is $\frac{-B-\sqrt{B^2-4AC}}{2A}$. □

This theorem implies that if a node is dominant, then its owner can receive more revenue by increasing its price. Otherwise, in the general case, there is an optimal price for the owner that maximizes its revenue. This maximum revenue can help the owner decide whether to process the current arrival rate. It also provides guidance to owners on how to optimally price their resources. Therefore, the amount of resources (supply) could change for each arrival rate. However, the supply is still bounded by the capacity of the system as external resources are not able to participate. Therefore, the supply is not following a true supply curve defined by the open market.

For resource allocation in distributed systems, it is normally assumed that all resource providers reveal their capacity and pricing parameters to the broker [19]. In our work, it can be observed that in order to determine the optimal price for a node, the broker only needs to know $S_{i,1}$ and $S_{i,2}$ besides the user arrival rate and the capacity of this node. On the other hand, an owner can determine its optimal price when provided with information on these two aggregated values *only*, rather than the service rate and price information from each individual node. Note that these nodes are non-cooperative and they are not willing to reveal their service rates and prices to other owners. Therefore, our pricing mechanism, which requires only aggregated information, has a higher chance to be accepted by owners.

Note that the optimal price for a node is determined by the set of nodes selected for load balancing and also their prices. Therefore, in the case that one or more of these owners "lie" to the broker, the optimal price that each of them gets is not "really" optimal. As a result, their revenues are likely to be affected, which in the long run may discourage them from lying.

3.3 Load Balancing Solutions with Preassigned Rates

In this section, we extend our pricing theory to a multi-user environment. Essentially, the arrival rate from a user may need to be balanced after a few other users have done so, that is, onto processors with already assigned arrival rates. Also, preassigned rates exist even for a single user if his arrival rate arrives at different time instances. Next, we will first explain how new arrival rate can be balanced with preassigned rates. Suppose that there is a new arrival rate ϕ and c_i denotes the preassigned arrival rate on node i. We consider two load balancing approaches: global and greedy. In the global approach, the cost is minimized for the whole system, that is,

$$\min \sum_{1\le i\le n} \frac{p_i(\beta_i + c_i)}{\mu_i - \beta_i - c_i}. \tag{9}$$

Using the Lagrange multiplier theorem, the solution for this constrained-minimum problem is

$$\beta_i = \mu_i - c_i - \sqrt{\mu_i p_i}\,\frac{S'_{i,1} + \mu_i - c_i - \phi}{S_{i,2} + \sqrt{\mu_i p_i}} \tag{10}$$

where $S'_{i,1} = \sum_{1\le j\le n, j\ne i} (\mu_j - c_j)$. The detailed proof is not given in this paper and it can be proved following the steps for the case without preassigned rates in [21]. Similarly, nodes are sorted in a list in non-decreasing order of their $\frac{\mu_i - c_i}{\sqrt{\mu_i p_i}}$ values to determine the set of nodes that should not be assigned any arrival rate.

We find out that this global approach can be implemented in a simple way. Suppose that the current arrival rate is ϕ. Instead of calculating β_i according to Equation 10, we can invoke MinC with arrival rate $\phi + \sum_{1\le i\le n} c_i$. That is, the overall arrival rate till this time. Let d_i denote the returned arrival rate fraction. Then β_i for this current arrival rate can be calculated by $d_i - c_i$. In this way, the preassigned rate (c_i) does not have to be known by MinC. The proof sketch on the correctness of this alternative way is given in the Appendix. Finally, we note that this alternative way will not work if the prices of nodes are changing, for example, with the proposed optimal pricing theory. The reason is that, as can been seen in the proof, the order of the nodes in the list (in terms of their $\frac{\mu_i}{p_i}$ values) is no longer fixed when their prices are changing.

In the greedy approach, the objective is to minimize the cost for the current arrival rate (from a certain user), that is,

$$\min \sum_{1\le i\le n} \frac{p_i \beta_i}{\mu_i - \beta_i - c_i}. \tag{11}$$

Using the Lagrange multiplier theorem, the solution for this constrained-minimum problem is

$$\beta_i = \mu_i - c_i - \sqrt{(\mu_i - c_i)p_i}\,\frac{S'_{i,1} + \mu_i - c_i - \phi}{S'_{i,2} + \sqrt{(\mu_i - c_i)p_i}} \tag{12}$$

where $S'_{i,2} = \sum_{1 \le j \le n, j \ne i} \sqrt{(\mu_j - c_j)p_j}$. Similarly, nodes are sorted in a list in non-decreasing order of their $\frac{\mu_i - c_i}{p_i}$ values to determine the set of nodes that should not be assigned any arrival rate.

3.4 Pricing Solutions with Preassigned Rates

In this section, we explain how to price nodes when they have preassigned rates. We consider the global and greedy load balancing approaches described above. For the global approach, similar to the optimal price without preassigned rates in Section 3.2 (and hence the proof is not given), the optimal price is still the (only) positive root of a quadratic function. For the quadratic function, A, B and C are different due to the preassigned arrival rates.

$A = -2\mu_i(S'_{i,1} - \phi)$
$B = \sqrt{\mu_i} S_{i,2} \left[\mu_i - c_i - 3(S'_{i,1} - \phi)\right]$
$C = 2(\mu_i - c_i)(S_{i,2})^2$

For the greedy approach, similarly, the optimal price is the (only) positive root of a quadratic function with

$A = -2(\mu_i - c_i)(S'_{i,1} - \phi)$
$B = \sqrt{\mu_i - c_i}\ S'_{i,2} \left[\mu_i - c_i - 3(S'_{i,1} - \phi)\right]$
$C = 2(\mu_i - c_i)(S'_{i,2})^2$

Similarly, the above optimal prices are for the general case where no node is dominant. Otherwise, the owner who dominates can receive more revenue by increasing its price. Further, to determine the optimal price for a node, only aggregate information from other nodes ($S'_{i,1}$ and $S_{i,2}$ (or $S'_{i,2}$)) is needed.

3.5 Complexity

We now analyze the complexity of the proposed framework. Note that when $S_{i,1}$ and $S_{i,2}$ (both take $O(n)$) are known, the optimal price for an owner can be computed in $O(1)$ (to calculate A, B, and C in Theorem 1). There are at most n nodes and hence the optimal prices for the set of nodes selected for load balancing can be calculated in $O(n)$ at most. Finally, MinC takes $O(n \log n)$. Therefore, the overall complexity is $O(n \log n)$.

4 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms (denoted as OptP), which integrate pricing with load balancing. We compare them with $MinC$ and $MinRT$. Further, we also compare with two alternative approaches where owners, given the optimal prices, can decide whether to use the optimal prices or how to charge based on the optimal prices. In the first approach, owners choose to adopt the optimal prices randomly. That is, on average the optimal prices are used for half of the jobs. This approach is denoted with "R" in the results. In the second approach, owners always set prices at the average of the optimal price and the last-round price. This approach is denoted with "A" in the results. The objective of comparing with these two approaches is to study the effect of owners' price decisions on the performance.

The scenario considered may have multiple users with arrival rates coming at different time instances. The arrival rate from a user is balanced on nodes that could have already assigned arrival rates. The greedy approach (default, without additional notation) and the global approach (denoted with "global") in Section III.C are used for load balancing. We consider a system with a large number of (1600) heterogeneous nodes. Service rates of these nodes are assumed to be uniformly distributed in the range $[2.0, 20.0]$. Initial prices of nodes can be set by their owners and in our experiments they are assumed to be uniformly distributed in the range $[0.1, 1.0]$. Note that our proposed pricing algorithms do not depend on the initial prices of nodes. Job arrival is assumed to follow Poisson process as has been used in [8], [11], [12]. For job arrival rates, we consider two cases. In the first case, the arrival rate is assumed to be exponentially distributed with a mean of 20.0. In the second case, the arrival rate is assumed to follow a heavy tail distribution as found out by previous studies in most computing systems according to [6]. In this case, a small number of very large jobs make up a significant portion of the total load. We use the Bounded Pareto distribution with the probability density function as given in [6]. For job execution time, we use exponential distribution as has been suggested in the literature [8], [9], [11]. The simulations were run over 1,000,000 jobs with different random number streams at each run. In order to obtain stable and accurate results with a confidence of 95%, each of our experiments was repeated 25 times and an average was taken.

The performance metrics used are the expected cost and response time per job where

cost= $\sum_{1 \le i \le n} \frac{p_i \beta_i}{\mu_i - \beta_i - c_i}$, RT= $\sum_{1 \le i \le n} \frac{\beta_i}{\mu_i - \beta_i - c_i}$ and

RT (max) = $\phi \cdot \max_{1 \le i \le n} \frac{1}{\mu_i - \beta_i - c_i}$.

RT and RT (max) differ in that the former reflects the sum of the expected response time for the load distributed on all used nodes while the latter reflects the maximum of the expected response time among the used nodes. Note that the expected cost equals to the revenue received by all nodes selected for this job.

Table I shows the performance of all algorithms for Exponential distributed arrival rates at light to moderate load. We observe that the proposed pricing algorithms, OptP (using the greedy load balancing algorithm) and OptP global (using the global load balancing algorithm), achieve a response time close to that of MinRT. For MinC, it results in a long response time as it prefers low-price nodes. For cost, MinRT results in a high cost as it prefers nodes with short

TABLE 1
Results for Exponential distributed arrival rates at light to moderate load

Algs	MinRT	MinRT global	MinC	MinC global	OptP	OptP global	OptP R	OptP global R	OptP A	OptP global A
RT	726	745	1000	861	768	752	769	754	765	747
RT (max)	729	925	2833	3019	1033	1209	1328	1588	1000	1148
Cost	397	410	282	304	355	352	351	349	358	356

TABLE 2
Results for Exponential distributed arrival rates at moderate to heavy load

Algs	MinRT	MinRT global	MinC	MinC global	OptP	OptP global	OptP R	OptP global R	OptP A	OptP global A
RT	3783	3819	3908	3853	3793	3825	3798	3831	3789	3822
RT (max)	4014	8491	21373	16549	4060	8644	4437	8991	4027	8632
Cost	2073	2101	2018	2050	1828	1834	1804	1813	1807	1816

TABLE 3
Results for heavy-tail distributed arrival rates at light to moderate load

Algs	MinRT	MinRT global	MinC	MinC global	OptP	OptP global	OptP R	OptP global R	OptP A	OptP global A
RT	751	771	1015	879	789	778	789	779	785	773
RT (max)	752	977	2997	3195	924	1145	1186	1655	942	1142
Cost	411	424	301	324	367	364	363	362	371	368

TABLE 4
Results for heavy-tail distributed arrival rates at moderate to heavy load

Algs	MinRT	MinRT global	MinC	MinC global	OptP	OptP global	OptP R	OptP global R	OptP A	OptP global A
RT	11942	11978	12065	12012	11952	11984	13360	13379	11948	11980
RT (max)	11946	27633	64665	51829	12103	27717	21248	41782	12093	27730
Cost	6545	6588	6489	6521	35421	35426	17283	17411	20233	20242

response time without considering their prices. As a result, the user may have to pay a high cost. On the other hand, by minimizing the cost, in MinC the expected revenue is the lowest and may not be acceptable to the provider.

By jointly maximizing the provider's revenue through pricing, the proposed integrated framework achieves a nice tradeoff in the expected cost for a win-win situation between the user and the provider. We also observe that at the light load the proposed algorithms, OptP and OptP global, perform closely with two alternative approaches (OptP R and OptP A). One exception is that in the random approach (R), the RT (max) is longer, which is not good to the users. Finally, the performance difference between using the global load balancing approach and the greedy load balancing approach is not significant for every algorithm.

Table II shows the performance of all algorithms for Exponential distributed arrival rates at moderate to heavy load. Firstly, it can be observed that both response time and cost are increased significantly. The proposed algorithms achieve a response time close to that of MinRT. For MinC, it results in a long response time (max) as it continues using low-price nodes even if they are saturated. For cost, MinRT results in a highest cost as it prefers nodes with short response time without considering their prices. The cost yielded by MinC is also high due to two reasons. Firstly, especially at heavy load, it has to balance the load on high-price nodes as the capacity of low-price nodes has been largely used. Secondly, as the response time yielded by MinC is long, a provider will receive a higher cost for the duration. The proposed integrated framework in fact yields a lower cost by considering the objectives of both the user and the provider. Note that it is not that MinRT and MinC could maximize the revenue (cost) but due to their design as just described. The cost yielded by the proposed pricing algorithms is more reasonable and is more acceptable to the user. We also observe that comparing to the two alternative approaches, the random approach (OptP R) still results in a longer RT (max). The optimal pricing algorithm (OptP) achieves a higher revenue than the average approach (OptP A), which takes a conservative approach by taking the average between the optimal price (for the current load) and the price charged for the last round. Finally, the performance between the global approach and the greedy approach now differ significantly in RT (max). This is because by greedily balancing the current load, a less expected response time could be achieved although the overall system performance may suffer.

Similar trends on performance can be observed for heavy-tail distributed arrival rates except that at heavy load, both response time and cost are increased

TABLE 5
Results for heavy-tail distributed arrival rates at moderate to heavy load with a limit on the price

Algs	OptP (1.5)	OptP global (1.5)	OptP (2.0)	OptP global (2.0)	OptP (2.5)	OptP global (2.5)
RT	12120	12026	11975	11990	11952	11984
RT (max)	81237	59896	88754	54568	12103	27726
Cost	8733	8749	9559	9571	9668	9680

significantly due to a small number of very large jobs. Also, at high load the cost yielded by the proposed pricing algorithms are much higher. This is because there are very large jobs and some nodes can get a large fraction of the arrival rates from them, when they charge at their optimal prices (which could be high). As a result, these nodes generate high revenue and also result in a high cost to users. To confirm the cause of this high cost at high load for heavy-tail distribution and enable us to control it in the proposed approaches, we carry out the following experiments.

In the following, we set a limit on the price that an owner can charge at each round. Specifically, the optimal price can only be used if it is not greater than x times of the last-round price. This is to avoid the case that an owner receives a large load charged at a high price, which results in a high cost. We note that a user may have a budget constraint or may choose another provider considering the cost, which means the provider may not receive any revenue at all. Also, this budget constraint may be facilitated by the broker. We vary x in the simulations and the results in Table V correspond to its value at 1.5, 2.0 and 2.5. It can be observed that by giving tight limits, the cost can be controlled. When we relax the limit, the cost increases gradually until at a certain point as we observed. The point is when the limit is more than five times and then the cost equals to the proposed optimal pricing approach without the limit. This is because that the limit is too loose and does not affect the price. This also confirms that the high cost at high load for the heavy-tail distribution is due to the high optimal price charged (for the large fraction of the arrival rate). We also observe that the RT does not change much with the limit. Therefore, the provider can tune the limit for a desired revenue while consider the cost acceptable to users. Finally, for the RT (max), we observe that for a tight limit, it becomes higher while beyond 2.5, it equals to the case without the limit. This is because for a tight limit, more nodes are charged at their last-round prices, which is (much) lower than their optimal prices. Consequently, the load balancing algorithm will allocate more load to these nodes as they are moving towards the end of the sorted list (see Section III. A). As a result, their capacities may be saturated by arriving jobs leading to long response time (RT (max)).

5 Discussions on a Distributed Pricing Algorithm and Its Potential for Cloud Computing

In this section, we briefly discuss how our optimal pricing theory can be used in a distributed manner where each owner is autonomous. An owner determines its optimal price and revenue using the aggregated information ($S_{i,1}$ and $S_{i,2}$) provided by the broker. Owners are synchronized and in each iteration, every owner calculates its optimal price and adjusts to it. Note that each owner assumes that the prices of all the other nodes are kept fixed. Therefore, owners can calculate their optimal prices independently and simultaneously. An owner may receive increased or decreased (or zero) fraction of the arrival rate and revenue depending on the new prices of other nodes. Therefore, the algorithm starts with the initially selected nodes and in the iterations the nodes in the set are dynamic. In each iteration, every owner sends its new price to the broker, which returns the (new) arrival rate fraction and the updated aggregated information needed to calculate the optimal price for the next iteration. Note that in this scenario the broker is not involved in pricing decisions.

This situation can be viewed as a non-cooperative game among decision makers (owners). Each owner optimizes its own revenue independently of the others. The *Nash equilibrium* [2] for the game is a strategy profile with the property that no owner can increase its expected revenue by changing its price given the other owners' prices. In other words a strategy profile is a Nash equilibrium if no owner can benefit by deviating unilaterally from its price to another feasible one. An important question is whether this algorithm can converge to the Nash equilibrium. In this algorithm, each owner iteratively adjusts its price to the new optimal price until no owner can receive more revenue by unilaterally changing its price (e.g., the Nash equilibrium is reached). That is, the expected revenues for the set of nodes used for load balancing all remain the same as the previous iteration. The only known results about the convergence to the Nash equilibrium are for distributed load balancing algorithms with linear and strictly increasing link costs [23]. The convergence proof for more than two players with general cost functions is still an open problem [8]. The authors of [8], [12] have demonstrated using simulation experiments that their distributed load balancing algorithms converge to the Nash equilibrium in distributed systems and computational grids.

This distributed autonomous pricing algorithm can be used for cloud environments where there might exist multiple providers. Therefore, there could be multiple brokers, such as the scenario described in InterCloud [24]. In such a decentralized architecture, these brokers may interact with each other and use the distributed pricing algorithm to autonomously determine prices through iterations. We note that for it to be used with virtualized environments, which is common for Cloud computing, the configurations of Virtual Machines (VMs) need to be considered in terms of allocation of their capability for user tasks. Specifically, node capacity needs to be priced based on an additional factor of VM configurations. A future evaluation of the proposed optimal pricing work in a realistic cloud setting, such as InterCloud or Hybrid Cloud [25], [26], [27], would be useful to investigate the application potential of this work in such environments.

6 Conclusions

In the paper, we addressed an important problem that integrates load balancing with pricing to provide a win-win situation between resource owners and users. We found that there exists an optimal price that maximizes the revenue for each owner. To determine it, only aggregated information on processing speeds and prices of computing nodes is required. We developed pricing algorithms for scenarios where the load arrives at the same time or at different time instances possibly from multiple users. We developed algorithms for a global approach with the objective to optimize the system-wide performance and a greedy approach with the objective to optimize the performance for the current load (from a user). Through simulation studies, we demonstrated the proposed algorithms can achieve a response time close to MinRT and a cost acceptable to both users and providers. Therefore, they perform the best considering the two objectives of time and cost. We do note that users can choose alternative algorithms if they consider just one objective or their jobs follow heavy-tail distribution and the arrival rate is moderate or heavy. We also note that a user can make the choice based on how much he would like to pay for better performance. Our optimal price theory can help an owner decide its optimal price and revenue, which helps it decide whether to process the current job and how to price its resource. In the future, we plan to test the proposed algorithms on actual platforms with realistic applications or workloads. Such an investigation together with virtual machines would help understand the potential of this work for cloud environments.

References

[1] I. Foster and C. Kesselman, Eds.. *The Grid: Blueprint for a Future Computing Infrastructure*. 2nd Edition. Morgan Kaufmann Publishers, 2004.

[2] J. Nash, “Non-Cooperative Games,” *The Annals of Mathematics*, Second Series, vol. 54, no. 2, pp. 286-295, Sep. 1951.

[3] M. Stokely, J. Winget, E. Keyes, C. Grimes, and B. Yolken, “Using a Market Economy to Provision Compute Resources Across Planet-wide Clusters,” in *Proceedings of the 23th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2009.

[4] L.M. Ni and K. Hwang, “Optimal Load Balancing in a Multiple Processor System with Many Job Classes,” *IEEE Transactions on Software Engineering*, vol. SE-11, no. 5, pp. 491-496, May 1985.

[5] H. Kameda, J. Li, C. Kim, and Y. Zhang. *Optimal Load Balancing in Distributed Computer Systems*. Springer Verlag, London, 1997.

[6] X. Tang and S. T. Chanson, “Optimizing Static Job Scheduling in a Network of Heterogeneous Computers,” in *Proceedings of the International Conference on Parallel Processing (ICPP)*, pp. 373-382, Aug. 2000.

[7] D. Grosu and A. T. Chronopoulos, “A Game-Theoretic Model and Algorithm for Load Balancing in Distributed Systems,” in *Proceedings of the 16th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2002.

[8] D. Grosu and A. T. Chronopoulos, “Noncooperative Load Balancing in Distributed Systems,” *Journal of Parallel and Distributed Computing*, vol. 65, no. 9, pp. 1022-1034, Sep. 2005.

[9] P. Ghosh, N. Roy, K. Basu, and S.K. Das, “A Game Theory Based Pricing Strategy for Job Allocation in Mobile Grids,” in *Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 26-30, 2004.

[10] S. Penmatsa and A. T. Chronopoulos, “Job Allocation Schemes in Computational Grids Based on Cost Optimization,” in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2005.

[11] P. Ghosh, K. Basu, and S.K. Das, “A Game Theory Based Pricing Strategy to Support Single/Multiple Job Allocation Schemes for Bandwidth-Constrained Distributed Computing Systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 3, pp. 289-306, Mar. 2007.

[12] R. Subrata, A.Y. Zomaya, and B. Landfeldt, “Game Theoretic Approach for Load Balancing in Computational Grids,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 66-76, Jan. 2008.

[13] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, Cambridge, MA, 1994.

[14] N. Roy, S.K. Das, K. Basu, and M. Kumar, “Enhancing Availability of Grid Computational Services to Ubiquitous Computing Applications,” in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2005.

[15] F.I. Popovici and J. Wilkes, “Profitable services in an uncertain world,” in *Proceedings of Supercomputing'05*, Nov. 2005.

[16] T. Sandholm, K. Lai, J.A. Ortiz, and J. Odeberg,“Market-Based Resource Allocation using Price Prediction in a High Performance Computing Grid for Scientific Applications,” in *Proceedings of HPDC*, pp. 132-143, 2006.

[17] Y.-K. Kwok, K. Hwang, and S. Song, “Selfish Grids: Game-Theoretic Modeling and NAS/PSA Benchmark Evaluation”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 5, pp. 621-636, May 2007.

[18] C. S. Yeo and R. Buyya, “Pricing for Utility-driven Resource Management and Allocation in Clusters,” *International Journal of High Performance Computing Applications*, vol. 21, no. 4, pp. 405-418, Nov. 2007.

[19] X. Bai, D.C. Marinescu, L. Boloni, H.J. Siegel, R.A. Daley, and I-J. Wang, “A macroeconomic model for resource allocation in large-scale distributed systems,” *Journal of Parallel and Distributed Computing*, vol. 68, no. 2, pp. 182-199, Feb. 2008.

[20] A. Hayrapetyan, E. Tardos, and T. Wexler, “A network pricing game for selfish traffic,” in *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing (PODC)*, 2005.

[21] Q. Zheng, C.K. Tham, and B. Veeravalli, “Dynamic Load Balancing and Pricing in Grid Computing with Communication Delay,” *Journal of Grid Computing*, special issue on Grid

Economics and Business Models, vol. 6, no. 3, pp. 239-253, Sep. 2008.

[22] J.A. Chandy, “An Analysis of Resource Costs in a Public Computing Grid,” in *Proceedings of the 16th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2009.

[23] T. Boulogne, E. Altman, and O. Pourtallier, “On the Convergence to Nash Equilibrium in Problems of Distributed Computing,” *Annals of Operation Research*, vol. 109, no. 1, pp. 279-291, Jan. 2002.

[24] R. Buyya, R. Ranjan, and R.N. Calheiros, “InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services,” in *Proceedings of ICA3PP*, 2010.

[25] T. Bicer, D. Chiu, and G. Agrawal, “Time and Cost Sensitive Data-Intensive Computing on Hybrid Clouds,” in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2012.

[26] R.V. Bossche, K. Vanmechelen, and J. Broeckhove, “Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads,” in *Proceedings of the 3rd International Conference on Cloud Computing*, 2010.

[27] S. Sotiriadis, N. Bessis, F. Xhafa, and N. Antonopoulos, “From meta-computing to interoperable infrastructures: A review of meta-schedulers for HPC, grid and cloud,” in *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications*, 2012.

PLACE PHOTO HERE

Qin Zheng received the B. Eng. degree in Information Engineering from Xi’an Jiaotong University, Xi’an, China in July 2001 and the Ph.D. degree in Electrical and Computer Engineering from National University of Singapore, Singapore in January 2006. He then worked as a Research Fellow in the same department till November 2007 when he joined the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include Cloud computing, MapReduce, and data analytics.

PLACE PHOTO HERE

Bharadwaj Veeravalli , Senior Member, IEEE IEEE-CS, received his BSc in Physics, from Madurai-Kamaraj University, India in 1987, Master’s in Electrical Communication Engineering from Indian Institute of Science, Bangalore, India in 1991 and PhD from Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India in 1994. He received Gold Medals for his Bachelor’s Degree overall performance and for an outstanding PhD thesis (Sabitha Chowdhary Gold Medal, IISc, Bangalore India) in the years 1987 and 1994, respectively. He did his post-doctoral research in the Department of Computer Science, Concordia University, Montreal, Canada, in 1996. He is currently with the Department of Electrical and Computer Engineering, Communications and Information Engineering (CIE) division, at The National University of Singapore, Singapore, as a tenured Associate Professor. His main stream research interests include, Cloud/Grid/Cluster Computing, Scheduling in Parallel and Distributed Systems, Bioinformatics Computational Biology, and Multimedia Computing. He is one of the earliest researchers in the field of Divisible Load Theory (DLT). He had successfully secured several externally funded projects and published over 120 papers in high-quality International Journals and Conferences. He has co-authored three research monographs in the areas of PDS, Distributed Databases (Competitive Algorithms), and Networked Multimedia Systems, in the years 1996, 2003, and 2005, respectively. He had guest edited a special issue on Cluster/Grid Computing for IJCA, USA journal in 2004. He is currently serving the Editorial Board of IEEE Transactions on SMC-A, Multimedia Tools Applications (MTAP) and Cluster Computing, as an Associate Editor. Until 2010 December he had served as an AE for IEEE Transactions on Computers. He was a Visiting Professor with HUST, Wuhan, China, from June 2007-May 2009. He had served (and serving) as a program committee member and as a General Co-Chair in IEEE ICON 2011 and IEEE ICON 2012. In September 2010, he delivered a keynote speech in fifth IEEE International Conference on Bio-Inspired Computing: Theory and Applications (BIC-TA) September 2010, held in Changsha, PR China. At present he is serving as a lead guest editor for a special issue with IEEE TC on Cloud of Clouds. Author’s academic profile, professional activities, main stream and peripheral research interests, research projects and collaborations, most recent list of publications, can be found in http://cnds.ece.nus.edu.sg/elebv